

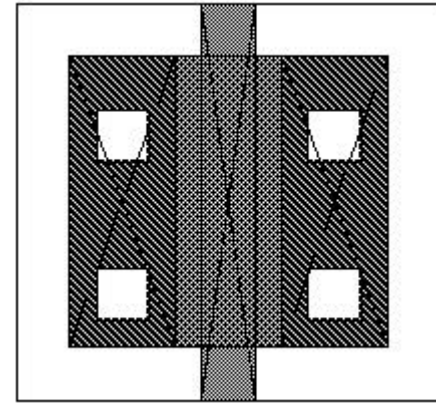
EE224

Class 9

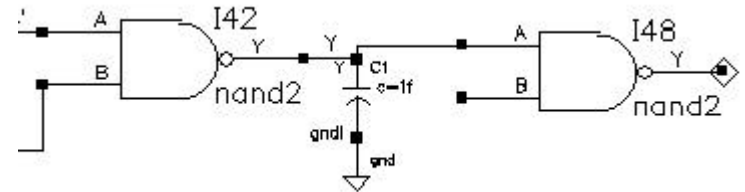
D. W Parent

Designers have many choices to make each of which affect the ultimate speed of a circuit.

Width of the transistors

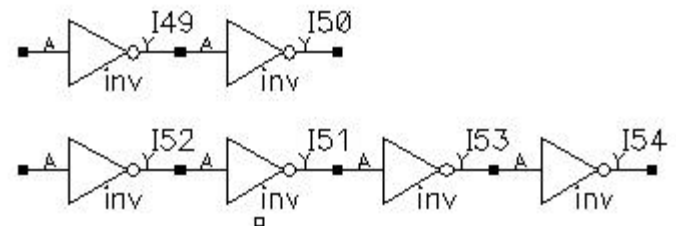


Wire capacitance between gates



The number of stages in a logic path

Which one?



There is a chick and the egg problem.

In order to know the width of the transistors  
I need to know Cinterconnect.

In order to know Cinterconnect I need to know the area of the circuits

In order to know the area I need to know the width.

I also need to decide which architecture to use.

I want to avoid doing several full blown designs. I want to evaluate which circuit and architecture is best to use before I design it.

Logical effort can be used to make these decisions.

Easy to evaluate one topology over another.

The excel sheet and logic effort are really the same thing.  
(with a few minor exceptions)

# Logic effort is based on the linear delay model

$$d = f + p$$

d is the the delay.

f is the effort delay or stage effort.

p is the parasitic delay if no load is attached.

