EECE 481

High-Speed CMOS Gate Design

(Logical Effort)

Lecture 10

Reza Molavi
Dept. of ECE
University of British Columbia
reza@ece.ubc.ca

Slides Courtesy : Dr. Res Saleh (UBC), Dr. D. Sengupta (AMD), Dr. B. Razavi (UCLA)

Optimal Path Delay Design - Review

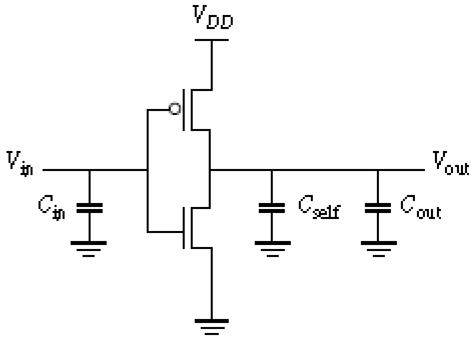
There are two unknowns, the number of gates and the size of each (*two degrees of freedom*)

$$C_{\mathrm{in}} = C_g(W_n + W_p) = C_g(W_n + 2W_n) = C_g(3W_n)$$

$$R_{\mathit{eff}} = R_{eqn}\left(\frac{L_n}{W_n}\right)$$

$$\tau_{\mathrm{inv}} = R_{\mathit{eff}}C_{\mathrm{in}} = R_{eqn}\left(\frac{L_n}{W_n}\right)C_g\,(3W_n) = 3R_{eqn}C_gL_n$$

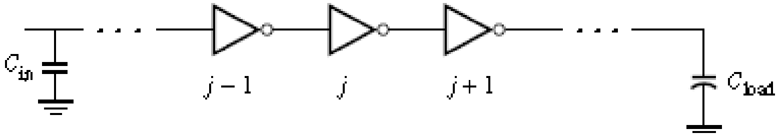
This is the *intrinsic delay* of a gate (specific tag) used many places



$$t_{\mathrm{delay}} = R_{\mathit{eff}}C_{\mathrm{in}}\left[\frac{C_{\mathrm{out}}}{C_{\mathrm{in}}} + \frac{C_{\mathrm{self}}}{C_{\mathrm{in}}}\right] = \tau_{\mathrm{inv}}\left[\frac{C_{\mathrm{out}}}{C_{\mathrm{in}}} + \gamma_{\mathrm{inv}}\right]$$

Gate Sizing for Optimal Path delay – All inverters

$$\text{total\_delay} = \sum_{j=1}^{N} \tau_{\text{inv}}\left(\frac{C_{j+1}}{C_j} + \gamma_{\text{inv}}\right)$$



$$\text{total\_delay} = \sum_{j} \tau_{\text{inv}}\left(\frac{C_g W_{j+1}}{C_g W_j} + \gamma_{\text{inv}}\right) = \sum_{j} \tau_{\text{inv}}\left(\frac{W_{j+1}}{W_j} + \gamma_{\text{inv}}\right)$$

Let’s consider two consecutive ones

$$D_j = \tau_{\text{inv}}\left(\frac{W_j}{W_{j-1}} + \gamma_{\text{inv}}\right) + \tau_{\text{inv}}\left(\frac{W_{j+1}}{W_j} + \gamma_{\text{inv}}\right)$$

Minimize the delay (find a W that makes

$dD_j/dW = 0$)

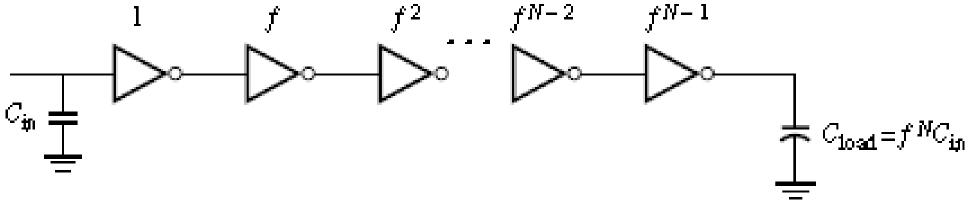
$$\frac{\partial D_j}{\partial W_j} = \tau_{\text{inv}}\frac{1}{W_{j-1}} - \tau_{\text{inv}}\frac{W_{j+1}}{W_j^2} = 0$$

