Topics

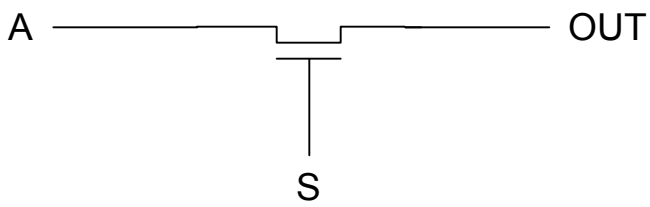
- CMOS Design
- Multi-input delay analysis

Dept. of Electrical and Computer Engineering
University of Connecticut

CMOS Logic Implementations

- Transmission Gate

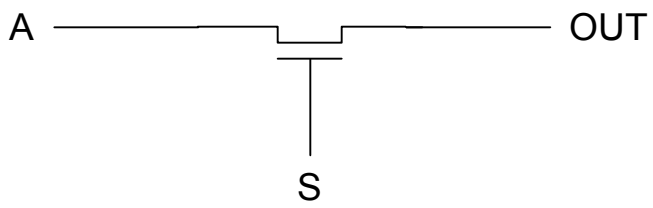
A	S	OUT
0	0	Z
0	1	0
1	0	Z
1	1	1



Dept. of Electrical and Computer Engineering
University of Connecticut

CMOS Logic Implementations

- Transmission Gate

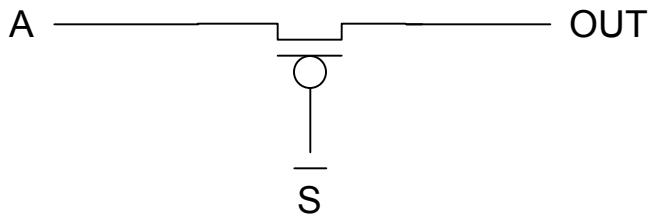


  - When S is low, the output is at high impedance
  - When S is high, the output follows A
    - However, OUT can only rise to V_{DD}-$V_{T,}$ at which point the transistor will shut off

Dept. of Electrical and Computer Engineering
University of Connecticut

CMOS Logic Implementations

- Transmission Gate



  - When S is high, the output is at high impedance
  - When S is low, the output follows A
    - However, OUT can only fall to $V_{T,}$ at which point the transistor will shut off

Dept. of Electrical and Computer Engineering
University of Connecticut

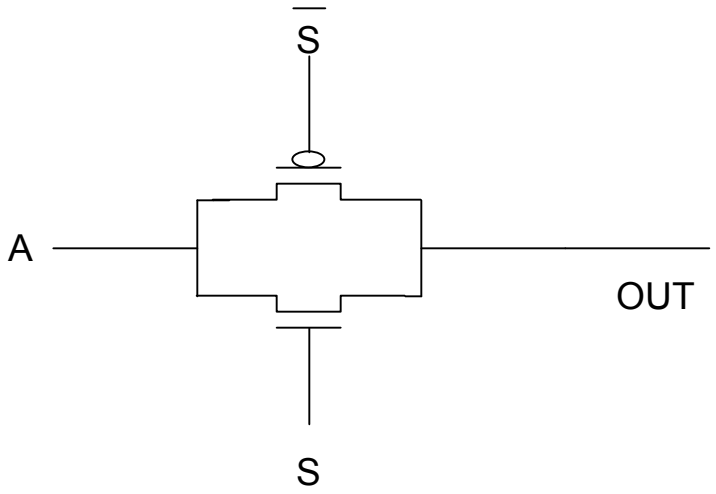
CMOS Logic Implementations

- Transmission gate
  - Requires both nMOS and pMOS transistors
  - Single nMOS or single pMOS will cause signal degradation