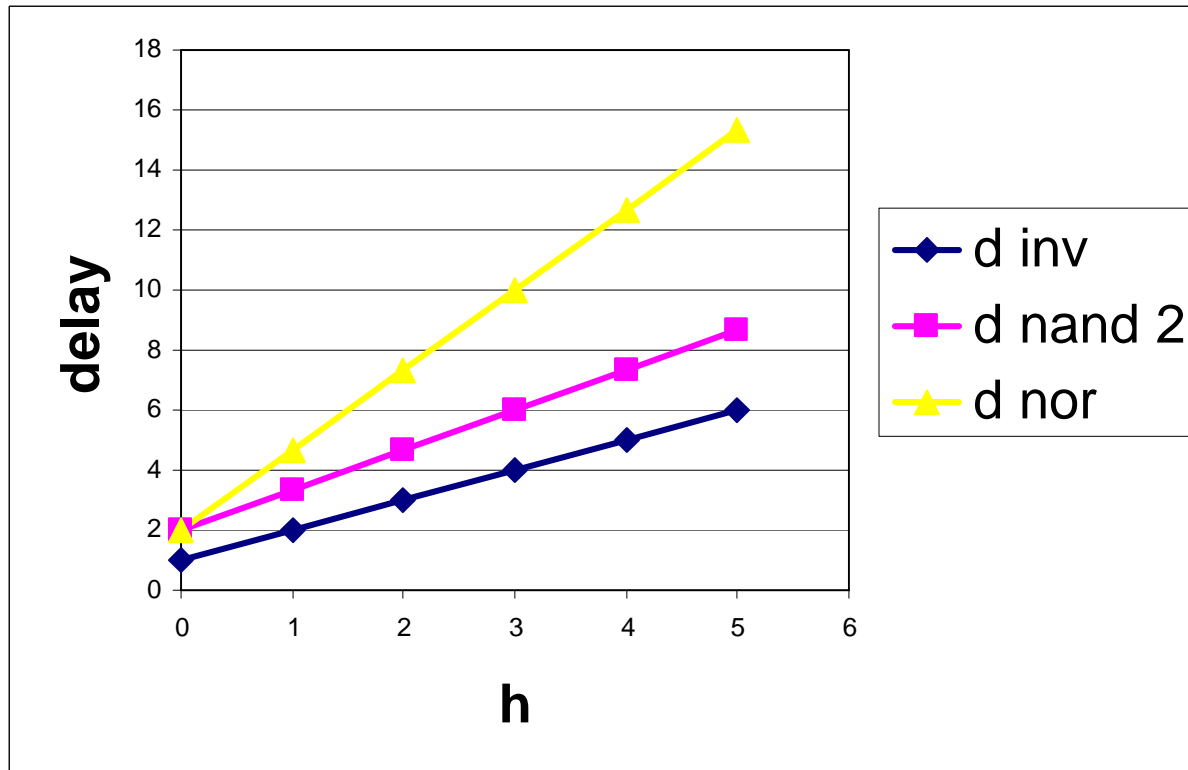
$$f = g \cdot h$$

g is the logical effort and for an inverter is defined to be 1.

h is the fanout or electrical effort.

$$h = \frac{C_{out}}{C_{in}} \quad h = \frac{C_{gload}}{C_{gdriver}}$$

Even though the nor and the nand have the same parasitic delay, due to lower hole mobilities the nor delay has a sharper slope



logical effort  
g

Gate Type	1	2	3	4n	
inverter	1				
NAND		4/3	5/3	6/3	$(n+2)/3$
NOR		5/3	7/3	9/3	$(2n+1)/3$
TriState-Mux		2	2	2	2
XOR		4	6	8	

parasitic delay  
p

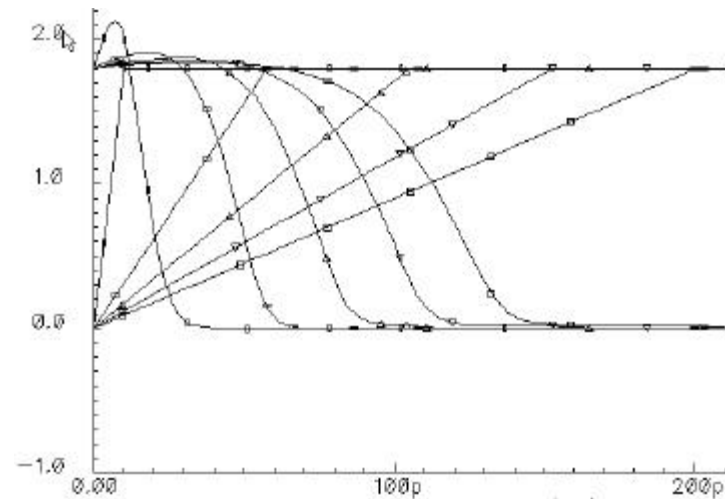
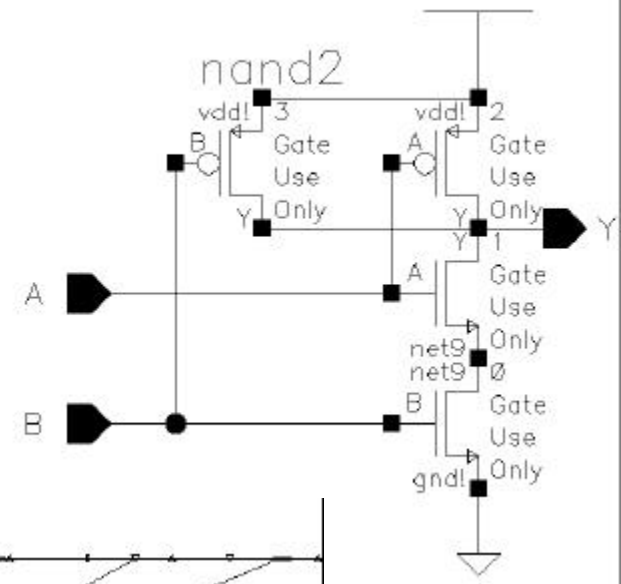
Gate Type	1	2	3	4n
inverter	1			
NAND		2	3	4n
NOR		2	3	4n
TriState-Mux		2	3	4n
XOR	2	4	6	8n

