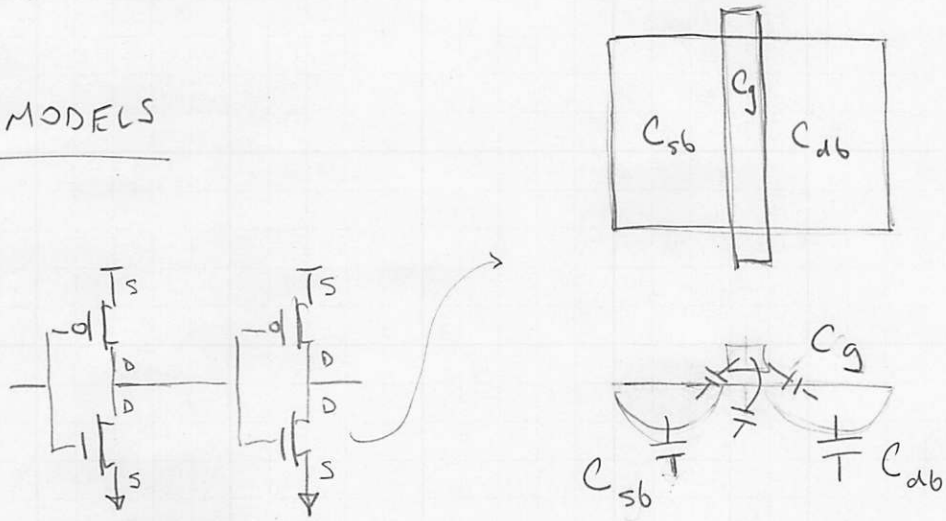
CMOS COMBINATIONAL LOGIC

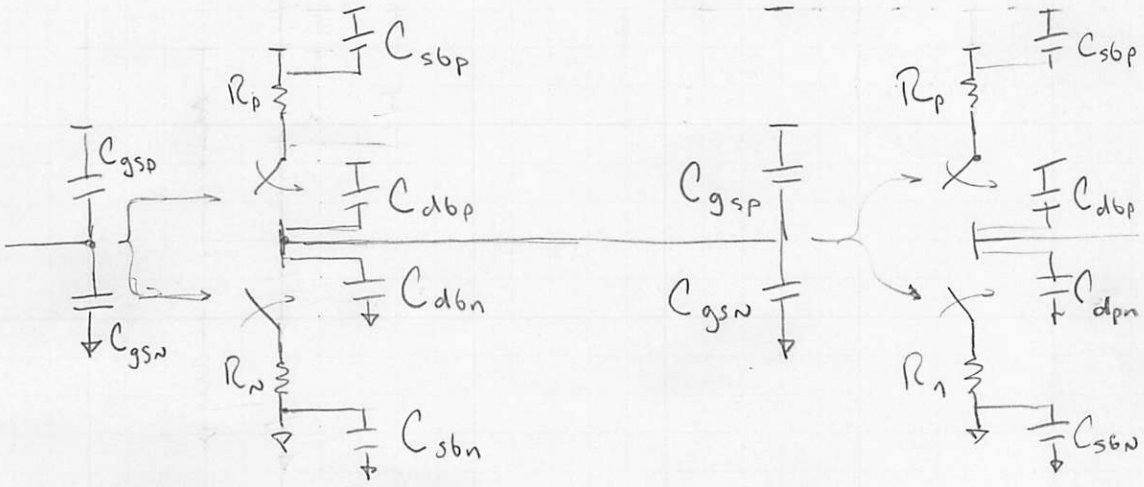
DELAY
- RC MODELS
- RC DELAY
- LOGICAL EFFORT

ENERGY

AREA

RC MODELS



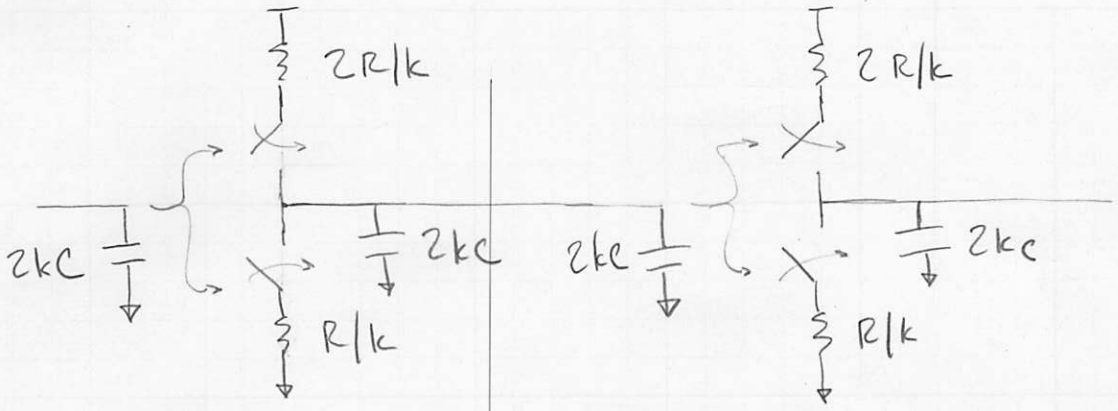


- C_{sb} CAPACITORS DO NOT ACTUALLY SWITCH, SO IGNORE
- LUMP $C_{Dbp} + C_{Dbn}$ SINCE BOTH TIED TO CONSTANT NODES
- LUMP $C_{gsp} + C_{gsn}$ SINCE BOTH TIED TO CONSTANT NODES
- ASSUME PMOS MOBILITY 2x WORSE THAN NMOS MOBILITY

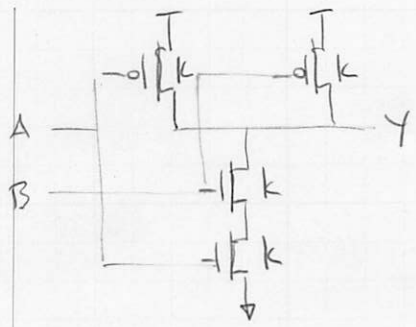
DEFINE C = GATE CAPACITANCE OF MINIMUM SIZED NMOS

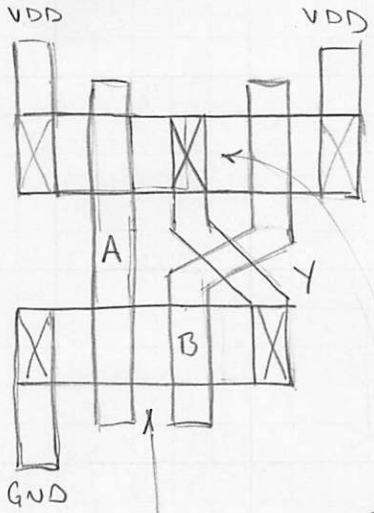
DEFINE R = EFFECTIVE RESISTANCE OF MINIMUM SIZED NMOS

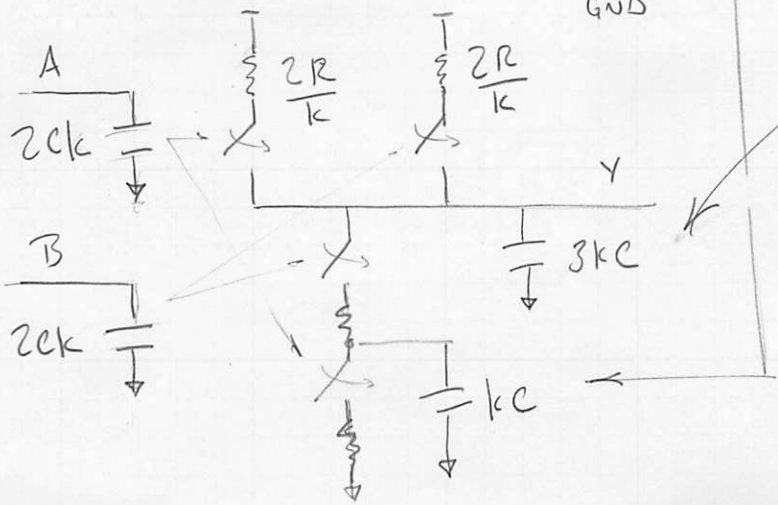
DEFINE k = HOW MUCH WIDER A TRANSISTOR IS COMPARED TO MIN NMOS



2INPUT NAND GATE



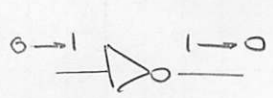


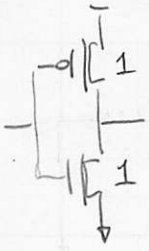


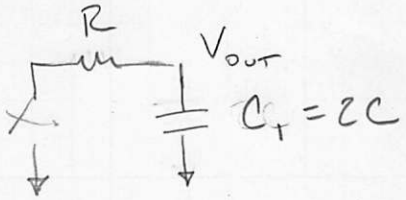
IF WE SHARE CONTACT DIFF THEN THIS HAS PARASITIC CAP NC NOT 2C! 3C ASSUMES NO DIFF SHARING!

SHARED DIFFUSION CAP

RC DELAY : INVERTER





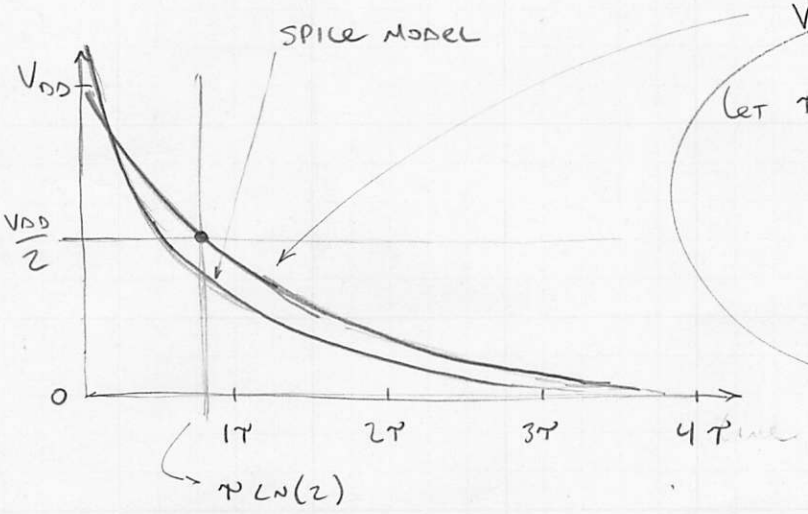


$$V_{OUT}(t) = V_{DD} e^{-t/RC_1}$$

Let $\tau = RC_1$

$$= \frac{V}{I} \cdot \frac{Q}{V} = \frac{Q}{I}$$

$$= \frac{Q}{Q/s} = \underline{\text{seconds}}$$



t_{PD} = PROPAGATION DELAY, TIME UNTIL $V_{OUT} = V_{DD}/2$

$= LN(2) \cdot R \cdot C_1$ Let $R' = LN(2) \cdot R$

$= R' C_1 = 2R'C$

We usually just assume effective resistance is scaled by LN(2)

$$\boxed{t_{PD} = 2RC}$$

OHM'S LAW $V = IR$
DEF of C is Q/V

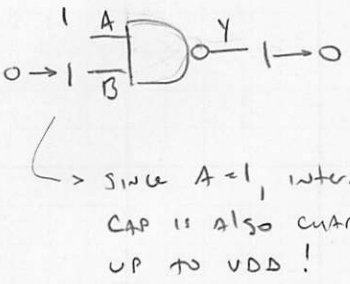
$$V_{OUT} = \frac{V_{DD}}{2} = V_{DD} e^{-t/\tau}$$

$$\frac{1}{V_{DD}} \frac{V_{DD}}{2} = e^{-t/\tau}$$

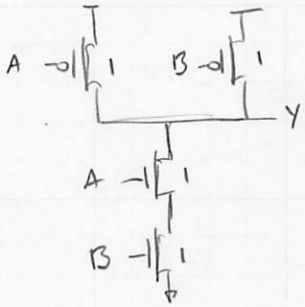
$$LN\left(\frac{1}{2}\right) = -\frac{t}{\tau}$$

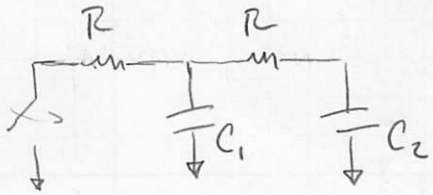
$$t = -\tau LN\left(\frac{1}{2}\right) = \tau LN(2)$$

RC DELAY: 2 INPUT NAND



→ SINCE A=1, INTERNAL CAP IS ALSO CHARGED UP TO VDD!