$$\therefore \frac{W_j}{W_{j-1}} = \frac{W_{j+1}}{W_j}$$

$$\therefore W_j = \sqrt{W_{j+1}W_{j-1}}$$

If the size of each gate is the geometric mean of the two gates (previous and after) the delay Is minimum!

Gate Sizing for Optimal Path delay - Review



Therefore, we can consider the size of gates a geometric sequence with factor *f*

$$f^N C_{in} = C_{load}$$

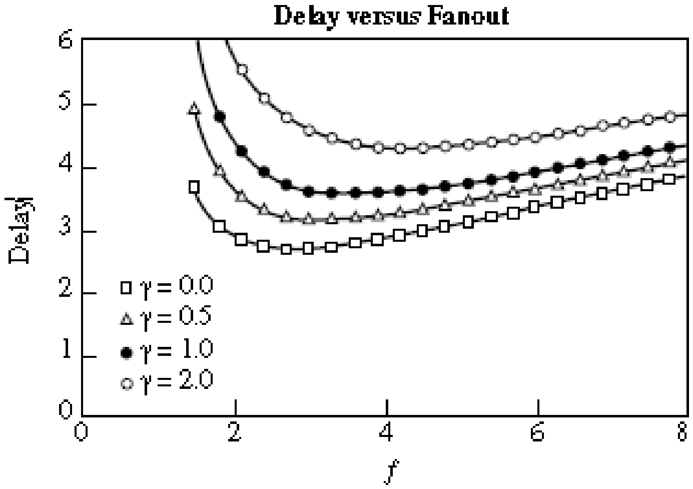
$$\therefore N = \frac{\ln(C_{load}/C_{in})}{\ln f}$$

$$\text{total\_delay} = N \times \tau_{inv}\left(\frac{C_j}{C_{j-1}} + \gamma_{inv}\right)$$

$$\text{total\_delay} = \frac{\ln(C_{load}/C_{in})}{\ln f} \times \tau_{inv}(f + \gamma_{inv})$$

What *f* makes the total delay a minimum?

Optimum Fan-out for CMOS Gates



The optimum value of *f* depends on *γ* (for *γ* = 0, it is *e*).

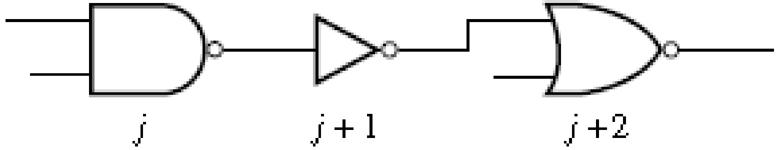
It is similar calculation if we have chain of NAND and NORs:

$$\tau_{\text{nand}} = R_{\mathit{eff}} C_{\text{in}} = R_{eqn}\left(\frac{L_n}{W_n}\right) 4 W_n C_g = 4 R_{eqn} C_g L_n$$

$$\text{total\_delay} = \sum_j \tau_{\text{nand}}\left(\frac{C_{j+1}}{C_j} + \gamma_{\text{nand}}\right)$$

$$\tau_{\text{nor}} = R_{\mathit{eff}} C_{\text{in}} = R_{eqn}\left(\frac{L_n}{W_n}\right) 5 W_n C_g = 5 R_{eqn} C_g L_n$$

Optimum Fan-out for CMOS Gates - Review



$$\text{total\_delay} = \tau_{\text{nand}}\left(\frac{C_{j+1}}{C_j} + \gamma_{\text{nand}}\right) + \tau_{\text{inv}}\left(\frac{C_{j+2}}{C_{j+1}} + \gamma_{\text{inv}}\right) + \tau_{\text{nor}}\left(\underbrace{\frac{C_{j+3}}{C_{j+2}}}_{\text{Fan-out portion of delay}} + \overbrace{\gamma_{\text{nor}}}^{\text{Parasitic portion of delay}}\right)$$

