

**EE434**  
**ASIC and Digital Systems**

**Final Exam**

**May 5, 2015. (1pm – 3pm)**

**Instructor: Dae Hyun Kim ([daehyun@eecs.wsu.edu](mailto:daehyun@eecs.wsu.edu))**

**Name:**

**WSU ID:**

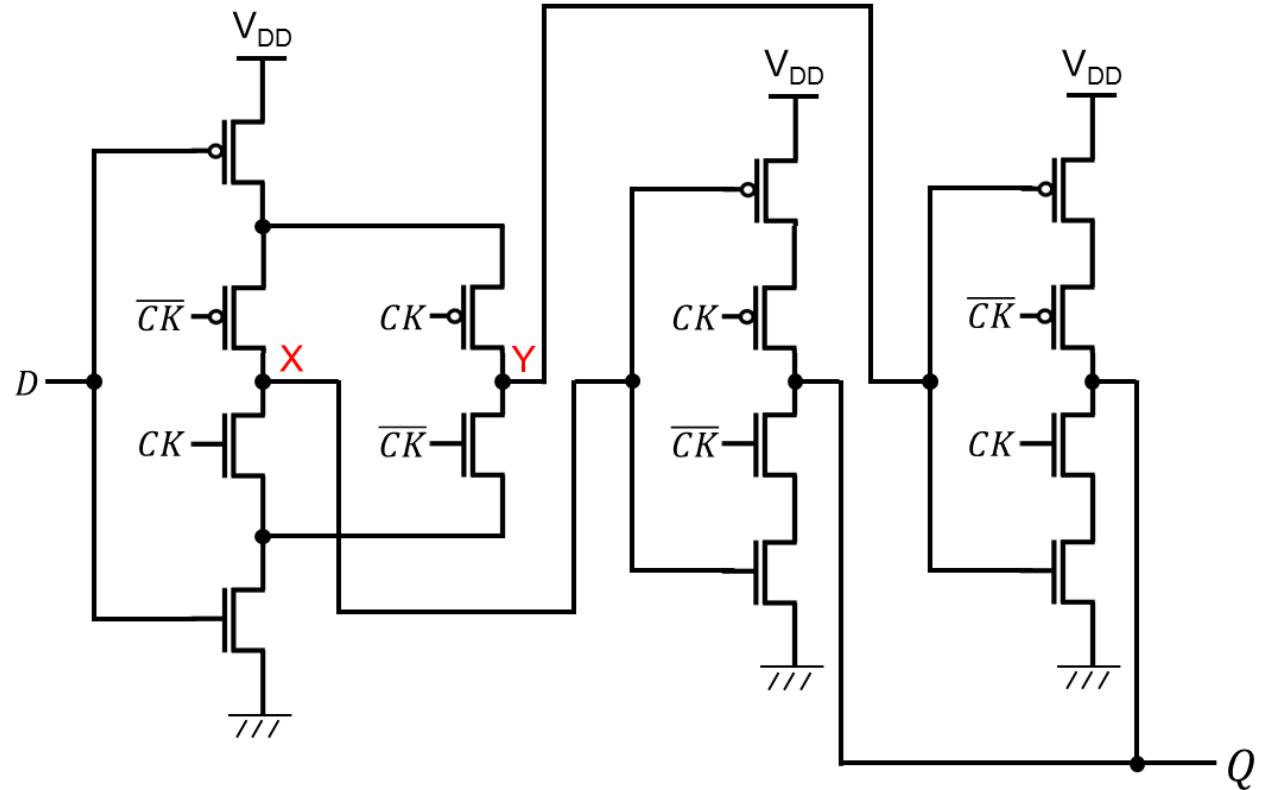
Problem	Points	
1	10	
2	10	
3	10	
4	10	
5	10	
6	10	
7	10	
8	10	
Total	80	

\* Allowed: Textbooks, cheat sheets, class notes, notebooks, calculators, watches

\* Not allowed: Electronic devices (smart phones, tablet PCs, laptops, etc.) except calculators and watches

### Problem #1 (CMOS gates, 10 points).

What does the following circuit do? Describe the function of the circuit in as much detail as possible (D: input, Q: output, CK: clock).



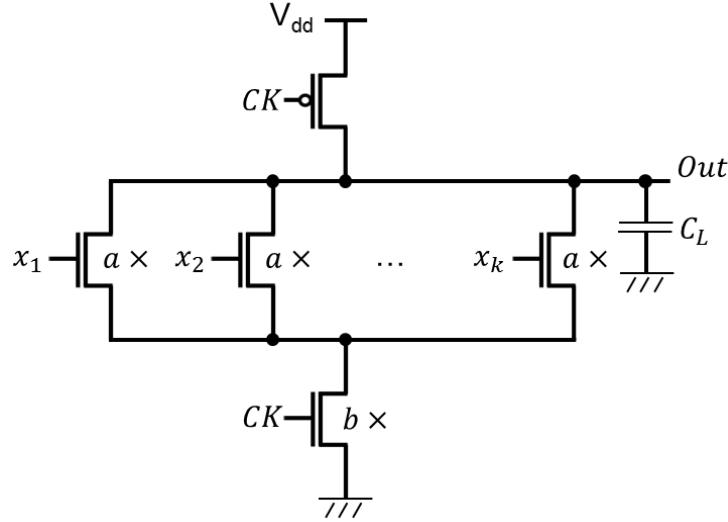
When  $CK$  goes to 0 (falling edge),  $X$  is floating,  $Y$  is  $\bar{D}$ , and  $Q$  is  $\bar{X}(= D)$ .

When  $CK$  goes to 1 (rising edge),  $Y$  is floating,  $X$  is  $\bar{D}$ , and  $Q$  is  $\bar{Y}(= D)$ .

Therefore, this is a dual-edge-triggered D flip flop (it captures  $D$  twice per clock cycle, at rising and falling edges).

## Problem #2 (Transistor Sizing Under Timing Constraints, 10 points).

Let's design a  $k$ -input NOR gate ( $k \geq 10$ ) using the dynamic CMOS design methodology. The following shows a schematic of the  $k$ -input NOR gate.



Our objective is to minimize the total width,  $Width = a \cdot k + b$  and satisfy given timing constraints at the same time. Two timing constraints are given to us as follows:

- Setup time: Elmore delay  $\leq 4 \cdot R_n \cdot C_L$
- Hold time: Elmore delay  $\geq \frac{1}{4} \cdot R_n \cdot C_L$

$R_n$  is the resistance of a 1X NMOS transistor.  $\mu_n = 2 \cdot \mu_p$ . Ignore all the parasitic capacitances. All the transistors for  $x_1 \sim x_k$  are upsized to  $aX$  and the transistor for  $CK$  is upsized to  $bX$  ( $a$  and  $b$  are real numbers). Find  $a$  and  $b$  minimizing the total width and satisfying the timing constraints.

The worst-case resistance of the pull-down network:  $\frac{R_n}{a} + \frac{R_n}{b}$

The worst-case delay:  $\left(\frac{R_n}{a} + \frac{R_n}{b}\right) C_L$

Constraints:  $\frac{1}{4} R_n C_L \leq \left(\frac{R_n}{a} + \frac{R_n}{b}\right) C_L \leq 4 R_n C_L \Rightarrow \frac{1}{4} \leq \left(\frac{1}{a} + \frac{1}{b}\right) \leq 4$

Width:  $W(a, b) = ak + b$

Let  $\frac{1}{a} + \frac{1}{b}$  be  $c$ , then  $W(a, b) = b + \frac{bk}{bc-1}$  where  $\frac{1}{4} \leq c \leq 4$ .

$$\frac{\partial W}{\partial b} = 1 + \frac{k(bc-1)-bkc}{(bc-1)^2} = 1 - \frac{k}{(bc-1)^2} = \frac{(bc-1)^2-k}{(bc-1)^2} = 0 \Rightarrow b = \frac{1}{c}(1 \pm \sqrt{k})$$

$b$	$\frac{1}{c}(1 - \sqrt{k})$	$\frac{1}{c}(1 + \sqrt{k})$
$\frac{\partial W}{\partial b}$	+	0

$k \geq 10, \frac{1}{4} \leq c \leq 4 \Rightarrow \frac{1}{c}(1 + \sqrt{k}) > 1$ , so  $W$  is minimized when  $b$  is  $\frac{1}{c}(1 + \sqrt{k})$ .

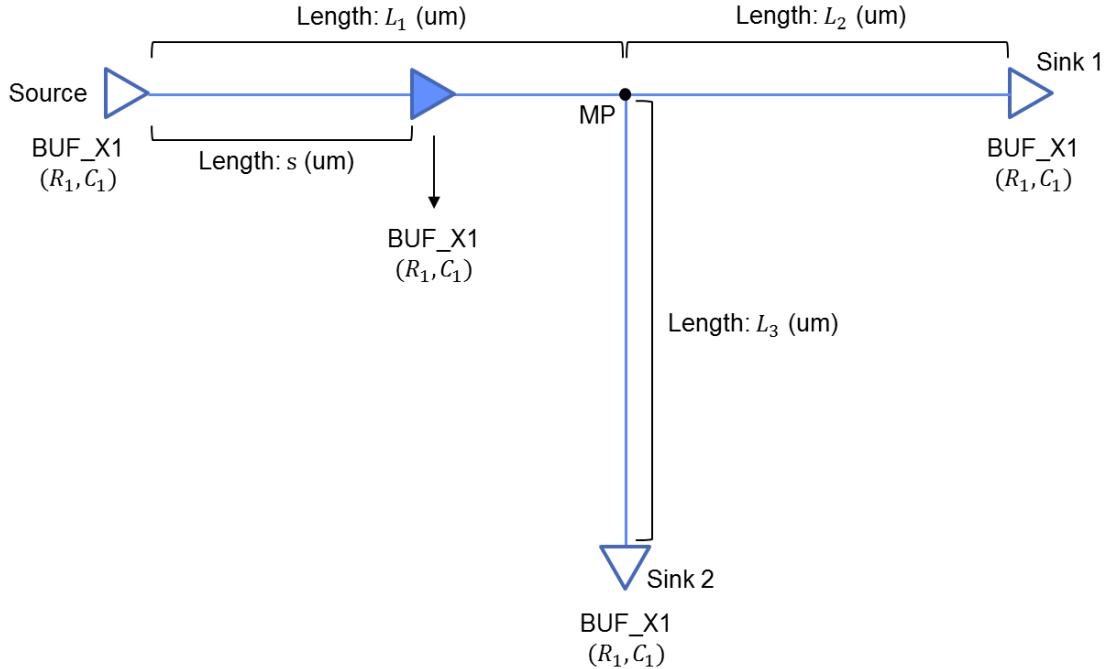
$$\Rightarrow a = \frac{b}{bc - 1} = \frac{1}{c} \left( 1 + \frac{1}{\sqrt{k}} \right)$$

$$W = ak + b = \frac{1}{c} \left( k + 2\sqrt{k} + 1 \right) = \frac{(\sqrt{k} + 1)^2}{c}$$

We set  $c$  to 4 to minimize  $W$ .

