Problem #2

a. Find the inverter size that will give equal rise/fall delay when driving a 1X load for the specified technology/Vdd.

b. For a string of 9 inverters driving NO LOAD, compare via simulation the average delay using the inverter sized in part (a) and to the delay of a string of inverters whose size is the square root of the size found in (a).

Inverter Variations

Vdd = 2.5V	Leda 0.25u	
	Tphl(ps)	Tplh (ps)
3/1 inverter	65.5	56.2
2/1 Inverter	59	59.6
1.4/1 inverter	56	58.5
1.0/1 inverter	55	60.3

2/1 inverter had approximately equal rise/fall delays

Inverter String Delay

Inverter Skew	Avg Delay(ns)	Stages	Dly/stage	%diff to 2/1	Cap(fF)	%diff to 2/1
Size 3/1	0.717	11	0.07	5.6%	88	27.5%
Size 2/1	0.679	11	0.06	0.0%	69	0.0%
Size 1.4/1	0.669	11	0.06	-1.5%	58	-15.9%
Size 1/1	0.68	11	0.06	0.1%	49	-29.0%

Used 11 stages.

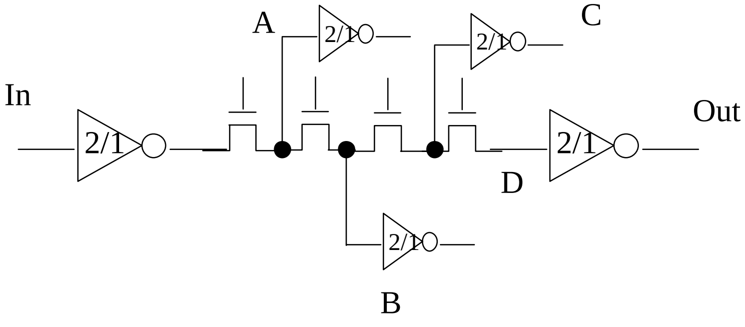
Skew did not make a significant different in terms of delay.

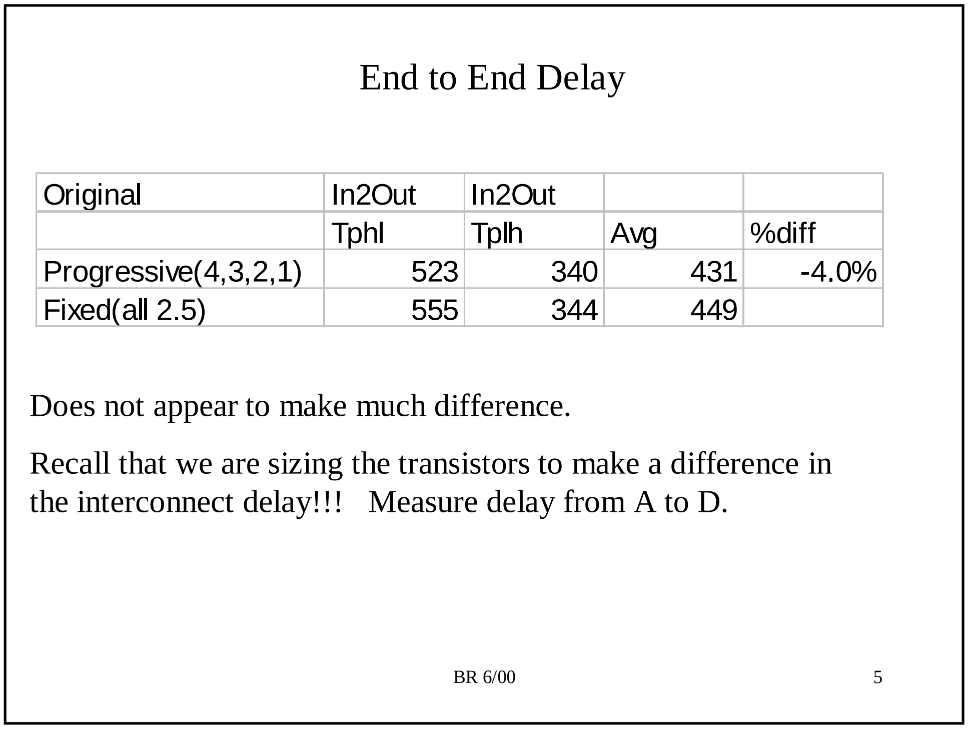
Makes a big difference in terms of power.

