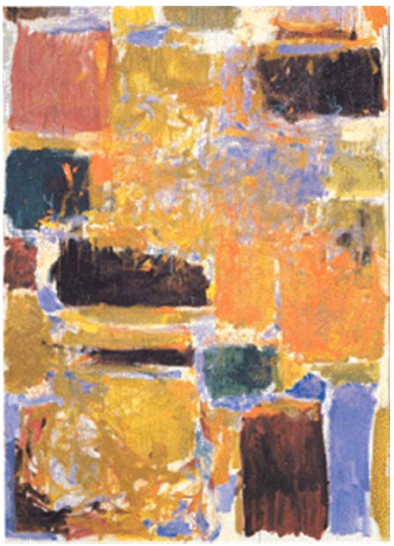


# EE141-Fall 2012 Digital Integrated Circuits

Lecture 7
Gate Delay and
Logical Effort



## Announcements

- Homework #3 due today
- Homework #4 due next Thursday

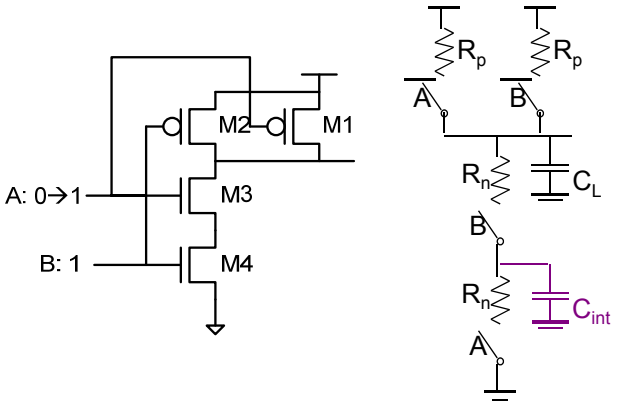
## Class Material

- Last lecture
  - Inverter delay optimization
- Today's lecture
  - Gate delay and logical effort
- Reading (Chapter 6)



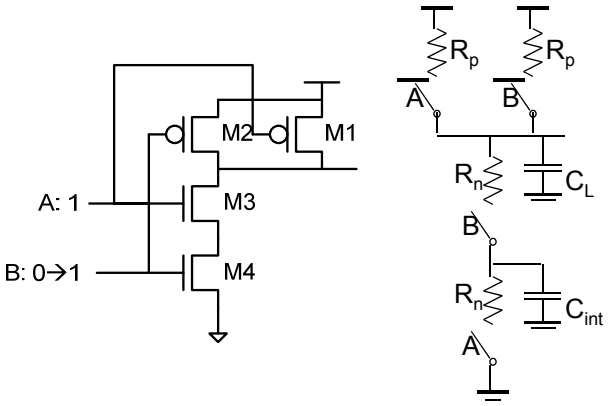
## Complex Gate Delay

- Use RC model to estimate delay:



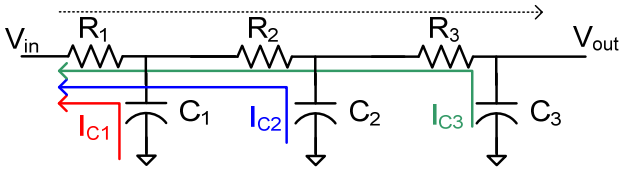
# Complex Gate Delay (2)

- Now what is the delay?



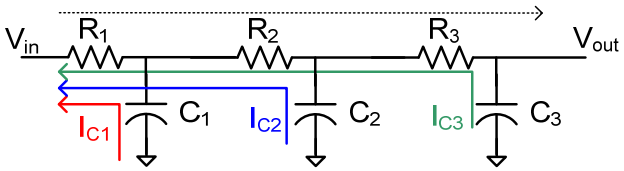



# Elmore Delay




- "Elmore delay": approximation for delay of arbitrary (complex) RC circuits
- To find "Elmore time constant":
  - For each capacitor, draw path of current from cap to input
  - Multiply C by sum of R's on current path that are common with path from $V_{in}$ to $V_{out}$
  - Add up RC products from all capacitors