# There are several things not modeled by logical effort

Input arrival time (some transitions are faster than others)

Finite slope of input signal

Feedback capacitance  
(C gate to drain overlap)



To estimate the delay of a digital path we used the path logic effort and the path electrical effort.

$$G = \prod g_i$$

$$H = \frac{C_{out}(path)}{C_{in}(path)}$$

We also need the path effort, and the path branching effort

$$F = \prod g_i \times h_i$$

$$B = \prod b_i$$

$$b_i = \frac{Conpath + Coffpath}{Conpath}$$

$$F = GBH$$

Now we can find the delay

$$D = GBH + P$$

$$P = \sum p_i$$

It seems this is the true equation to use.

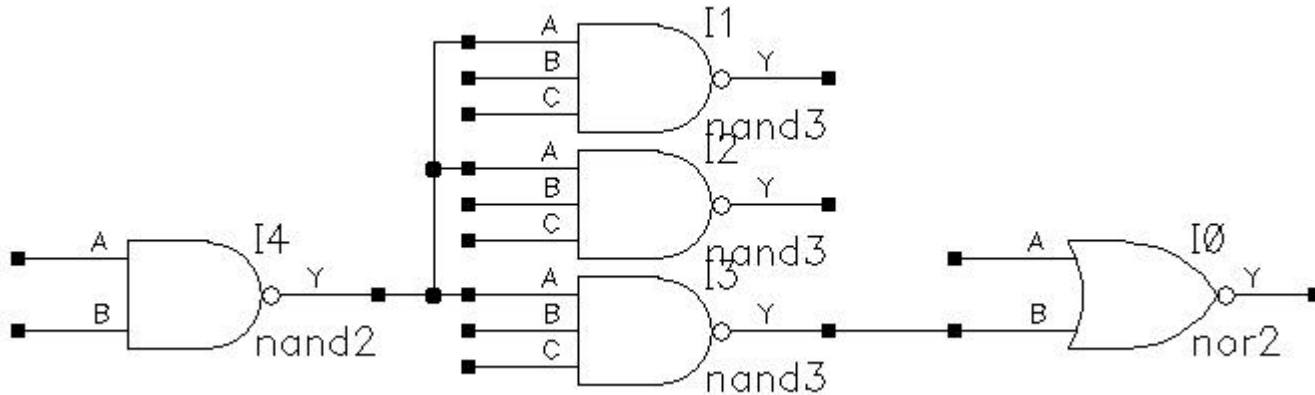
$$D = N \times \hat{f} + P$$

N is the number of stages

$$\hat{f} = \sqrt[N]{GBH}$$

Note we need to know the input capacitance.

Let's do an example:



$$P=2+3+2=7$$

$$G=4/3 \times 5/3 \times 5/3=100/27$$

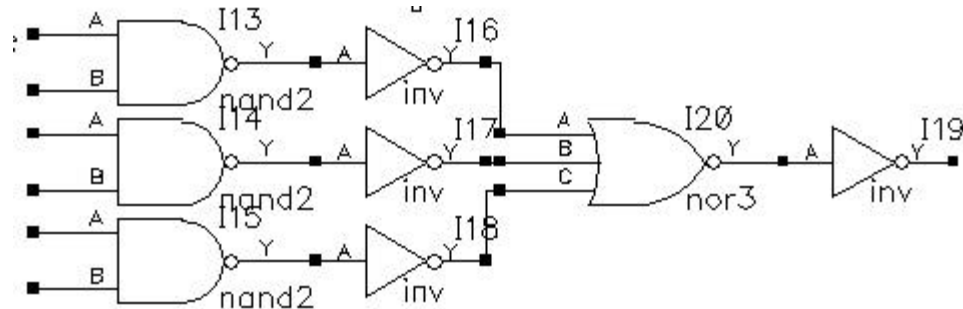
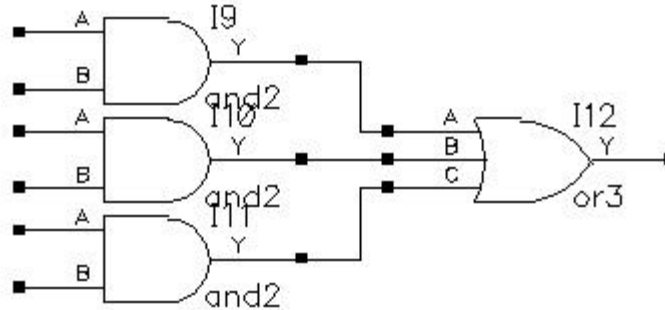
$$H=45/8$$

