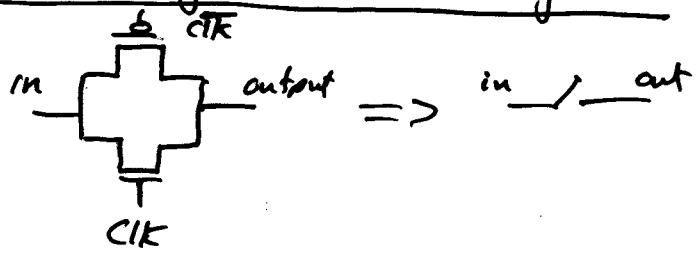
Implementation of Logic Functions using TG's

TG:



1.) Indentify the control signals
2.) Build a truth table
3.) Convert the truth table to a multiplexer-style design by creating a path for each row of T.T. to the output.
4.) The desired outputs are routed from the input to output.

Example 1

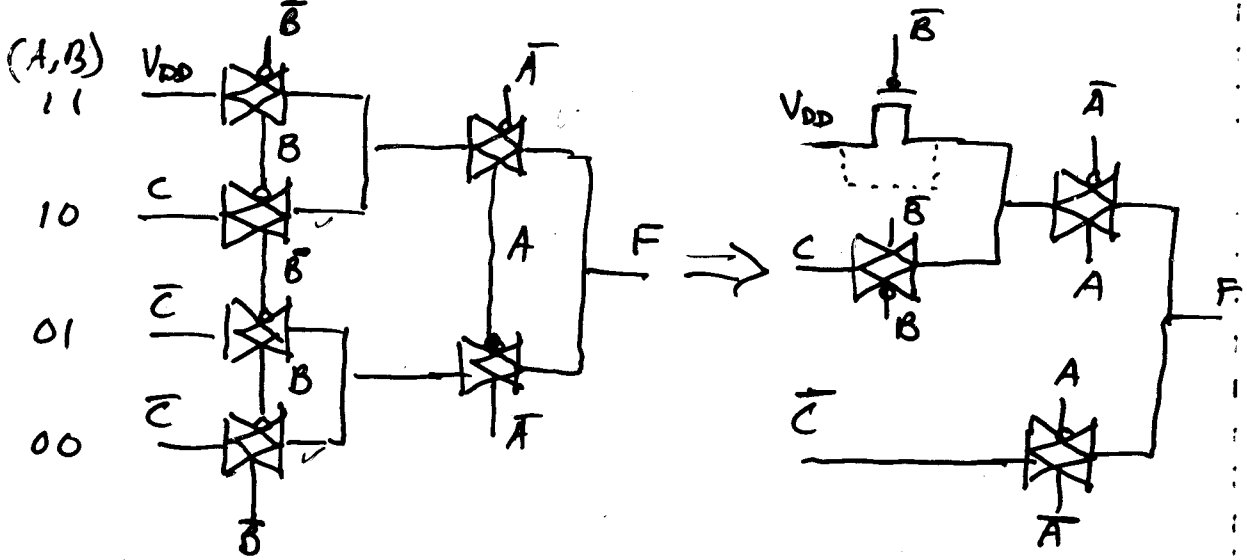
$F = AB + A\bar{B}C + \bar{A}\bar{C}$

1.) Let A & B be control signals.

2.)

A	B	F
0	0	$\bar{C}$
0	1	$\bar{C}$
1	0	C
1	1	1

3.) Two-level MUX



Note that one can combine TG's with static logic gates to achieve efficient implementations of logic functions.

Example 2 (Prob. 7.4c)