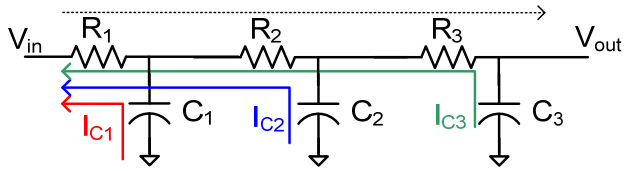
## Elmore Delay (Formal Method)




- Set $V_{in}$ to (incremental) ground
- Apply current I across C, measure $V_{out}$
- Calculate $R_{eff} = V_{out}/I$
- Time constant due to that C is $R_{eff}$*C

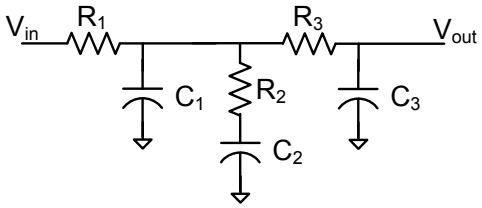


## Simplified Model: Elmore Delay




$$\tau_{Elmore} = R_1C_1 + (R_1 + R_2)C_2 + (R_1 + R_2 + R_3)C_3$$

## Another Elmore Delay Example

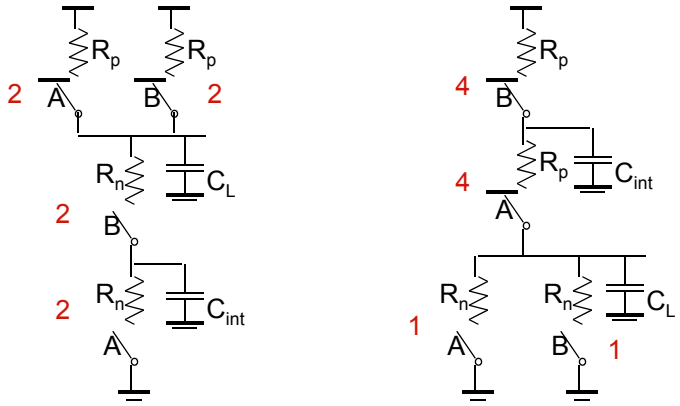



$\tau_{Elmore} =$



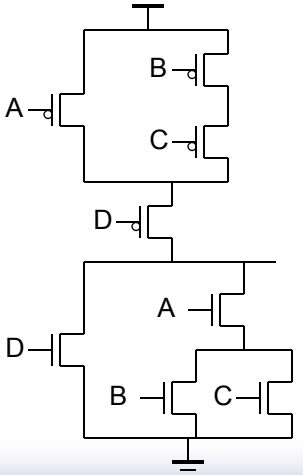
## Another Example Gate

## Gate Sizing

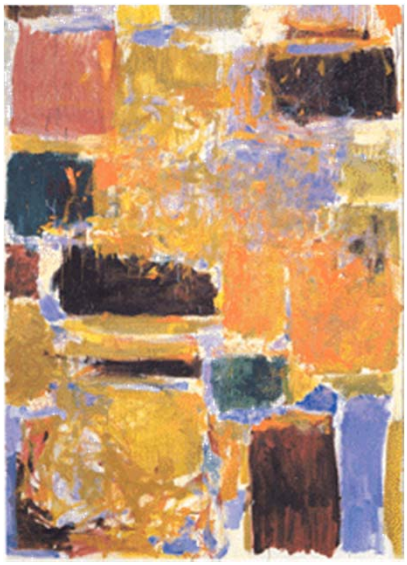





## Sizing Example




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

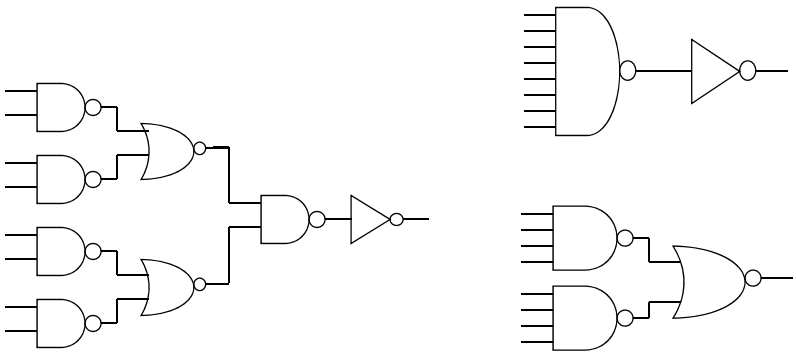


## *Logical Effort*



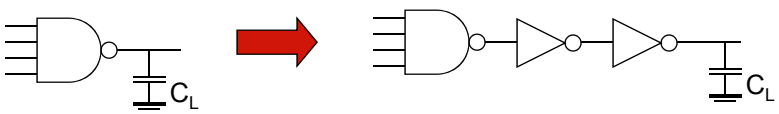
## *Question #1*

- All of these are decoders
  - Which one is “best”?