Answer)  $a = \frac{1}{4} \left( 1 + \frac{1}{\sqrt{k}} \right)$        $b = \frac{1}{4} (1 + \sqrt{k})$

### Problem #3 (Buffer Insertion, 10 points).



A source drives two sinks through a net and you are supposed to insert a buffer between the source and the branch point (MP) as shown in the above figure. Find an optimal location of the buffer minimizing the total delay, i.e., represent “s” as a function of the following parameters:

- Output resistance of  $\text{BUF\_X1}$ :  $R_1$
- Input capacitance of  $\text{BUF\_X1}$ :  $C_1$
- Unit wire resistance:  $r$  ( $\Omega/\mu\text{m}$ )
- Unit wire capacitance:  $c$  ( $F/\mu\text{m}$ )
- $\frac{C_1}{c} + L_2 + L_3 < L_1$

$$\text{Delay of the first segment: } \tau_1 = R_1 \cdot (c \cdot s + C_1) + (r \cdot s) \cdot C_1 + \frac{1}{2} \cdot r \cdot c \cdot s^2$$

**Delay of the second segment:**  $\tau_2 = R_1 \cdot (c \cdot (L_1 - s) + c \cdot L_2 + c \cdot L_3 + 2C_1) + (r \cdot (L_1 - s)) \cdot (c \cdot L_2 + c \cdot L_3 + 2C_1) + \frac{1}{2} \cdot r \cdot c \cdot (L_1 - s)^2 + k$  (where  $k$  is a constant. Note that the delay of the downstream of MP is not related to the location of the buffer.)

$$\tau = \tau_1 + \tau_2$$

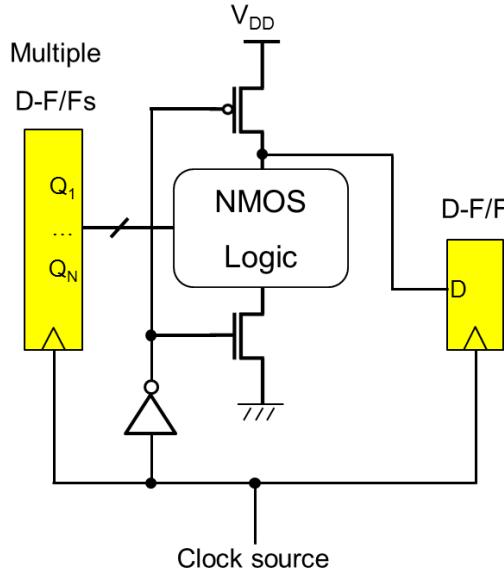
$$\frac{d\tau}{ds} = R_1 \cdot c + r \cdot C_1 + r \cdot c \cdot s - R_1 \cdot c - r(c \cdot L_2 + c \cdot L_3 + 2C_1) - r \cdot c \cdot (L_1 - s) = 0$$

$$s = \frac{r \cdot C_1 + r \cdot c \cdot L_1 + r \cdot c \cdot L_2 + r \cdot c \cdot L_3}{2 \cdot r \cdot c} = \frac{C_1 + c \cdot L_1 + c \cdot L_2 + c \cdot L_3}{2c}$$

$$\therefore s = \frac{1}{2} \left( \frac{C_1}{c} + L_1 + L_2 + L_3 \right)$$

## Problem #4 (Timing Analysis for Dynamic CMOS Circuits, 10 points).

The following figure shows a dynamic CMOS circuit between two pipeline stages.



- Setup time:  $T_s$
- Clock period:  $T_{CLK}$
- D-F/F internal delay:  $T_{CQ}$
- Clock skew: 0
- NMOS logic delay:  $T_{logic}$
- Inverter delay:  $T_v$
- Clock duty cycle: 50%

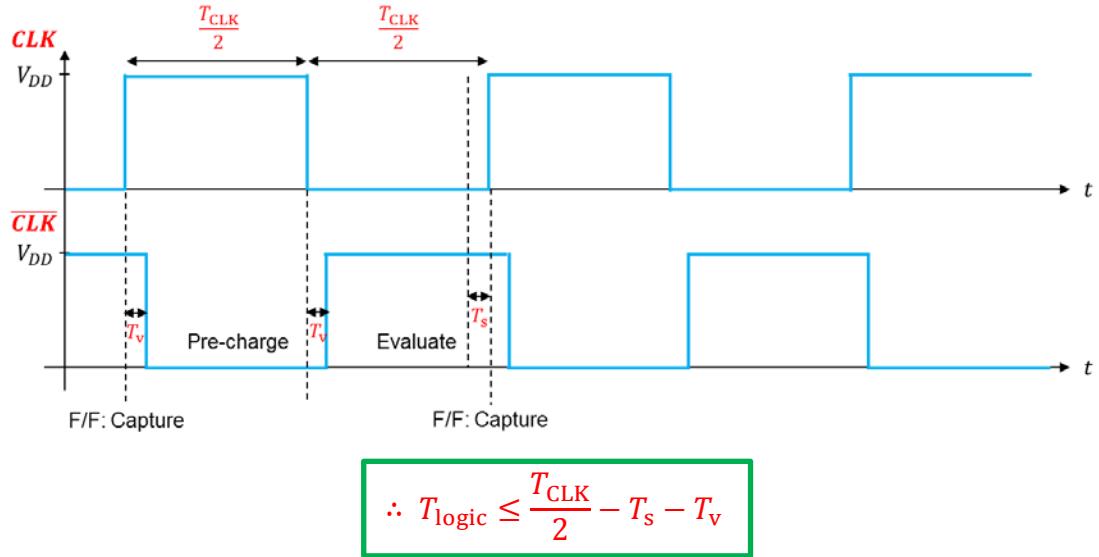
When the clock goes from low to high, the F/Fs capture their input signals. At the same time, the dynamic CMOS circuit starts pre-charging the output node. When the clock goes from high to low, the dynamic CMOS circuit starts evaluating its inputs. The delay of the dynamic CMOS circuit ( $T_{logic}$ ) is actually the time spent to discharge the output node. Derive a new setup time constraint (inequality) for the dynamic CMOS circuit shown above.

The following is the setup time inequality for the single-edge F/F operation:

$$T_s \leq T_{CLK} + T_{skew} - T_{logic} - T_{CQ}$$

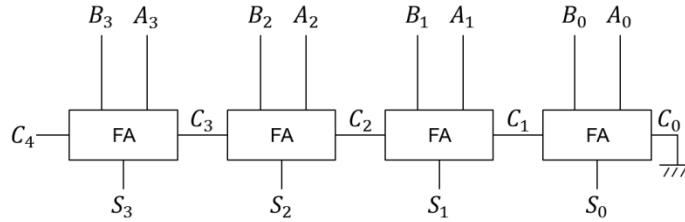
As the following figure shows, when the clock goes from low to high, the D-F/Fs capture their input signals. After  $T_v$ ,  $\overline{CLK}$  goes from high to low and the dynamic CMOS circuit starts pre-charging the output capacitor. When the clock goes from high to low,  $\overline{CLK}$  goes from low to high after  $T_v$  and the dynamic CMOS circuits starts evaluating the

logic. The D-F/Fs will capture their input signals at the next rising edge, so the evaluation should be done at least  $T_s$  before the edge.

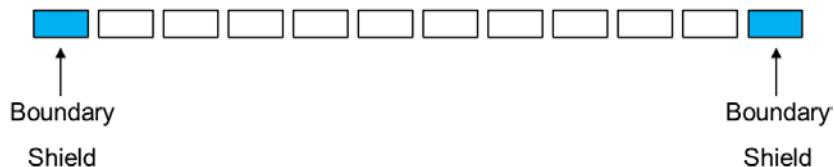


## Problem #5 (Timing Analysis and Coupling in an RCA, 10 points).

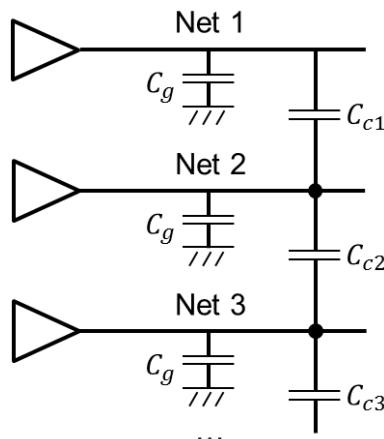
Let's design a four-bit ripple carry adder (RCA) with  $C_0$  tied to ground as shown below.



Due to some physical design constraints, only ten routing tracks are available for the eight primary input signals as follows:

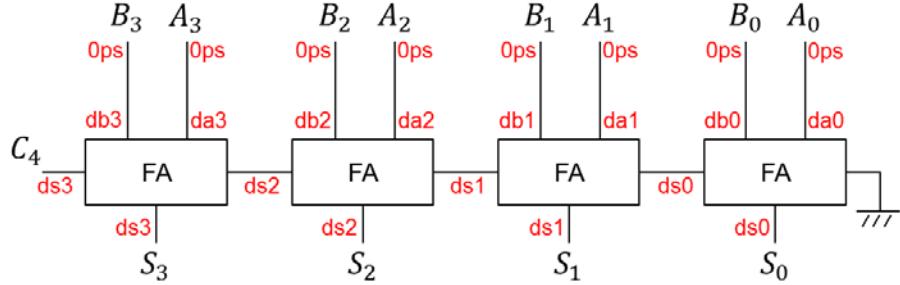


You are supposed to use eight routing tracks for signal and the other two routing tracks for shielding tied to  $V_{DD}$  or  $V_{SS}$ . Wire resistance is negligible and each wire has a ground capacitor ( $C_g$ ) and a coupling capacitor as follows:



- Output resistance of the buffers driving the wires:  $500\Omega$
- $C_g: 50fF$
- $C_c: 25fF$
- Delay of a full adder (from its inputs to both its Sum and Carry-Out):  $40ps$
- Wire delay:  $2 \cdot R \cdot C$  (where  $R$  is the output resistance of the buffer driving the net and  $C$  is the total capacitance of the wire).

Assign the eight primary input signals and the two shielding to the ten routing tracks and compute the delay from the primary inputs to  $S_3$  or  $C_4$ . You should minimize the delay from the primary inputs to  $S_3$  or  $C_4$  when you assign the signals to the routing tracks. (See the next page for an example).



$$ds0 = \text{MAX}(da0, db0) + \varepsilon \text{ where } \varepsilon \text{ is the full adder delay.}$$

$$ds1 = \text{MAX}(da1, db1, ds0) + \varepsilon$$

$$ds2 = \text{MAX}(da2, db2, ds1) + \varepsilon$$

$$ds3 = \text{MAX}(da3, db3, ds2) + \varepsilon$$

### Coupling capacitance



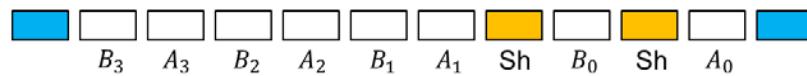
$$(B_3 \ A_3 \ B_2 \ A_2 \ B_1 \ A_1 \ B_0 \ A_0) = (3C_c, 3, 3, 3, 3, 4, 4, 3)$$

$$\Rightarrow \text{delay} = (125, 125, 125, 125, 125, 150, 150, 125)$$



$$\text{Case 1: } (B_3 \ A_3 \ B_2 \ A_2 \ B_1 \ A_1 \ B_0 \ A_0) = (3C_c, 4, 4, 3, 3, 3, 3, 3)$$

$$\Rightarrow \text{delay} = (125, 150, 150, 125, 125, 125, 125, 125)$$



$$\text{Case 2: } (B_3 \ A_3 \ B_2 \ A_2 \ B_1 \ A_1 \ B_0 \ A_0) = (3C_c, 4, 4, 4, 4, 3, 2, 2)$$

$$\Rightarrow \text{delay} = (125, 150, 150, 150, 150, 125, 100, 100)$$

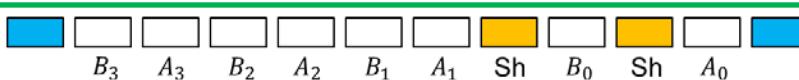
Case 1:  $ds_0 = 125+40 = 165\text{ps}$      $ds_1 = 165+40 = 205\text{ps}$      $ds_2 = 205+40 = 245\text{ps}$

$ds_3 = 245+40 = 285\text{ps}$

Case 2:  $ds_0 = 100+40 = 140\text{ps}$      $ds_1 = 150+40 = 190\text{ps}$      $ds_2 = 190+40 = 230\text{ps}$

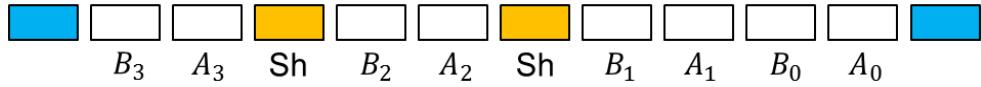
$ds_3 = 230+40 = 270\text{ps}$

Thus, Case 2 is the best.



Delay = 270ps

Example) Suppose the following is my assignment result.