$C_1 = C$

$C_2 = 3C$

COMPLICATES 2ND ORDER MODEL

APPROXIMATION

$$\tau = \tau_1 + \tau_2 = RC_1 + (R+R)C_2$$

$$= RC + (2R)(3C)$$

$$= RC + 6RC = 7RC \quad (3.5\times \text{ slower than inverter})$$

BEST WHEN ONE τ MUCH LARGER THAN OTHER τ

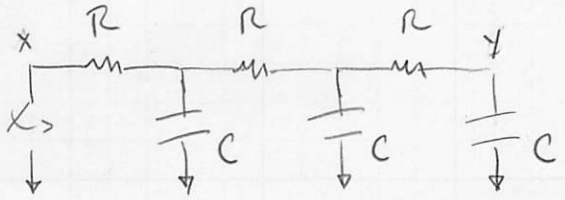
EVEN IF $\tau_1 = \tau_2$, ERROR < 15%

GENERALIZE TO ELMORE DELAY

$$t_{PD} = \sum_{i}^{\text{all nodes}} R_{ij} C_i$$

C_i: CAPACITANCE AT NODE i

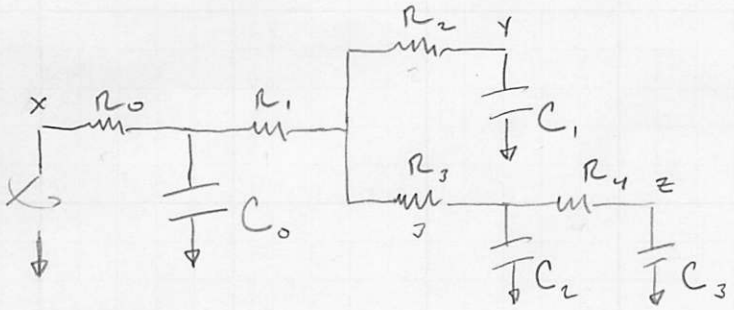
R_{ij}: EFFECTIVE RESISTANCE ON SHARED PATH FROM INPUT TO NODE i AND NODE j, WHERE j IS OUTPUT



$$t_{PD} = RC + 2RC + 3RC$$

$$= 6RC$$

ELMORE DELAY OF TREES



ASSUME ALL RESISTANCES ARE R AND ALL CAPACITANCES ARE C

DELAY OF PATH FROM X TO Y IS IMPACTED BY BRANCH TO Z
DELAY OF PATH FROM X TO Z IS IMPACTED BY BRANCH TO Y

FOR PATH X TO Y WE ALSO LUMP $C_2 + C_3$ AND USE SHARED RESISTANCE, $R_0 + R_1$

SIMILARLY FOR PATH X TO Z WE LUMP C_1 AND USE SHARED RESISTANCE, $R_0 + R_1$

THIS EXTRA TERM FACTORS IN DELAY OF "BRANCH"

$$t_{PD,XY} = R_0 C_0 + (R_0 + R_1 + R_2) C_1 + \underbrace{(R_0 + R_1)(C_2 + C_3)}_{\text{DELAY DUE TO EXTRA BRANCH}}$$

$$= RC + 3RC + 4RC$$

$$= 8RC$$

$$t_{PD,XZ} = R_0 C_0 + (R_0 + R_1 + R_3) C_2 + (R_0 + R_1 + R_3 + R_4) C_3 + \underbrace{(R_0 + R_1) C_1}_{\text{DELAY DUE TO EXTRA BRANCH}}$$

$$= RC + 3RC + 4RC + 2RC$$

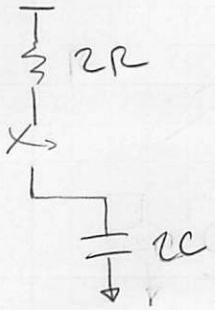
$$= 10RC$$

RISE/FALL TIMES : INVERTER

FROM EARLIER $t_{PD,1\to0} = 2RC$

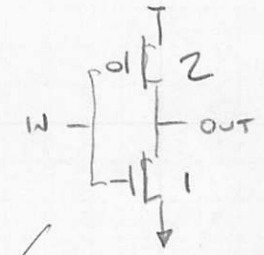
$t_{PD,0\to1} = 4RC$

} UNEQUAL RISE/FALL TIMES



FOR EQUAL RISE/FALL TIMES, THE EFFECTIVE RESISTANCE OF PULLUP MUST EQUAL EFFECTIVE RESISTANCE OF PULL DOWN

IF WE ASSUME PMOS MOBILITY 2x WORSE THAN NMOS, THEN PMOS MUST BE 2x SIZE OF NMOS IN AN INVERTER FOR EQUAL RISE/FALL

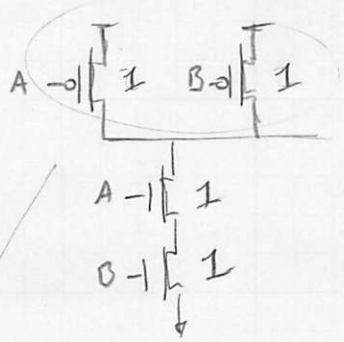


$t_{PD,1\to0} = 3RC$

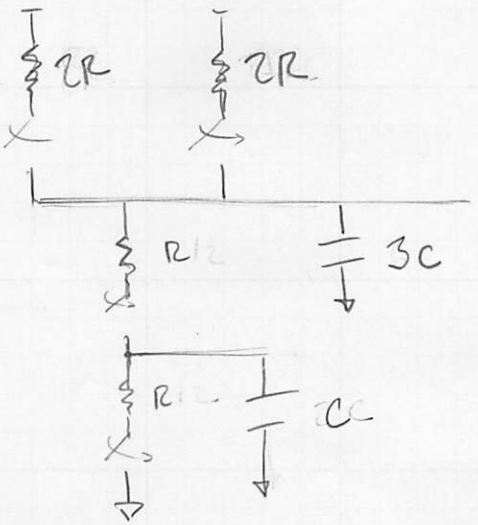
$t_{PD,0\to1} = 3RC$

RISE/FALL TIMES : 2INPUT NAND

SIZE TRANSISTORS SO WORST CASE EFFECTIVE RESISTANCE EQUAL IN BOTH PULL UP AND PULL DOWN NETWORKS



ASSUMES WORST CASE WHERE ONLY A SINGLE PMOS IS PULLING UP OUTPUT NODE



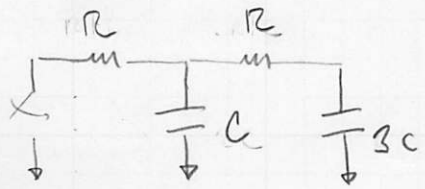
WORST CASE EFFECTIVE RESISTANCE = 2R

WORST CASE EFFECTIVE RESISTANCE = 2R

$t_{PD,1\to0}$

$A = 1$

$B = 0 \to 1$



$t_{PD} = RC + (R+R)3C$

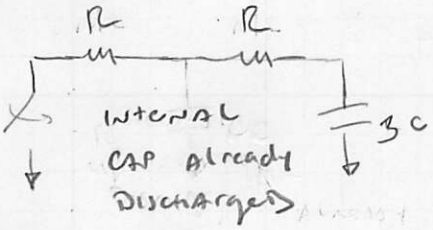
$= RC + 6RC$

$= 7RC$

$t_{PD,1\to0}$

$A = 0 \to 1$

$B = 1$



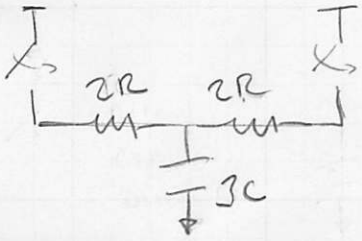
$t_{PD} = (R+R)3C$

$= 6RC$

$t_{PD,0\to1}$

$A = 1 \to 0$

$B = 1 \to 0$



$t_{PD} = (2R)3C$

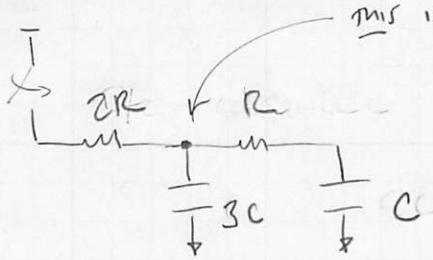
$= 3RC$

$t_{PD,0\to1}$

$A = 1$

$B = 1 \to 0$

THIS IS OUTPUT NODE!



$t_{PD} = 2R \cdot 3C + 2RC$

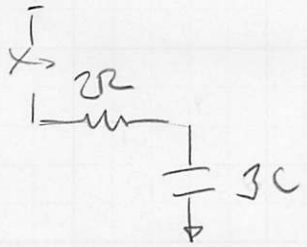
$= 6RC + 2RC$

$= 8RC$

$t_{PD,0\to1}$

$A = 1 \to 0$

$B = 1$

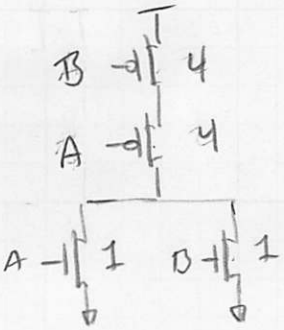


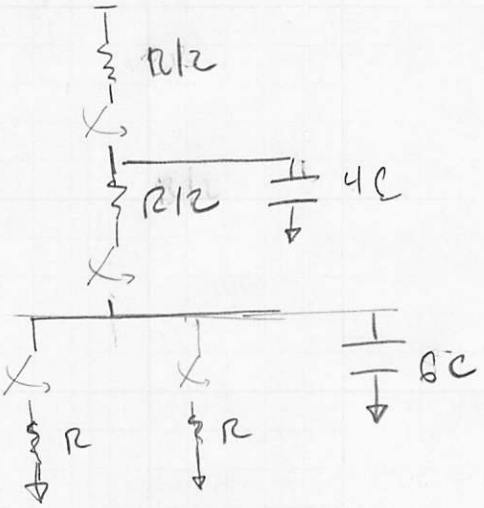
$t_{PD} = 6RC$

	$1\to0$		$0\to1$	
	WORST	BEST	WORST	BEST
NO INTERNAL CAP	6RC		6RC	3RC
W/ INTERNAL CAP	6RC	7RC	8RC	3RC

SO DELAY DEPENDS ON INTERNAL CAPACITANCE AND THE ORDER IN WHICH INPUTS ARRIVE OR WHICH INPUTS CHANGE

RISE/FALL TIMES: 2INPUT NOR

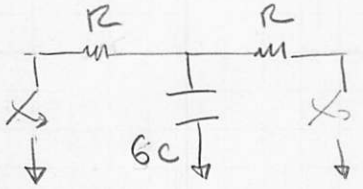




$t_{PD}, 1 \rightarrow 0$

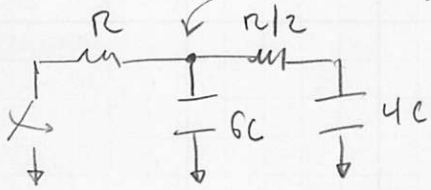
$A = 0 \rightarrow 1$

$B = 0 \rightarrow 1$



$t_{PD} = \frac{R}{2} 6C$

$= 3RC$



THIS IS OUTPUT NODE!

$t_{PD} = 6RC + R4C$

$= 6RC + 4RC$

$= 10RC$

$t_{PD}, 1 \rightarrow 0$

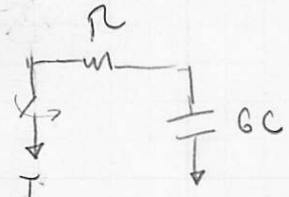
$A = 0$

$B = 0 \rightarrow 1$

$t_{PD}, 1 \rightarrow 0$

$A = 0 \rightarrow 1$

$B = 0$

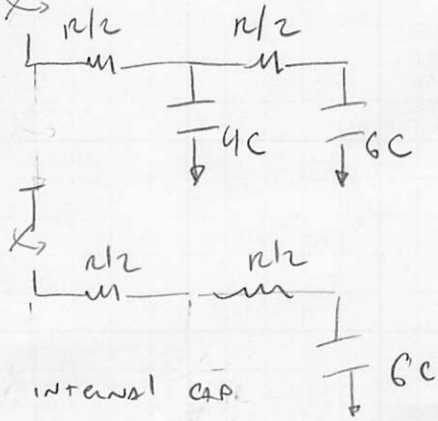


$t_{PD} = 6RC$

$t_{PD}, 0 \rightarrow 1$

$A = 0$

$B = 1 \rightarrow 0$



$t_{PD} = \frac{R}{2} 4C + \left(\frac{R}{2} + \frac{R}{2}\right) 6C$

$= 2RC + 6RC$

$= 8RC$

$t_{PD} = 6RC$

$t_{PD}, 0 \rightarrow 1$

$A = 1 \rightarrow 0$

$B = 0$

INTERNAL CAP ALREADY CHARGED UP!