Conclusion: Use skewed gates to save power, sacrifice noise margin at one of the rails.

Problem #3

For the problem below, determine via simulation if progressive sizing or uniform sizing of pass transistors yields minimum delay. Note that last inverter has no load. Limit NMOS pass size to 4x size.



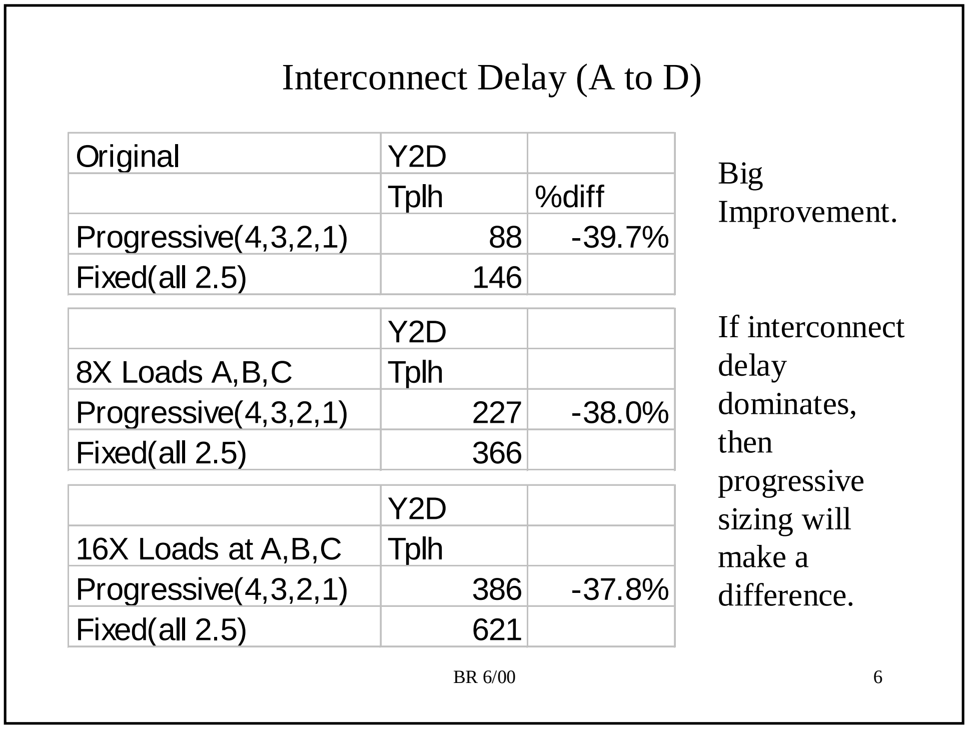


End to End Delay

Original	In2Out	In2Out		
	Tphl	Tplh	Avg	%diff
Progressive(4,3,2,1)	523	340	431	-4.0%
Fixed(all 2.5)	555	344	449	

Does not appear to make much difference.

Recall that we are sizing the transistors to make a difference in the interconnect delay!!! Measure delay from A to D.



Interconnect Delay (A to D)

Original	Y2D	
	Tplh	%diff
Progressive(4,3,2,1)	88	-39.7%
Fixed(all 2.5)	146	

Big Improvement.

	Y2D	
8X Loads A,B,C	Tplh	
Progressive(4,3,2,1)	227	-38.0%
Fixed(all 2.5)	366	

	Y2D	
16X Loads at A,B,C	Tplh	
Progressive(4,3,2,1)	386	-37.8%
Fixed(all 2.5)	621	

If interconnect delay dominates, then progressive sizing will make a difference.