The delay through stages j and j + 1 is given by

$$D_{j+1} = \tau_{\text{nand}}\left(\frac{C_{j+1}}{C_j} + \gamma_{\text{nand}}\right) + \tau_{\text{inv}}\left(\frac{C_{j+2}}{C_{j+1}} + \gamma_{\text{inv}}\right)$$

For minimum delay, we take the derivative with respect to C_{j+1}:

$$\frac{\partial D_{j+1}}{\partial C_{j+1}} = \tau_{\text{nand}}\left(\frac{1}{C_j}\right) - \tau_{\text{inv}}\left(\frac{C_{j+2}}{C_{j+1}^2}\right) = 0$$

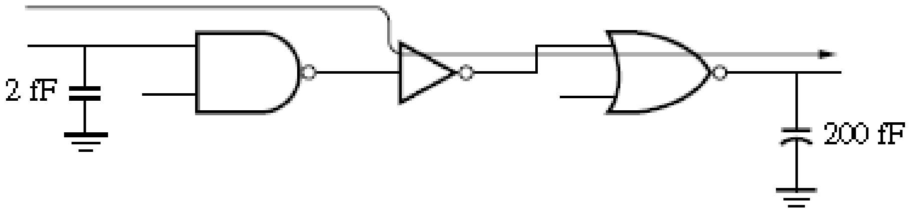
$$\therefore \tau_{\text{nand}}\left(\frac{C_{j+1}}{C_j}\right) = \tau_{\text{inv}}\left(\frac{C_{j+2}}{C_{j+1}}\right)$$

$$\therefore \tau_{\text{nand}} FO_j = \tau_{\text{inv}} FO_{j+1} \longleftarrow$$

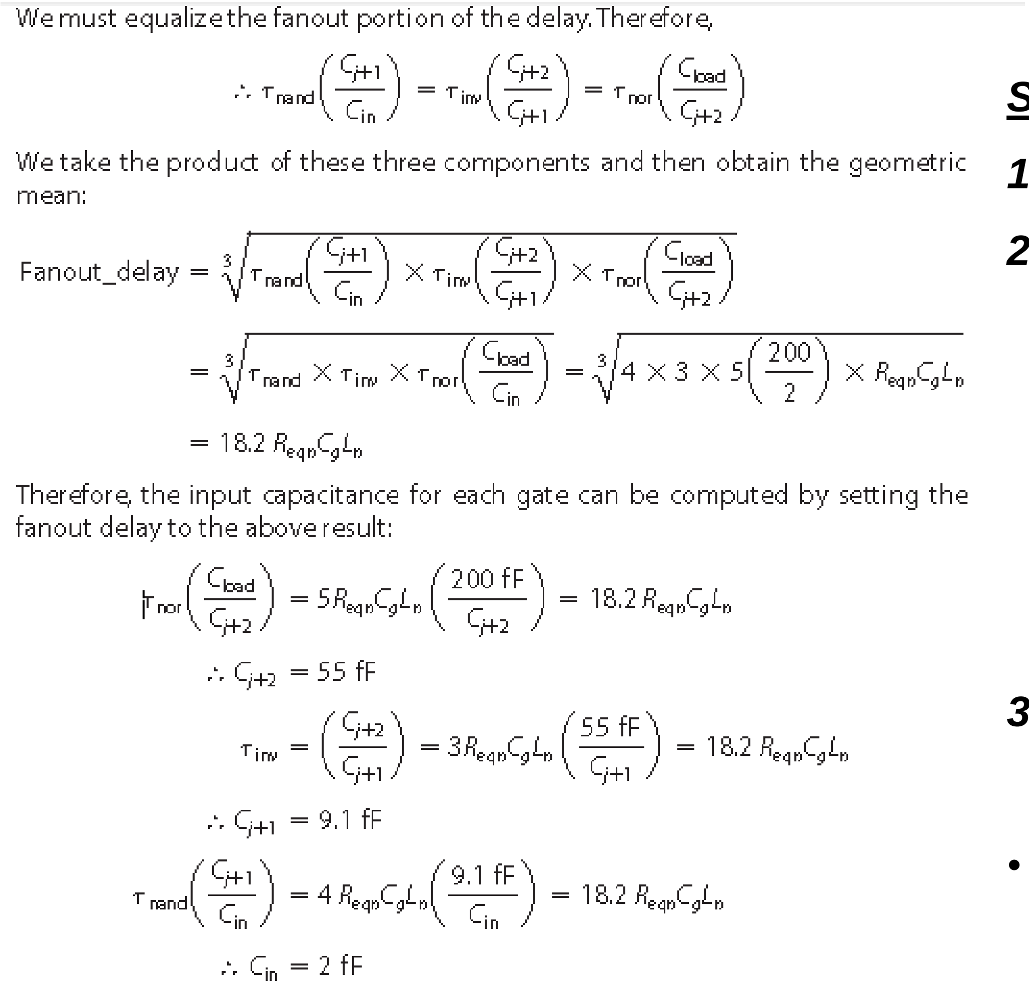
When we have mixture of gates minimum delay is achieved when the Fan-out portion of delays are equal for all gates