		$t_{PD,1\to0}$ WORST	$t_{PD,1\to0}$ BEST	$t_{PD,0\to1}$ WORST	$t_{PD,0\to1}$ BEST
INVERTER		3RC		3RC	
NAND	NO INTERNAL CAP	6RC		6RC	3RC
	W/ INTERNAL CAP	7RC	6RC	8RC	3RC
NOR	NO INTERNAL CAP	6RC	3RC	6RC	
	W/ INTERNAL CAP	10RC	3RC	8RC	6RC

IS THIS A FAIR COMPARISON? NO, WE ARE NOT NORMALIZING ANY THING ACROSS THESE GATES. NEED TO EITHER NORMALIZE

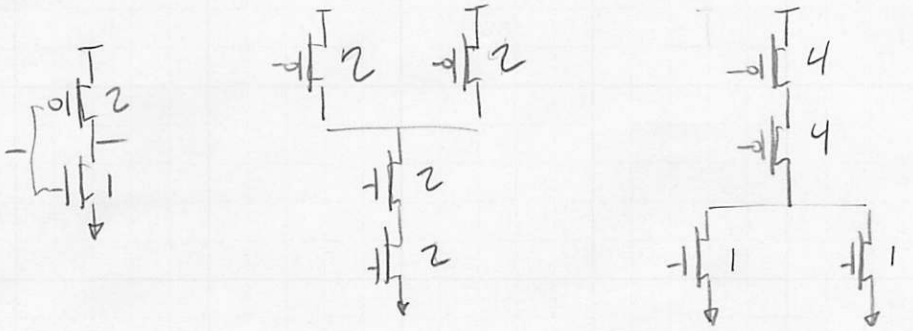
1) INPUT GATE CAP (IE. LOAD ON PREVIOUS GATE)

2) DRIVE STRENGTH (IE. EFFECTIVE RESISTANCE)

EFFECTIVE RESISTANCE OF ALL 3 GATES

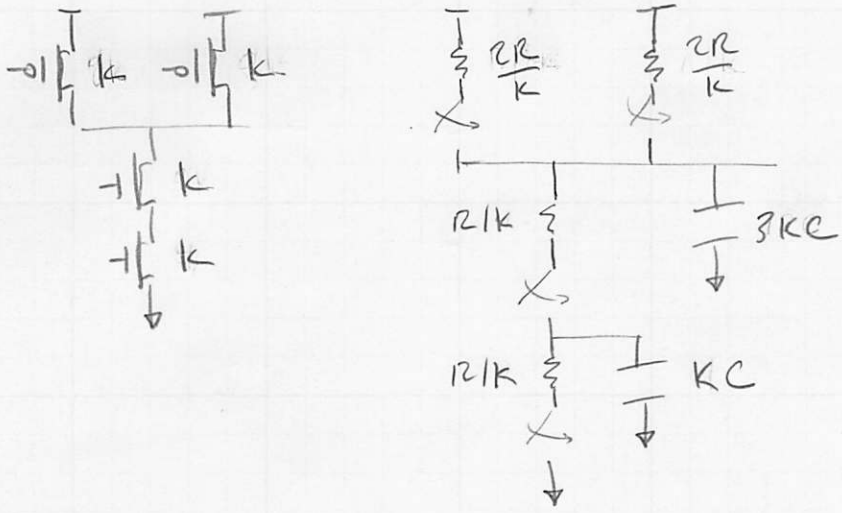
INV	R
NAND	2R ← TWICE THE EFFECTIVE RESISTANCE
NOR	R ← SO HALF THE DRIVE CURRENT

HERE ARE ALL THREE GATES SIZED TO HAVE EQUAL RISE AND FALL TIMES AND THE SAME DRIVE STRENGTH



R_{EFF} =	R	R	R
C_{IN} =	3C	4C	5C
C_P =	3C	6C	6C
$t_{PD,1\to0}$ =	3RC	6RC	6RC) IGNORE INT CAP
$t_{PD,0\to1}$ =	3RC	6RC	6RC) WORST CASE

LARGER GATES



WORST CASE

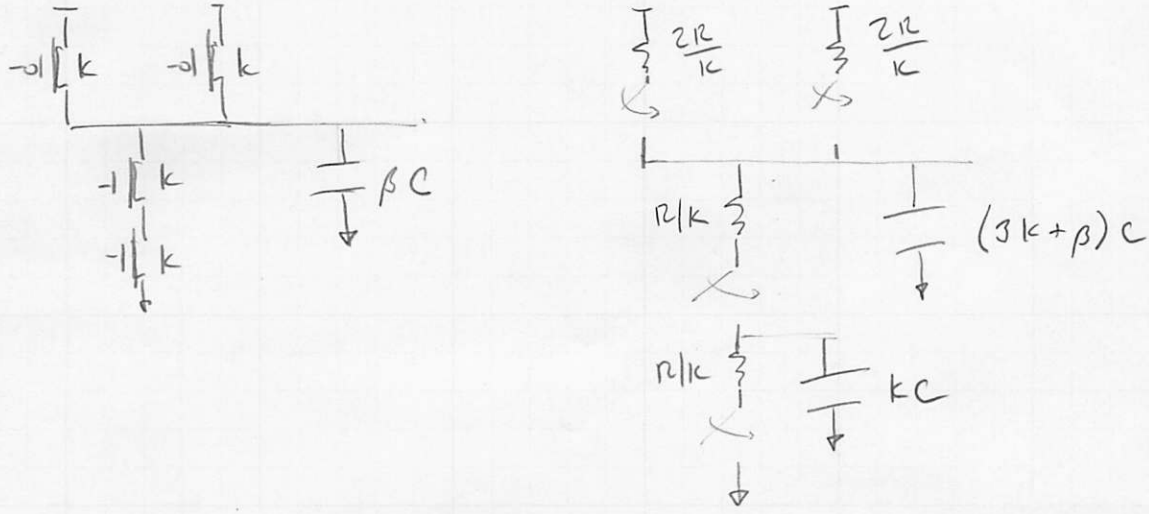
$$t_{PD,1\to0} = \frac{R}{k}kC + \left(\frac{R}{k} + \frac{R}{k}\right)3kC = 7RC$$

$$t_{PD,0\to1} = \frac{2R}{k}\cdot 3kC + \left(\frac{2R}{k}\right)kC = 9RC$$

SAME AS BEFORE!

THIS IS THE PARASITIC DELAY, IT IS INDEPENDENT OF SIZE

LARGER LOADS



$t_{PD,1\to0}$

$$\frac{R}{k}\cdot kC + \left(\frac{R}{k}+\frac{R}{k}\right)(3k+\beta)C$$

$$RC + 2\frac{R}{k}(3k+\beta)C$$

$$RC + 6RC + 2\frac{R}{k}\beta C$$

$$\underline{7RC} + \underline{\frac{\beta}{k}2RC}$$

↳ EFFORT DELAY: DEPENDS ON COMPLEXITY OF GATE, SIZE OF GATE, AND WHAT IT IS DRIVING

→ PARASITIC DELAY: INHERENT DELAY WHEN NO LOAD IS ATTACHED, INDEPENDENT OF SIZING

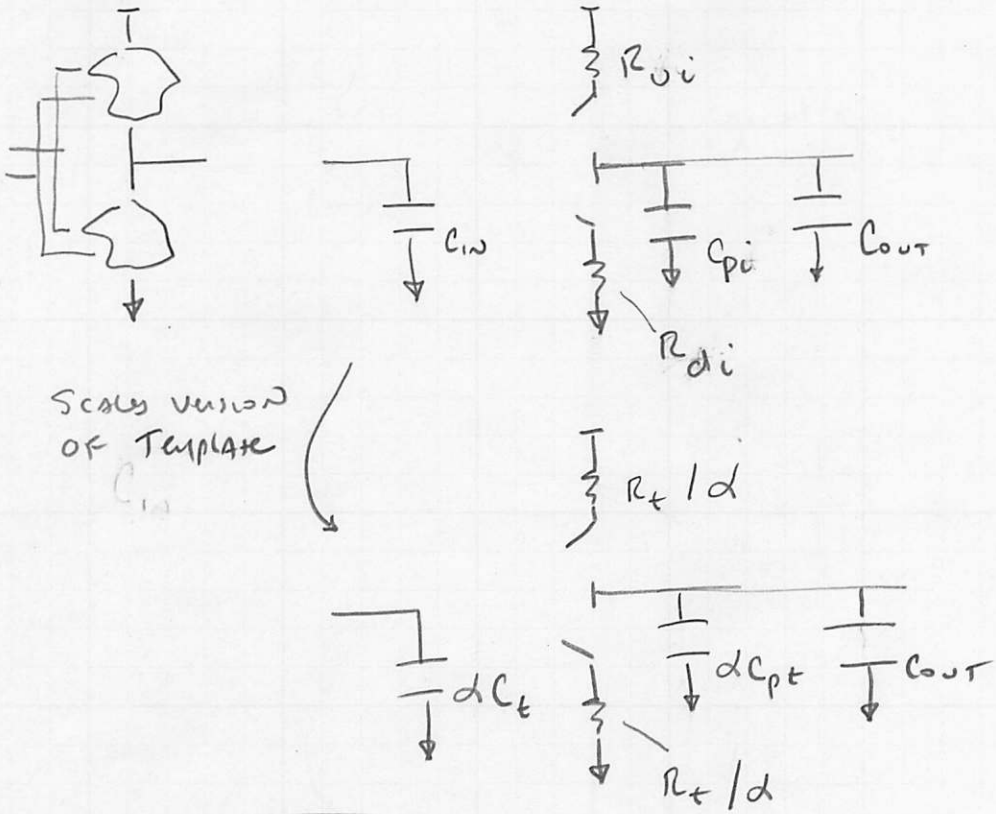
NOTE: INCREASING β, INCREASES EFFORT DELAY

INCREASING k, DECREASES EFFORT DELAY

BUT will INCREASE β of PREVIOUS GATE!

Deriving Linear Decay Mode (logical effort)

- Abstract all CMOS logic gates as



$C_{IN} = \alpha C_t$

$R_i = R_{ui} = R_{di} = R_t/\alpha$

$C_{pi} = \alpha C_{pt}$

α is subtly different from k
α is how much larger gate is compares to the template, where the template is defined to have EQ rise/fall + same drive strength as min inverter

$$t_{pd} = d_{ABS} = R_i (C_{out} + C_{pi})$$

$$= \frac{R_t}{\alpha} C_{IN} \left(\frac{C_{OUT}}{C_{IN}}\right) + \frac{R_t}{\alpha} (\alpha C_{pt})$$

$$= \frac{R_t}{\alpha} \alpha C_t \left(\frac{C_{OUT}}{C_{IN}}\right) + \frac{R_t}{\alpha} (\alpha C_{pt})$$

$$= R_t C_t \left(\frac{C_{OUT}}{C_{IN}}\right) + R_t C_{pt}$$

$$d_{ABS} = \underbrace{R_t C_t \left(\frac{C_{OUT}}{C_{IN}}\right)}_{\text{ABS effort delay}} + \underbrace{R_t C_{pt}}_{\text{ABS parasitic delay}}$$

ABS parasitic delay independent of d and C_{OUT}!

d is "hidden" in just C_{IN}

Let $\tau = R_{INV} C_{INV}$ "delay units"

$g = R_t C_t / R_{INV} C_{INV}$ logical effort

$h = C_{OUT} / C_{IN}$ electrical effort

$p = R_t C_{pt} / R_{INV} C_{INV}$ parasitic delay

$$d_{ABS} = R_{INV} C_{INV} \left(\frac{R_t C_t}{R_{INV} C_{INV}}\right)\left(\frac{C_{OUT}}{C_{INV}}\right) + R_{INV} C_{INV} \left(\frac{R_t C_{pt}}{R_{INV} C_{INV}}\right)$$

$$\boxed{d_{ABS} = \tau (gh + p)}$$

$$d_{ABS} = \tau d$$

Ratio of load to input cap

If C_{OUT} ↑ delay ↑

To reduce delay need to ↑ C_{IN}

- Complexity of gate topology
- Ratio of gate time constant to INV time constant
- How much worse is gate at producing current compared to INV with same input cap
- How much more input cap is necessary to drive same current as min. INV.

= 1 for INV!