Problem #4

Implement the function:

F = (A and not(B) and not(C)) or (not(A) and C and D)

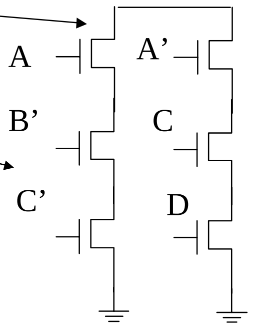
in DCVSL. Use the minimum number of transistors.

Assume A,B,C,D and their complements are available.

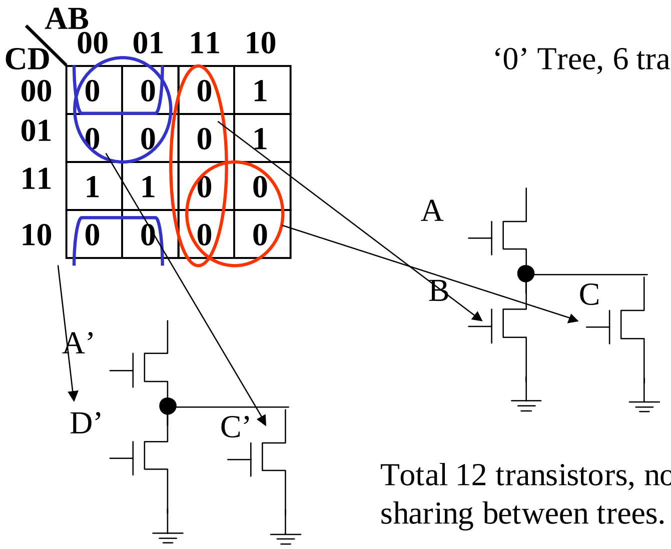
AB’C’ + A’CD

CD \ AB	00	01	11	10
00	0	0	0	1
01	0	0	0	1
11	1	1	0	0
10	0	0	0	0

‘1’ Tree, 6 transistors



AB’C’ + A’CD



CD \ AB	00	01	11	10
00	0	0	0	1
01	0	0	0	1
11	1	1	0	0
10	0	0	0	0

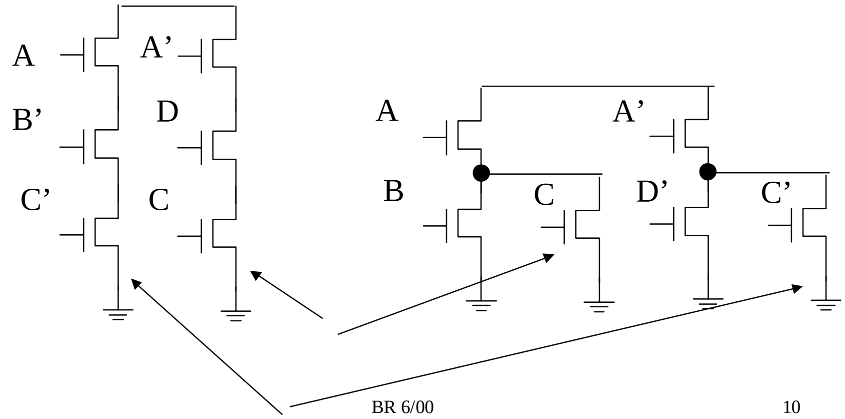
‘0’ Tree, 6 transistors

Total 12 transistors, no sharing between trees.

