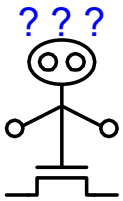
Introduction to CMOS VLSI Design

Logical Effort Part A

Lecture by Jay Brockman
University of Notre Dame Fall 2008
Modified by Peter Kogge Fall 2010,2011,2015, 2018
Based on lecture slides by David Harris, Harvey Mudd College
http://www.cmosvlsi.com/coursematerials.html

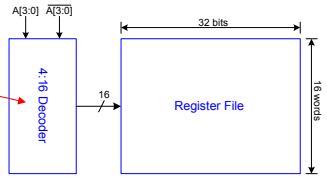
Introduction

- Chip designers face a bewildering array of choices
  - What is the best circuit topology?
  - What gives least delay?
  - How wide should the transistors be?
- **Logical effort:** a method to make these decisions
  - Simple model of delay
  - "Back-of-the-envelope"
  - Rapid comparisons between alternatives
  - Emphasizes remarkable symmetries



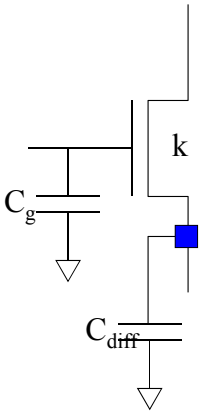
Motivating Example

- Ben Bitdiddle is the memory designer for the Motoroil 68W86, an embedded automotive processor. Help Ben design the *decoder* for a 16x32 register file.



- Decoder specifications:
  - 16 word register file
  - Each word is 32 bits wide
  - Each file bit presents load of 3 unit-sized transistors
  - Both true & complementary address inputs A[3:0] available
  - Each address input may drive 10 unit-sized transistors
- Ben needs to decide:
  - How many stages to use in decoder driver output buffers?
  - How large should each gate be?
  - How fast can decoder operate?

Some Review



- k: transistor's width is k times unit width
- On resistance is ***1/k*** times unit transistor
- C_g is capacitance of gate to body
  - Unit transistor's C_g is "1 unit"
  - $C_{g\text{-}k}$ is ***k*** times $C_{g\text{-}unit}$ of unit transistor
- C_{diff} is capacitance to body from a contacted source or drain
  - Approximately = C_g
- Diffusion capacitance of uncontacted source or drain is less (approx ½)
  - For simplicity assume the same

N vs P Widths

- To make rise ≈ fall time, want ~ equality between
  - On resistance of N side of circuit
  - On resistance of P side of circuit
- Remember
  - Unit width PMOS ≈ 2X unit width NMOS
  - Resistors in series add
  - Resistors in parallel = 1/sum(reciprocal of Rs)
- When multiple paths are possible
  - Reflecting multiple possible input conditions
  - Choose one with max resistance

Elmore Delay Model

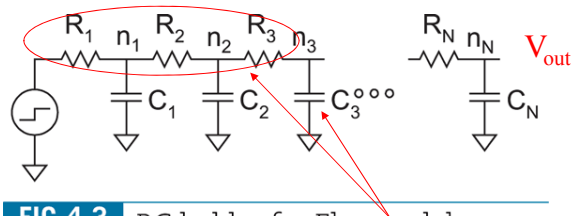
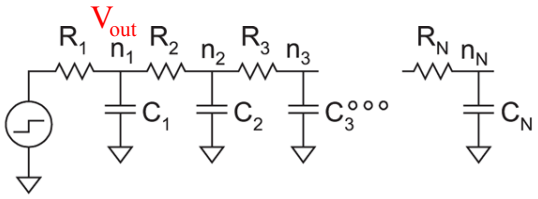


FIG 4.3 RC ladder for Elmore delay

$$\text{Delay} = \sum_{1} R_{n-i} C_i \sim \sum_{i=1}^{N} C_i \sum_{j=1}^{i} R_j$$

What If Output Not At End?



Ignore R2 thru Rn, & just sum C's

$$\text{Delay} = R_1 * \sum C_i$$

In reality, C2, … CN "shielded" by R's
(don't have to charge all way to voltage at Y)
Thus a "conservative" estimate

Resistance Examples

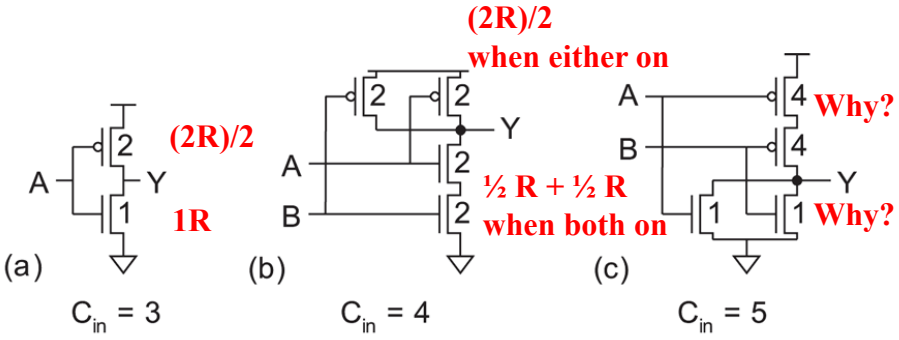


FIG 4.9 Logic gates sized for unit resistance

What's Cin: the Input Capacitance?