Dept. of Electrical and Computer Engineering
University of Connecticut

CMOS Logic Implementations

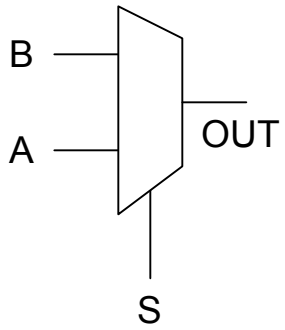
- Transmission Gate



Dept. of Electrical and Computer Engineering
University of Connecticut

CMOS Logic Implementations

- Multiplexer



A	B	S	OUT
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

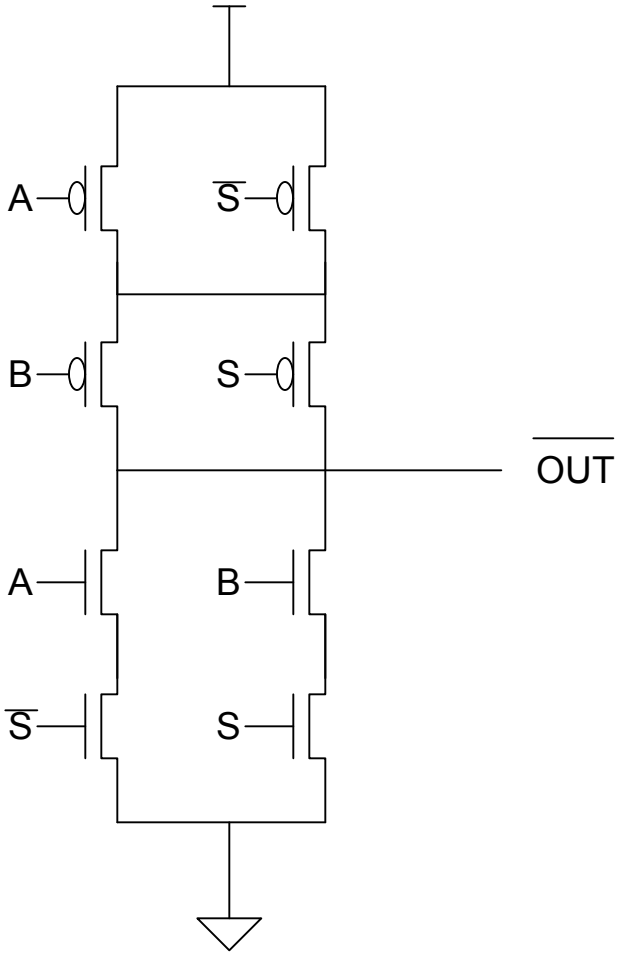
$$OUT = A\overline{S} + BS$$

Dept. of Electrical and Computer Engineering
University of Connecticut

CMOS Logic Implementation

- Multiplexer

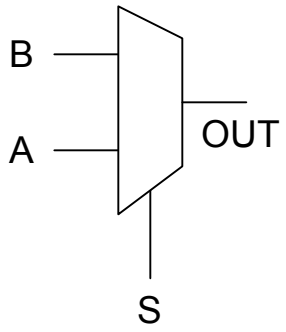
$$OUT = A\overline{S} + BS$$

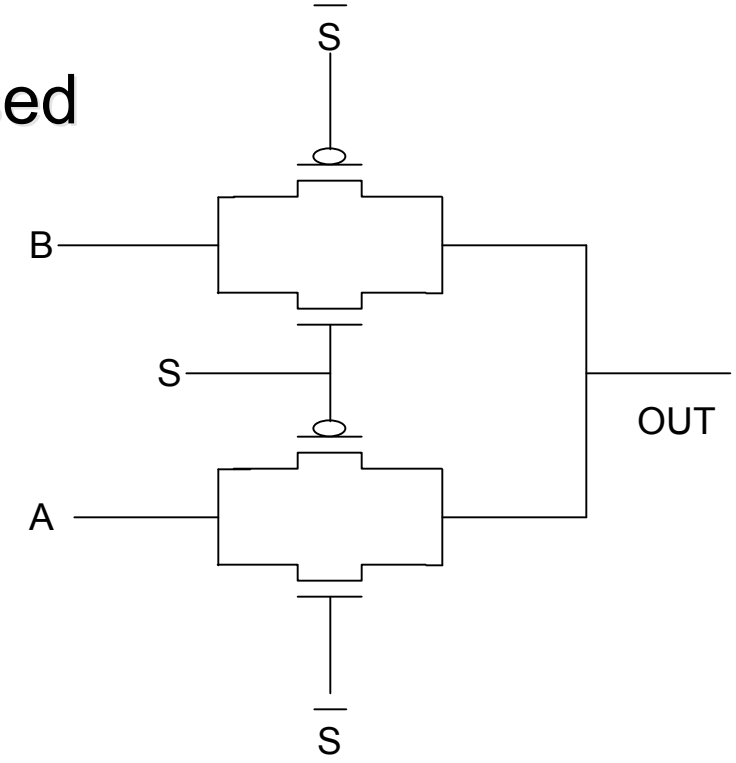


Dept. of Electrical and Computer Engineering
University of Connecticut

CMOS Logic Implementations

- Multiplexer
  - Transmission gate based

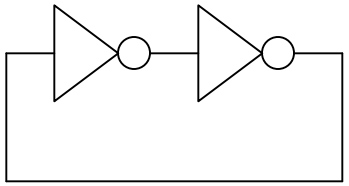




Dept. of Electrical and Computer Engineering
University of Connecticut

CMOS Logic Implementations

- Memory
  - Use feedback loops to store bits

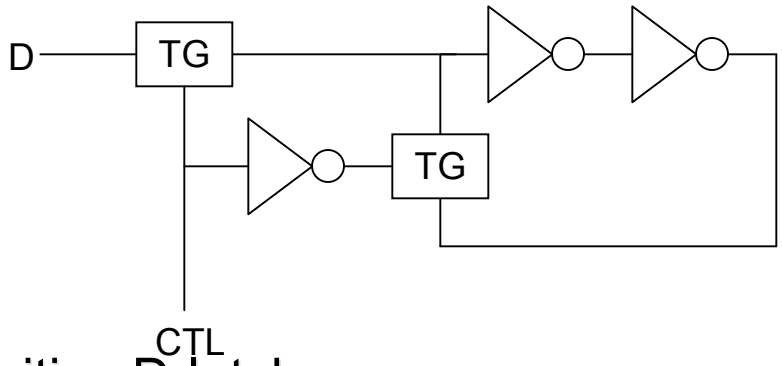


  - How do you get the bit in the loop?