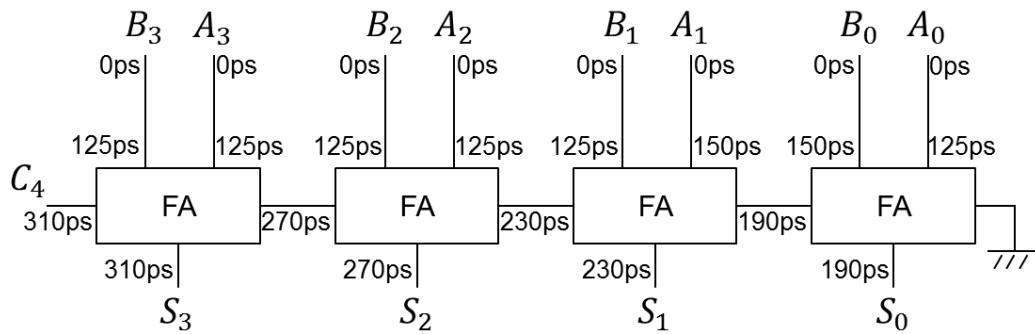


The following shows the total capacitance of each wire:

- $B_3: C_g + 3C_c$        $A_3: C_g + 3C_c$        $B_2: C_g + 3C_c$        $A_2: C_g + 3C_c$
- $B_1: C_g + 3C_c$        $A_1: C_g + 4C_c$        $B_0: C_g + 4C_c$        $A_0: C_g + 3C_c$

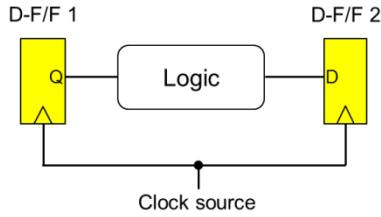
The following shows the arrival time at each node:

- $B_3, A_3, B_2, A_2, B_1, A_0: 2RC = 2R(C_g + 3C_c) = 2 \cdot (500\Omega) \cdot (50fF + 3 \cdot 25fF) = 125ps$
- $A_1, B_0: 2RC = 2R(C_g + 4C_c) = 2 \cdot (500\Omega) \cdot (50fF + 4 \cdot 25fF) = 150ps$



Thus, the delay is 310ps.

## Problem #6 (Timing Analysis under PVT Variation, 10 points).



- Delay from the clock source to D-F/F 1 (and D-F/F 2):  $cd_1$  (and  $cd_2$ )
- Setup time of the F/Fs:  $T_s$
- Hold time of the F/Fs:  $T_h$
- D-F/F internal delay:  $T_{CQ}$
- Clock skew:  $T_{\text{skew}} = cd_2 - cd_1$
- Logic delay:  $T_{\text{logic}}$
- Clock period:  $T_{\text{CLK}}$

Ideally, the following inequalities should be satisfied:

1. Setup time:  $T_s \leq T_{\text{CLK}} + T_{\text{skew}} - T_{\text{logic}} - T_{CQ}$
2. Hold time:  $T_h \leq T_{CQ} + T_{\text{logic}} - T_{\text{skew}}$

Process-voltage-temperature (PVT) variation causes serious problems such as delay variation. For example, a transistor can be faster or slower than predicted due to process variation (i.e.,  $\mu_p$  and  $\mu_n$  change) and wire delay can be increased or decreased depending on the operating temperature. The following shows variations in the timing values due to PVT variation ( $\Delta_1, \Delta_2, \Delta_3, \Delta_4 > 0$ ):

- $cd_1 \rightarrow cd_1 \pm \Delta_1$
- $cd_2 \rightarrow cd_2 \pm \Delta_2$
- $T_{CQ} \rightarrow T_{CQ} \pm \Delta_3$
- $T_{\text{logic}} \rightarrow T_{\text{logic}} \pm \Delta_4$

Derive a new setup time and a new hold time constraints (inequalities) that should be satisfied under the PVT variations. The new inequalities should consist of the following constants and variables only:

- $T_s, T_h, T_{CQ}, T_{\text{CLK}}, T_{\text{skew}}, T_{\text{logic}}, \Delta_1, \Delta_2, \Delta_3, \Delta_4$

The new range of  $T_{\text{skew}}$ :  $[(cd_2 - \Delta_2) - (cd_1 + \Delta_1), (cd_2 + \Delta_2) - (cd_1 - \Delta_1)]$

The new range of  $T_{CQ}$ :  $[T_{CQ} - \Delta_3, T_{CQ} + \Delta_3]$

The new range of  $T_{\text{logic}}$ :  $[T_{\text{logic}} - \Delta_4, T_{\text{logic}} + \Delta_4]$

Setup time)

Rewrite the setup time inequality as follows:  $T_{\text{logic}} \leq T_{\text{CLK}} + T_{\text{skew}} - T_s - T_{\text{CQ}}$

To derive a new setup time inequality, let's first focus on the right term. The smallest value the right term can have is  $T_r = T_{\text{CLK}} + \{(cd_2 - \Delta_2) - (cd_1 + \Delta_1)\} - T_s - \{T_{\text{CQ}} + \Delta_3\}$ .

The new logic delay should be less than or equal to  $T_r$ .

$$\begin{aligned} [T_{\text{logic}} - \Delta_4, T_{\text{logic}} + \Delta_4] &\leq T_r \\ \therefore T_{\text{logic}} &\leq T_{\text{CLK}} + \{(cd_2 - \Delta_2) - (cd_1 + \Delta_1)\} - T_s - \{T_{\text{CQ}} + \Delta_3\} - \Delta_4 \\ T_{\text{logic}} &\leq (T_{\text{CLK}} + T_{\text{skew}} - T_s - T_{\text{CQ}}) - (\Delta_1 + \Delta_2 + \Delta_3 + \Delta_4) \end{aligned}$$

Hold time)

Rewrite the hold time inequality as follows:  $T_{\text{logic}} \geq T_h - T_{\text{CQ}} + T_{\text{skew}}$

To derive a new hold time inequality, let's first focus on the right term. The largest value the right term can have is  $T_r = T_h - \{T_{\text{CQ}} - \Delta_3\} + \{(cd_2 + \Delta_2) - (cd_1 - \Delta_1)\}$ .

The new logic delay should be greater than or equal to  $T_r$ .

$$\begin{aligned} [T_{\text{logic}} - \Delta_4, T_{\text{logic}} + \Delta_4] &\geq T_r \\ \therefore T_{\text{logic}} &\geq T_h - \{T_{\text{CQ}} - \Delta_3\} + \{(cd_2 + \Delta_2) - (cd_1 - \Delta_1)\} + \Delta_4 \\ T_{\text{logic}} &\geq (T_h - T_{\text{CQ}} + T_{\text{skew}}) + (\Delta_1 + \Delta_2 + \Delta_3 + \Delta_4) \end{aligned}$$

## Problem #7 (High-Speed Adder, 10 points).

Compute the sum of A and B and Cin using the conditional sum adder.

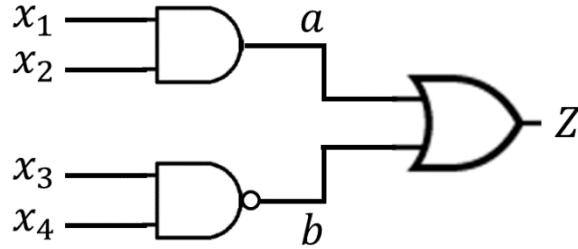
- A = 65534 (1111111111111110)
- B = 13421 (0011010001101101)
- Cin = 1

	$i:$	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	$A_i:$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
	$B_i:$	0	0	1	1	0	1	0	0	0	1	1	0	1	1	0	1
Step 1	$S_i^0:$	1	1	0	0	1	0	1	1	1	0	0	1	0	0	1	
	$CO_i^0:$	0	0	1	1	0	1	0	0	0	1	1	0	1	1	0	
Step 2	$S_i^1:$	0	0	1	1	0	1	0	0	0	1	1	0	1	1	0	0
	$CO_i^1:$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Step 3	$S_i^0:$	1	1	1	0	0	0	1	1	0	0	0	1	1	0		
	$CO_i^0:$	0		1		1		0		1		1		1			
Step 4	$S_i^1:$	0	0	1	1	0	1	0	0	0	1	1	0	1	1	0	0
	$CO_i^1:$	1		1		1		1		1		1		1			
Result		0	0	1	1	0	1	0	0	0	1	1	0	1	1	0	0

Carry out: 1

### Problem #8 (Testing, 10 points).

We want to detect stuck-at-0 and stuck-at-1 faults at all the primary inputs,  $x_1, x_2, x_3, x_4$ , and the two internal nodes,  $a, b$ . Computation of  $Z_f$  to detect a stuck-at-0/1 fault at an internal node can be done by setting the value of the node to constant 0 (for stuck-at-0 faults) or 1 (for stuck-at-1 faults). Find a minimal set of test vectors that can detect all the s-a-0 and s-a-1 faults at  $x_1, x_2, x_3, x_4, a$ , and  $b$ .



$$Z = x_1 x_2 + \overline{x_3 x_4}$$

$$x_1 \text{ s-a-0: } Z_f = \overline{x_3 x_4} \Rightarrow Z \oplus Z_f = (x_1 x_2 + \overline{x_3 x_4}) \oplus \overline{x_3 x_4} = 1 \Rightarrow (x_1 x_2 x_3 x_4) = (1 1 1 1)$$

$$x_1 \text{ s-a-1: } Z_f = x_2 + \overline{x_3 x_4} \Rightarrow Z \oplus Z_f = (x_1 x_2 + \overline{x_3 x_4}) \oplus (x_2 + \overline{x_3 x_4}) = 1 \\ \Rightarrow (x_1 x_2 x_3 x_4) = (0 1 1 1)$$

$$x_2 \text{ s-a-0: } Z_f = \overline{x_3 x_4} \Rightarrow (x_1 x_2 x_3 x_4) = (1 1 1 1)$$

$$x_2 \text{ s-a-1: } Z_f = x_1 + \overline{x_3 x_4} \Rightarrow (x_1 x_2 x_3 x_4) = (1 0 1 1)$$

$x_3$  s-a-0:

$$Z_f = 1 \Rightarrow Z \oplus Z_f = (x_1 x_2 + \overline{x_3 x_4}) \oplus 1 = 1 \Rightarrow (x_1 x_2 x_3 x_4) = (0 0 1 1) \text{ or } (0 1 1 1) \text{ or } (1 0 1 1)$$

$$x_3 \text{ s-a-1: } Z_f = x_1 x_2 + \overline{x_4} \Rightarrow Z \oplus Z_f = (x_1 x_2 + \overline{x_3 x_4}) \oplus (x_1 x_2 + \overline{x_4}) = 1 \Rightarrow (x_1 x_2 x_3 x_4) = (0 0 0 1) \text{ or } (0 1 0 1) \text{ or } (1 0 0 1)$$

$$x_4 \text{ s-a-0: } Z_f = 1 \Rightarrow (x_1 x_2 x_3 x_4) = (0 0 1 1) \text{ or } (0 1 1 1) \text{ or } (1 0 1 1)$$

$$x_4 \text{ s-a-1: } Z_f = x_1 x_2 + \overline{x_3} \Rightarrow (x_1 x_2 x_3 x_4) = (0 0 1 0) \text{ or } (0 1 1 0) \text{ or } (1 0 1 0)$$