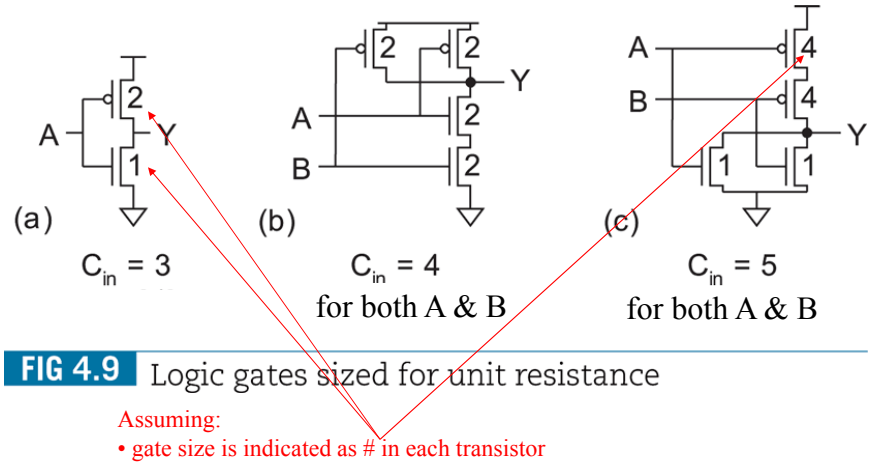


FIG 4.9 Logic gates sized for unit resistance

Assuming:
- gate size is indicated as # in each transistor
- gate capacitance of unit transistor 1 $C_g = C$

General Input Capacitance: Minimum Transistors

# of Inputs	NAND	NOR
1	3C:Inverter	
2	4C	5C
3	?	?
4	?	?
N	?	?

How About for K inputs: (K+2)C (2K+1)C

Sample Circuit Extraction

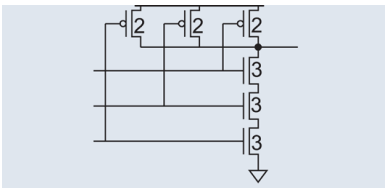


FIG 4.1 3-input NAND gate with unit rise and fall resistance

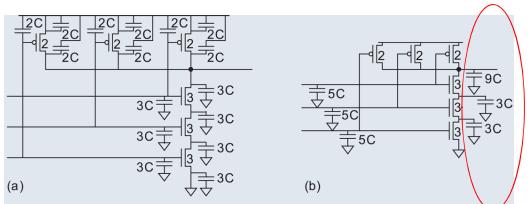


FIG 4.2 3-input NAND gate annotated with capacitances
Assuming all diffusion nodes contacted

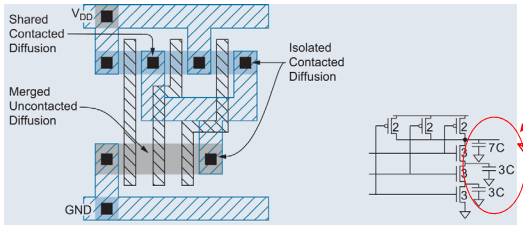


FIG 4.5 3-input NAND annotated with diffusion capacitances extracted from the layout

Real world
Slightly
smaller

Key Definitions

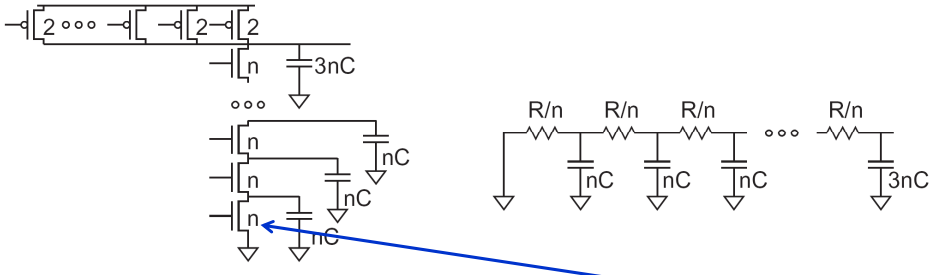
- **Parasitic Delay**: gate delay when no load attached
  - Count only diffusion capacitances
- **Contamination Delay**: Minimum (fastest) time from some input changes until any output starts to change
  - Look for input pattern that minimizes delay
- **Logical Effort** of some gate: ratio of its input capacitance to input capacitance of inverter that can deliver same output current
  - I.e. has same pullup/pulldown resistance
- **Ideal Delay**: ideal inverter (with no parasitic diffusion capacitance) driving an identical inverter
- **Effort Delay**: Delay imposed by external load