If we assume template has same effective resistance as minimum inverter then

$R_t = R_{INV}$

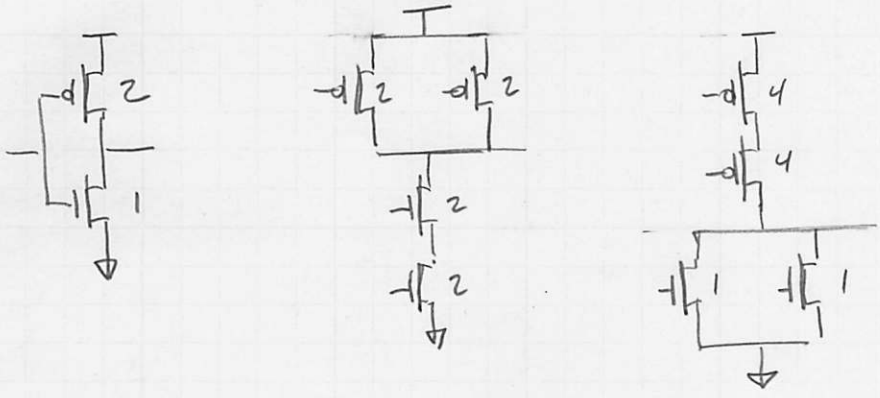
so $g = \frac{R_t C_t}{R_{INV} C_{INV}} = \frac{C_t}{C_{INV}}$

$p_{INV} = R_t C_{pt} / R_{INV} C_{INV} = C_{pt} / C_{INV}$

Ratio of diff cap / µm vs gate cap / µm

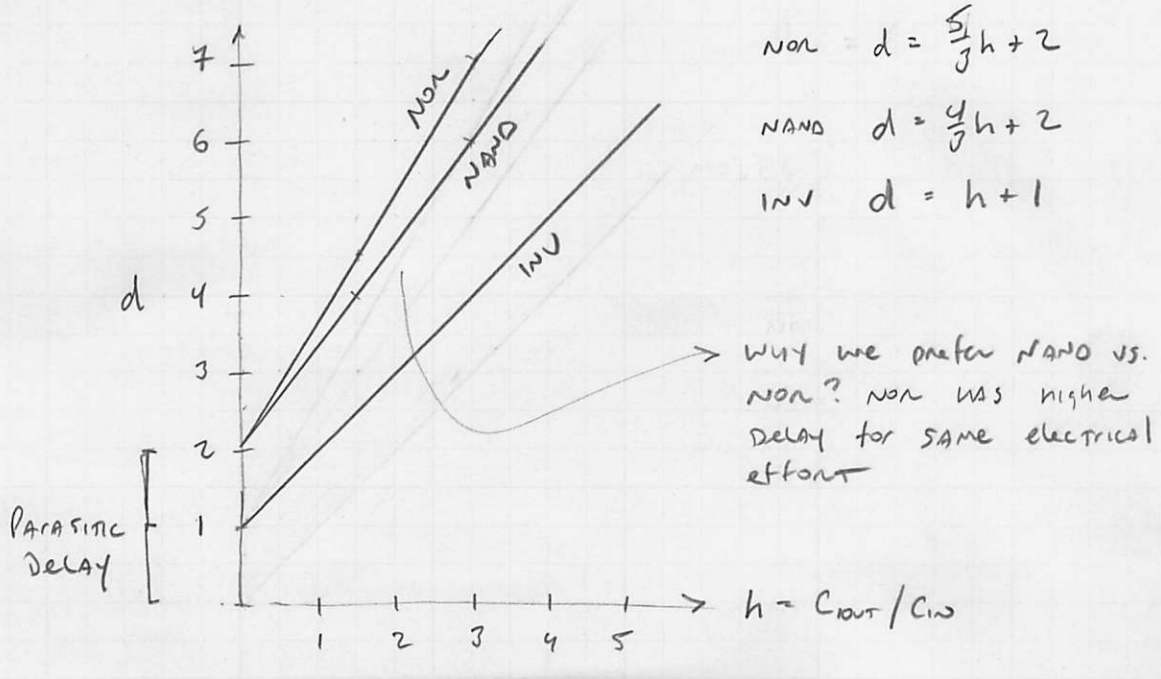
Crudely assume 1 for INV

TEMPLATES



R_{EFF}	R	R	R
C_{IN}	$3C$	$4C$	$5C$
$\frac{R_t C_t}{R_{INV} C_{INV}}$	$\frac{R3C}{R3C}$	$\frac{R4C}{R3C}$	$\frac{R5C}{R3C}$
g	1	$4/3$	$5/3$
$\frac{R_t C_{pt}}{R_{INV} C_{INV}}$	1	$\frac{6}{3} = 2$	$\frac{6}{3} = 2$

RECALL $d = gh + p$ (LINEAR DELAY MODEL)
PLOT d AS A FUNCTION OF h



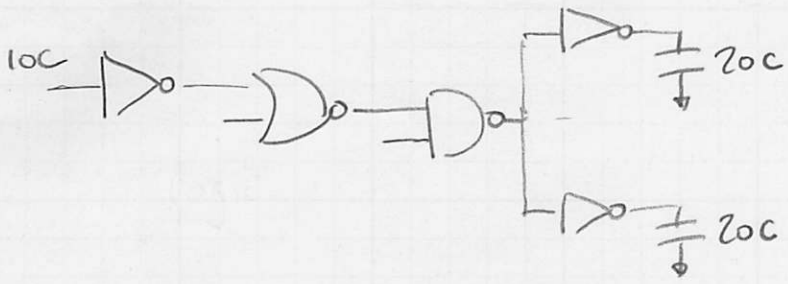
MANY, MANY APPROXIMATIONS

- ELMORE DELAY
- $p = 1$ FOR INV (IE. 1/μm DIFF CAP ≈ 1/μm GATE CAP)
- IGNORE INTERNAL PARASITIC CAP
- EQUAL RISE AND FALL TIME
- $\mu_p = \frac{1}{2}\mu_n$
- IGNORE ACTUAL RISE/FALL TIMES
- IGNORE INPUT ARRIVAL TIME
- IGNORE VELOCITY SATURATION

STILL REASONABLY GOOD RESULTS WHEN USING LOGICAL EFFORT FOR SIZING EVEN IN MODERN TECHNOLOGIES

HELPS DESIGNERS BUILD THE RIGHT INTUITION

MULTISTAGE LOGIC NETWORKS



PATH DELAY IS THE SUM OF THE DELAYS OF EACH STAGE

$$D = \sum d_i = \sum g_i h_i + \sum p_i$$

KEY QUESTIONS:

- HOW SHOULD WE SIZE GATES TO MINIMIZE TOTAL DELAY?
- HOW SHOULD WE CHANGE THE TOPOLOGY TO MINIMIZE DELAY?

LET'S DEVELOP SOME METRICS THAT ARE INDEPENDENT OF SIZING

PATH LOGICAL EFFORT

$$G = \prod g_i$$

PATH ELECTRICAL EFFORT

$$H = \frac{C_{out}}{C_{in}}$$

SO FOR ABOVE EXAMPLE:

$$G = 1 \cdot \frac{5}{3} \cdot \frac{4}{3} \cdot 1 = 20/9 = 2.22$$

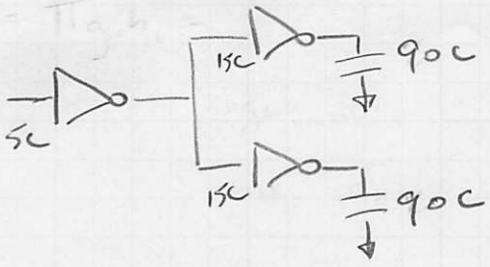
$$H = 20C / 10C = 2$$

PATH EFFORT IS THE PRODUCT OF STAGE EFFORTS

$$F = \prod f_i = \prod g_i h_i$$

So since STAGE effort $f = gh$, does PATH effort $F = GH$?

Consider simple example:



$G = 1 \cdot 1 = 1$

$H = 90C / 5C = 18$

$F = \Pi g_i h_i = 1 \times 6 \times 1 \times 6 = 36$

↑ 6 NOT 3 because Cout for first INV is 30C!

BUT $GH = 18 \neq 36$

So in this example $F = 2GH$

↑ we call this the "branching" effort

Key idea is some of the drive current is directed off the PATH we are analyzing! Recall elmore delay for trees.

STAGE branch eff. $$b = \frac{C_{ONPATH} + C_{OFFPATH}}{C_{ONPATH}}$$

PATH branch eff. $B = \Pi b_i$

for above example $b = \frac{15+15}{15} = 2$

So PATH EFFORT

$$F = \Pi f_i = \Pi g_i h_i = GBH$$

NOTE that PATH effort depends on circuit topology and loading of entire PATH but NOT size of transistors in network

ALSO NOTE that PATH effort DOES NOT change if ADD or remove inverters!

WITH THESE METRICS WE CAN NOW ANSWER THE TWO EARLIER QUESTIONS.

Q1: HOW SHOULD WE SIZE GATES TO MINIMIZE TOTAL DELAY?

START WITH PATH DELAY EQUATION:

$$D = \sum d_i = \sum g_i h_i + \sum p_i$$

INDEPENDENT VARIABLES ARE h's (i.e. INTERNAL GATE SIZINGS). WE WANT TO CHOOSE h's TO MINIMIZE D. SO WE CAN TAKE THE PARTIAL DERIVATIVE OF D WITH RESPECT TO h's, SET TO ZERO, AND SOLVE FOR OPTIMUM h's.

CONSIDER TWO STAGE PATH

ASSUME C_1 AND C_3 ARE GIVEN

INPUT CAP	C_1	C_2
LOGICAL EFF	g_1	g_2
PARASITIC DELAY	p_1	p_2

$$D = (g_1 h_1 + p_1) + (g_2 h_2 + p_2)$$

NOTE THAT h_1 AND h_2 ARE CONSTRAINED SINCE C_1 AND C_3 ARE GIVEN AND INPUT CAP OF GATE 2 IS OUTPUT CAP FOR GATE 1

$$h_1 = C_2 / C_1 \qquad h_2 = C_3 / C_2$$

$$H = h_1 h_2 = \frac{C_3}{C_1}$$ H IS A CONSTANT SINCE C_1 AND C_3 GIVEN

SUBSTITUTE $h_2 = H / h_1$ INTO DELAY EQUATION

$$D = (g_1 h_1 + p_1) + \left(\frac{g_2 H}{h_1} + p_2\right)$$

TAKE PARTIAL DERIVATIVE WITH RESPECT TO ONLY VARIABLE h_1

$$\frac{\partial D}{\partial h_1} = g_1 - \frac{g_2 H}{h_1^2} = 0$$

$$g_1 h_1^2 = g_2 H$$
$$g_1 h_1^2 = g_2 h_1 h_2$$
$$g_1 h_1 = g_2 h_2$$
$$f_1 = f_2$$

→ DELAY IS MINIMIZED WHEN STAGE EFFORT IS SAME IN BOTH STAGES!

This generalizes to paths with any number of stages and paths with branching effort.

FASTEST DESIGN ALWAYS EQUALIZES EFFORT IN EACH STAGE.

For a general path, optimal delay is:

$$\hat{f} = F^{1/N}$$

TAKE TOTAL PATH EFFORT AND DIVIDE EVENLY ACROSS N STAGES

$$\boxed{\hat{D} = N F^{1/N} + P}$$

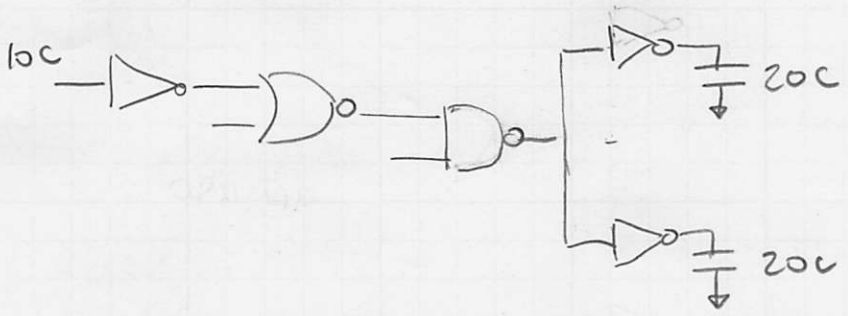
MIN PATH DELAY W/ OPTIMAL SIZING

Method for optimal sizing

1. CALCULATE PATH EFFORT $F = GBH$
2. CALCULATE EFFORT FOR EACH STAGE $\hat{f} = F^{1/N}$
3. ESTIMATE MIN DELAY W/ OPT SIZING $\hat{D} = N\hat{f} + P$
4. STARTING WITH LAST STAGE, WORK BACKWARDS SIZING EACH GATE

$$\hat{f} = gh = g\frac{C_{out}}{C_{in}} \qquad C_{in} = \frac{g}{\hat{f}} C_{out}$$

EXAMPLE OF OPTIMAL SIZING



g	1	5/3	4/3	1	
h	1	1	2	1	N = 4
p	1	2	2	1	

Step 1) $F = GBH = \left(1 \cdot \frac{5}{3} \cdot \frac{4}{3} \cdot 1\right)\left(1 \cdot 1 \cdot 2 \cdot 1\right)\left(\frac{20C}{10C}\right)$

$= 2.22 \times 2 \times 2 = 8.89$

Step 2) $\hat{f} = F^{1/N} = 1.73$

Step 3) $\hat{D} = 4 \cdot (1.73) + (1 + 2 + 2 + 1)$

$= 6.92 + 6 = 12.92$

$\hat{D}_{ABS} = 12.92\,\tau$

Step 4) $C_{in,} = \frac{1}{1.73}\,20C = 11.56C$

Due to branching!