Optimum Fan-out for CMOS Gates

Example: Find the device sizes that optimize the delay through the indicated path for the circuit below.



Optimum Fan-out for CMOS Gates



We must equalize the fanout portion of the delay. Therefore,

$$\therefore \tau_{nand}\left(\frac{C_{j+1}}{C_{in}}\right) = \tau_{inv}\left(\frac{C_{j+2}}{C_{j+1}}\right) = \tau_{nor}\left(\frac{C_{load}}{C_{j+2}}\right)$$

We take the product of these three components and then obtain the geometric mean:

$$\text{Fanout\_delay} = \sqrt[3]{\tau_{nand}\left(\frac{C_{j+1}}{C_{in}}\right) \times \tau_{inv}\left(\frac{C_{j+2}}{C_{j+1}}\right) \times \tau_{nor}\left(\frac{C_{load}}{C_{j+2}}\right)}$$

$$= \sqrt[3]{\tau_{nand} \times \tau_{inv} \times \tau_{nor}\left(\frac{C_{load}}{C_{in}}\right)} = \sqrt[3]{4 \times 3 \times 5\left(\frac{200}{2}\right)} \times R_{eqn}C_gL_n$$

$$= 18.2\ R_{eqn}C_gL_n$$

Therefore, the input capacitance for each gate can be computed by setting the fanout delay to the above result:

$$\tau_{nor}\left(\frac{C_{load}}{C_{j+2}}\right) = 5R_{eqn}C_gL_n\left(\frac{200\ \text{fF}}{C_{j+2}}\right) = 18.2\ R_{eqn}C_gL_n$$

$$\therefore C_{j+2} = 55\ \text{fF}$$

$$\tau_{inv} = \left(\frac{C_{j+2}}{C_{j+1}}\right) = 3R_{eqn}C_gL_n\left(\frac{55\ \text{fF}}{C_{j+1}}\right) = 18.2\ R_{eqn}C_gL_n$$

$$\therefore C_{j+1} = 9.1\ \text{fF}$$

$$\tau_{nand}\left(\frac{C_{j+1}}{C_{in}}\right) = 4\ R_{eqn}C_gL_n\left(\frac{9.1\ \text{fF}}{C_{in}}\right) = 18.2\ R_{eqn}C_gL_n$$

$$\therefore C_{in} = 2\ \text{fF}$$

<u>Steps to solve</u>

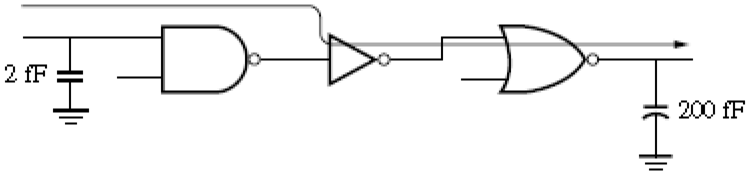
1. ***Make the fan-out portions equal = X***
2. ***Multiply all of them = X^n***
    1. ***This will make the internal fan-outs vanish and only remaining Terms are the input/output load and Intrinsic delays, take nth root to find X***
3. ***Work backwards from the load to find the size of each gate and then device***