Approximate Parasitic Delay of Gates

- Parasitic delay = delay when it drives zero load
- Normalized to multiples of p_{inv} = 3RC

Parasitic Delay (Normalized)					
Gate type	Number of inputs				
	1	2	3	4	n
Inverter	1				
NAND		2	3	4	n
NOR		2	3	4	n
Tristate / mux	2	4	6	8	2n
XOR, XNOR		4	6	8	

Real Life Parasitic Delay



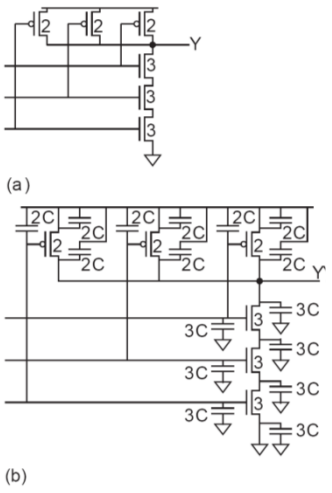
Applying Elmore model when all but **lowest**=1 & lowest:0→1:

$t_{pd} = (n^2/2 + 5n/2)RC$

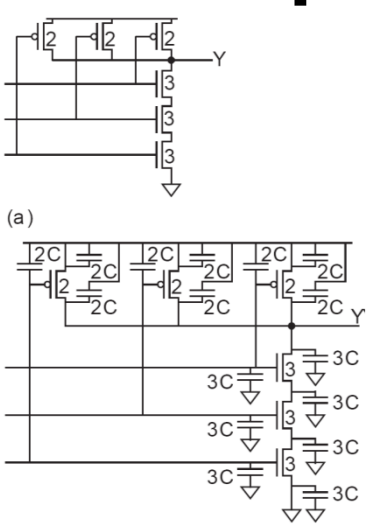
Note the square term!!!!
Big gates are ***slooooowwwwweeeeerrrrrr***

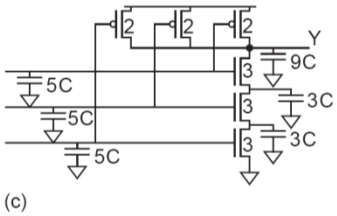
Contamination Delay

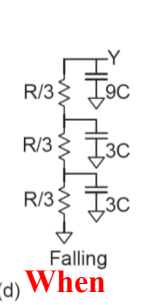
- **Minimum time** from some input change until any output starts to change for any input pattern
  - A function of load capacitance
- E.g. 3NAND: On fall, best if both bottom NMOS Ts on
  - Diffusion cap already drained
  - Only R effective left
  - Delay = ?
- On rise, best when ALL 3 PMOS turn on
  - Resistances in parallel
  - Delay = ?



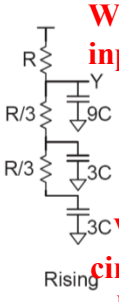
3 Input NAND







When all inputs high



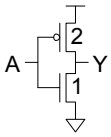
When one input low

What is circuit for all inputs low?

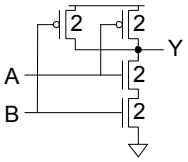
FIGURE 4.7 Equivalent circuits for a 3-input NAND gate

Logical Effort

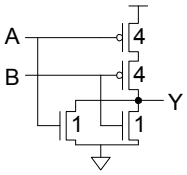
- ***Logical effort g:*** ratio of input capacitance of a gate to the input capacitance of an inverter *delivering the same output current* (same pullup/down resistance).
- Measure from delay vs. fanout plots
- Or *estimate* by counting transistor widths
- **AND divide by 3 to "normalize"**



C_{in} = 3
g = 3/3



C_{in} = 4
g = 4/3



C_{in} = 5
g = 5/3

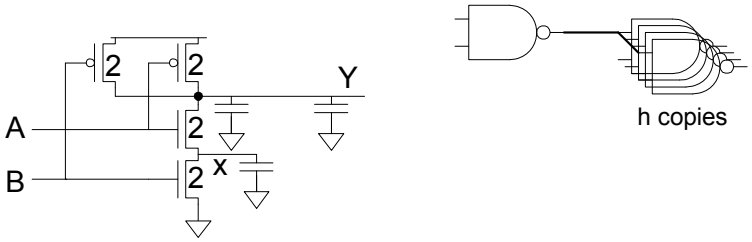
Logical Effort of Gates

Logical Effort (Normalized)					
Gate type	Number of inputs				
	1	2	3	4	n
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	2
XOR, XNOR		4, 4	6, 12, 6	8, 16, 16, 8	?