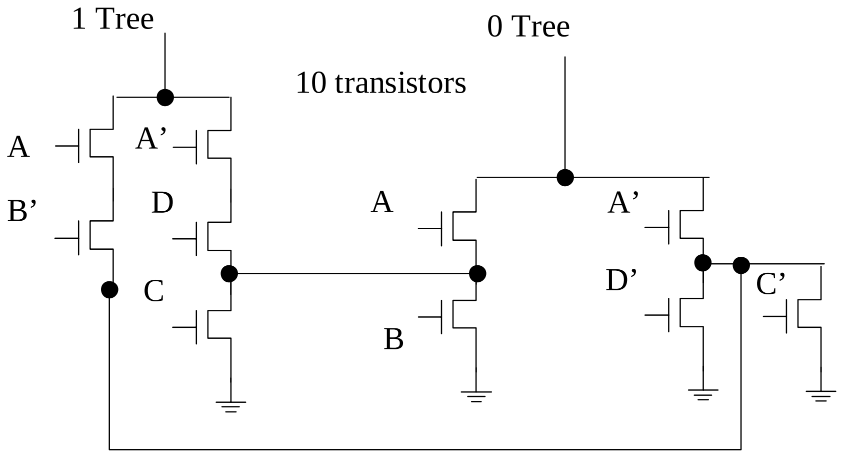
Share Between Trees

1 Tree

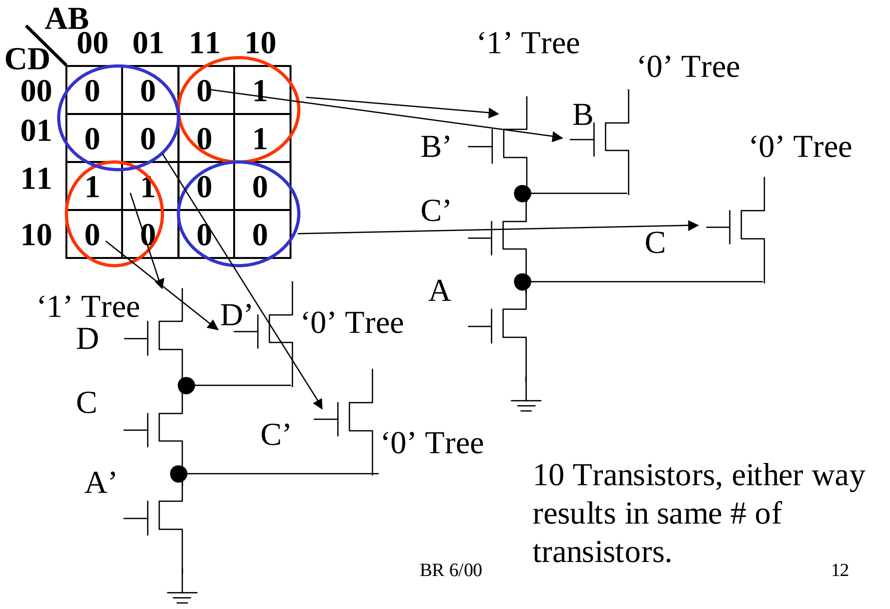
0 Tree



Share Between Trees



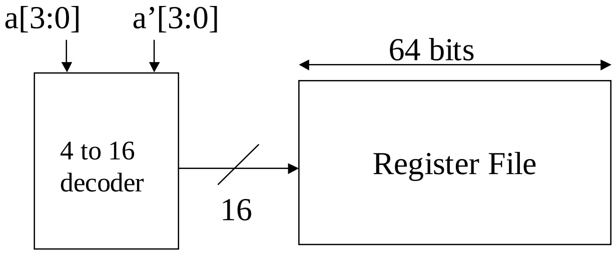
AB’C’ + A’CD



CD \ AB	00	01	11	10
00	0	0	0	1
01	0	0	0	1
11	1	1	0	0
10	0	0	0	0

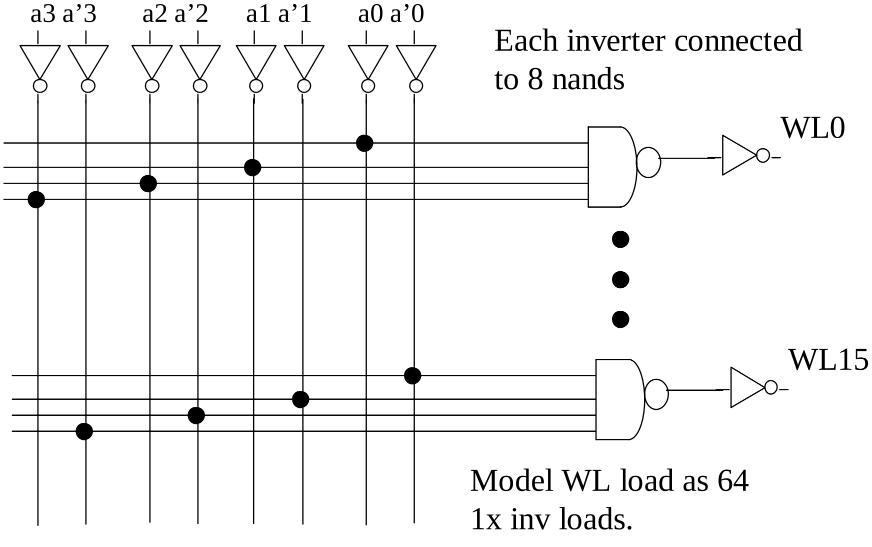
10 Transistors, either way results in same # of transistors.