$$B=(C+2C)/C \times (C+C)/C=6$$

$$F=GBH$$

$$D=N \times F^{1/N} + P = 3 \times 5 + 7 = 22$$

And another:  
 $f=ab+bc+de$



$$P=2+1+3+1=7$$

$$G=4/3 \times 1 \times 7/3 \times 1 = 28/9$$

$$H=45/8$$

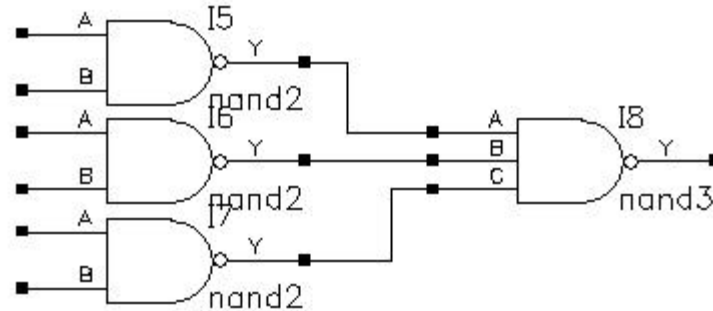
$$B=1$$

$$F=GBH=17.5$$

$$N=4$$

$$D=N \times F^{1/N} + P = 4 \times 2 + 7 = 15$$

And another:  
 $f=ab+bc+de$



$$P=2+3=5$$

$$G=4/3 \times 5/3 \times 1 = 20/9$$

$$H=45/8$$

$$B=1$$

$$F=GBH=12.5$$

$$N=2$$

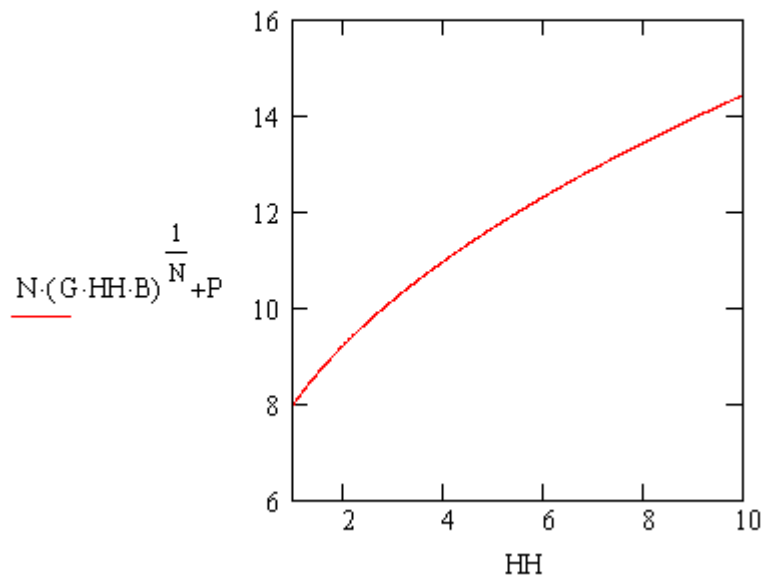
$$D=N \times F^{1/N} + P = 2 \times 3.53 + 5 = 12$$

This is faster!