$C_{in} = \frac{4/3}{1.73}\,11.56C \times 2 = 17.81C$

$C_{in} = \frac{5/3}{1.73}\,17.81C = 17.16C$

$C_{in} = \frac{1}{1.73}\,17.16C = 9.92C$

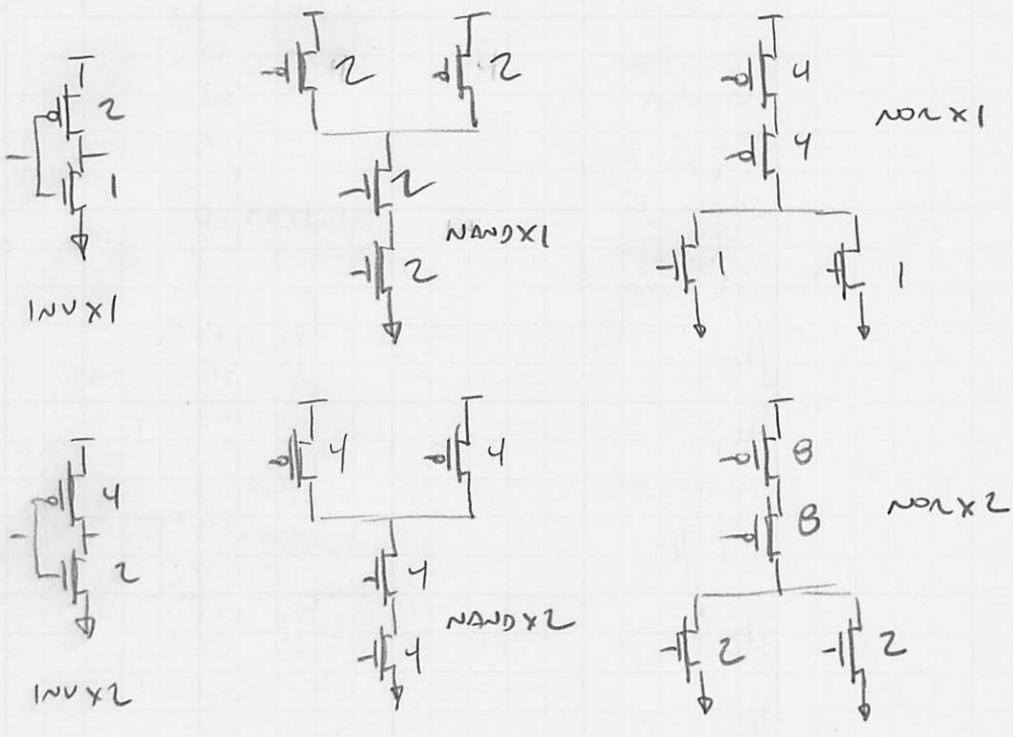
Close to given input cap of 10C

THIS ASSUMES WE CAN SIZE GATE ARBITRARILY. IN A FULL CUSTOM DESIGN. WHAT IF USING STANDARD CELL LIB?

ASSUME STANDARD CELL LIB WITH FOLLOWING GATES

INVX1,	INVX2	INVX3	INVX4	INVX5	INVX6
NANDX1	NANDX2	NANDX4			
NORX1	NORX2	NORX4			

WHAT DOES X1, X2, X4 MEAN? X2 MEANS TWICE THE DRIVE STRENGTH OF AN X1 INVERTER. SO NANDX2 MEANS $\alpha = 2$



ASSUME WE HAVE DETERMINED OPTIMAL SIZING IS C_{IN}, HOW DO WE FIGURE OUT WHICH CELL TO USE?

remember $g = \frac{R_t C_t}{R_{INV} C_{INV}}$ IF WE ASSUME $R_t = R_{INV}$

$$g = C_t / C_{INV} \longrightarrow C_t = g C_{INV}$$

AND $C_{IN} = \alpha C_t$

SO $C_{IN} = \alpha g C_{INV}$

$$\alpha = C_{IN} / g C_{INV} \quad \text{SINCE } C_{INV} = 3C$$

$$\boxed{\alpha = \frac{C_{IN}}{g\,3C}}$$

Given optimum C_{in} from Before, what is α?

C_{in}	g	α		gate?
11.56 C	1	$11.56/(1\cdot 3)$	= 3.85	INVx4
17.81 C	4/3	$17.81/(\frac{4}{3}\cdot 3)$	= 4.45	NANDx4
17.16 C	5/3	$17.16/(\frac{5}{3}\cdot 3)$	= 3.55	NORx4
9.92 C	1	$9.92/(1\cdot 3)$	= 3.3	INVx3

(must be $\leq 10C$ for input constraint)

Recalculate actual delay given these gates

First calculate actual C_{in} for each std cell gate

INVx4 $C_{in} = \alpha g 3C = 4\cdot 1\cdot 3C = 12C$

NORx4 $C_{in} = \alpha g 3C = 4\cdot \frac{4}{3} 3C = 16C$

NANDx4 $C_{in} = \alpha g 3C = 4\cdot \frac{5}{3} 3C = 20C$

INVx3 $C_{in} = \alpha g 3C = 3\cdot 1\cdot 3C = 9C$

Now use path delay equation

$$D = \Sigma gh + \Sigma p$$

$$= \left(1\cdot \frac{20}{12}\right) + \left(\frac{4}{3}\cdot \frac{12\times 2}{16}\right) + \left(\frac{5}{3}\cdot \frac{16}{20}\right) + \left(1\cdot \frac{20}{9}\right) + (1+2+2+1)$$

$$= 1.67 + 2 + 1.33 + 2.22 + 6$$

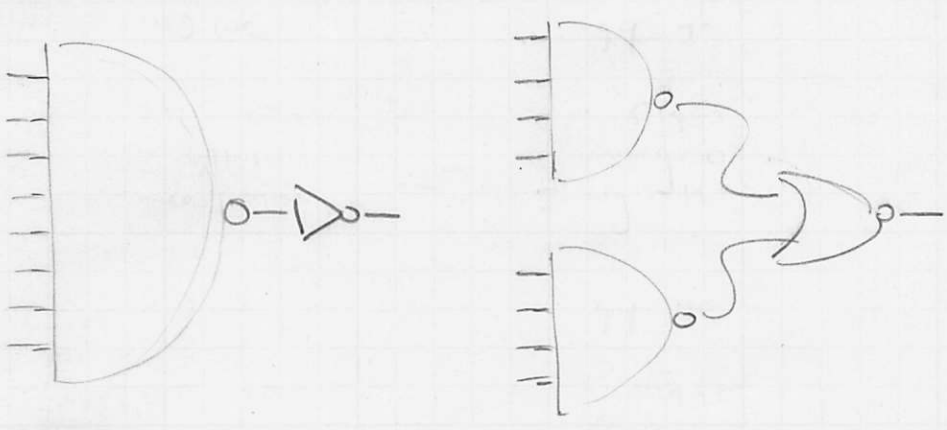
$$= 7.22 + 6$$

$$= 13.22$$

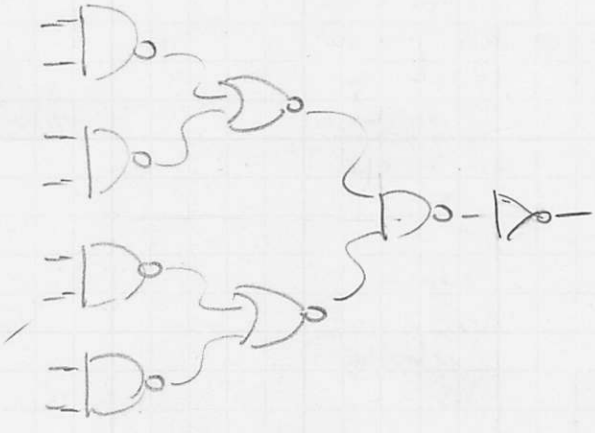
Compare with optimal which is 12.92 off by 2.3%

Q2: How should we change topology to minimize delay?

Assume we want to implement an eight input AND gate. Calculate minimum delay assuming optimal sizing for following three topologies assuming $H=1$ and $H=12$



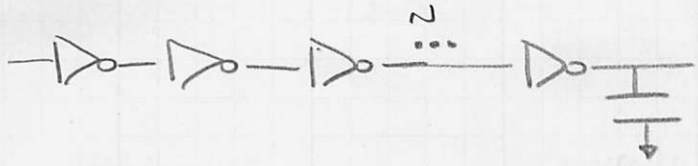
g	10/3	1		g	2	5/3
p	8	1		p	4	2



g	4/3	5/3	4/3	1
p	2	2	2	1

	H=1			H=12		
topology	$NF^{1/N}$	P	$\hat{D}$	$NF^{1/N}$	P	$\hat{D}$
8-NAND	3.65	9	12.65	12.64	9	21.64
2x 4-NAND	3.65	6	9.65	12.64	6	18.64
4x 2-NAND	5.25	7	12.25	9.77	7	16.77

Optimal Number of Stages using Inverters?



$$\hat{D} = N F^{\frac{1}{N}} + N P_{INV}$$

$$\frac{\partial \hat{D}}{\partial N} = F^{\frac{1}{N}} - F^{\frac{1}{N}} \ln\left(F^{\frac{1}{N}}\right) + P_{INV} = 0$$

If $P_{INV} = 0$

$$F^{\frac{1}{N}} - F^{\frac{1}{N}} \ln\left(F^{\frac{1}{N}}\right) = 0$$

$$\ln\left(F^{\frac{1}{N}}\right) = 1$$

$$F^{\frac{1}{N}} = e \quad \text{or in other words,} \quad \hat{f} = e$$

So if we assume $P_{INV} = 0$, optimal number of stages results in a stage effort of e for every stage. Since $g = 1$ for an inverter this means $h = 2.718$ for every stage.

If $P_{INV} = 1$ then we need to solve following nonlinear eq:

$$F^{\frac{1}{N}} - F^{\frac{1}{N}} \ln\left(F^{\frac{1}{N}}\right) + 1 = 0$$

Let $\rho = F^{\frac{1}{\hat{N}}}$ where $\hat{N}$ is optimal

$$1 + \rho(1 - \ln \rho) = 0$$

We can find numerically that $\rho \approx 3.59$

So optimal number of stages results in stage effort of 3.59 when we take into account parasitics. We can roughly approximate 3.59 to be 4

$$F^{\frac{1}{\hat{N}}} \approx 4$$

$$\log\left(F^{\frac{1}{\hat{N}}}\right) = \log(4)$$

$$\frac{1}{\hat{N}} \log(F) = \log(4)$$

$$\hat{N} = \frac{\log(F)}{\log(4)} = \log_4(F)$$

So optimal number of stages for inverter chain is roughly:

$$\hat{N} = \log_4(F)$$

Since $G=1$ and $B=1$ for inverter chain

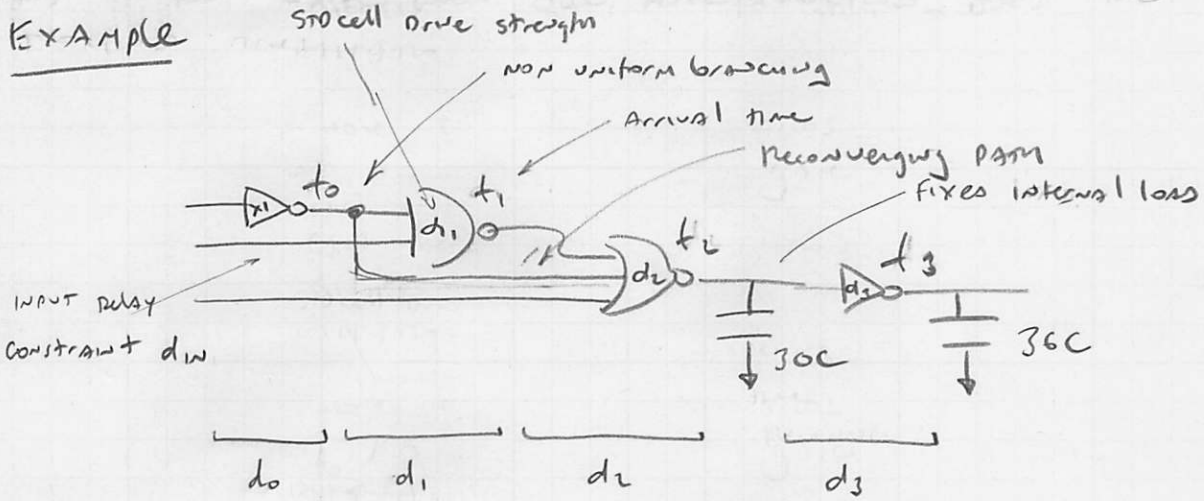
$$\hat{N} = \log_4(H)$$

Not too bad of an estimate even for realistic paths on gates that are not inverters

Logical effort can help give us intuition on how to size gates + choose a topology to minimize delay but it has many limitations.