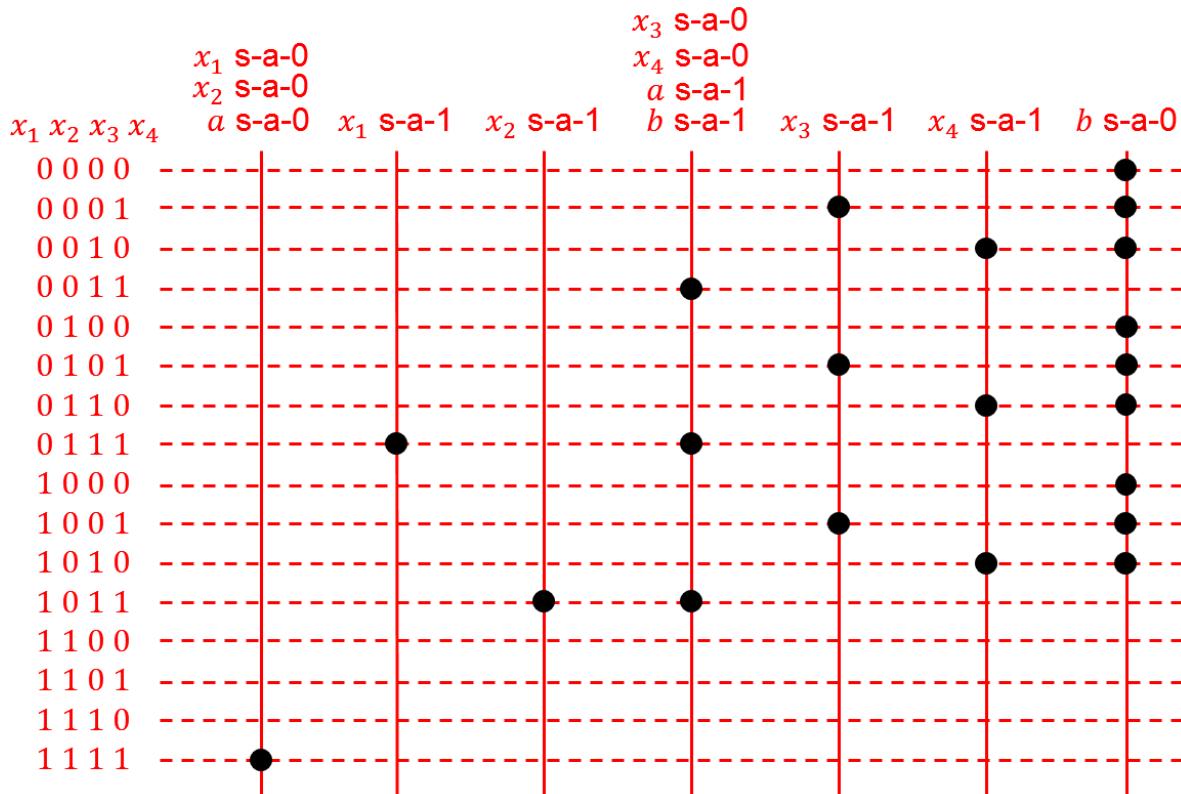
$$a \text{ s-a-0: } Z_f = \overline{x_3 x_4} \Rightarrow (x_1 x_2 x_3 x_4) = (1 1 1 1)$$

$$a \text{ s-a-1: } Z_f = 1 \Rightarrow (x_1 x_2 x_3 x_4) = (0 0 1 1) \text{ or } (0 1 1 1) \text{ or } (1 0 1 1)$$

$$b \text{ s-a-0: } Z_f = x_1 x_2 \Rightarrow Z \oplus Z_f = (x_1 x_2 + \overline{x_3 x_4}) \oplus (x_1 x_2) = 1 \Rightarrow (x_1 x_2 x_3 x_4) = (0 0 0 0) \text{ or } (0 1 0 0) \text{ or } (1 0 0 0) \text{ or } (0 0 0 1) \text{ or } (0 1 0 1) \text{ or } (1 0 0 1) \text{ or } (0 0 1 0)$$

or  $(0\ 1\ 1\ 0)$  or  $(1\ 0\ 1\ 0)$

$b\ s-a-1: Z_f = 1 \Rightarrow (x_1\ x_2\ x_3\ x_4) = (0\ 0\ 1\ 1) \text{ or } (0\ 1\ 1\ 1) \text{ or } (1\ 0\ 1\ 1)$



- 1) Needs  $(1\ 1\ 1\ 1)$  to cover the 1<sup>st</sup> column.
- 2) Needs  $(0\ 1\ 1\ 1)$  to cover the 2<sup>nd</sup> column. This also covers the 4<sup>th</sup> column.
- 3) Needs  $(1\ 0\ 1\ 1)$  to cover the 3<sup>rd</sup> column.
- 4) Needs two more vectors to cover the 5<sup>th</sup>, 6<sup>th</sup>, and 7<sup>th</sup> columns.

=>  $(0\ 0\ 1\ 0)$  and  $(0\ 0\ 0\ 1)$  (there exist total 18 different minimal sets).

Notice that  $(1\ 1\ 1\ 1)$ ,  $(0\ 1\ 1\ 1)$ , and  $(1\ 0\ 1\ 1)$  must be selected.

**EE434**  
**ASIC and Digital Systems**

**Midterm Exam 1**

**February 25, 2015. (5:10pm – 6pm)**

**Instructor: Dae Hyun Kim ([daehyun@eeecs.wsu.edu](mailto:daehyun@eeecs.wsu.edu))**

**Name:**

**WSU ID:**

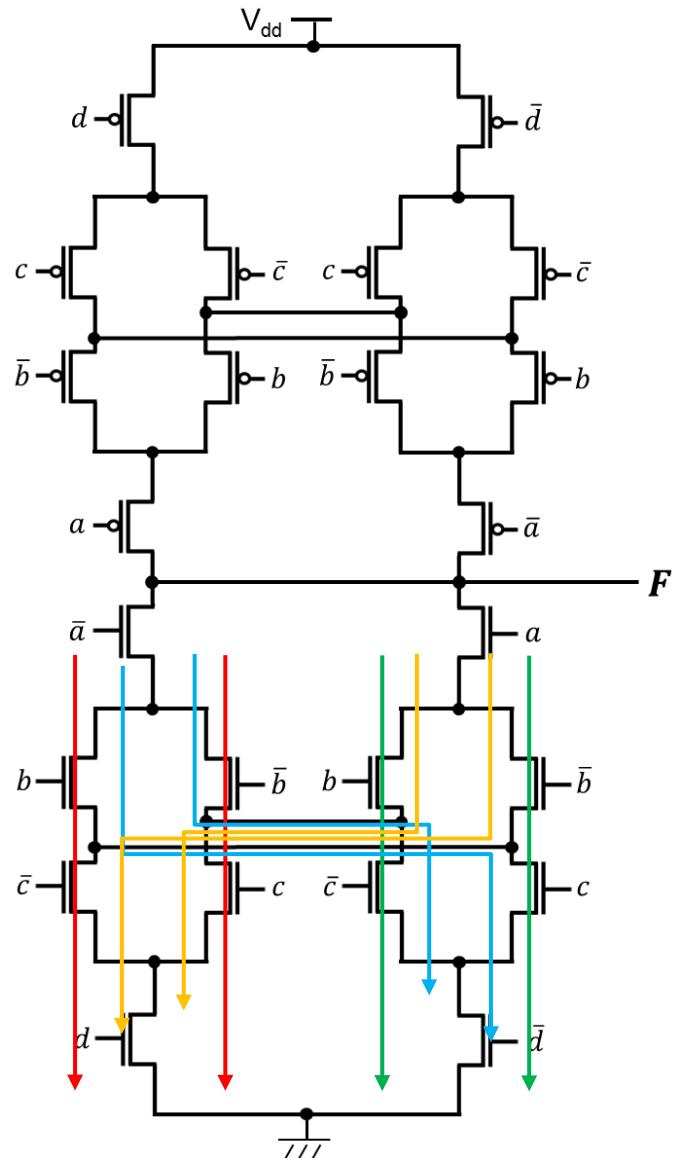
Problem	Points	
1	10	
2	10	
3	10	
4	10	
5	10	
6	10	
7	10	
8	10	
Total	80	

\* Allowed: Textbooks, cheat sheets, class notes, notebooks, calculators, watches

\* Not allowed: Electronic devices (smart phones, tablet PCs, laptops, etc.) except calculators and watches

### Problem #1 (Static CMOS gates, 10 points).

Represent  $F$  as a function of  $a, b, c$ , and  $d$ .

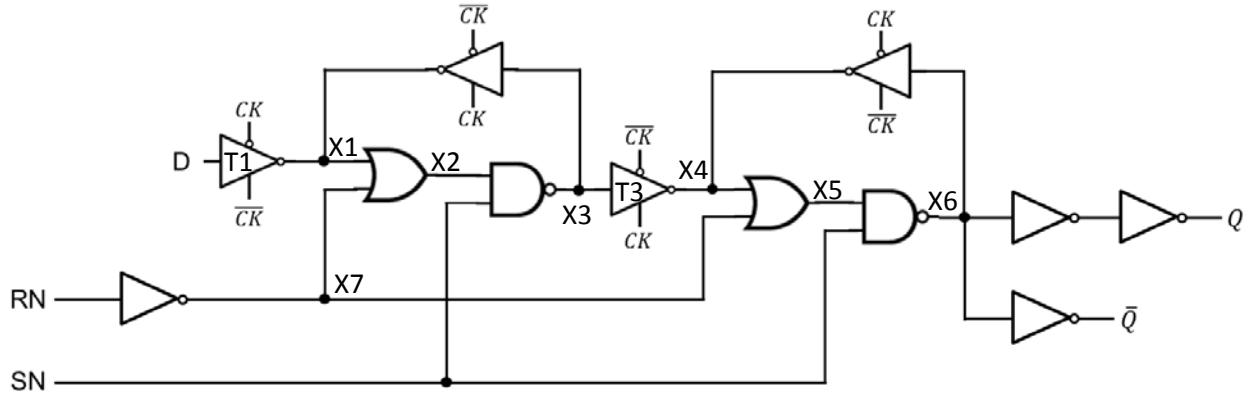


$$\begin{aligned}
 \bar{F} &= \underline{\bar{a}d(b\bar{c} + \bar{b}c)} + \underline{a\bar{d}(b\bar{c} + \bar{b}c)} + \underline{\bar{a}\bar{d}(bc + \bar{b}\bar{c})} + \underline{ad(bc + \bar{b}\bar{c})} \\
 &= (\bar{a}d + a\bar{d})(b\bar{c} + \bar{b}c) + (\bar{a}\bar{d} + ad)(bc + \bar{b}\bar{c}) \\
 &= (a \oplus d)(b \oplus c) + (\overline{a \oplus d})(\overline{b \oplus c}) \\
 &= (a \oplus d) \oplus (b \oplus c)
 \end{aligned}$$

$$\therefore F = a \oplus b \oplus c \oplus d$$

## Problem #2 (Static CMOS gates, 10 points).

What does the following circuit do? Describe the function of the circuit in as much detail as possible.



1.  $SN=0 \rightarrow X6=1 \rightarrow Q=1$ : SN is an active-low asynchronous set.
2.  $SN=1, RN=0 \rightarrow X7=1 \rightarrow X5=1 \rightarrow X6=0 \rightarrow Q=0$ : RN is an active-low synchronous reset.
3.  $SN=1, RN=1 \rightarrow X7=0$

If  $CK=0 \rightarrow T3$  is off  $\rightarrow Q=X6, X4=\overline{X6}, X5=X4, X6=\overline{X5}=\overline{X4}=X6 \rightarrow Q^+ = Q$

If  $CK=1 \rightarrow T1$  is off, T1 captures D,  $X1=\overline{D} \rightarrow X2=\overline{D} \rightarrow X3=D$

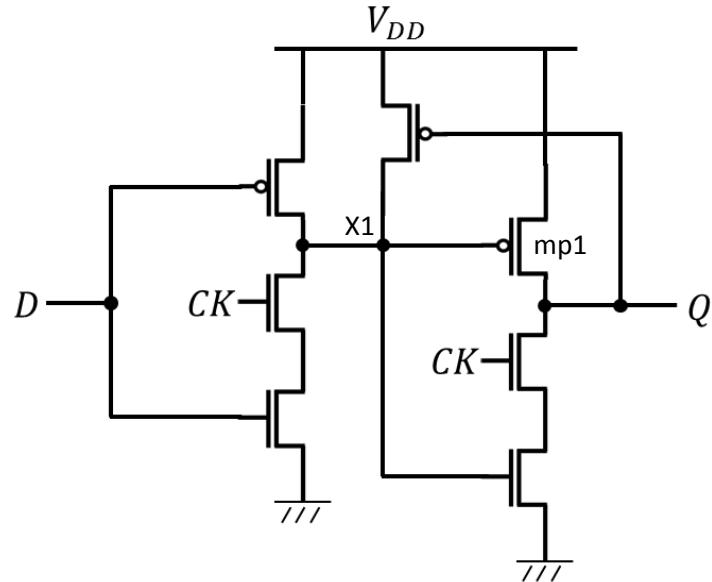
$T3$  is ON,  $X4=\overline{X3}=\overline{D} \rightarrow X5=X4 \rightarrow X6=\overline{X5}=D \rightarrow Q^+ = D$

$\rightarrow$  If  $SN=1$  and  $RN=1$ , the circuit is just a positive edge-triggered D FF.

so, this is a positive edge-triggered D-FF with asynchronous active-low set and reset signals (set dominates reset).