Dept. of Electrical and Computer Engineering
University of Connecticut

CMOS Logic Implementations

- Memory
  - Use transmission gates to control entry to the loop



  - Level sensitive D-latch

Dept. of Electrical and Computer Engineering
University of Connecticut

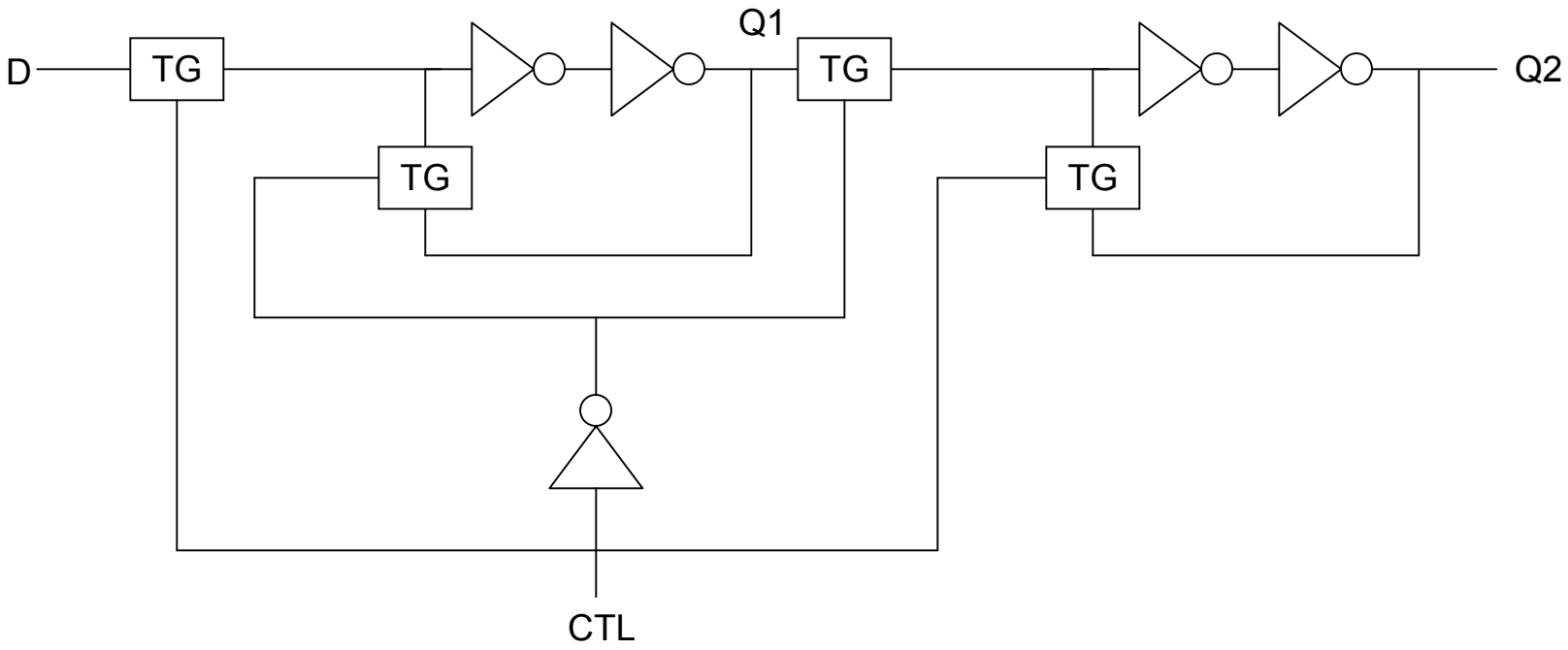
CMOS Logic Implementations

- Latches
  - Level sensitive latches do not allow you to isolate output from input when control is high.
  - Solution is to use a master-slave setup where master latches input and then slave latches the output

Dept. of Electrical and Computer Engineering
University of Connecticut

CMOS Logic Implementations

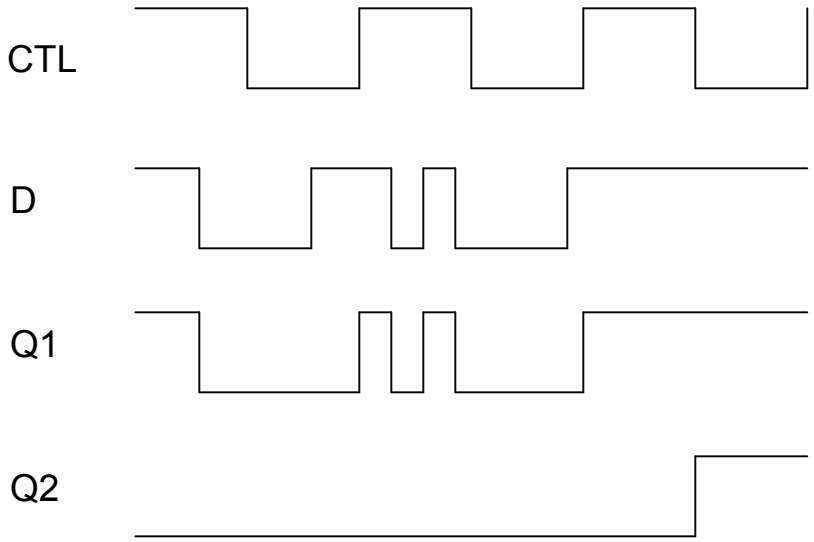
- Master -slave latch



Dept. of Electrical and Computer Engineering
University of Connecticut

CMOS Logic Implementations

- Master -slave latch



- Negative edge-triggered flip-flop

Dept. of Electrical and Computer Engineering
University of Connecticut

CMOS Logic

- Setup time is how much time before the clock edge that the data should be ready
  - If setup time is not met, the data will not have time to get through transmission gate and into the feedback loop
- Hold time is the time that the data is required to be stable after the clock edge.
  - If hold time is not met, invalid data may get past the transmission gate and into the feedback loop.

Dept. of Electrical and Computer Engineering
University of Connecticut

CMOS Logic

- Flip-Flops

D	Q0	Q1
0	X	0
1	X	1

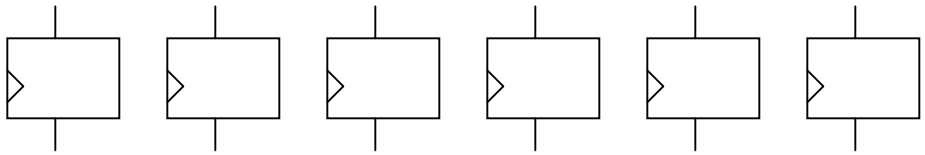
S	R	Q0	Q1
0	0	0	0
0	0	1	1
0	1	X	0
1	0	X	1
1	1	X	X

J	K	Q0	Q1
0	0	0	0
0	0	1	1
0	1	X	0
1	0	X	1
1	1	0	1
1	1	1	0

Dept. of Electrical and Computer Engineering
University of Connecticut

CMOS Logic

- Registers
  - N-bit collection of flip-flops



Dept. of Electrical and Computer Engineering
University of Connecticut

CMOS Logic Gate Design

- How do you characterize delay for a gate more complicated than an inverter?

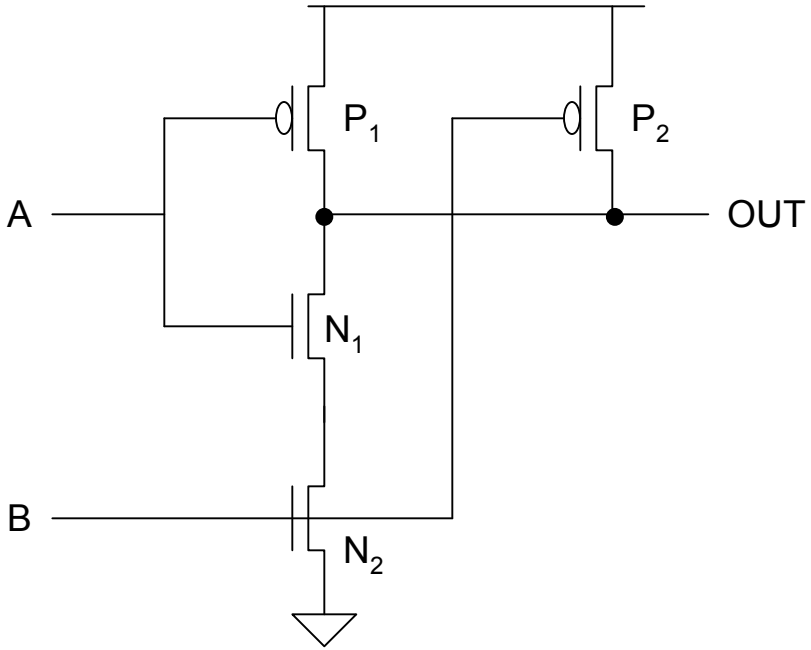
$$t_d = A\frac{C_L}{k_{eff}}$$

- k_{eff} is determined by the structure of the logic gate

Dept. of Electrical and Computer Engineering
University of Connecticut

Delay time analysis

- Multi-input gates



•For pull down (falling delay time)