To deal with more complicated scenarios we can also write the delay equations for each gate in system and minimize the latest arrival time.

EXAMPLE



Let's write our linear delay equation as a function of α

$$d = gh + p \qquad g = \frac{R_T C_T}{R_{INV} C_{INV}} \qquad C_{IN} = \alpha C_T \qquad C_T = \frac{C_{IN}}{\alpha}$$

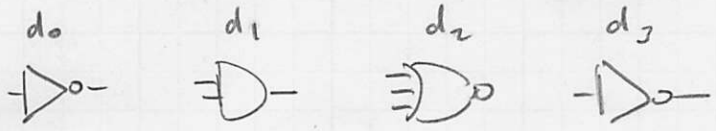
$$d = \frac{C_{IN}}{\alpha C_{INV}} \cdot \frac{C_{OUT}}{C_{IN}} + p \qquad g = \frac{C_{IN}}{\alpha C_{INV}}$$

$C_{IN} = 3C$

$$\boxed{d = \frac{C_{OUT}}{3\alpha} + p}$$ Delay as function of α (std cell drive)

Recall $\boxed{C_{IN} = 3g\alpha}$

Now write delay equations for each stage



α	1	α_1	α_2	α_3
g	1	4/3	7/3	1
$C_{IN} = 3g\alpha$	3	$4\alpha_1$	$7\alpha_2$	$3\alpha_3$
p	1	2	3	1

$$d_0 = \frac{(4\alpha_1 + 7\alpha_2)}{3 \cdot 1} + 1 = \frac{4}{3}\alpha_1 + \frac{7}{3}\alpha_2 + 1$$

$$d_1 = \frac{(7\alpha_2)}{3 \cdot \alpha_1} + 2 = \frac{7}{3}\frac{\alpha_2}{\alpha_1} + 2$$

$$d_2 = \frac{(30 + 3\alpha_3)}{3\alpha_2} + 3 = \frac{10}{\alpha_2} + \frac{\alpha_3}{\alpha_2} + 3$$

$$d_3 = \frac{(36)}{3\alpha_3} + 1 = \frac{12}{\alpha_3} + 1$$

Arrival Times

$$t_0 = d_0$$

$$t_1 = \max(t_0, d_{IN}) + d_1$$

$$t_2 = \max(t_0, t_1) + d_2$$

$$t_3 = t_2 + d_3$$

$$t_3 = \max(t_0, t_1) + d_2 + d_3$$

$$= \max(d_0, \max(d_0, d_{IN}) + d_1) + d_2 + d_3$$

$$t_3 = MAX(d_0, MAX(d_0, d_{IN}) + d_1) + d_2 + d_3$$

MINIMIZE t_3 SUBJECT TO ABOVE CONSTRAINTS WITH d_1 d_2 d_3 AS THE INDEPENDENT VARIABLES

ACTUALLY IN SYNTHESIS WE REALLY WANT TO MINIMIZE AREA (OR ENERGY) SUBJECT TO CONSTRAINT ON t_3.

SO WE COULD CRAFT OPTIMIZATION PROBLEM TO BE MINIMIZE SUM OF d_1 d_2 d_3 (PROXY FOR AREA) SUBJECT TO CONSTRAINT:

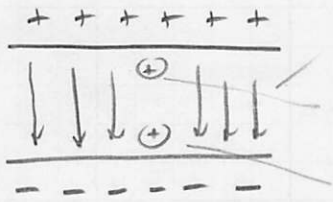
$$t_{\substack{CLK \\ PERIOD}} > MAX(d_0, MAX(d_0, d_{IN}) + d_{IN}) + d_2 + d_3$$

↳ CLOCK PERIOD CONSTRAINT

Energy

Energy is a measure of work

Power is the rate at which work is DONE



Electric pot energy — CAPACITY for DOING work which ARISES from position of charge IN E FIELD (JOULES)

Electric Potential — Electric pot energy of a positive per unit charge

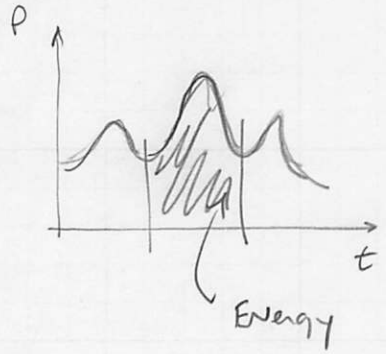
(VOLTS, $1V = 1J/C$, $\Delta V = \Delta E/Q$)

Current — RATE AT which charge flows past position

(AMPS, $1A = 1C/s$, $I = Q/\Delta t$)

Power — RATE AT which electric energy is supplies or consumed

(WATTS, $1W = 1J/s$, $P = \Delta E/\Delta t = \frac{\Delta V \cdot Q}{Q/I} = VI$)



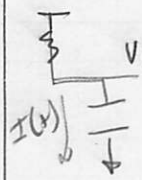
$$E = \int_0^T P(t)\,dt$$

ENERGY STORED ON CAP

Energy on capacitor

$$E_{Cap} = \int_0^{\infty} P(t)\,dt = \int_0^{\infty} V(t)\,I(t)\,dt$$

$$= \int_0^{\infty} V(t)\,\frac{dQ}{dt}\,dt = \int_0^{\infty} V(t)\,\frac{C\,dv}{dt}\,dt$$



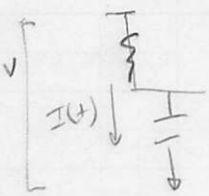
$C = \frac{dQ}{dv}$

$$= C\int_0^{V_{dd}} V(t)\,dV = \frac{1}{2} C V_{dd}^2$$

$\int_0^z x\,dv = \frac{1}{2}z^2$

So on $1 \rightarrow 0$ input transition $\frac{1}{2} C V_{dd}^2$ stores on capacitor + this energy is released on $0 \rightarrow 1$ transition

ENERGY DELIVERED FROM PWR SUPPLY



$$E_{Supply} = \int_0^{\infty} P(t)\,dt = \int_0^{\infty} V_{dd}\,I(t)\,dt$$

constant

$dQ = C\,dv$

$$= V_{dd}\int_0^{\infty} \frac{dQ}{dt}\,dt = V_{dd}\int_0^{\infty} C\,\frac{dv}{dt}\,dt$$

$$= C V_{dd}\int_0^{V_{dd}} dv = C V_{dd}^2$$

$\int_0^z dv = z$

half energy dissipated as heat in PMOS

other half dissipated as heat in NMOS

On avg each bit transition requires $\frac{1}{2} C V_{dd}^2$

- $1 \rightarrow 0$ uses $C V_{dd}^2$
- $0 \rightarrow 1$ uses ϕ

$E_{node} = \alpha\,\frac{1}{2} C V_{dd}^2$

α: probability of bit toggle

Power

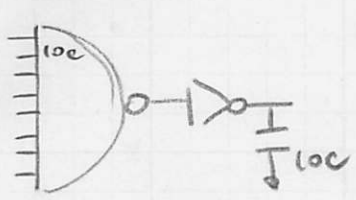
$$P_{TOT} = P_{SWITCHING} + P_{STATIC}$$

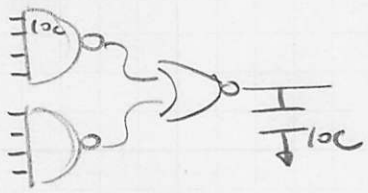
$$= \alpha f \frac{1}{2} C V_{dd}^2 + V_{dd} I_{off}$$

↳ # transitions, sometimes just # 1→0 transitions but then won't be factor of $\frac{1}{2}$

Compare Energy

Need to find total switched cap in worst case





g	10/3	1	2	5/3
p	8	1	4	2
G	10/3		10/3	
H	1		1	
B	1		1	
F	10/3		10/3	
$\hat{f}$	1.8		1.8	

$C_{INV,g} = \frac{1}{1.8} 10 = 5.6$

$C_{NAND,g} = \frac{10/3}{1.8} 5.6 = 10.4$

$C_{TOT,g} = C_{INV,g} + B \cdot C_{NAND,g}$

$= 88.8\ C$

$C_{NOR,g} = \frac{5/3}{1.8} 10 = 9.3$

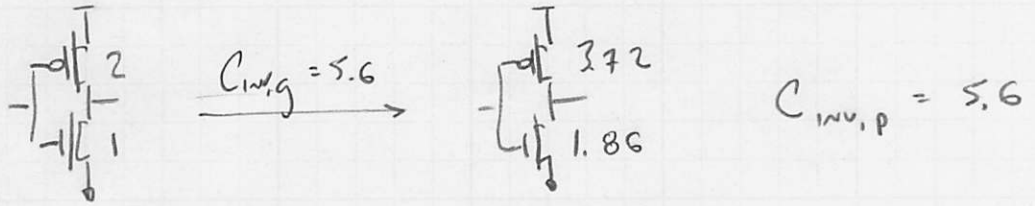
$C_{NAND,g} = \frac{2}{1.8} 9.3 = 10.3$

$C_{TOT,g} = 2 C_{NOR,g} + 8 C_{NAND,g}$

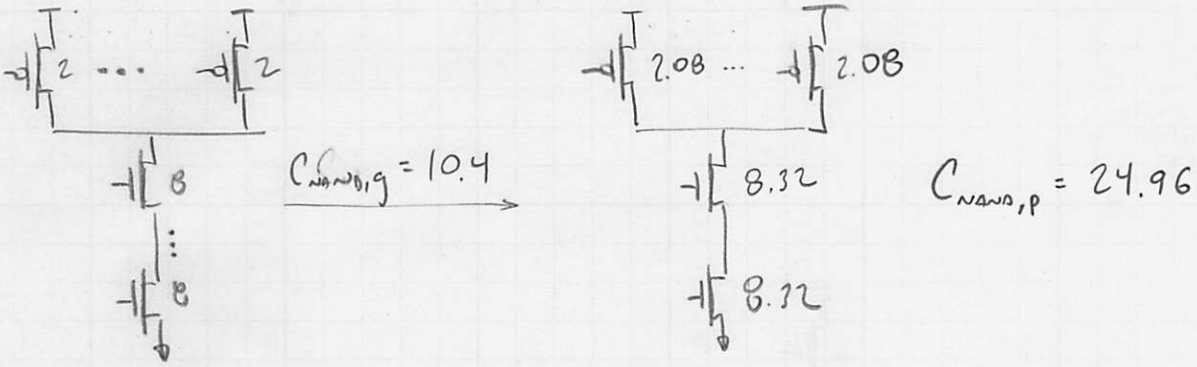
$= 101\ C$

To determine parasitic cap need to understand how gate cap is distributed across transistors.