$F = \overline{(A+B+C)} + \bar{A}\bar{B} = \bar{A}\bar{B}\bar{C} + \bar{A}B$

1.) Let A & B be control signals

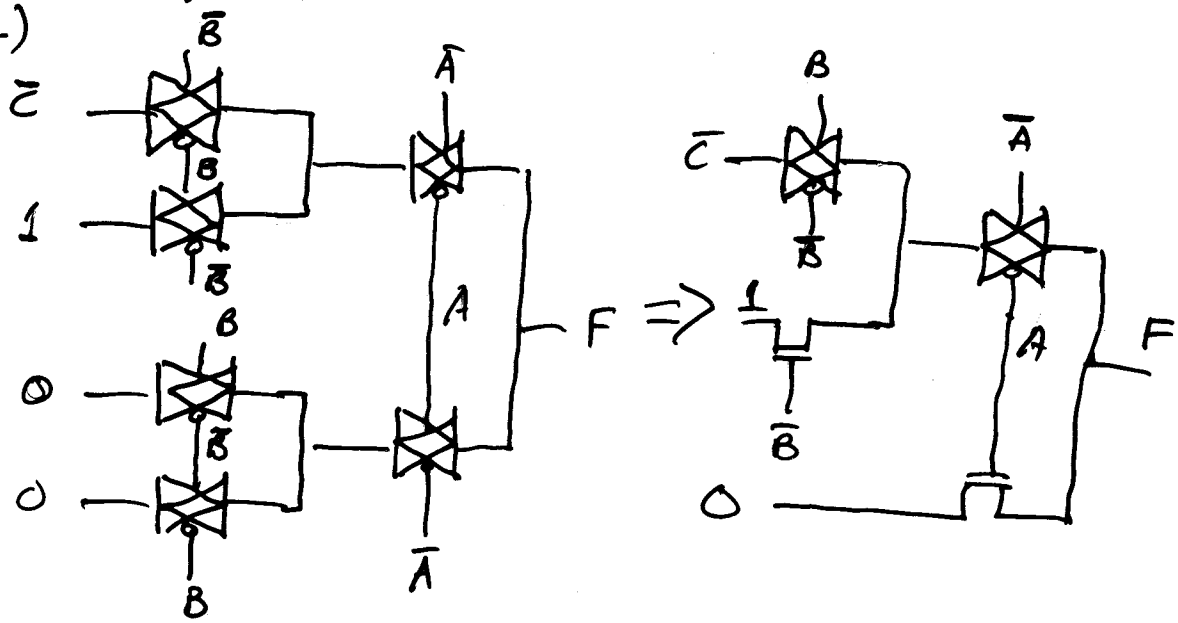
2.)

A	B	C	F
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

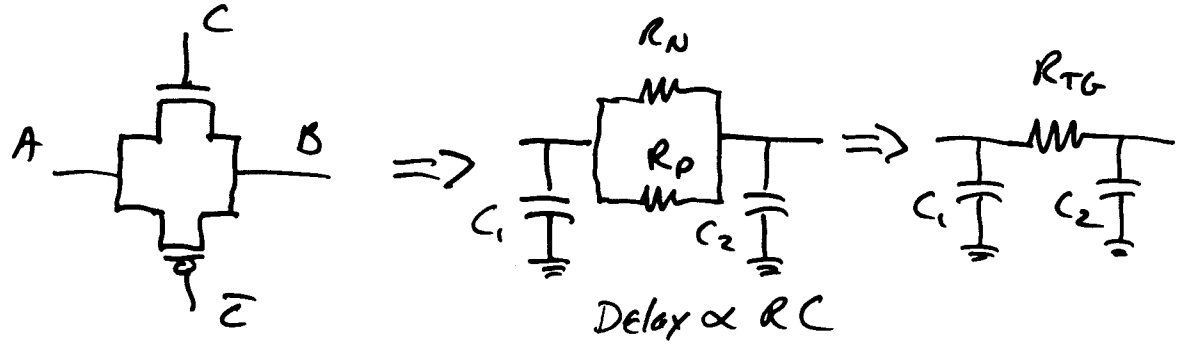
$\Rightarrow$

A	B	F
0	0	$\bar{C}$
0	1	1
1	0	0
1	1	0

3.)



CMOS Transmission Gate Delays



b) What is $R_{TG}(R)$?

a.) For the propagation of V_{DD}

$$R_{TG} = R_N \| R_P \cong (2R_{eqn}) \| R_{eqp} = 2R_{eqn} \| 2.4R_{eqn} \approx R_{eqn}$$

↑ This approximates the NMOS shutting off

b.) For propagation of a zero.

$$R_{TG} = R_N \| R_P \cong R_{eqn} \| 2(2.4R_{eqn}) = R_{eqn} \| 4.8R_{eqn} \cong R_{eqn}$$

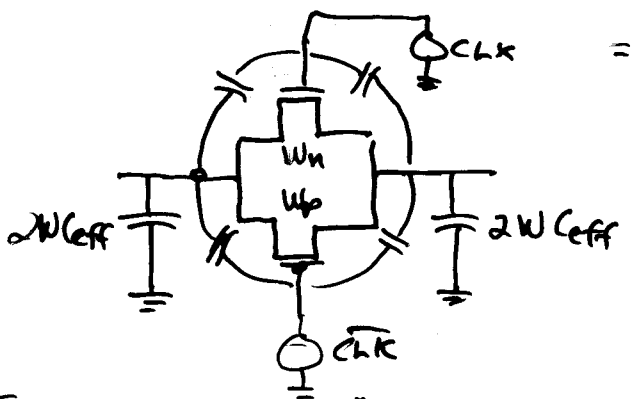
↑ Approximating the R_{eqp} for a PMOS shutting off

2.) What is C_1 and C_2 ($C_1 = C_2$)?