- ***This technique gives you the size first and then you calculate the delay***

Logical Effort Method (normalized delay)

One way to simplify delay calculation is to normalize the total delay w.r.t τ_{inV}

$\tau_{inV} = R_{eqn}L*3C_g = 12.5k *3*2fF/\mu m*.1\mu m = 7.5$ ps



$$\frac{\text{total\_delay}}{\tau_{inv}} = \overset{LE_{nand}}{\boxed{\frac{\tau_{nand}}{\tau_{inv}}}}\left(\frac{C_{j+1}}{C_j} + \gamma_{nand}\right) + \overset{LE_{nor}}{\boxed{\frac{\tau_{inv}}{\tau_{inv}}}}\left(\frac{C_{j+2}}{C_{j+1}} + \gamma_{inv}\right) + \overset{LE_{nor}}{\boxed{\frac{\tau_{nor}}{\tau_{inv}}}}\left(\frac{C_{j+3}}{C_{j+2}} + \gamma_{nor}\right)$$

In normalized form, the delay equation is

$$D = (LE_{nand}FO_1 + P_{nand}) + (LE_{inv}FO_2 + P_{inv}) + (\overset{SE \xrightarrow{\quad} \text{Stage Effort}}{\boxed{LE_{nor}FO_3}} + P_{nor})$$

we need to learn how to calculate ***LE***, and ***P*** for different gates

Ex. If path delay is 13D therefore the total delay 13*7.5ps = 97.5ps

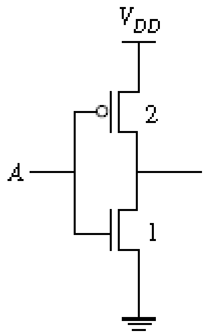
* All we need to do (for delay optimization) is to make **SE** equal for cascaded gates in a chain

Logical Effort – Calculating *LE*

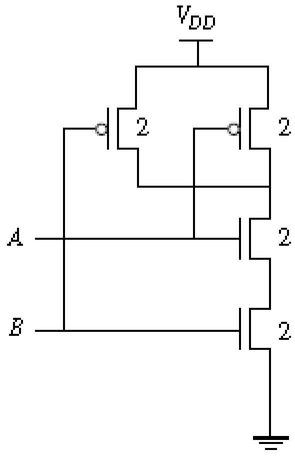
In order to calculate LE for any gate we first make its on-resistance (both pull-up and pull-down paths) equal to an inverter (if needed scale devices) and then take the ratio of input capacitors

For the NAND gate: $LE = \dfrac{(C_{\text{in}})_{\text{nand}}}{(C_{\text{in}})_{\text{inv}}} = \dfrac{2+2}{3} = \dfrac{4}{3}$

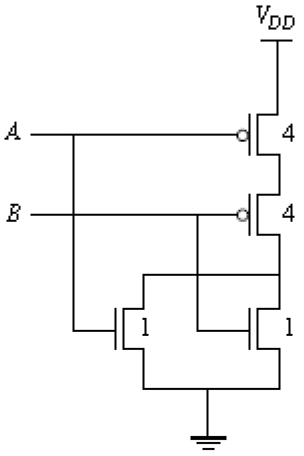
For the NOR gate: $LE = \dfrac{(C_{\text{in}})_{\text{nor}}}{(C_{\text{in}})_{\text{inv}}} = \dfrac{4+1}{3} = \dfrac{5}{3}$