### Problem #3 (CMOS Logic, 10 points).

What is the functionality of the following circuit? Describe the functionality in as much detail as possible.



$$CK=1 \rightarrow X1=\bar{D} \rightarrow Q=\bar{X1}=D$$

If  $D=0 \rightarrow X1=1 \rightarrow Q=0 \rightarrow X1=1$  (no conflict)

If  $D=1 \rightarrow X1=0 \rightarrow Q=1$

$CK=0 \rightarrow$  The PDN of the right half is off.

If  $D=0 \rightarrow X1=1 \rightarrow mp1$  is off  $\rightarrow Q$  is floating (hold)

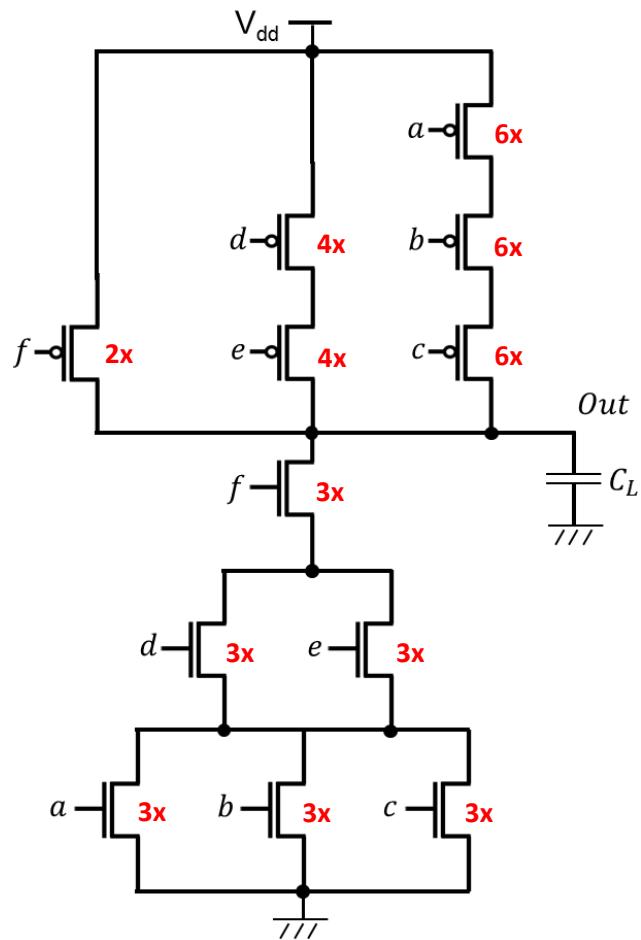
If  $D=1, Q=0 \rightarrow X1=1 \rightarrow mp1$  is off  $\rightarrow Q$  is floating (hold)

If  $D=1, Q=1 \rightarrow X1$  is floating (hold)  $\rightarrow Q$  is floating (hold)

so, this is a **tri-state D-latch**.

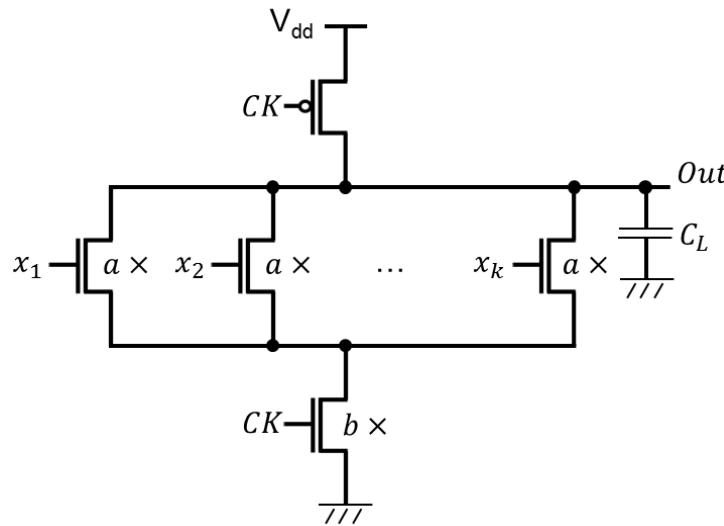
### Problem #4 (Transistor Sizing, 10 points).

Size the transistors in the following gate.  $R_n$  is the resistance of a 1X NMOS transistor.  $\mu_n = 2 \cdot \mu_p$ . Ignore all the parasitic capacitances. Target time constant:  $\tau_{target} = R_n \cdot C_L$ . Try to minimize the total area.



## Problem #5 (Transistor Sizing, 10 points).

We want to design a  $k$ -input NOR gate. However, the static CMOS gate design methodology is not suitable for the design of the  $k$ -input NOR gate due to area overhead in the pull-up network and the body-bias effect. Therefore, we are going to design it using the dynamic CMOS design methodology. The following shows a schematic of the  $k$ -input NOR gate.



$R_n$  is the resistance of a 1X NMOS transistor.  $\mu_n = 2 \cdot \mu_p$ . Ignore all the parasitic capacitances. Target time constant:  $\tau_{target} = R_n \cdot C_L$ . All the transistors for  $x_1 \sim x_k$  are upsized to  $aX$  and the transistor for  $CK$  is upsized to  $bX$  ( $a$  and  $b$  are *real* numbers). We minimize the total width,  $Width = a \cdot k + b$ . Find  $a$  and  $b$  (i.e., derive  $a$  (and  $b$ ) as a function of  $k$ ) minimizing the total width.

$$\text{Constraint: } \left( \frac{R_n}{a} + \frac{R_n}{b} \right) C_L = R_n C_L \Rightarrow \frac{1}{a} + \frac{1}{b} = 1$$

$$W = ak + b = ak + \frac{a}{a-1}$$

$$1) W' = k + \frac{(a-1)-a}{(a-1)^2} = k - \frac{1}{(a-1)^2} = 0 \Rightarrow a = 1 + \frac{1}{\sqrt{k}} \Rightarrow b = 1 + \sqrt{k}$$

2)  $ak + b = c \Rightarrow$  Two functions,  $b = -ak + c$  and  $b = \frac{a}{a-1}$ , should meet at a single point (confirm this by drawing their graphs).

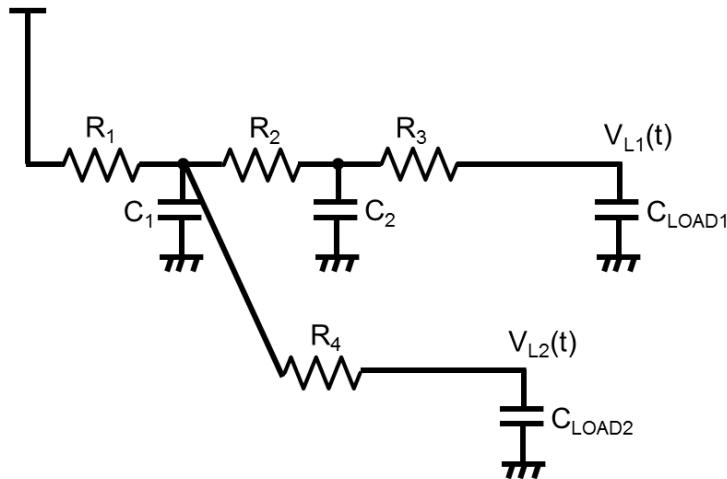
$\Rightarrow -ak + c = \frac{a}{a-1}$  should have a single root.  $\Rightarrow ka^2 - (k + c - 1)a + c = 0$  has a single root.  $\Rightarrow (k + c - 1)^2 - 4kc = 0 \Rightarrow (k - c - 1)^2 = 0 \Rightarrow c = k + 1 \pm 2\sqrt{k} \Rightarrow$

$$a = 1 \pm \frac{1}{\sqrt{k}} \Rightarrow a > 1, \text{ so } a = 1 + \frac{1}{\sqrt{k}} \Rightarrow b = 1 + \sqrt{k}$$

### Problem #6 (Elmore Delay, 10 points).

6-1. Compute Elmore delay at LOAD1 and LOAD2, i.e., represent the delay at LOAD1 (and LOAD2) as a function of  $R_1 \sim R_4$ ,  $C_1$ ,  $C_2$ ,  $C_{LOAD1}$ , and  $C_{LOAD2}$ .

$$V(t) = V_{DD} \cdot u(t)$$



At LOAD1:  $R_1(C_1 + C_2 + C_{LOAD1} + C_{LOAD2}) + R_2(C_2 + C_{LOAD1}) + R_3 C_{LOAD1}$

At LOAD2:  $R_1(C_1 + C_2 + C_{LOAD1} + C_{LOAD2}) + R_4 C_{LOAD2}$

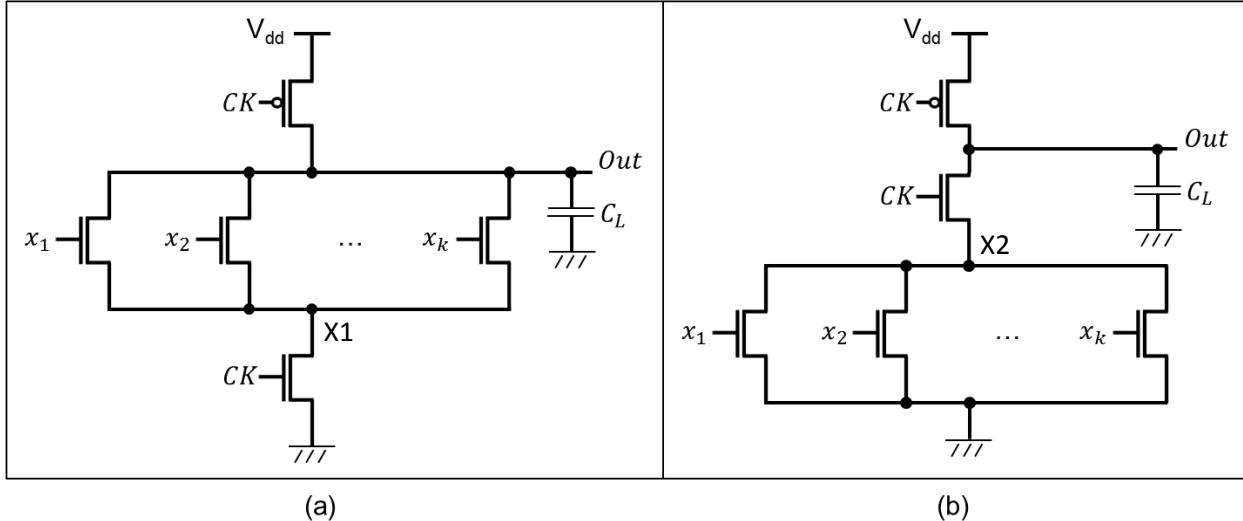
6-2. Compute Elmore delay at LOAD1 for  $R_1 = R_2 = R_3 = 1k\Omega$ ,  $C_1 = C_2 = C_{LOAD1} = 10fF$ ,  $R_4 = 0.1k\Omega$ , and  $C_{LOAD2} = 1pF$ . Then, compute Elmore delay at LOAD1 for  $R_1 = R_2 = R_3 = 1k\Omega$ ,  $C_1 = C_2 = C_{LOAD1} = 10fF$ ,  $R_4 = 10M\Omega$ , and  $C_{LOAD2} = 1pF$ . This result is called “resistive shielding”. Discuss a limitation of the Elmore delay model in terms of the resistive shielding effect.

At LOAD1:  $1k(10f+10f+10f+1p) + 1k(10f+10f) + 1k(10f) = (1k*1030f) + (1k*20f) + (1k*10f) = 1030ps + 20ps + 10ps = 1060ps$

The Elmore delay at LOAD1 is not affected by  $R_4$ . If  $R_4$  is sufficiently small, the Elmore delay at LOAD1 should include  $C_{LOAD2}$ . However, if  $R_4$  is sufficiently large, we can assume that the Load2 branch does not exist, so the Elmore delay at LOAD1 should not include  $C_{LOAD2}$ . The Elmore delay model does not distinguish these two cases.

### Problem #7 (Dynamic CMOS, 10 points).

Compare the following implementations for a dynamic-CMOS  $k$ -input NOR gate. Are there any problems in (a)? in (b)?