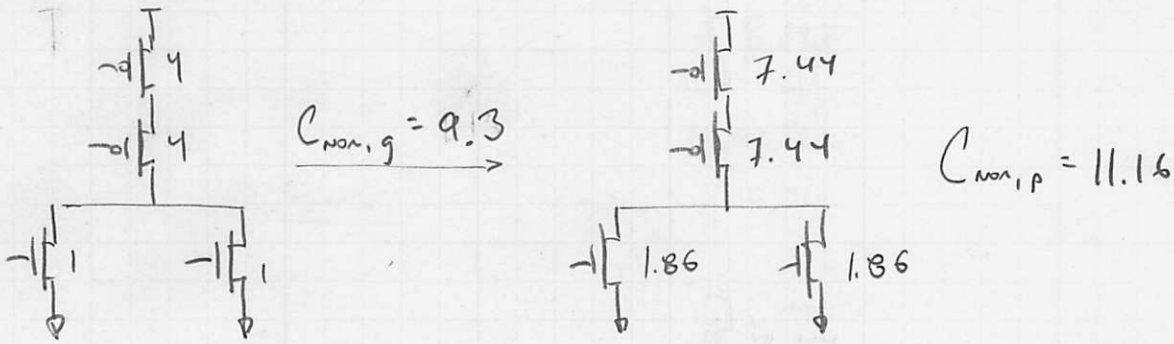
INVERTER



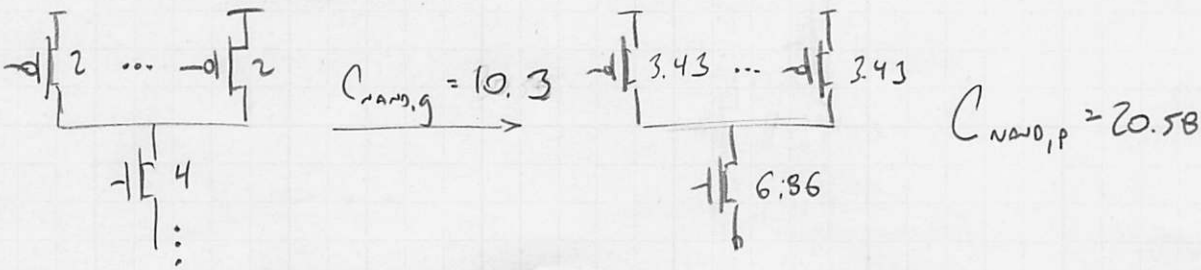
8 INPUT NAND GATE



2 INPUT NOR GATE



4 INPUT NAND GATE



$E = \alpha \frac{1}{2} C V_{dd}^2$, ASSUME $\alpha = 0.1$ AND $V_{dd} = 1V$ for both, so only difference is amount of switched capacitance

FOR 8-INPUT NAND TOPOLOGY

$$C_{tot} = C_{tot,g} + C_{tot,p} = 88.8 + (5.6 + 24.96) = 119.36\ C$$

FOR 4-INPUT NAND TOPOLOGY

$$C_{tot} = C_{tot,g} + C_{tot,p} = 101 + (11.16 + 20.58) = 132.74\ C$$

SO SECOND TOPOLOGY REQUIRES ~10% MORE ENERGY IN THE WORST CASE WHERE ALL CAPACITANCE IS SWITCHED. THIS IGNORES THE ENERGY REQUIRED FOR SWITCHING THE OUTPUT LOAD.

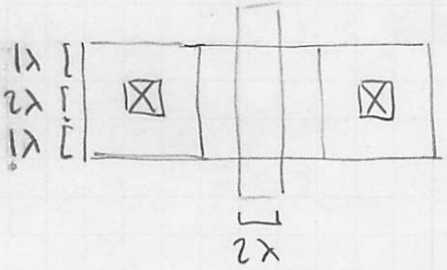
THESE ENERGY ESTIMATES ARE IN UNITS OF C (I.E. THE GATE CAP FOR A MINIMUM NMOS TRANSISTOR). WHAT IF WE WANT TO ESTIMATE THE ABSOLUTE ENERGY IN JOULES?

PAGE 312, TABLE 8.5 IN WESTE & HARRIS

C_g IN IBM 90nm $\approx$ 1-2 fF/μm $\quad V_{dd} = 1V$

ASSUME $W_{min} \approx 9\lambda$

$W = 4\lambda$ TOO SMALL NOWDAYS SO LET'S ASSUME 9λ



$\lambda = 45$ nm

$W_{min} = 9 \cdot 45 = 405$ nm $= 0.4\ \mu$m

SINCE $C_g \approx 1\text{-}2$ fF/μm AND $W_{min} \approx 0.4\ \mu$m EST $C = 0.5$ fF

FOR FIRST TOPOLOGY (ASSUME $\alpha = 0.1$)

$$E = \alpha \frac{1}{2} C V_{dd}^2 = \frac{1}{2}(120\ C)\ \frac{0.5\ fF}{C}\ (1V)^2 = 3\ fJ$$

$$P = \alpha f \frac{1}{2} C V_{dd}^2 = (0.5 \times 10^9)(30 \times 10^{-15}) = 1.5\ \mu W$$

f = 500 MHz

ACTIVITY FACTORS

- Previous example just used fixed $\alpha = 0.1$ for all nodes.
- More accurate to track activity factors thru topology

- Assume inputs have completely random data

	P
0 → 0	0.25
0 → 1	0.25
1 → 0	0.25
1 → 1	0.25

P_i = Prob node is one

$\bar{P}_i = 1 - P_i$ = Prob node is zero

$$\alpha = P_{i-1}\bar{P}_i + \bar{P}_{i-1}P_i$$

($P_{i-1}\bar{P}_i$: cycle i; $\bar{P}_{i-1}P_i$: cycle $i-1$)

for uncorrelated data

$$\alpha = P_i\bar{P}_i + \bar{P}_i P_i = 0.5$$

$$\alpha' = \bar{P}_i P_i = 0.25 \qquad \alpha' = \frac{1}{2}\alpha$$

α = bit transition

α' = node 0 → 1 (not input!)

$\alpha \, f \frac{1}{2} C V_{dd}^2$

$\alpha' f C V_{dd}^2$

NAND 2 $\quad \alpha'_{out} = \bar{P}_{out} P_{out} = (P_A P_B)(1 - P_A P_B)$

NAND 8 $\quad \alpha'_{out} = (P^8)(1 - P^8) = (0.0039)(0.996) \neq 0.0039$

Assume $P_A = P_B = \ldots = P$

NAND 2

$$\alpha_{out}' = \overline{P}_{out} P_{out} = (P_A P_B)(1 - P_A P_B) =$$

$\overline{P}_{out}$: probability output is zero; P_{out}: probability output is one

output of NAND 2 is zero if both inputs one
otherwise output of NAND 2 is one

→ Assume inputs are random data

$$P_A = 0.5 \quad P_B = 0.5$$

$$\alpha_{out}' = (P_A P_B)(1 - P_A P_B) = (0.5 \times 0.5)(1 - 0.5 \times 0.5)$$
$$= (0.25)(1 - 0.25) = 0.25 \times 0.75 = 0.6$$

NAND 8

$$\alpha_{out}' = \overline{P}_{out} P_{out} = \left(P_{in}^{8}\right)\left(1 - P_{in}^{8}\right)$$

output is only zero if all 8 inputs are one
otherwise output of NAND8 is one

$$\alpha_{out}' = \left(P_{in}^{8}\right)\left(1 - P_{in}^{8}\right) = \left(0.5^{8}\right)\left(1 - 0.5^{8}\right)$$
$$= (0.0039)(0.996) = 0.0039$$

During lecture today, I mentioned that _adding_ inverters can sometimes _reduce_ the path delay. This might seems counter intuitive based on what you learned in ECE 2300. In ECE 2300, gates had a _constant_ delay. So every inverter might always have a delay of 1 tau, and every NAND2 gate might always have a delay of 2 tau. In fact, we used a similar simplification when estimating the critical path in ECE 4750. If we assume a constant delay model, then adding a pair of inverters would indeed _always_ slow down the path delay. Adding a pair of inverters would simply increase the total propagation delay.

Based on what we have learned in ECE 5745 so far, it should be clear that the constant delay model is a significant oversimplication. The delay of a gate depends on many things including its size, the load capactance at the output, when inputs arrive, the rise/fall time of the inputs, layout details, etc. Our RC modeling and method of logical effort use a _linear_ delay model which is a little more reasonable than a _constant_ delay model (but of course is still a significant simplification!). So the delay of a gate is:

```
  d = gh + p
```

The logical effort (g) and the parasitic delay (p) depend only on the template, while the electrical effort (h) depends on both the size of the gate (Cin) and the load capacitance at the output (Cout).

Let's look in more detail at the example we were discussing in lecture to demonstrate how _adding_ intervers can sometimes _reduce_ the path delay. Assume after synthesis we have the following two-gate path:

```
 --|NAND
   |NAND---I>o----.---
 --|NAND          |
 4C              --- 1000C
  NAND2X1 INVX4  ---
                  |
                  V
```

So we have a X1 two-input NAND gate (NAND2X1) and a X4 inverter driving a load of 1000C. The synthesis tool optimized the design assuming the inverter was driving a modest load, but after place-and-route, it turned out that the inverter has to drive a cross-chip global wire and thus a very large fixed capacitance.

What is the delay of this two-gate path?

```
  D = ( g0*h0 + g1*h1 ) + ( p0 + p1 )
    = ( 4/3 * 12/4 + 1 * 1000/12 ) + ( 2 + 1 )
    = ( 4 + 83.3 ) + 3
    = 90.3 tau
```

Recall that the minimum delay will occur when the stage effort is equal across all stages. Notice that the stage effort of the two stages is not even close to being equal which suggests this sizing is suboptimal.

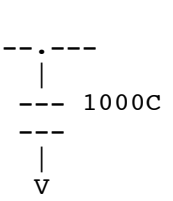
The place-and-route tool can potentially reduce the path delay using "buffer resizing". So let's assume the tool wants to increase the size of the inverter. Let's use logical effort to figure out the optimal sizing.

```
  F  = GHB = 4/3 * 1000/4 * 1 = 333
  f' = F^(1/N) = (333)^(1/2) = 18.25
  D' = N*F^(1/N) + P = 2*18.25 + ( 2 + 1 ) = 39.5 tau
```

Note that I am using f' instead of f "hat" and D' instead of D "hat". The path delay is significatly lower if we can resize the inverter. Let's figure out how large the final inverter needs to be to achieve this optimal delay.

```
  C_in,1 = (g/f') * C_load = (1/18.25) * 1000 = 54.8C
```

That is a pretty big inverter! The inverter's NMOS would be 18.27 times the minimum width and the inverter's PMOS would be 36.53 times the minimum width. Assume our standard cell library has an INVX1, INVX2, INVX4, INVX8, INVX16, INVX32, and INVX64. Let's choose the INVX16 for the final inverter (which is a little smaller than the optimal full-custom sizing).



What is the new delay of the new path?

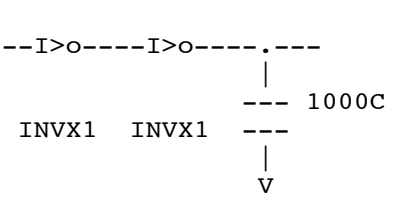
```
  D = ( g0*h0 + g1*h1 ) + ( p0 + p1 )
    = ( 4/3 * 48/4 + 1 * 1000/48 ) + ( 2 + 1 )
    = ( 16 + 20.8 ) + 3
    = 39.8 tau
```

The delay using the standard cell is a little slower and the stage effort is not exactly balanced, but buffer resizing does still significantly reduce the path delay.

The place-and-route tool can potentially further reduce the path delay using "buffer insertion". Let's quickly estimate the optimal number of stages.

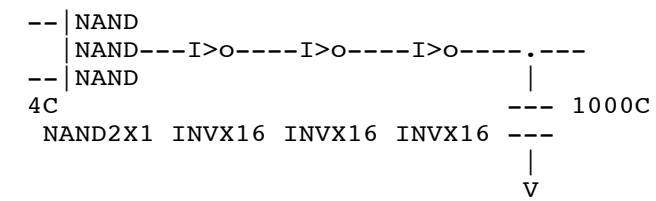
```
  log_4(F) = log_4(333) = 4.2
```

So the rough estimate of the optimal number of stages if 4, but we are only using two stages. Let's add two INVX1 gates at the end of the path to see if that helps.




```
  D = ( g0*h0 + g1*h1 + g2*h2 + g3*h3 ) + ( p0 + p1 + p2 + p3 )
    = ( 4/3 * 48/4 + 1 * 3/48 + 1 * 3/3 + 1 * 1000/3 ) + ( 2 + 1 + 1 + 1 )
    = ( 16 + 0.0625 + 1 + 333.3 ) + 5
    = 355 tau
```

Yeow -- this is a bad idea. The delay is 9x worse! Instead of driving the large load capacitance with an INVX16, now we are driving this large load capacitance with an INVX1. Very bad idea. What if we add two more INVX16 gates at the end of the path?




```
  D = ( g0*h0 + g1*h1 + g2*h2 + g3*h3 ) + ( p0 + p1 + p2 + p3 )
    = ( 4/3 * 48/4 + 1 * 48/48 + 1 * 48/48 + 1 * 1000/48 ) + ( 2 + 1 + 1 + 1
)
    = ( 16 + 1 + 1 + 20.8 ) + 5
    = 43.8 tau
```

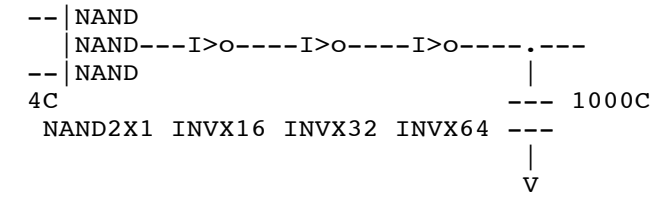
This is better than the original design, but slower than the optimized two-gate design with buffer resizing. The key is that we don't want to add more inverters. We want to add more inverters and then properly resize the gates to ensure we are balancing the stage efforts appropriately. We can just use the method of logical effort to find the optimal delay and the optimal sizing.

```
  F  = GHB = 4/3 * 1000/4 * 1 = 333
  f' = F^(1/N) = (333)^(1/4) = 4.27
  D' = N*F^(1/N) + P = 4*4.27 + ( 2 + 1 + 1 + 1 ) = 22.1 tau
```

So we have further reduced the delay by using a combination of buffer insertion and buffer resizing. Let's figure out how large each inverter needs to be to achieve this optimal delay.

```
  C_in,3 = (g/f') * C_load = (1/4.27) * 1000 = 234C
  C_in,2 = (g/f') * C_in,3 = (1/4.27) * 234  = 55C
  C_in,1 = (g/f') * C_in,2 = (1/4.27) * 55   = 13C
```

Yeow -- that is a big final inverter! Our INVX64 has a C_in of 192C so that will be the best we can do. Let's size our inverters as follows:



And now let's calculate the delay again:

```
  D = ( g0*h0 + g1*h1 + g2*h2 + g3*h3 ) + ( p0 + p1 + p2 + p3 )
    = ( 4/3 * 48/4 + 1 * 96/48 + 1 * 192/96 + 1 * 1000/192 ) + ( 2 + 1 + 1 +
1 )
    = ( 16 + 2 + 2 + 5.2 ) + 5
    = 27.2 tau
```

The original design without buffer insertion/resizing had a delay of 90.3 tau while the new design with buffer insertion/resizing has a delay of only 27.2, and improvement of 3.3x! So clearly _adding_ inverters can _reduce_ the path delay, but (as we saw with some of our counter-examples) this is only true if you properly size the gates!