$LE = 1$



$LE = 4/3$



$LE = 5/3$

Logical Effort – Calculation *P*

For Inverter:

$$P = LE_{\text{inv}} \times \gamma_{\text{inv}} = LE \times \frac{C_{\text{self}}}{C_{\text{in}}} = LE \times \frac{C_{\text{eff}} 3W}{C_g 3W} = LE \times \frac{C_{\text{eff}}}{C_g}$$

$$P_{\text{inv}} = (1) \times \frac{1 \text{ fF}/\mu\text{m}}{2 \text{ fF}/\mu\text{m}} = \frac{1}{2}$$

For NAND:

$$P_{\text{nand}} = LE_{\text{nand}} \times \gamma_{\text{nand}} = LE \times \frac{C_{\text{self}}}{C_{\text{in}}}$$

$$= LE \times \frac{C_{\text{eff}}(2W + 2W + 2W)}{C_g(2W + 2W)} = LE \times \frac{C_{\text{eff}}}{C_g} \times \frac{3}{2} = \frac{4}{3} \times \frac{1}{2} \times \frac{3}{2} = 1$$

For NOR

$$P_{\text{nor}} = LE_{\text{nor}} \times \gamma_{\text{nor}} = LE \times \frac{C_{\text{self}}}{C_{\text{in}}}$$

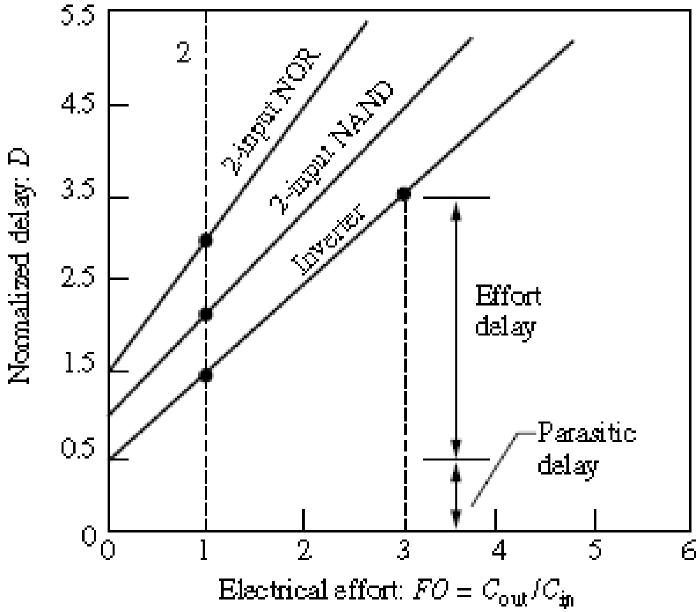
$$= LE \times \frac{C_{\text{eff}}(W + 4W + 4W)}{C_g(W + 4W)} = LE \times \frac{C_{\text{eff}}}{C_g} \times \frac{9}{5} = \frac{5}{3} \times \frac{1}{2} \times \frac{9}{5} = \frac{3}{2}$$