Problem #1: Sizing Problem



16 x 64 register file

Decoder



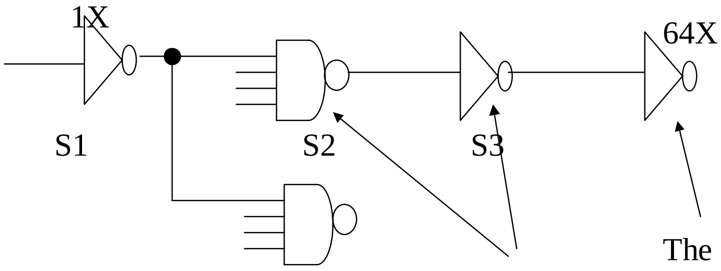
Each inverter connected to 8 nands

Model WL load as 64 1x inv loads.

Problem Statement

- Using the logical effort model, determine ‘p’, tau for a 4-input nand using Leda 0.25u process, Vdd = 2.5 v.
- Size the inverters, nand gate in the previous slide using the logical effort model.
- Predict the delay, and then compare simulated delay to predicted delay.

The Problem



1X

64X

S1

S2

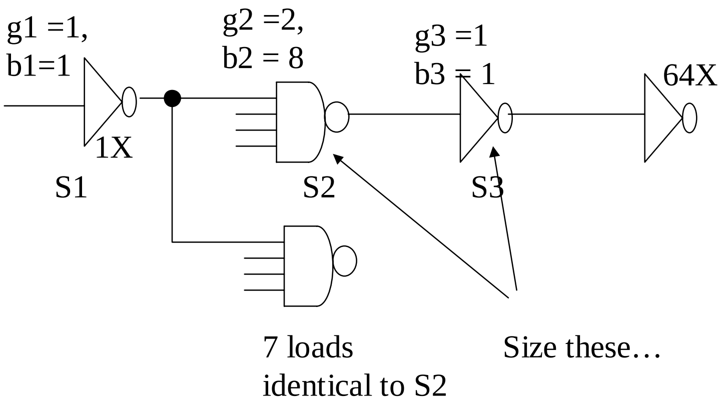
S3

The Load.

7 loads identical to S2

Size these…

The Problem



g1 =1,
b1=1

g2 =2,
b2 = 8

g3 =1
b3 = 1

64X

1X

S1

S2

S3

7 loads
identical to S2

Size these…

Path Logical effort G = g1*g2*g3 = 2.
Path Electrical Effort H = Cout/Cin = 64/1 = 64.
Path Branch Effort B = b1*b2*b3 = 8

Sizing

Path Effort F = G*H*B = 2 * 64 * 8 = 1024.

Min Delay when stage effort = Fmin = $(F)^{1/N}$

N =3,   so Fmin = $(1023)^{1/3}$ = 10.1

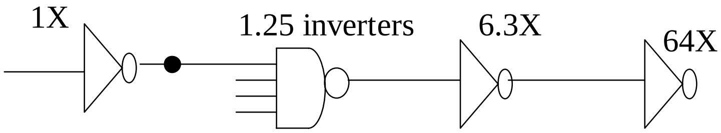
For S3, want   Fmin = b3*g3*h3 = 1 * 1 * 64/Cin
so   Cin(s3) =   64/10.1 = 6.3

Cin(s2) = (6.3 * 2 * 1)/10.1 = 1.25

Check Cin(s1), should be 1.