$$k_{neff} = \frac{1}{\frac{1}{k_{n1}} + \frac{1}{k_{n2}}} = \frac{k_n}{2}$$

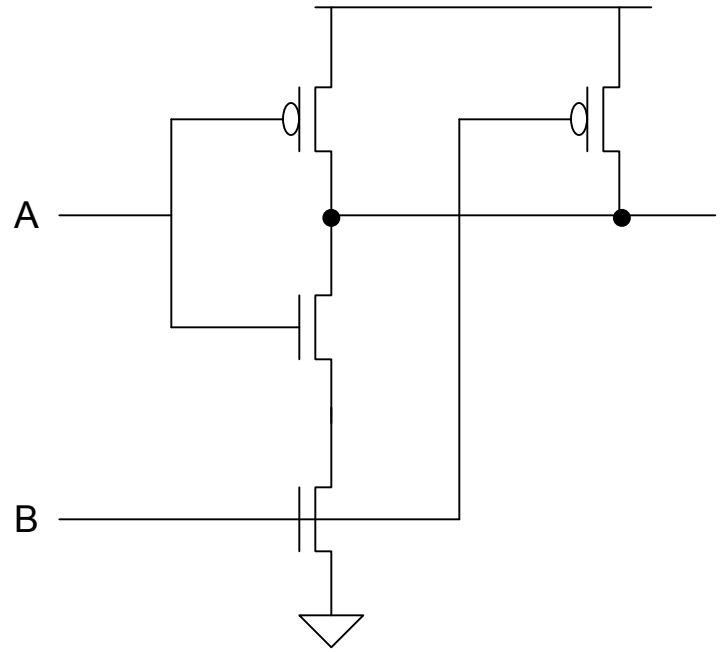
•For pull up (rising delay time)