Logical Effort – Interpreting *LE* and *P*

Stage Delay = LE * EE (FO) + P

EE or Electrical effort is the Fan-out of the gate

Let’s plot Stage delay vs. FO for different gates

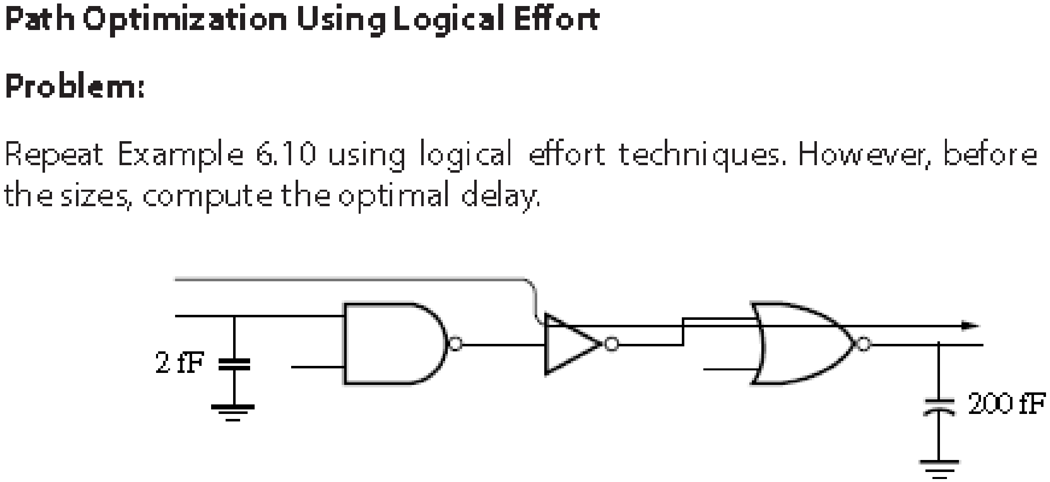


The graph indicates that for a fixed FO (EE) inverters shows the lowest delay, next NAND and finally NOR (Why NOR is the slowest?)

In other words, in optimizing the delay when we make the SE equal for all gates, NOR will drive the lowest Fan-out (load)

**intuitively speaking*, the weakest guys (gate) in a team (chain) should do the least amount of work (Fan-out) to deliver the fastest results (unfortunately, no ethics in CMOS world!)

Logical Effort – Example



Path Optimization Using Logical Effort

Problem:

Repeat Example 6.10 using logical effort techniques. However, before specifying the sizes, compute the optimal delay.

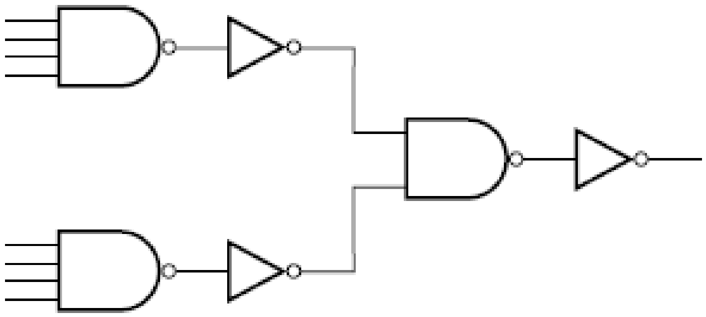
(Board Notes)

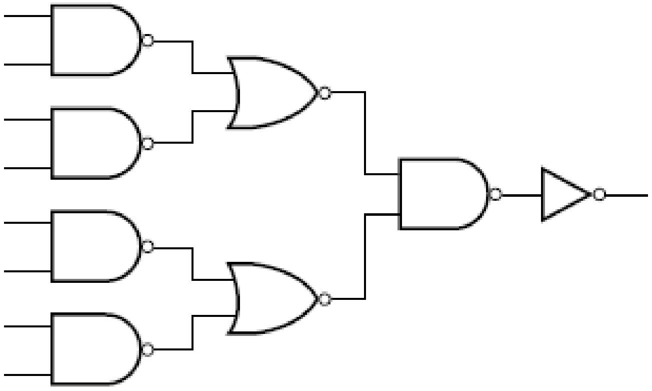
Logical Effort – Example2

In Lecture 8 we learnt that a one-stage 8-input CMOS AND is not a good idea and

We presented two alternative implementations, now can analyze them quantitatively and find

Better of the two in terms of optimum delay.