Cin(s1) = (1.25 * 1 * 8)/10.1 =   1

Final Sizes



How to size 4-input Nand?

Reference inverter was a 2/1 inverter. Total Cap load of 2/1 inverter is P+N = 2 + 1 = 3.

Want (Nand Load)/ (1X inv) = 1.25
(P + N) / (3) = 1.25.

Leave P transistor of Nand4 at ‘2’. Size N.
(2 + N) /(3) = 1.25
N = 1.75 (of course, this is 1.75 * Lmin)

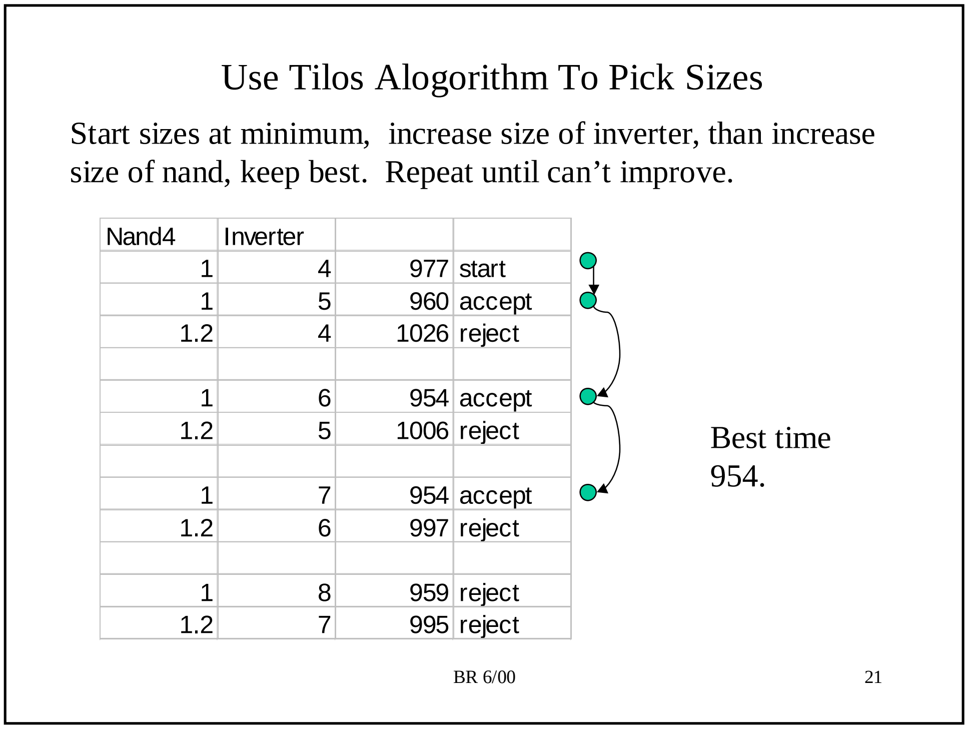
Measured vs. Predicted

	Stg 1	Stg2	Stg3	Total
Predicted	140	278	140	558
Measured	141	325	373	839

Times in ps, used Tau = 8.4, Pinv = 6.6, Pnand4 = 23.

Agreement:
1st stage fine, 2nd stage ok, 3rd stage lousy

But was the sizes a good pick???? Could we do better???