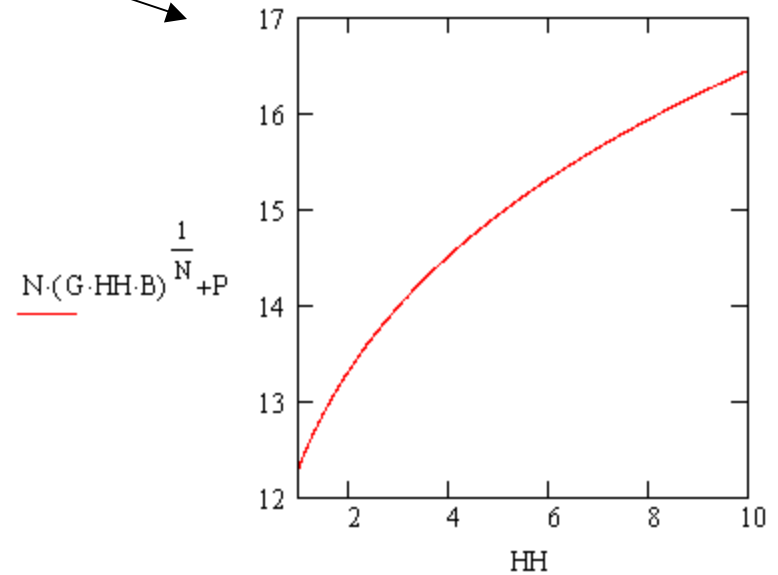
One draw back is you need to know what the  $C_i$  of the path is.

In the last example we saw that the nand/nand was faster than the and or.

$$N = 2 \quad G = 2.222 \quad B = 1$$



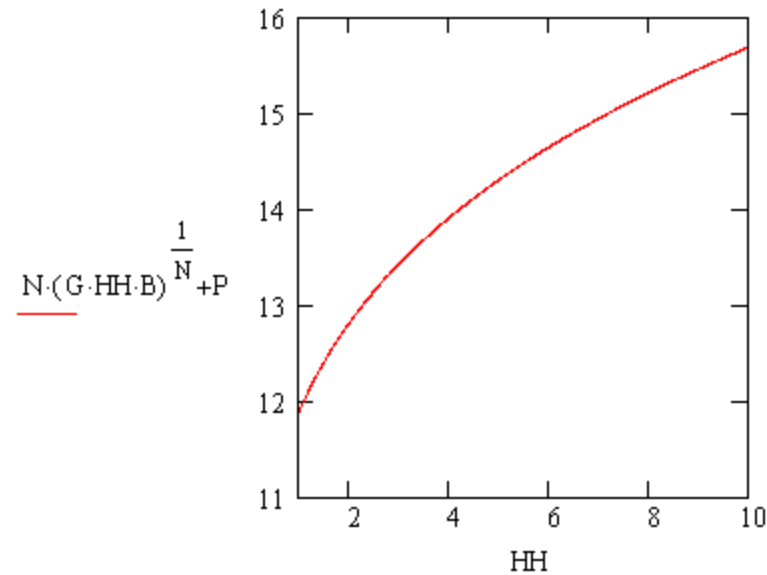
$$B := 1 \quad P := 7 \quad G := \frac{28}{9}$$