## Question #2

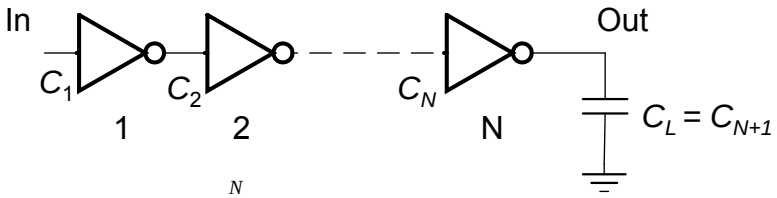
- Is it better to drive a big capacitive load directly with the NAND gate, or after some buffering?




- Method to answer both of these questions:
  - Logical effort
  - Extension of buffer sizing problem



## Buffer Chain Review



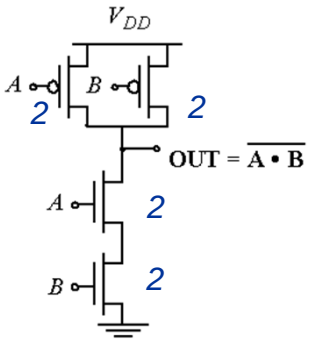

$$Delay = t_{inv}\sum_{i=1}^{N}(\gamma + f_i)$$

$f_i = C_{i+1}/C_i$

For given $N$: $C_{i+1}/C_i = C_i/C_{i-1}$
To find $N$: $C_{i+1}/C_i \sim 4$

# Delay of NAND Gate

$C_{dnand} = 6C_D$
$C_{gnand} = 4C_G = (4/3)\, C_{ginv}$
$C_D/C_G = \gamma$

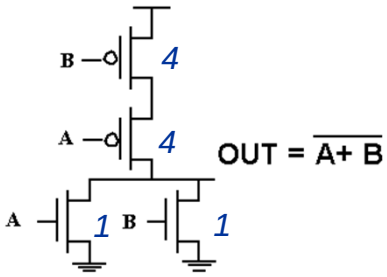



$t_{pNAND} = kR_W(C_{int} + C_L)$
$= k(R_{sq,n}*L/W)(WC_{dnand} + C_L)$
$= k(R_{sq,n}*L*C_{gnand})(C_{dnand}/C_{gnand} + C_L/(WC_{gnand}))$
$= k(R_{sq,n}*L*4*C_g)(3/2*\gamma + C_L/(WC_{gnand}))$
$= (4/3)\, t_{inv}\,(3/2\gamma + f)$



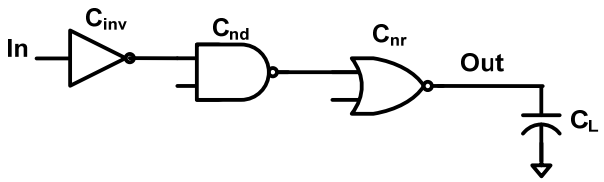
# Delay of NOR Gate

$C_{dnor} = 6C_D$
$C_{gnor} = 5C_G = (5/3)\, C_{ginv}$
$C_D/C_G = \gamma$




$t_{pNAND} = kR(C_{int} + C_L)$
$= k(R_{sq,n}*L/W)(WC_{dnor} + C_L)$
$= k(R_{sq,n}*L*C_{gnor})(C_{dnor}/C_{gnor} + C_L/(WC_{gnor}))$
$= k(R_{sq,n}*L*5*C_g)(6/5*\gamma + C_L/(WC_{gnor}))$
$= (5/3)\, t_{inv}\,(6/5\gamma + f)$