It is useful in our pen-and-paper analysis to have a good estimate for some of the technology parameters in our target process. For example, when estimating the power and energy of a circuit, we need to know the supply voltage and gate capacitance (since all of our switched capacitance estimtes will be in units of C). West & Harris provides a methodology for using SPICE simulations to estimate various technology parameters. We can also look in the .lib file for our 45nm standard cell library, since this file was itself generated from many SPICE simulations.

The following snippet shows the entry in the .lib file for our cannonical inverter (INV_X1). We can see that the nominal supply voltages is 1.1V and that the total input gate capacitance for this inverter is estimated to be 1.7fF.

```
  library (NangateOpenCellLibrary) {
    ...

    /* Units Attributes */
    time_unit                : "1ns";
    leakage_power_unit       : "1nW";
    voltage_unit             : "1V";
    current_unit             : "1mA";
    pulling_resistance_unit  : "1kohm";
    capacitive_load_unit       (1,ff);

    /* Op Conditions */
    nom_process              : 1.00;
    nom_temperature          : 25.00;
    nom_voltage              : 1.10;

    /*****************************************************************
     Module     : INV_X1
     Cell Descr : Combinational cell (INV_X1) with drive strength X1
    *****************************************************************/

    cell (INV_X1) {
      drive_strength         : 1;
      area                   : 0.532000;

      ...
      pin (A) {
        direction            : input;
        related_power_pin    : "VDD";
        related_ground_pin   : "VSS";
        capacitance          : 1.700230;
        fall_capacitance     : 1.549360;
        rise_capacitance     : 1.700230;
      }
      ...
```

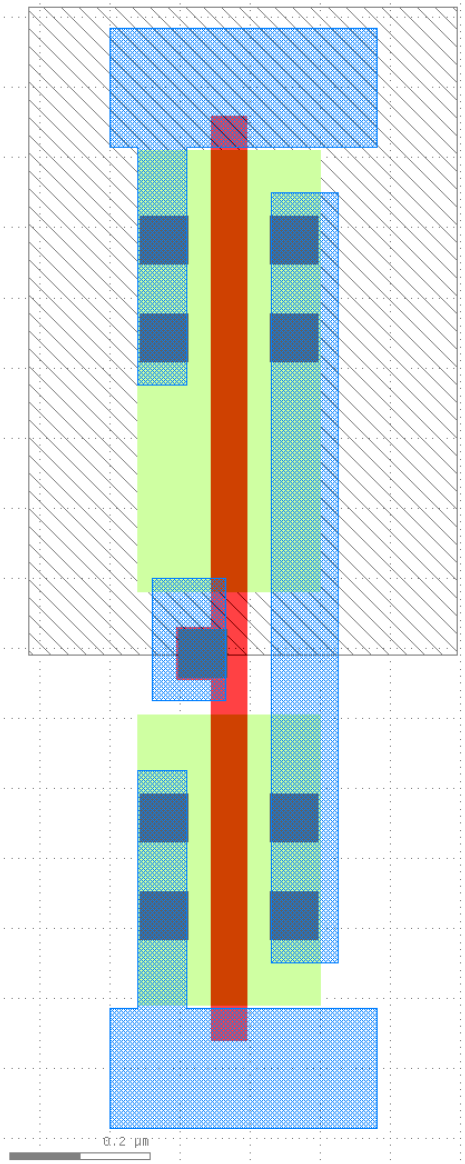
However, we need an estimate for C which is the gate cap for the NMOS in this cannonical inverter. We can figure out C if we take a closer look at the layout and SPICE deck for this inverter. The following page shows the layout for an INV_X1, INV_X2, and INV_X4 gate along with the corresponding SPICE deck for an INV_X1 and INV_X2 gate.

Notice how the layout uses multiple parallel "fingers" to implement a single larger "logical" transistor. So an X2 gate has two fingers and an X4 has four fingers. The SPICE deck has the exact length and width of each transistors (we could also just measure the layout). Notice that both transistors have a width of 50nm even though this is a 45nm process! It is not unsual for standard-cell libraries to use slightly longer than

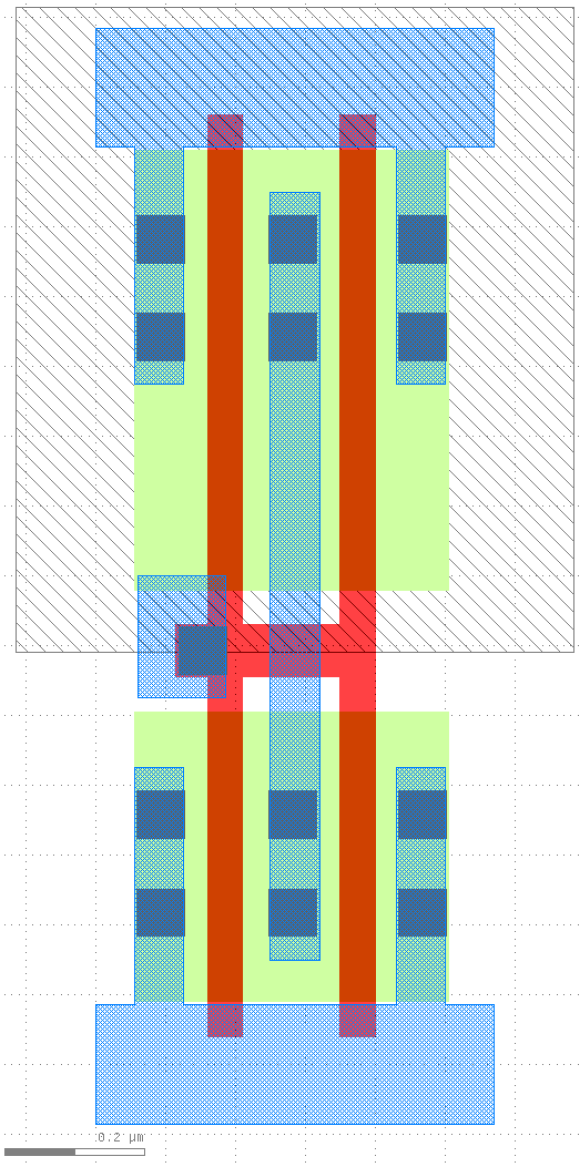
minimum transistors, since this geometry offers a nice compromise between performance and power consumption. The PMOS width is 630nm and the NMOS width is 415nm. Notice that the PMOS is definitely not twice the width of the NMOS (it is only 630/415 = 1.5x). This is probably because the mobility of an NMOS transistor is not exactly 2x the mobility of a PMOS transistor, and also because the standard-cell library is choosing to offer slightly unequal rise/fall times to offer reduced energy and area. Also notice that the NMOS in this inverter is 415/45 is about 9x the technology node size. This is ratio is a very reasonable size.

With this information we can now estimate C. We know the total gate cap for an INV_X1 gate is 1.7fF, and we know the ratio of how much of that gate cap comes from the NMOS is 415/(415+630) = 0.4. So C is 0.4 * 1.7fF = 0.68fF. To make our analysis simpler we will just roughly estimate the supply voltage as 1V and C as 0.5fF.

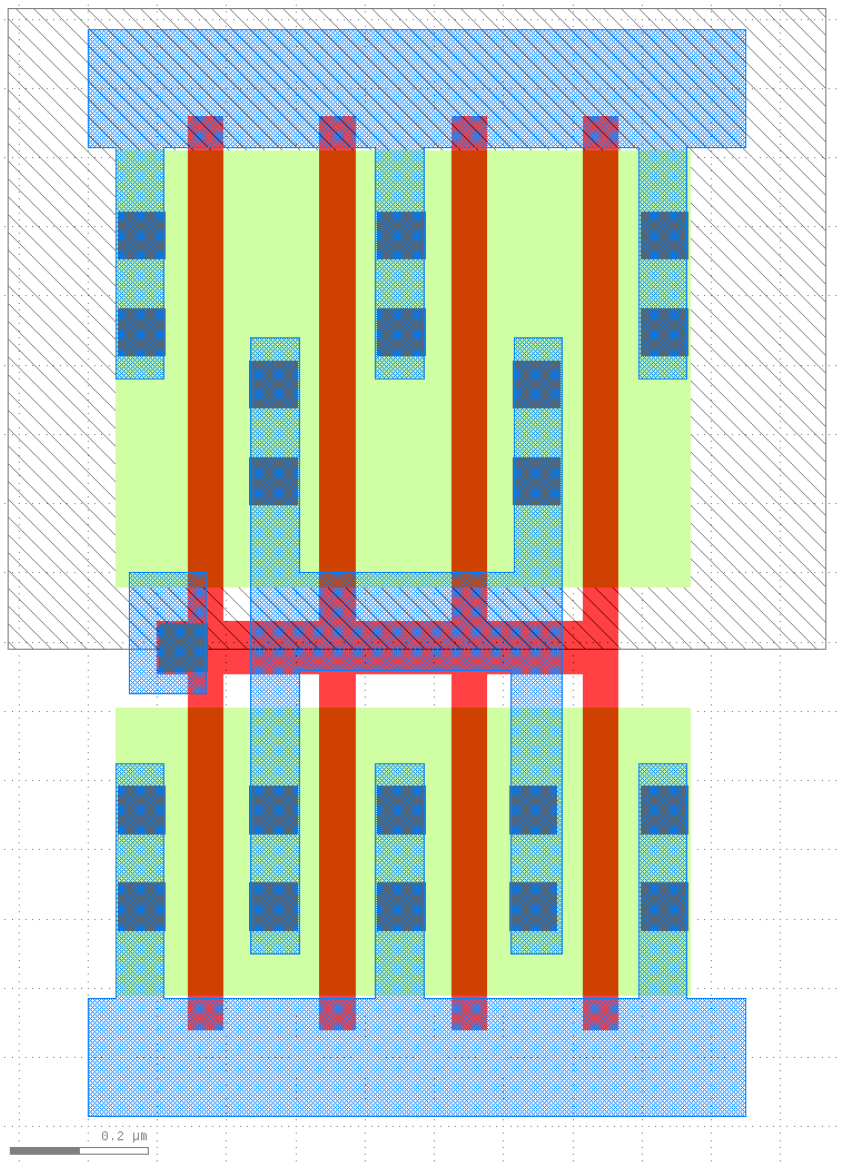
INV_X1



INV_X2



INV_X4



```
.SUBCKT INV_X1 A ZN VDD VSS
*.PININFO A:I ZN:O VDD:P VSS:G
*.EQN ZN=!A
M_i_0 ZN A VSS VSS NMOS_VTL W=0.415U L=0.050U
M_i_1 ZN A VDD VDD PMOS_VTL W=0.630U L=0.050U
.ENDS
```

```
.SUBCKT INV_X2 A ZN VDD VSS
*.PININFO A:I ZN:O VDD:P VSS:G
*.EQN ZN=!A
M_i_0_0_x2_0 ZN A VSS VSS NMOS_VTL W=0.415U L=0.050U
M_i_0_0_x2_1 VSS A ZN VSS NMOS_VTL W=0.415U L=0.050U
M_i_1_0_x2_0 ZN A VDD VDD PMOS_VTL W=0.630U L=0.050U
M_i_1_0_x2_1 VDD A ZN VDD PMOS_VTL W=0.630U L=0.050U
.ENDS
```