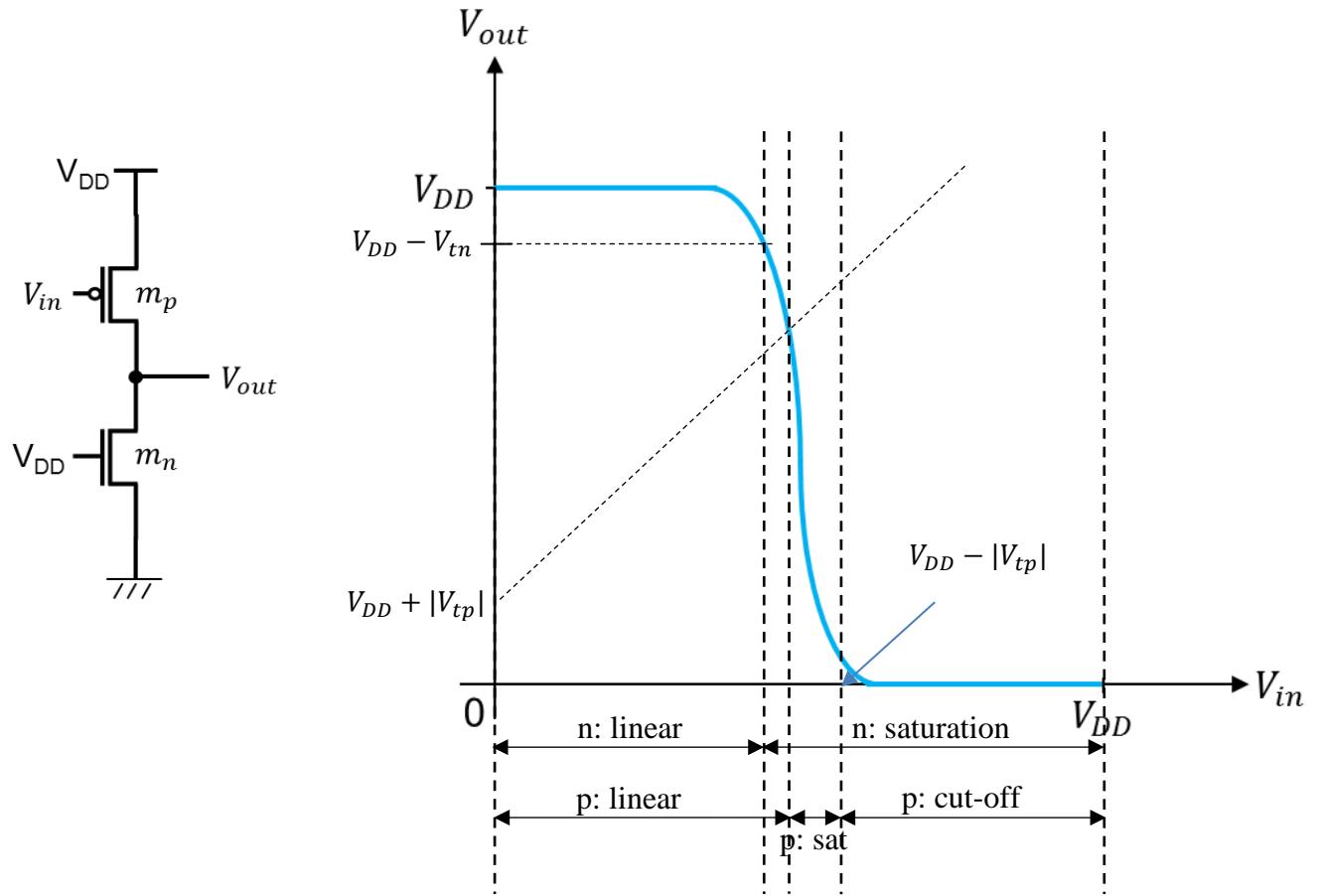


(a) does not have any charge sharing problem. When  $CK=0$ , the PMOS transistor charges the output capacitor and  $X1$  (if any  $x_i$  is 1).

(b) suffers from charge sharing. When  $CK=0$ , the PMOS transistor charges the output capacitor only. When  $CK=1$  and  $x_1=\dots=x_k=0$ , charge sharing happens between the output capacitor and the parasitic capacitor at  $X2$ .

### Problem #8 (DC Characteristics, 10 points).

The following circuit is called “pseudo-PMOS”. Sketch a DC characteristic curve of the pseudo-PMOS inverter and properly split the curve into regions. In each region, show the status (cut-off, linear, saturation) of each transistor ( $m_n$  and  $m_p$ ).



1)  $m_n$ :  $V_{gs} - V_t = V_{DD} - V_{tn} > 0$  (always),  $V_{ds} = V_{out}$

i)  $V_{gs} - V_t < V_{ds} \Rightarrow V_{DD} - V_{tn} < V_{out} \Rightarrow$  saturation

ii)  $V_{DD} - V_{tn} > V_{out} \Rightarrow$  linear

2)  $m_p$ :  $|V_{gs}| - |V_t| = V_{DD} - V_{in} - |V_{tp}|$ ,  $|V_{ds}| = V_{DD} - V_{out}$

i)  $|V_{gs}| - |V_t| < 0 \Rightarrow V_{in} > V_{DD} - |V_{tp}| \Rightarrow$  cut-off

ii)  $V_{gs} - V_t < V_{ds} \Rightarrow V_{out} < V_{in} + |V_{tp}| \Rightarrow$  saturation

iii)  $V_{gs} - V_t > V_{ds} \Rightarrow V_{out} > V_{in} + |V_{tp}| \Rightarrow$  linear

\*) Depending on the size of the NMOS and PMOS transistors, the highest output voltage might be less than  $V_{DD} - V_{tn}$ . In this case, the NMOS transistor is in the linear mode even when  $V_{in}$  is close to 0.

**EE434**  
**ASIC and Digital Systems**

**Midterm Exam 2**

**April 8, 2015. (5:10pm – 6pm)**

**Instructor: Dae Hyun Kim ([daehyun@eecs.wsu.edu](mailto:daehyun@eecs.wsu.edu))**

**Name:**

**WSU ID:**

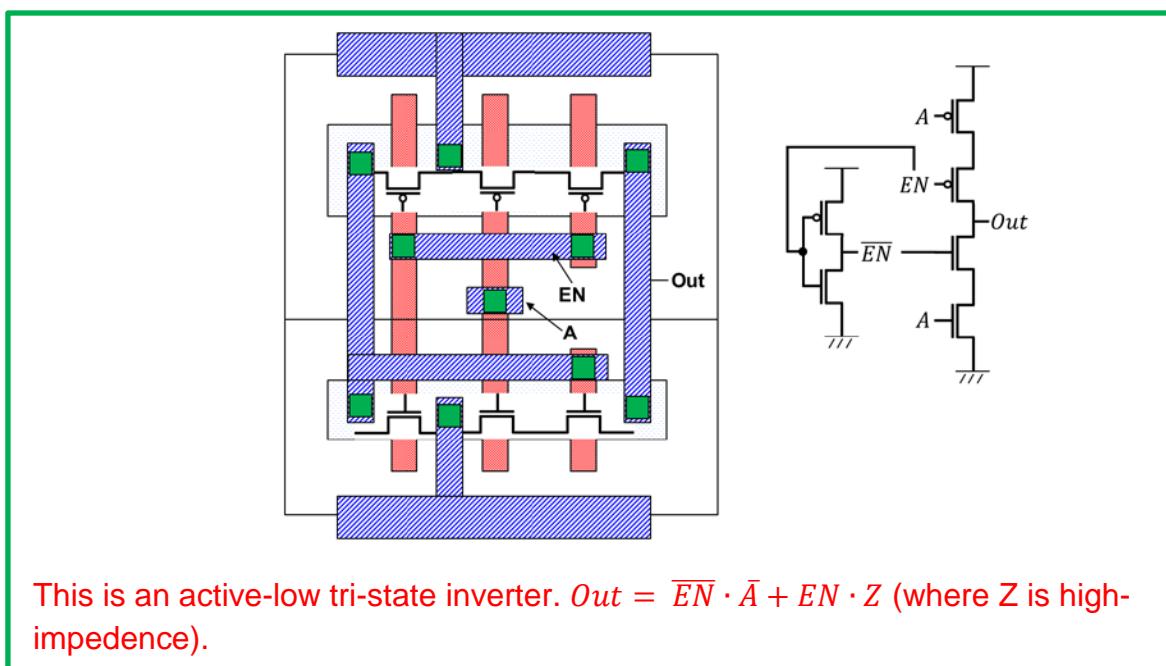
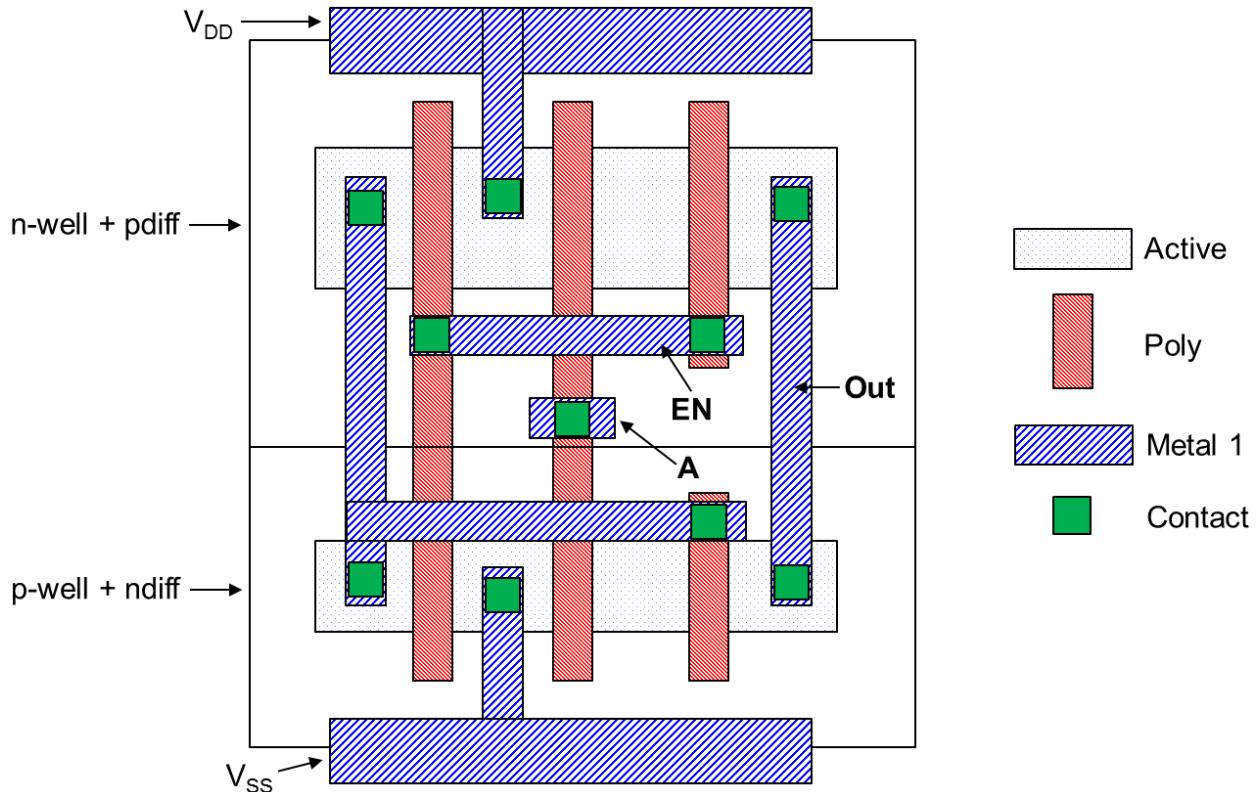
Problem	Points	
1	20	
2-1	13	
2-2	7	
3-1	10	
3-2	10	
4	20	
Total	80	

\* Allowed: Textbooks, cheat sheets, class notes, notebooks, calculators, watches

\* Not allowed: Electronic devices (smart phones, tablet PCs, laptops, etc.) except calculators and watches

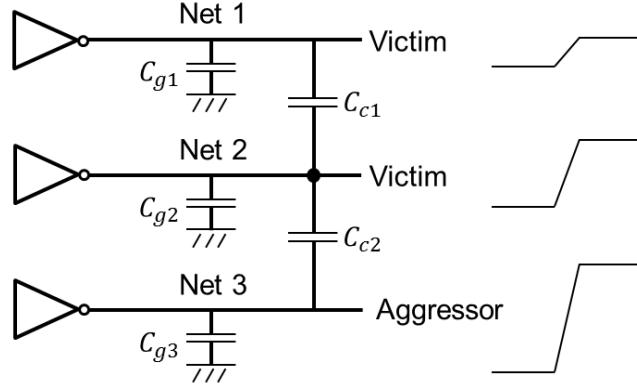
## Problem #1 (Layout, 20 points).