## Sizing a Chain



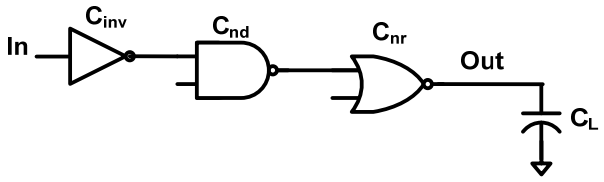

$t_{p,tot} = t_{inv}[(\gamma+C_{nd}/C_{inv}) + 4/3*(3/2*\gamma+C_{nr}/C_{nd}) + 5/3*(6/5*\gamma+C_L/C_{nr})]$

$= t_{inv}[(\gamma + C_{nd}/C_{inv}) + (2\gamma + (4/3)*C_{nr}/C_{nd}) + (2\gamma + (5/3)*C_L/C_{nr})]$

- You already know how to solve this!
  - NAND and NOR are just like inverters
  - Except that their fanout looks larger – i.e., get more delay than an inverter for same fanout



## Logical Effort




$t_{p,tot} = t_{inv}[(\gamma+C_{nd}/C_{inv}) + 4/3*(3/2*\gamma+C_{nr}/C_{nd}) + 5/3*(6/5*\gamma+C_L/C_{nr})]$

$= t_{inv}[(\gamma + C_{nd}/C_{inv}) + (2\gamma + (4/3)*C_{nr}/C_{nd}) + (2\gamma + (5/3)*C_L/C_{nr})]$

**$p_{gate}$**
parasitic delay ≠ f(W)

**LE**
logical effort ≠ f(W)

**f**
electrical fanout

$$t_{pgate} = t_{inv}\,(p_{gate} + LE \cdot f)$$

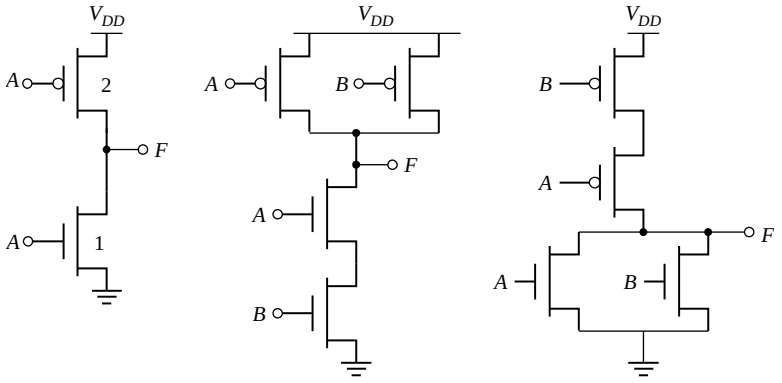
# Finding Logical Effort of a Gate

- Inverter has the smallest logical effort and intrinsic delay of all static CMOS gates
  - So normalize everything to inverter
- Logical effort LE is defined as:
  - $(R_{eq,gate}C_{in,gate})/(R_{eq,inv}C_{in,inv})$
  - Easiest way to calculate (usually):
    - Size gate to deliver same current as an inverter, take ratio of gate input capacitance to inverter capacitance
- LE increases with gate complexity



# Finding Logical Effort of a Gate

**Calculating LE by sizing for same drive strength:**




Inverter

***LE* = 1**

2-input NAND

***LE* =**

2-input NOR

***LE* =**

## Gate Sizing Convention

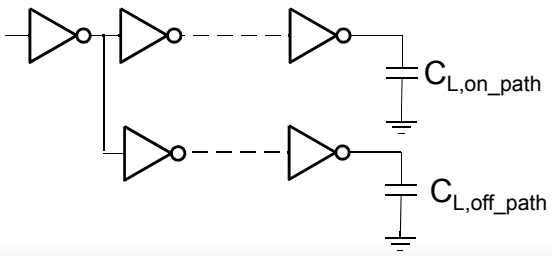
- Need to set a convention:
  - What does a gate of size '2' mean?
- For an inverter it is clear:
  - $C_{inv} = 2$, $R_{inv} = ½$
- For a gate, two possibilities:
  - $C_{gate} = 2C_{inv} \rightarrow R_{gate} = (LE/2)*R_{inv}$
  - $R_{gate} = R_{inv}/2 \rightarrow C_{gate} = (2*LE)*C_{inv}$
- In my notes, size ≡ $C_{gate}/C_{inv}$
  - Size 2 gate has twice the input capacitance of a unit inverter