In this case the all nand is all ways faster

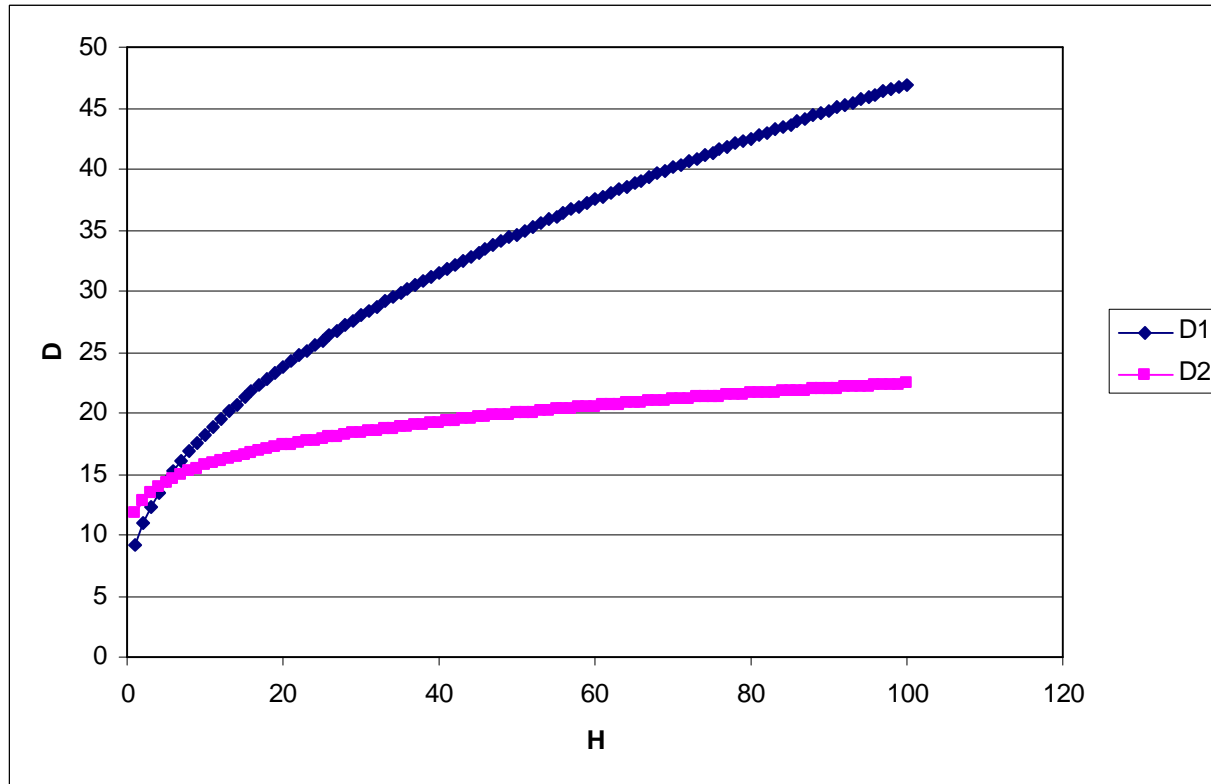
# Will adding two inverters speed up the design?