Represent *Out* as a Boolean function of *EN* and *A* or describe the function of the following layout in as much detail as possible (Primary inputs: *A*, *EN*. Primary output: *Out*).

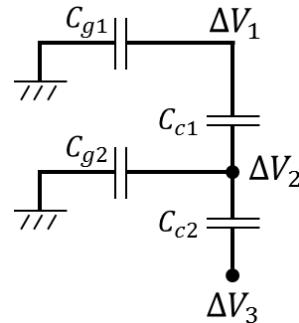


## Problem #2 (Coupling Analysis, 20 points).

Three nets are coupled through  $C_{c1}$  and  $C_{c2}$  as shown in the following figure:



Net 3 is the only aggressor and Net 2 and Net 1 are victims. Although Net 1 is not directly connected to Net 3, Net 1 is affected by the potential change of Net 2 when Net 3 switches. The above figure can be simplified as follows:



1 – 12 points) Derive  $\Delta V_2$  and  $\Delta V_1$  as a function of  $\Delta V_3$ ,  $C_{g1}$ ,  $C_{g2}$ ,  $C_{c1}$ , and  $C_{c2}$ .

$$i_{32} = C_{c2} \frac{d(V_3 - V_2)}{dt} = C_{g2} \frac{dV_2}{dt} + C_{c1} \frac{d(V_2 - V_1)}{dt}$$

$$i_{21} = C_{c1} \frac{d(V_2 - V_1)}{dt} = C_{g1} \frac{dV_1}{dt}$$

From  $i_{21}$ :  $C_{c1}(\Delta V_2 - \Delta V_1) = C_{g1}\Delta V_1 \rightarrow \Delta V_1 = \frac{C_{c1}}{C_{g1} + C_{c1}}\Delta V_2$

From  $i_{32}$ :  $C_{c2}(\Delta V_3 - \Delta V_2) = C_{g2}\Delta V_2 + C_{c1}(\Delta V_2 - \Delta V_1)$

$$\rightarrow C_{c2}\Delta V_3 = (C_{g2} + C_{c1} + C_{c2})\Delta V_2 - C_{c1}\Delta V_1 = \left( C_{g2} + C_{c1} + C_{c2} - \frac{C_{c1}^2}{C_{g1} + C_{c1}} \right) \Delta V_2$$

$$\rightarrow C_{c2}\Delta V_3 = \left( \frac{C_{g1}C_{g2} + C_{g2}C_{c1} + C_{g1}C_{c1} + C_{g1}C_{c2} + C_{c1}C_{c2}}{C_{g1} + C_{c1}} \right) \Delta V_2$$

$$\begin{aligned} \therefore \Delta V_2 &= C_{c2} \left( \frac{C_{g1} + C_{c1}}{C_{g1}C_{g2} + C_{g2}C_{c1} + C_{g1}C_{c1} + C_{g1}C_{c2} + C_{c1}C_{c2}} \right) \Delta V_3 \\ \Delta V_1 &= \left( \frac{C_{c1}C_{c2}}{C_{g1}C_{g2} + C_{g2}C_{c1} + C_{g1}C_{c1} + C_{g1}C_{c2} + C_{c1}C_{c2}} \right) \Delta V_3 \end{aligned}$$

2 – 8 points) True/False questions (Hint: Use your intuition or the formulas you derived in the above problem).

- a) If  $C_{g1}$  increases,  $\Delta V_1$  increases (true/false).

$$\Delta V_1 = \left( \frac{C_{c1}C_{c2}}{(C_{g2} + C_{c1} + C_{c2})C_{g1} + C_{g2}C_{c1} + C_{c1}C_{c2}} \right) \Delta V_3 \rightarrow \Delta V_1 \text{ decreases.}$$

- b) If  $C_{g1}$  increases,  $\Delta V_2$  increases (true/false).

$$\Delta V_2 = \frac{C_{c2}}{\left( C_{g2} + C_{c1} + C_{c2} - \frac{C_{c1}^2}{C_{g1} + C_{c1}} \right)} \Delta V_3 \rightarrow \Delta V_2 \text{ decreases.}$$

- c) If  $C_{g2}$  increases,  $\Delta V_1$  increases (true/false).

$$\Delta V_1 = \left( \frac{C_{c1}C_{c2}}{(C_{g1} + C_{c1})C_{g2} + C_{g1}C_{c1} + C_{g1}C_{c2} + C_{c1}C_{c2}} \right) \Delta V_3 \rightarrow \Delta V_1 \text{ decreases.}$$

- d) If  $C_{g2}$  increases,  $\Delta V_2$  increases (true/false).

$$\Delta V_2 = C_{c2} \left( \frac{C_{g1} + C_{c1}}{(C_{g1} + C_{c1})C_{g2} + C_{g1}C_{c1} + C_{g1}C_{c2} + C_{c1}C_{c2}} \right) \Delta V_3 \rightarrow \Delta V_2 \text{ decreases.}$$

- e) If  $C_{c1}$  increases,  $\Delta V_1$  increases (true/false).

$$\Delta V_1 = \left( \frac{C_{c2}}{K_1 + \frac{K_2}{C_{c1}}} \right) \Delta V_3 \rightarrow \Delta V_1 \text{ increases.}$$

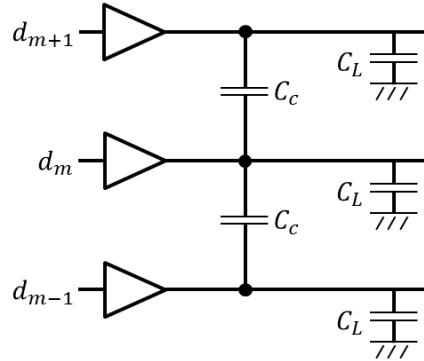
- f) If  $C_{c2}$  increases,  $\Delta V_1$  increases (true/false).

$$\Delta V_1 = \left( \frac{C_{c1}}{K_1 + \frac{K_2}{C_{c2}}} \right) \Delta V_3 \rightarrow \Delta V_1 \text{ increases.}$$

- g) If  $C_{c2}$  increases,  $\Delta V_2$  increases (true/false).

$$\Delta V_2 = \frac{1}{\left( 1 + \frac{K}{C_{c2}} \right)} \Delta V_3 \rightarrow \Delta V_2 \text{ increases.}$$

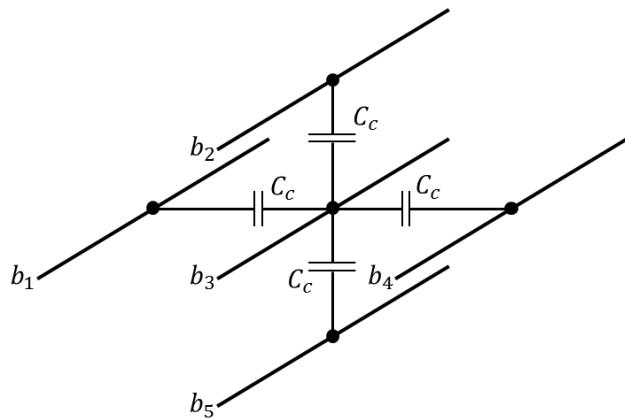
### Problem #3 (Coupling Minimization, 20 points).



1 – 10 points) Compute effective capacitance for the net in the middle ( $d_m$ ) for the following transition patterns:

Transition patterns ( $d_{m+1} \ d_m \ d_{m-1}$ )	Effective cap of $d_m$
$010 \rightarrow 000$	$C_L + 2C_c$
$010 \rightarrow 001$	$C_L + 3C_c$
$010 \rightarrow 100$	$C_L + 3C_c$
$010 \rightarrow 101$	$C_L + 4C_c$

2 – 10 points) A bus consisting of five bits ( $b_1 \ b_2 \ b_3 \ b_4 \ b_5$ ) is routed in three metal layers. Due to some unknown reasons, four of them ( $b_1 \ b_2 \ b_4 \ b_5$ ) are routed in parallel with  $b_3$ . The following shows the coupling capacitance among the five nets.



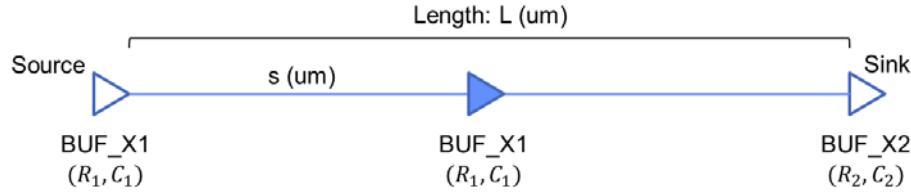
Due to the coupling between  $b_3$  and  $b_k$ , the worst-case effective coupling capacitance that  $b_3$  experiences will be  $8 \cdot C_c$ . List all transition patterns that make  $b_3$  experience  $8 \cdot C_c$  and  $7 \cdot C_c$ .

$8C_c: 00100 \leftrightarrow 11011$

$7C_c: 00100 \leftrightarrow 11010 \ / \ 00100 \leftrightarrow 11001 \ / \ 00100 \leftrightarrow 10011 \ / \ 00100 \leftrightarrow 01011$

$11011 \leftrightarrow 00101 \ / \ 11011 \leftrightarrow 00110 \ / \ 11011 \leftrightarrow 01100 \ / \ 11011 \leftrightarrow 10100$

## Problem #4 (Buffer Insertion, 20 points).



A source (type: BUF\_X1) drives a sink (type: BUF\_X2) through a net and you are supposed to insert a buffer (type: BUF\_X1) between them as shown in the above figure. Find an optimal location of the buffer minimizing the total delay, i.e., represent “ $s$ ” as a function of the following parameters.

- Output resistance of BUF\_X1:  $R_1$
- Input capacitance of BUF\_X1:  $C_1$
- Input capacitance of BUF\_X2:  $C_2$
- Total length of the net:  $L$  (um)
- Total wire resistance:  $R_w$
- Total wire capacitance:  $C_w$
- $(C_w + C_2 > C_1)$

$$\begin{aligned}
 \text{Delay} = \tau &= \left( R_1 \left( C_w \cdot \frac{s}{L} + C_1 \right) + R_w \cdot \frac{s}{L} \cdot C_1 + \frac{1}{2} \left( R_w \frac{s}{L} \right) \left( C_w \frac{s}{L} \right) \right) \\
 &\quad + \left( R_1 \left( C_w \cdot \frac{L-s}{L} + C_2 \right) + R_w \cdot \frac{L-s}{L} \cdot C_2 + \frac{1}{2} \left( R_w \frac{L-s}{L} \right) \left( C_w \frac{L-s}{L} \right) \right) \\
 &= R_1 C_w + R_1 C_1 + R_1 C_2 + R_w \left( \frac{s}{L} \cdot C_1 + \frac{L-s}{L} \cdot C_2 \right) + \frac{1}{2L^2} R_w C_w (s^2 + s^2 - 2Ls + L^2) \\
 \frac{d\tau}{ds} &= \frac{R_w C_1}{L} - \frac{R_w C_2}{L} + \frac{R_w C_w}{2L^2} (4s - 2L) = 0 \\
 s \left( \frac{2R_w C_w}{L^2} \right) &= \frac{R_w C_w}{L} + \frac{R_w (C_2 - C_1)}{L}
 \end{aligned}$$