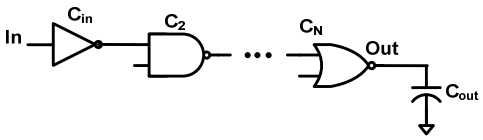
## Adding Branching

Branching effort:

$$b = \frac{C_{L,on-path} + C_{L,off-path}}{C_{L,on-path}}$$



## Chain of Arbitrary Gates




$$Delay = \sum_{i=1}^{N}\left(p_i + LE_i \cdot f_i\right)$$

Effective fanout: $EF_i = LE_i f_i$

Path electrical fanout: $F = C_{out}/C_{in}$

Path logical effort: $\Pi LE = LE_1 LE_2 \ldots LE_N$

Branching effort: $\Pi B = b_1 b_2 \ldots b_N$

Path effort: $PE = \Pi LE\ \Pi B\ F$

Path delay $D = \Sigma d_i = \Sigma p_i + \Sigma EF_i$



## Optimum EF/Stage

- Just like buffer chain, but use (PE, EF) instead of (F, f):

$$EF^N = PE$$

$$EF = \sqrt[N]{PE}$$

Effective fanouts: $LE_1 f_1 = LE_2 f_2 = \ldots = LE_N f_N$

Minimum path delay

$$\hat{D} = \sum_{i=1}^{N}\left(LE_i f_i + p_i\right) = N \cdot PE^{1/N} + \sum_{i=1}^{N} p_i$$

## Optimal Number of Stages

For a given load,
and given input capacitance of the first gate
Find optimal number of stages and optimal sizing

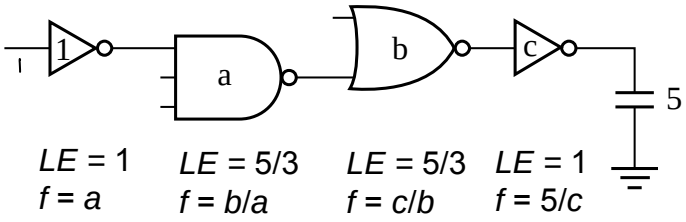
$$D = N \cdot PE^{1/N} + \sum p_i$$

*Remember: we can always add inverters to the end of the chain*

The 'best effective fanout' $EF = PE^{1/\hat{N}}$ is still around 4
(3.6 with $\gamma$=1)



## Example: Optimize Path




*LE* = 1
*f* = *a*

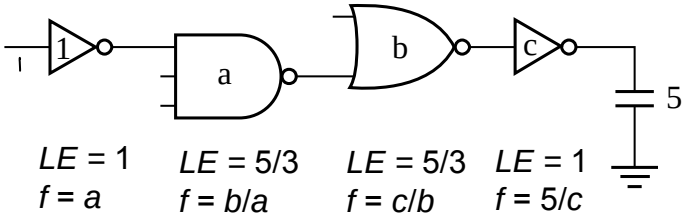
*LE* = 5/3
*f* = *b*/*a*

*LE* = 5/3
*f* = *c*/*b*

*LE* = 1
*f* = 5/*c*

*Electrical fanout, F* =
$\Pi$ *LE* =
*PE* =
*EF/stage* =
*a* =
*b* =
c =

## Example: Optimize Path




| $LE = 1$ | $LE = 5/3$ | $LE = 5/3$ | $LE = 1$ |
|---|---|---|---|
| $f = a$ | $f = b/a$ | $f = c/b$ | $f = 5/c$ |

*Electrical fanout, F* = 5
$\Pi LE = 1 \cdot (5/3) \cdot (5/3) \cdot 1 = (25/9)$
*PE* = $(\Pi LE) \cdot F = (125/9)$
*EF/stage* = (125/9)^(1/4) = 1.93
*a* = 1.93
*b* = 2.23
c = 2.59

*5/c = 1.93*
*(5/3)c/b = 1.93*
*(5/3)b/a = 1.93*



## Method of Logical Effort

- Compute the path effort: PE = $(\Pi LE)BF$
- Find the best number of stages $N \sim \log_4 PE$
- Compute the effective fanout/stage $EF = PE^{1/N}$
- Sketch the path with this number of stages
- Work either from either end, find sizes:
  $C_{in} = C_{out} * LE/EF$

Reference: Sutherland, Sproull, Harris, "Logical Effort, Morgan-Kaufmann 1999.

## *Next Lecture*

- Applying logical effort to a decoder