$$k_{peff} = k_p$$

Dept. of Electrical and Computer Engineering
University of Connecticut

CMOS Logic Implementations

- NAND



$$k_{eff-n} = \frac{k_n}{2}$$

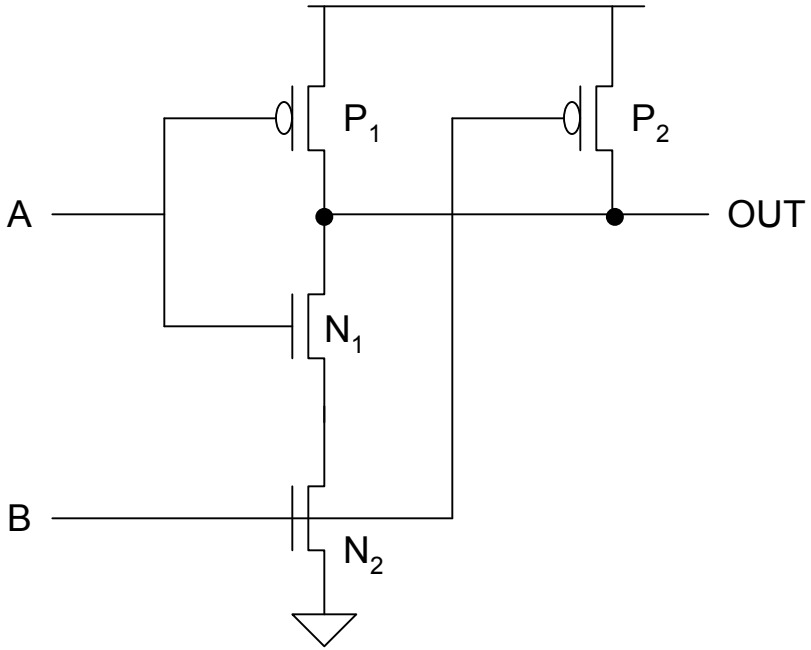
$$k_{eff-p} = k_p$$

Falling delay has doubled

Dept. of Electrical and Computer Engineering
University of Connecticut

Delay time analysis

- Multi-input gates



•For equal delay times

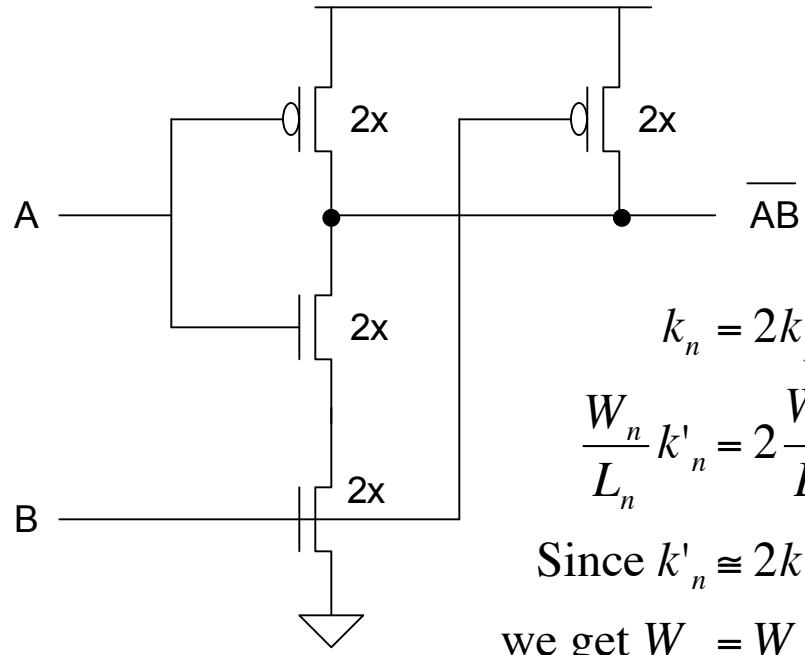
$$k_n = 2k_p$$

$$\frac{W_n}{L_n} k'_n = 2\frac{W_p}{L_p} k'_p$$

Dept. of Electrical and Computer Engineering
University of Connecticut

CMOS Logic Implementations

- NAND



$$k_n = 2k_p$$

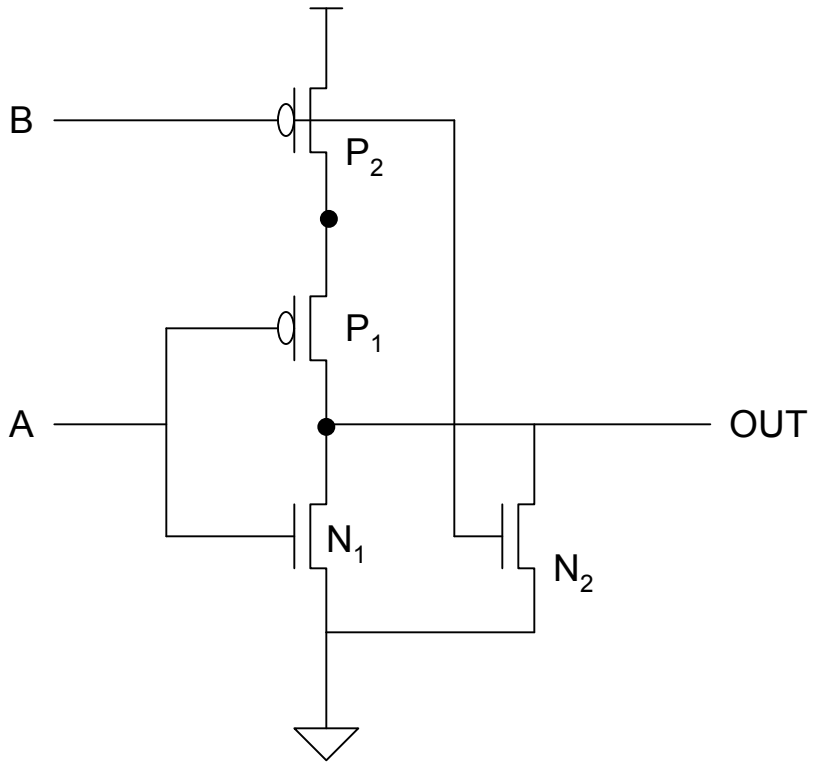
$$\frac{W_n}{L_n} k'_n = 2\frac{W_p}{L_p} k'_p$$

Since $k'_n \cong 2k'_p$ and assume $L_n = L_p$

we get $W_n = W_p$

Dept. of Electrical and Computer Engineering
University of Connecticut

Delay time analysis



•For pull down (falling delay time)

$k_{n-eff} = k_n$ (single transistor on)

•For pull up (rising delay time)

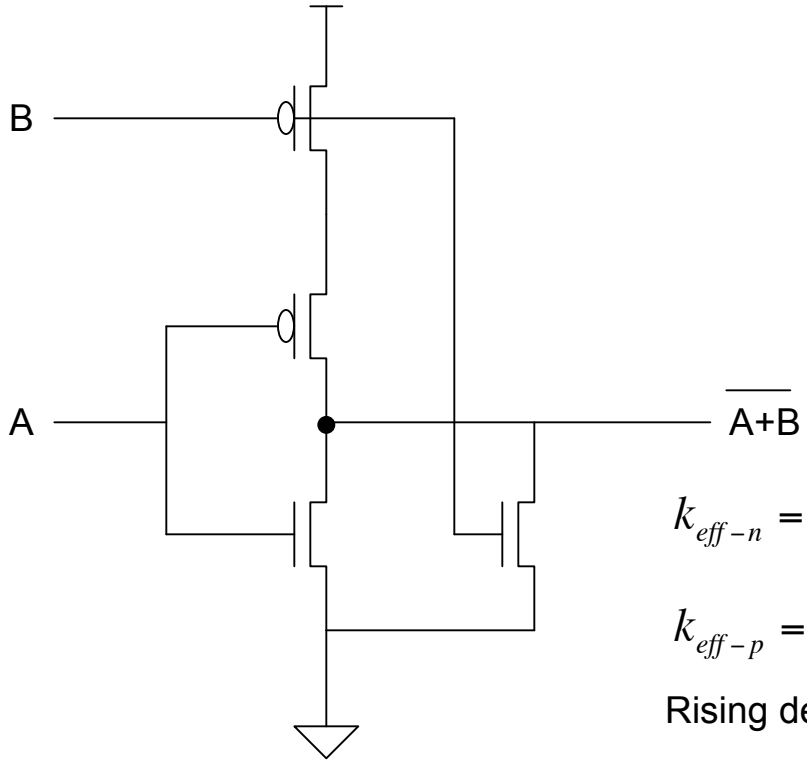
$$k_{p-eff} = \frac{1}{\frac{1}{k_{p1}} + \frac{1}{k_{p2}}}$$