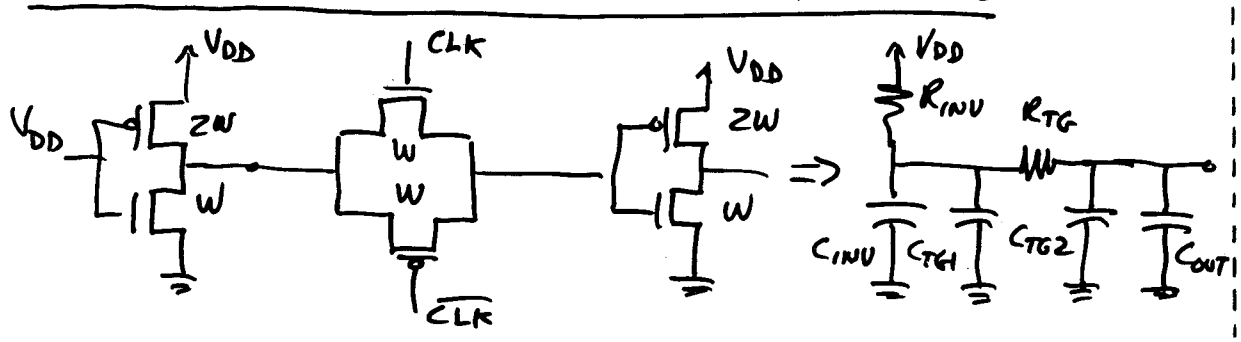
a.) In off state: $C_1 = C_{in} = C_2 = C_{out} = C_{eff}(W_n + W_p) = 2WC_{eff}$

b.) In on state: $C_1 = C_2 = C_{eff}(W_n + W_p) + \frac{1}{2}C_g(W_n + W_p)$

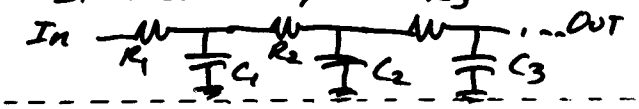
$= 2WC_{eff} + WC_g$



Transmission Gate Driven and Loaded by Inverters



Elmore's Delay:



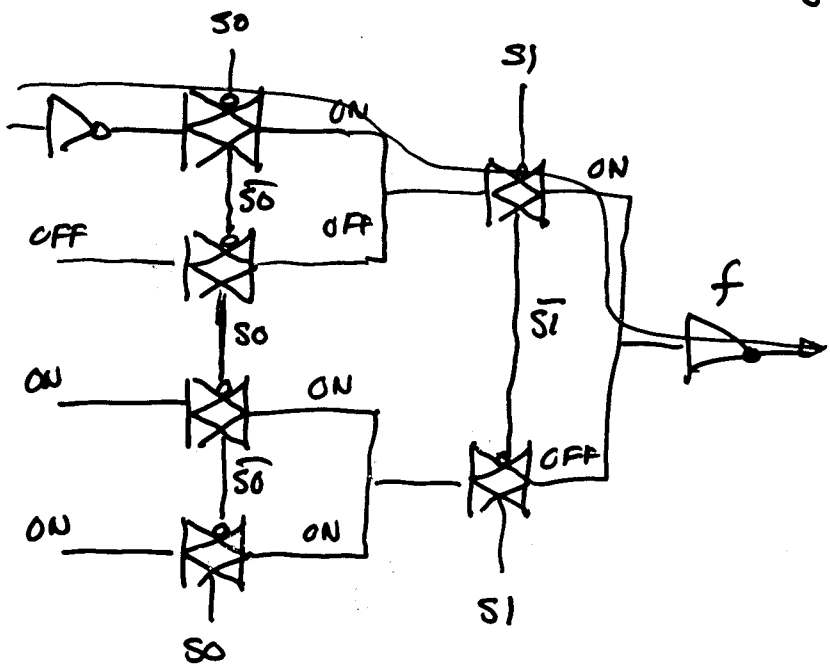
Elmore's Delay $= R_1C_1 + (R_1 + R_2)C_2 + (R_1 + R_2 + R_3)C_3 + \cdots$

For the above,

Elmore Delay = $R_{INV}(C_{INV}+C_{TG1})+(R_{INV}+R_{TG})(C_{TG2}+C_{out})$

Example 7.4

Find the Elmore delay for the following -



To be cont'd