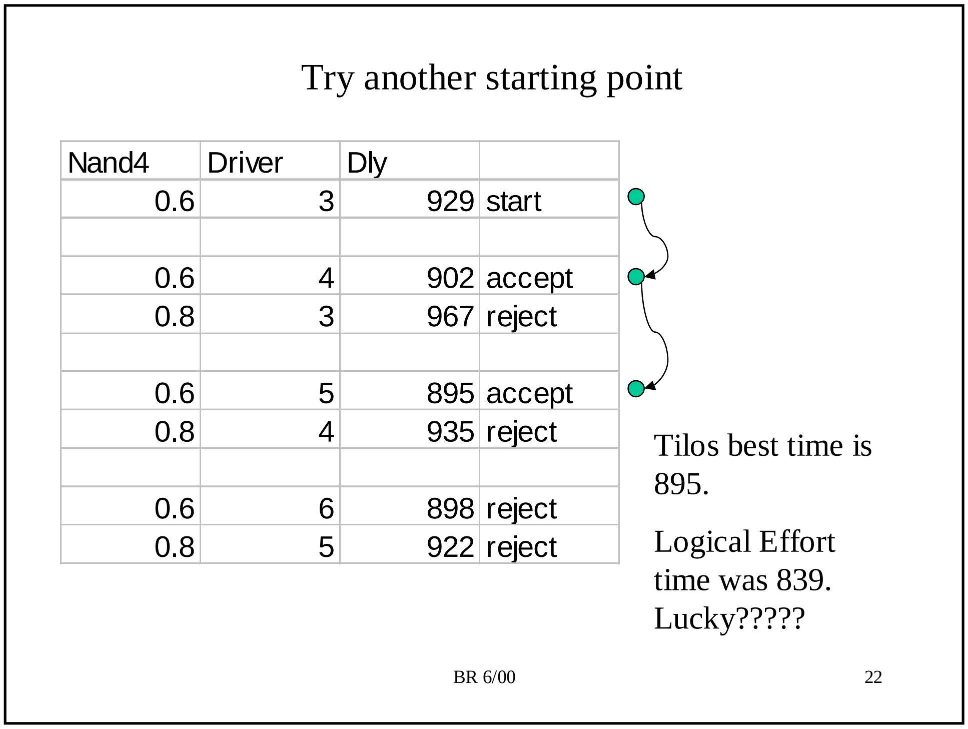


Use Tilos Alogorithm To Pick Sizes

Start sizes at minimum, increase size of inverter, than increase size of nand, keep best. Repeat until can't improve.

Nand4	Inverter		
1	4	977	start
1	5	960	accept
1.2	4	1026	reject
1	6	954	accept
1.2	5	1006	reject
1	7	954	accept
1.2	6	997	reject
1	8	959	reject
1.2	7	995	reject

Best time 954.



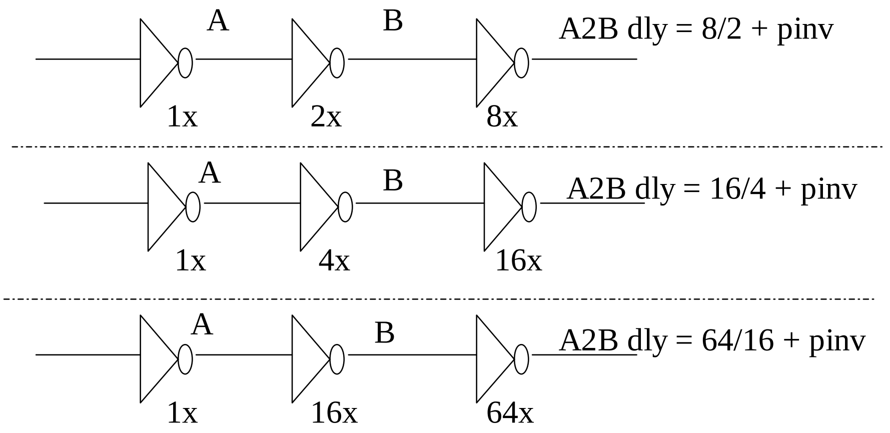
Try another starting point

Nand4	Driver	Dly	
0.6	3	929	start
0.6	4	902	accept
0.8	3	967	reject
0.6	5	895	accept
0.8	4	935	reject
0.6	6	898	reject
0.8	5	922	reject

Tilos best time is 895.

Logical Effort time was 839. Lucky?????

Why is Logical Effort lousy at Absolute Times?



A2B dly = 8/2 + pinv

A2B dly = 16/4 + pinv

A2B dly = 64/16 + pinv

Logical Effort model says A2B delay same in all cases. We know this is not the case (why??)

Logical Effort

- In this case produced sizes that resulted in low path delay, even if absolute delay prediction was wrong.
- Need more experience with this model, especially at submicron, before judging it further.