$$= \frac{k_p}{2}$$

Dept. of Electrical and Computer Engineering
University of Connecticut

CMOS Logic Implementations

- NOR



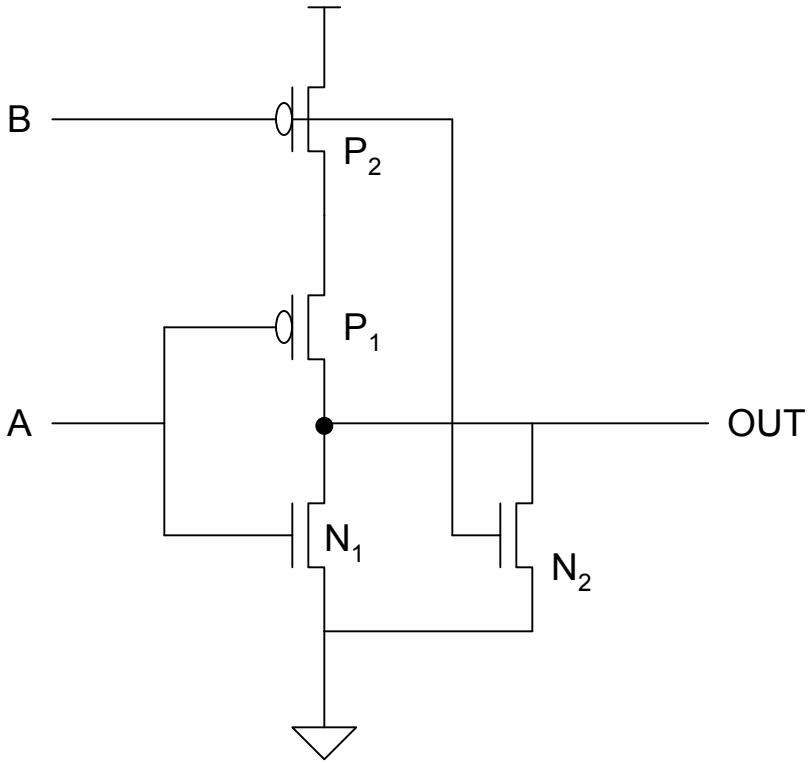
$$k_{eff-n} = k_n$$

$$k_{eff-p} = \frac{k_p}{2}$$

Rising delay has doubled

Dept. of Electrical and Computer Engineering
University of Connecticut

Delay time analysis



•For equal delay times

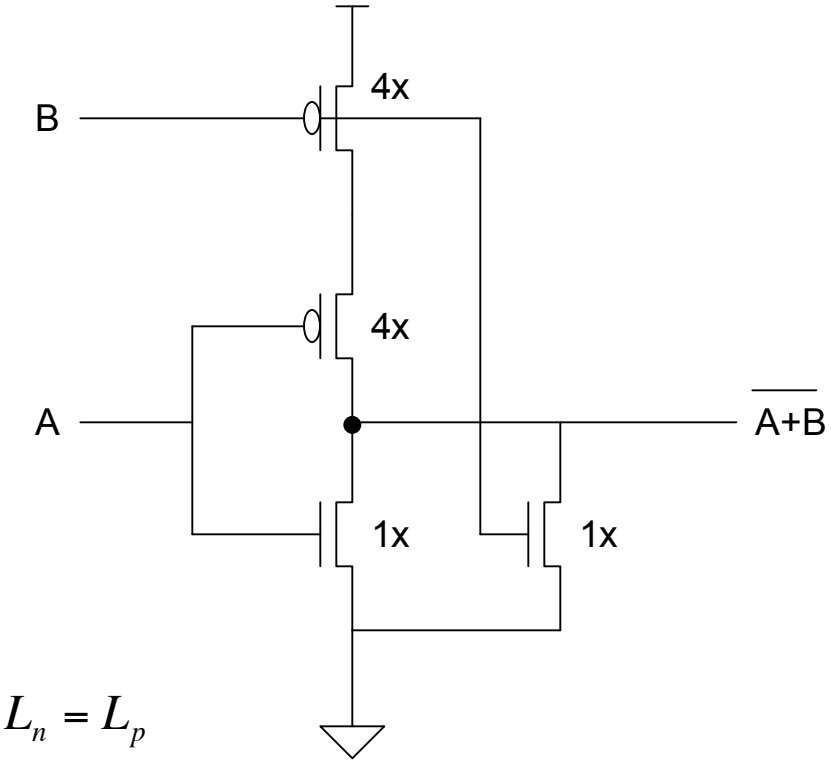
$$2k_n = k_p$$

$$2\frac{W_n}{L_n}k'_n = \frac{W_p}{L_p}k'_p$$

Dept. of Electrical and Computer Engineering
University of Connecticut

CMOS Logic Implementations

- NOR



$$2k_n = k_p$$

$$2\frac{W_n}{L_n}k'_n = \frac{W_p}{L_p}k'_p$$

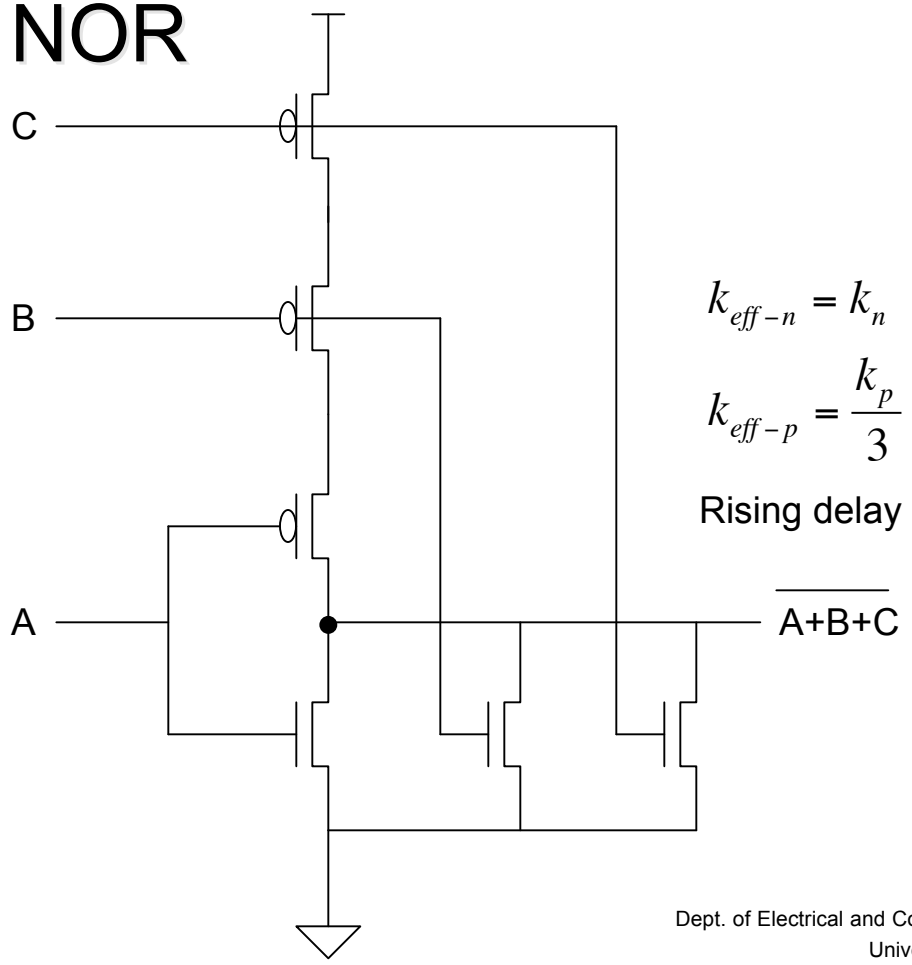
Since $k'_n \cong 2k'_p$ and assume $L_n = L_p$