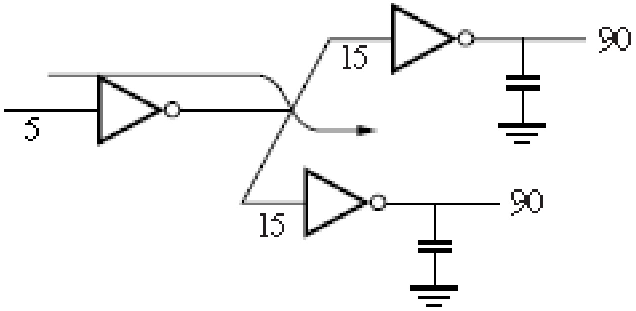




(Board Notes)

Branching Effort

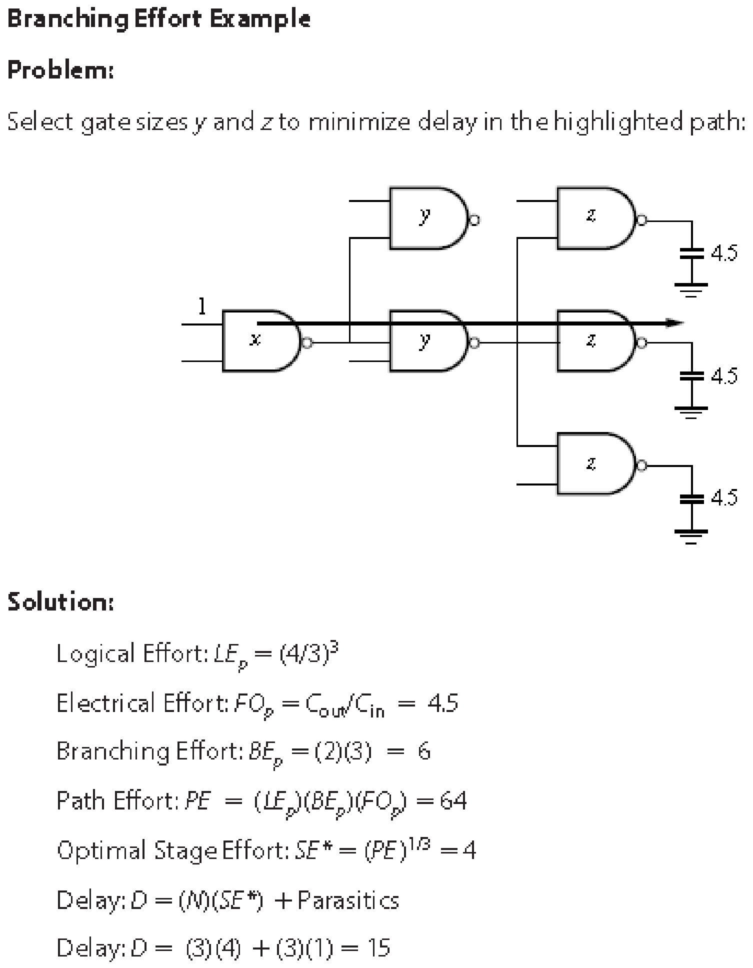
Sometimes a signal drives more than one path, if the paths are identical we can consider Their effect by multiplying up the entire path effort by number of identical paths (also called Branching effort)



$$\text{total\_path\_effort} = \prod(LE \times BE \times FO) = \prod(LE \times BE) \times \frac{C_{\text{load}}}{C_{\text{in}}}$$

If the other paths are not identical (*sideload*), you should write the total Delay and optimize gate sizes based on the minimizing the delay locally (between every two stages derive the optimum gate size from δD/ δsize) = 0, Review Example **6.16**

Example with Branching effort



Branching Effort Example

Problem:

Select gate sizes *y* and *z* to minimize delay in the highlighted path:

1
x
y
y
z
z
z
4.5
4.5
4.5

Solution:

Logical Effort: $LE_p = (4/3)^3$

Electrical Effort: $FO_p = C_{out}/C_{in} = 4.5$

Branching Effort: $BE_p = (2)(3) = 6$

Path Effort: $PE = (LE_p)(BE_p)(FO_p) = 64$

Optimal Stage Effort: $SE^* = (PE)^{1/3} = 4$

Delay: $D = (N)(SE^*) + \text{Parasitics}$

Delay: $D = (3)(4) + (3)(1) = 15$

Note that in backward size calculations

We have to consider the effect of branching

Effort on the size of each previous stage