N := 4    G = 2.222    B = 1    P := 7

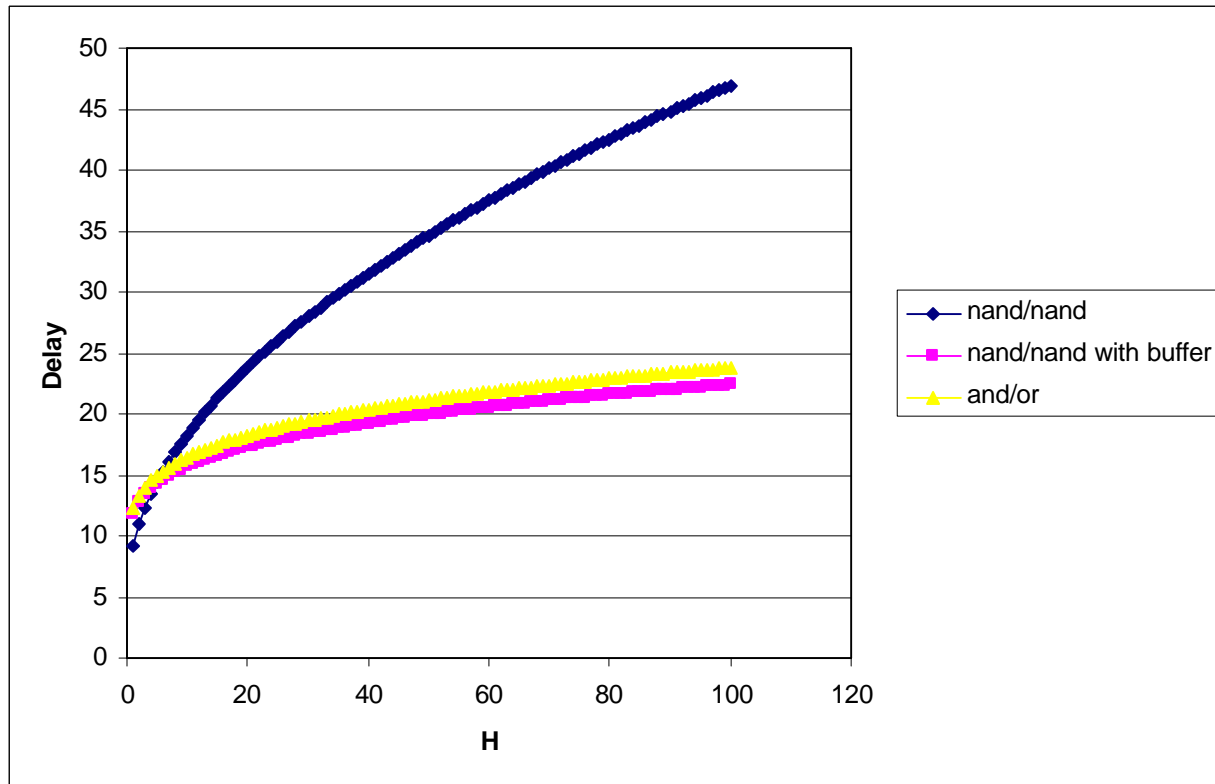


It does not seem so, but what if we had to drive a really large load?

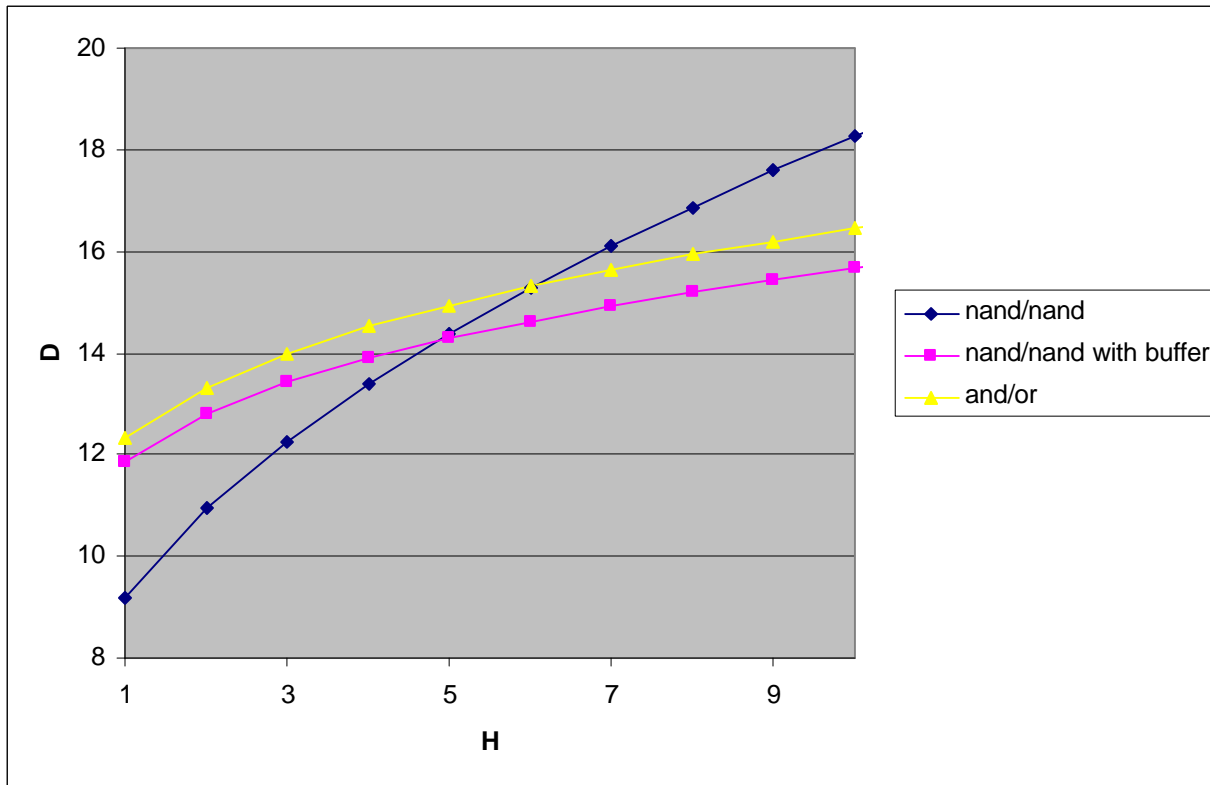
Driving a large load will mean a very large value of H of over 100



# Should we revisit the and or?



# Should we revisit the and or?



# Summary

- We learned about how to estimate delay for a digital CMOS circuit
- We learned how to compare various circuits for delay
- We learned that large loads need buffers