we get $4W_n = W_p$

Dept. of Electrical and Computer Engineering
University of Connecticut

CMOS Logic Implementations

- Multi-input NOR



$$k_{eff-n} = k_n$$

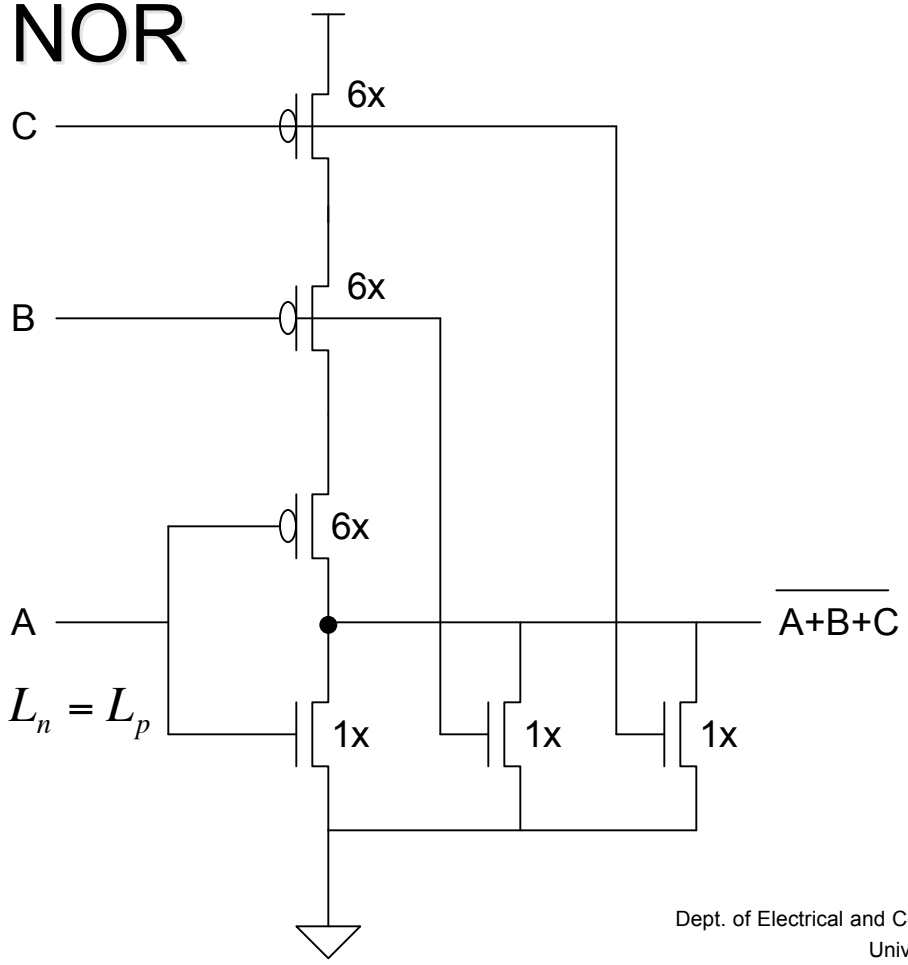
$$k_{eff-p} = \frac{k_p}{3}$$

Rising delay has tripled

Dept. of Electrical and Computer Engineering
University of Connecticut

CMOS Logic Implementations

- Multi-input NOR



$$3k_n = k_p$$

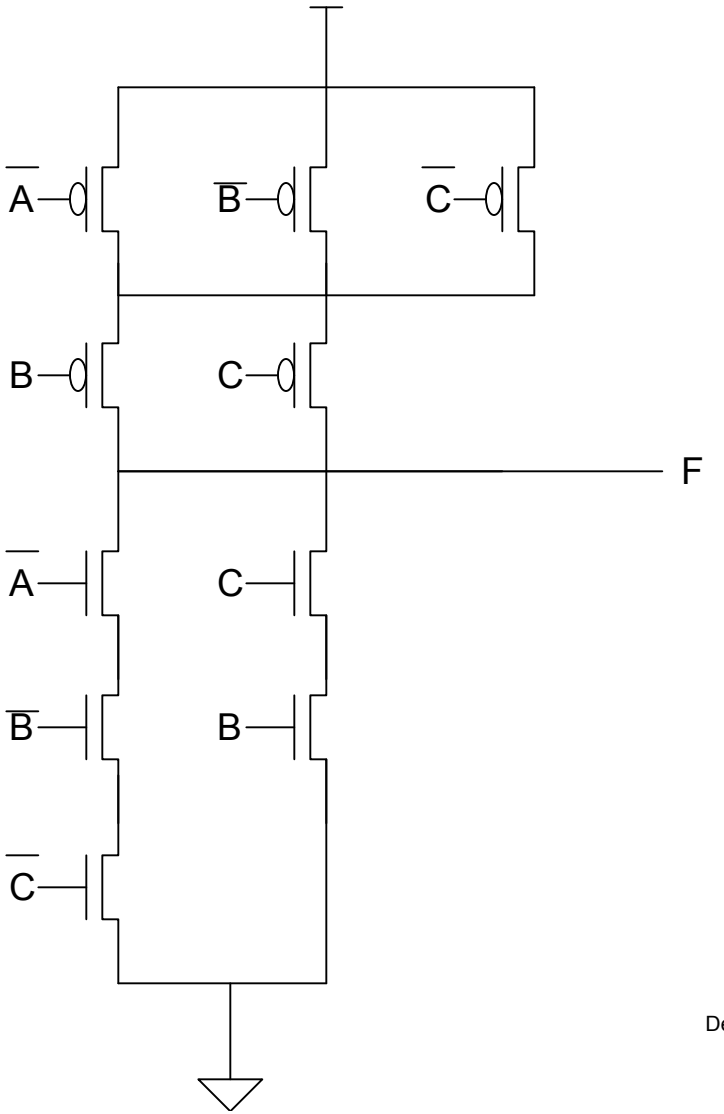
$$3\frac{W_n}{L_n}k'_n = \frac{W_p}{L_p}k'_p$$