All inputs are not the same!

Now Consider Loaded Gates

- Estimate worst-case rising and falling delay of 2-input NAND driving *h* identical 2in NAND gates.



Summary Delay Estimation

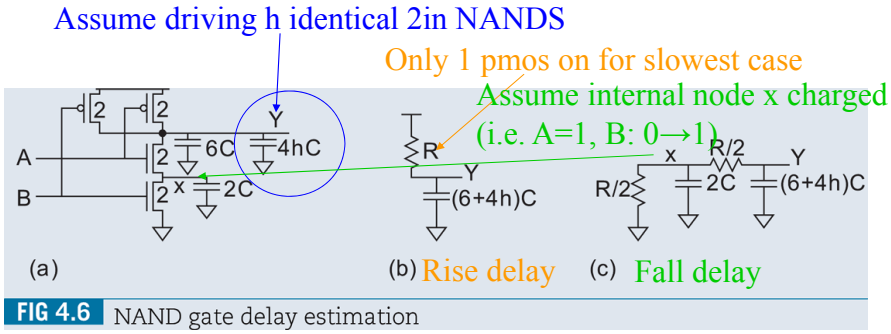


FIG 4.6 NAND gate delay estimation

Elmore delays:
Worst Case Rise = R(6+4h)C = (6+4h)RC
Worst Case Fall = (R/2)(2C) + R*(6+4h)C = (7+4h)RC

What If All Transistors *k Wider*?

- What happens to capacitances? Resistance?
- Assume output load = 4h'C
  - h'=h if driving small gates; = kh if larger gates

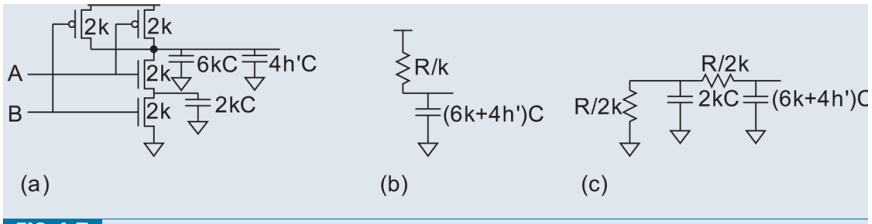


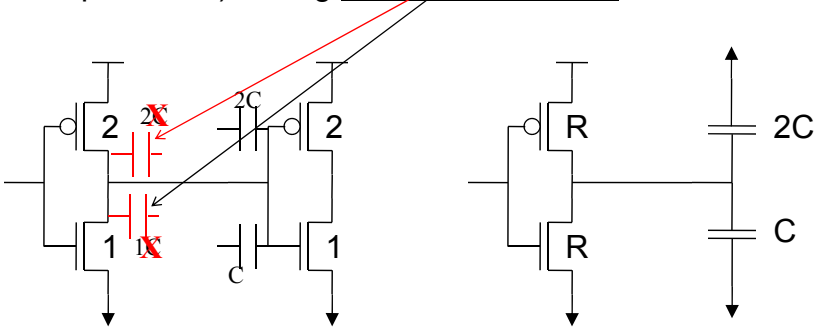
FIG 4.7 Scaled NAND gate delay estimation

Elmore delays if driving h copies of scaled gates:
Worst Case Rise = (6+4h)RC
Worst Case Fall = (7+4h)RC

The same as before: *WHY*?