Since $k'_n \cong 2k'_p$ and assume $L_n = L_p$

we get $6W_n = W_p$

Dept. of Electrical and Computer Engineering
University of Connecticut

CMOS Logic Implementations



$$k_{eff-n} = \frac{k_n}{3}$$

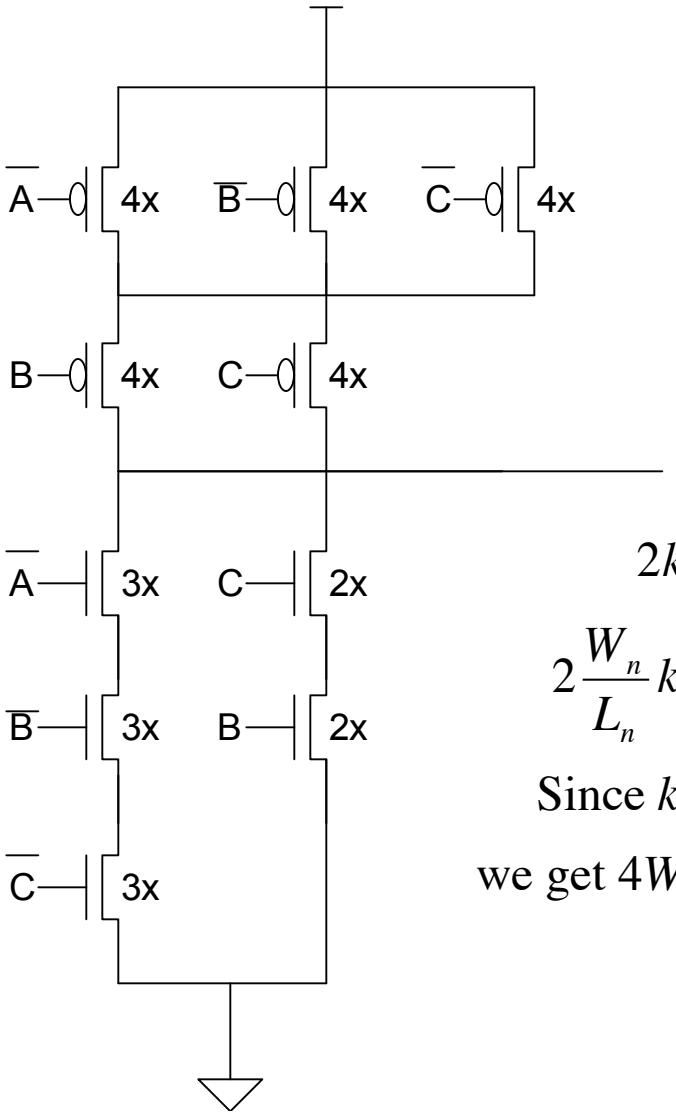
$$k_{eff-p} = \frac{k_p}{2}$$

Falling delay has tripled

Rising delay has doubled

Dept. of Electrical and Computer Engineering
University of Connecticut

CMOS Logic Implementations



$$2k_n = 3k_p$$

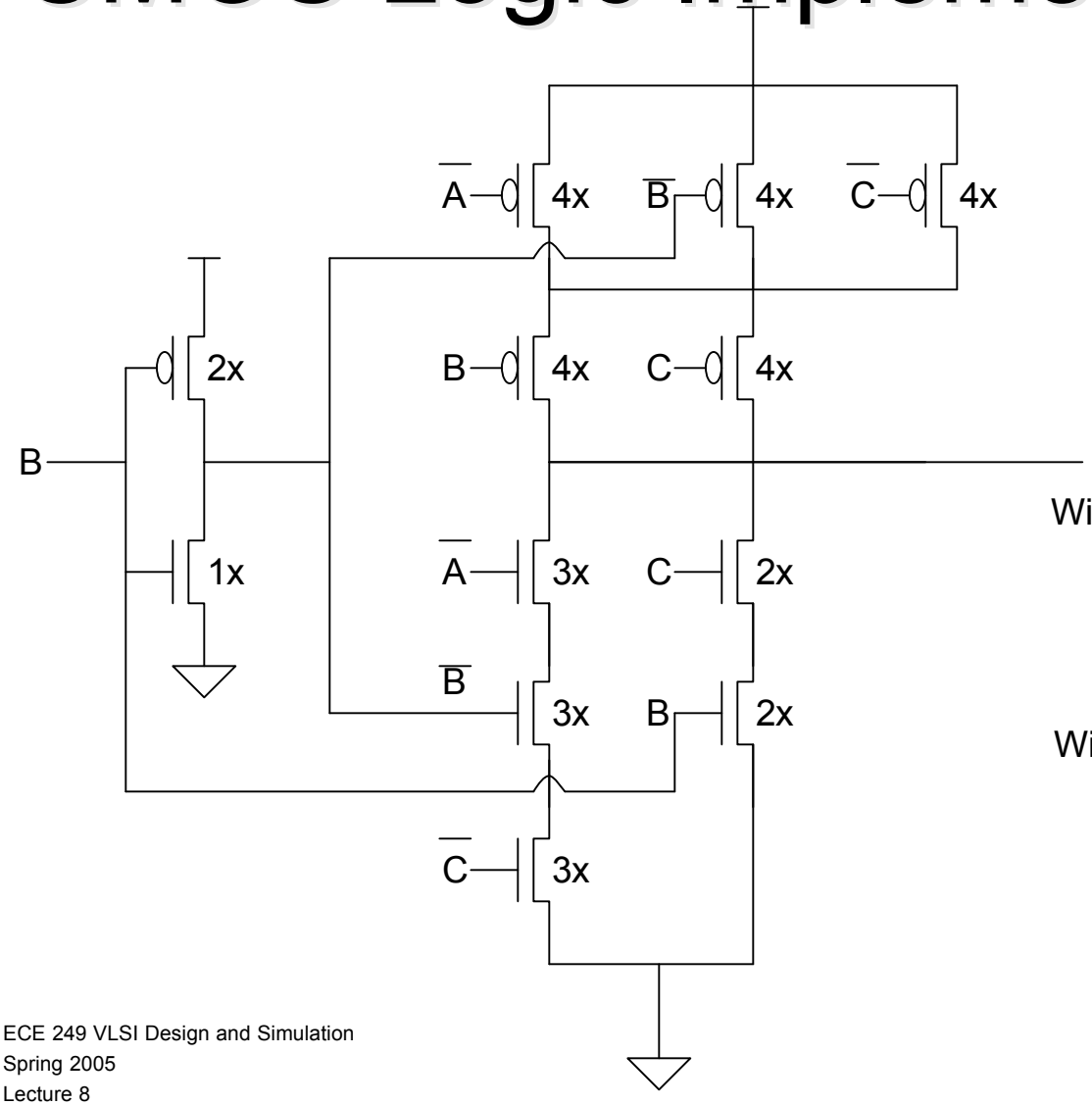
$$2\frac{W_n}{L_n}k'_n = 3\frac{W_p}{L_p}k'_p$$

Since $k'_n \cong 2k'_p$ and assume $L_n = L_p$

we get $4W_n = 3W_p$

Dept. of Electrical and Computer Engineering
University of Connecticut

CMOS Logic Implementations



Without sizing

$$t_{p1} = A\left(\frac{C_{in}}{k}\right) + A\left(\frac{C_L}{\frac{k}{3}}\right)$$

With sizing

$$t_{p2} = A\left(\frac{\frac{7}{3}C_{in}}{k}\right) + A\left(\frac{C_L}{k}\right)$$

Dept. of Electrical and Computer Engineering
University of Connecticut

CMOS Logic Gate Delays

- Increasing transistor sizes can reduce the delays, but it
  - Increases area
  - Increases load capacitance for driving gates
- Multi-input gates may not always be good

Dept. of Electrical and Computer Engineering
University of Connecticut

Next class

- Gate Delays
- Logical Effort
- Chapter 6.2

Dept. of Electrical and Computer Engineering
University of Connecticut