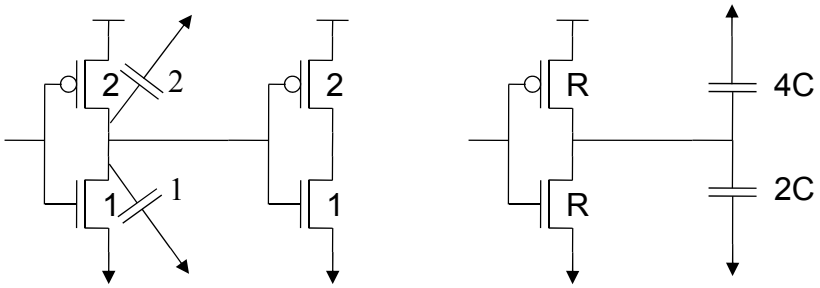
Ideal Gate Delay τ

- Imagine ***ideal inverter*** (with *no* parasitic diffusion capacitance) driving *an identical inverter*



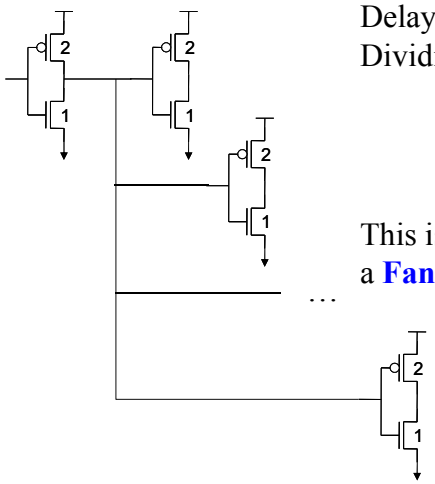
τ = 3RC = Ideal Inverter Delay

What If We Count Diffusion?



- delay = 6RC = 3RC (from diffusion) + 3RC (from load)
- Let's convert to **normalized delay**
  - Divide by τ = 3RC
  - Get 2τ
- So *real* gate has TWICE delay of our "*ideal*"

Sample Inverters



Capacitance = 3C (diffusion)
+ h3C (combined load)
Delay = R*(3C + h3C)
Dividing by t = 3RC yields

$$1 + h$$

This is ***normalized delay*** of
a **Fanout of h invertor** – denoted FO"h"

Linear Delay Model

$$\text{delay } d = p + f = p + hg$$

Prior delay had two parts

- ***p*: *Parasitic* delay**
  - 3RC for inverter
  - *Independent* of load
- ***f*: *Effort delay***
  - 3h RC
  - *Proportional* to total load capacitance
- = h*g if driving *identical* circuits
  - h = # of copies of gate (also called "**Fanout**")
  - g = "**Logical Effort**" (function of gate complexity)
    - 3RC for simple inverter

Normalized Linear Delay

- Remember
  - 3RC = parasitic delay of inverter
  - 3RC + 3hRC = delay if driving h copies
- **Normalized delay**: divide delay by 3RC
  - Measure of how much "slower" a circuit is than an inverter
  - = 1 + h for inverter driving h inverters

What If We're Not Driving Identical Copies of Self?

- Let C_{in} = input capacitance of gate *doing the driving*
- Let C_{out} = total capacitance *of load* presented to gate
- Express **fanout h** as C_{out}/C_{in}
  - Also called the **electrical effort**
  - How many *equivalent* copies of driving gate the actual load is

Delay in a Logic Gate

- Express delays in process-independent unit

$$d = \frac{d_{abs}}{\tau}$$

τ = 3RC for ideal invertor
≈ 12 ps in 180 nm process
40 ps in 0.6 μm process

Delay in a Logic Gate

- Express delays in process-independent unit

$$d = \frac{d_{abs}}{\tau}$$

- Delay has two components

$$d = f + p$$

f: property of driven circuit

p: property of driving gate

Delay in a Logic Gate

- Express delays in process-independent unit

$$d = \frac{d_{abs}}{\tau}$$

- Delay has two components

$$d = f + p$$

- **Parasitic delay *p***
  - Represents delay of gate driving no load
  - Set by internal parasitic capacitance

Question?

- What happens to Parasitic Delay p if we scale all transistors by same factor?
  - Drops
  - Stays the same
  - Increases

Delay in a Logic Gate

- Express delays in *process-independent unit*

$$d = \frac{d_{abs}}{\tau}$$

- Delay has two components

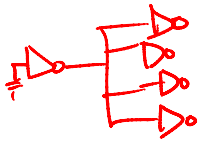
$$d = f + p$$

- **Effort delay** $f = gh$ (a.k.a. stage effort)
  - Again has two components
- g: ***logical effort***
  - Measures *relative* ability of gate to deliver current
  - $g \equiv 1$ for unit inverter

Delay in a Logic Gate

- Express delays in process-independent unit

$$d = \frac{d_{abs}}{\tau}$$



- Delay has two components

$$d = f + p$$

- Effort delay $f = gh$ (a.k.a. stage effort)
  - Again has two components

- *h*: ***electrical effort*** = C_{out} / C_{in}
  - Ratio of output to input capacitance of driving gate
  - Sometimes called **fanout**

Question?

- What happens to Logical Effort g if we scale all transistors by same factor?
  - Drops
  - Stays the same
  - Increases