$$\therefore s = \frac{L(C_w + C_2 - C_1)}{2C_w}$$

**EE434**  
**ASIC and Digital Systems**

**Final Exam**

**May 5, 2016. (1pm – 3pm)**

**Instructor: Dae Hyun Kim ([daehyun@eecs.wsu.edu](mailto:daehyun@eecs.wsu.edu))**

**Name:**

**WSU ID:**

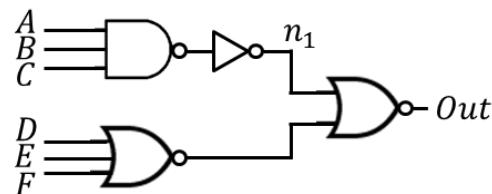
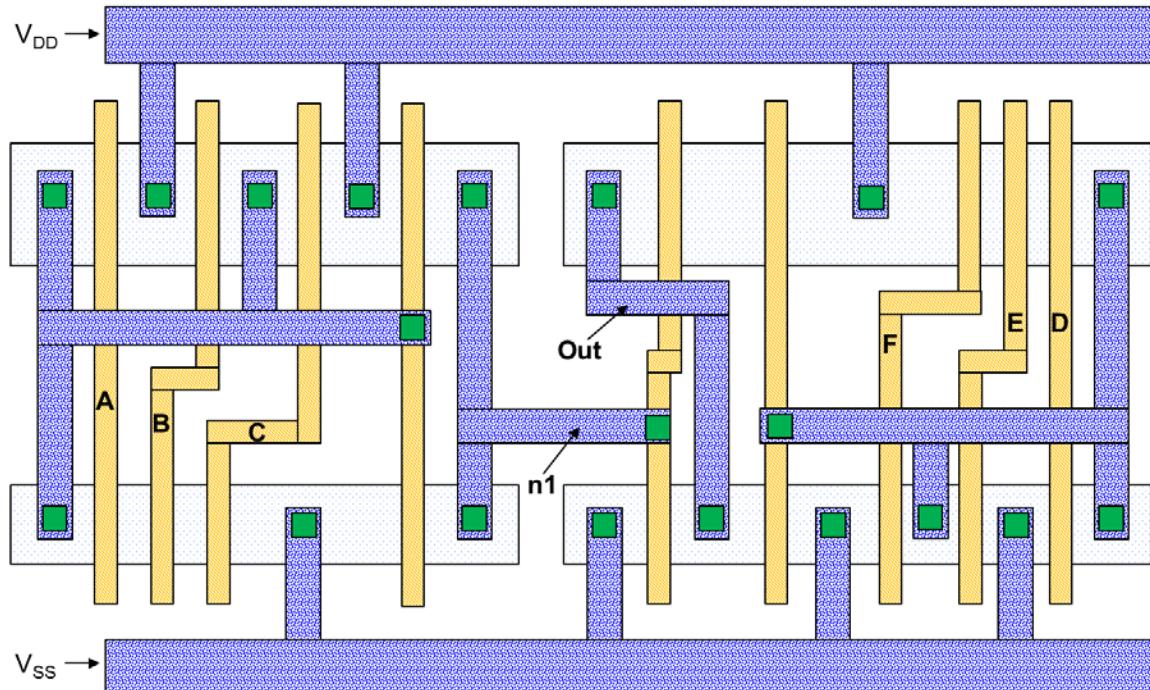
Problem	Points	
1	10	
2	10	
3	10	
4	10	
5	10	
6	10	
7	10	
8	10	
Total	80	

\* Allowed: Textbooks, cheat sheets, class notes, notebooks, calculators, watches

\* Not allowed: Electronic devices (smart phones, tablet PCs, laptops, etc.) except calculators and watches

## Problem #1 (Layout + Testing, 10 points)

The following layout consists of six primary inputs (A, B, C, D, E, F) and a primary output (Out). n1 is an internal node. Find all input vectors that can detect a stuck-at-0 fault at node n1.



$$Out = \overline{ABC} + \overline{D+E+F} = \overline{ABC} \cdot (D+E+F)$$

$$Out_f = D + E + F$$

$$Out \oplus Out_f = 1 \rightarrow ABC = 1, D + E + F = 1$$

$$\rightarrow ABCDEF = 111001, 111010, 111011, 111100, 111101, 111110, 111111$$

## Problem #2 (Testing, 10 points)

A combinational logic has  $n$  inputs ( $x_1, x_2, \dots, x_n$ ) and an output ( $Z$ ).  $Z$  is a Boolean function of the inputs,  $Z = g(x_1, \dots, x_n, \overline{x_1}, \dots, \overline{x_n}, AND, OR)$ . To find input vectors that can detect a stuck-at- $v$  fault  $f$  at a node, we compute  $Z_f$  by setting the value of the node to  $v$  and solving  $Z \oplus Z_f = 1$ . Let  $T_1$  be a set of all input vectors that detect fault  $f_1$  and  $T_2$  be a set of all input vectors that detect fault  $f_2$  ( $f_1 \neq f_2$ ). Prove that if  $Z_{f1} \neq Z_{f2}$ ,  $T_1$  cannot be equal to  $T_2$ .

(Example: A three-input AND gate has three inputs  $a, b, c$  and an output  $Z$ . Let  $f_1$  be a stuck-at-0 fault at input  $a$  and  $f_2$  be a stuck-at-1 fault at input  $b$ . Then,  $Z = a \cdot b \cdot c$ ,  $Z_{f1} = 0$ ,  $Z_{f2} = a \cdot c$  ( $Z_{f1} \neq Z_{f2}$ ). In this case,  $T_1 = \{111\}$  and  $T_2 = \{101\}$ , so  $T_1 \neq T_2$ .)

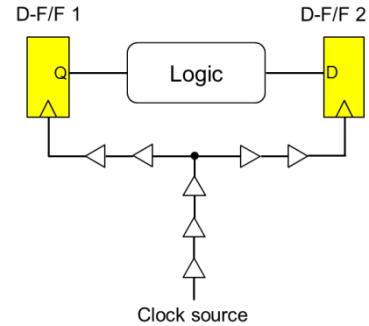
First, we prove that if  $Z_{f1} = Z_{f2}$ ,  $T_1 = T_2$ .

We get  $T_1$  from  $Z \oplus Z_{f1} = 1$  and  $T_2$  from  $Z \oplus Z_{f2} = 1$ . Thus, if  $Z_{f1} = Z_{f2}$ ,  $T_1 = T_2$ .

Now, suppose there exists a case where  $T_1 = T_2$  when  $Z_{f1} \neq Z_{f2}$  (i.e., if  $Z_{f1} \neq Z_{f2}$ , then  $T_1 = T_2$ ). The contrapositive of this statement is “If  $T_1 \neq T_2$ , then  $Z_{f1} = Z_{f2}$ ”. However, this is a contradiction because if  $Z_{f1} = Z_{f2}$ , then  $T_1 = T_2$  by the proof shown above. Thus, the original statement “if  $Z_{f1} \neq Z_{f2}$ , then  $T_1 = T_2$ ” is also false, i.e., if  $Z_{f1} \neq Z_{f2}$ ,  $T_1$  cannot be equal to  $T_2$ .

### Problem #3 (Timing Analysis, 10 points).

- Setup time of the F/Fs:  $T_s$
- Hold time of the F/Fs:  $T_h$
- D-F/F internal delay:  $T_{CQ}$
- Clock skew:  $T_{skew} = \text{delay from the clock source to D-FF2} - \text{delay from the clock source to D-FF1}$
- Logic delay:  $T_{logic}$
- Clock period:  $T_{CLK}$
- Buffer delay:  $T_b$

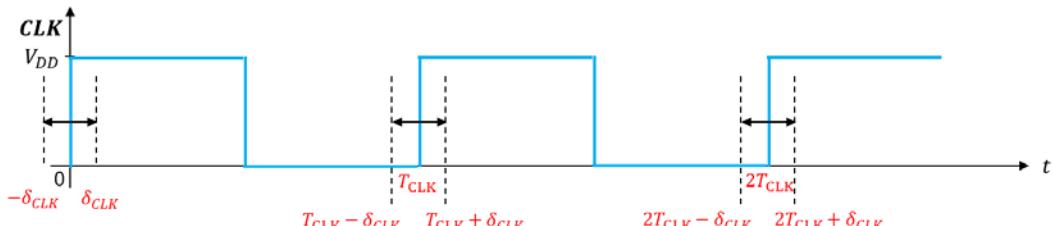


There are five buffers between the clock source and the clock pin of D-FF1 (and D-FF2) as shown in the figure. Ideally, the following inequalities should be satisfied:

- Setup time:  $T_s \leq T_{CLK} + T_{skew} - T_{logic} - T_{CQ}$
- Hold time:  $T_h \leq T_{CQ} + T_{logic} - T_{skew}$

However, there exist uncertainties such as delay variations due to temperature, so we should incorporate those uncertainties (variations) into the setup and hold time inequalities. The followings show the variation sources we are going to consider:

- $T_b \rightarrow T_b \pm \delta_b$
- $T_{CQ} \rightarrow T_{CQ} \pm \delta_{CQ}$
- $T_{logic} \rightarrow T_{logic} \pm \delta_{logic}$
- Clock jitter:  $\pm \delta_{CLK}$



Derive a new setup time and a new hold time constraints (inequalities) that should be satisfied under the variations.

Setup time constraint:

Suppose the first clock leaves the clock source at time 0 and the second clock leaves at time  $T_{CLK}$  ideally. Then, the minimum arrival time of the second clock at DFF2 is  $T_{CLK} - \delta_{CLK} + 5T_b - 5\delta_b$  and the maximum arrival time of the first clock at DFF1 is  $\delta_{CLK} + 5T_b + 5\delta_b$ , so the minimum skew is  $-2\delta_{CLK} - 10\delta_b$ . The maximum logic+CQ delay is  $T_{logic} + \delta_{logic} + T_{CQ} + \delta_{CQ}$ . Thus, the new setup time constraint is

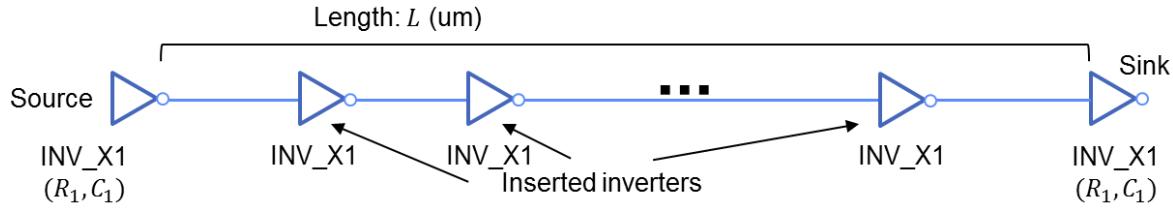
$$T_s \leq T_{CLK} - 2\delta_{CLK} - 10\delta_b - (T_{logic} + \delta_{logic} + T_{CQ} + \delta_{CQ})$$

Hold time constraint:

The minimum arrival time of the first clock at DFF1 is  $-\delta_{CLK} + 5T_b - 5\delta_b$  and the maximum arrival time of the first clock at DFF2 is  $\delta_{CLK} + 5T_b + 5\delta_b$ , so the maximum skew is  $2\delta_{CLK} + 10\delta_b$ . The minimum logic+CQ delay is  $T_{logic} + \delta_{logic} - T_{CQ} - \delta_{CQ}$ . Thus, the new hold time constraint is

$$T_h \leq (T_{logic} - \delta_{logic} + T_{CQ} - \delta_{CQ}) - (2\delta_{CLK} + 10\delta_b)$$

## Problem #4 (Interconnect Optimization, 10 points)



A buffer is composed of two inverters, so it consumes more power than an inverter. Thus, we can insert inverters instead of buffers to optimize a net while minimizing power consumption. However, only an even number of inverters can be inserted (if the inverter count is odd, there will be signal inversion).

In the above figure, the driver, the sink, and the inserted inverters have the same input capacitance ( $C_1$ ) and output resistance ( $R_1$ ), so the inserted inverters should be evenly distributed between the source and the sink. Now, suppose the optimal number of inverters we find is  $k$  where  $k$  is odd. Since  $k$  is odd, we have to insert either  $k - 1$  or  $k + 1$  inverters. Which will lead to a better result (shorter delay)?  $k - 1$  or  $k + 1$ ? Show all the details about your answer.

- Net length:  $L$  (um)
- Inverter output resistance and input capacitance:  $R_1, C_1$
- Unit wire resistance and capacitance:  $r, c$
- Inverter delay:  $d$

Suppose we insert  $p - 1$  inverters and evenly distribute them. Then, the delay will be

$$\begin{aligned}\tau(p-1) &= p \cdot \left\{ R_1 \left( \frac{cL}{p} + C_1 \right) + \frac{rL}{p} C_1 + 0.5rc \left( \frac{L}{p} \right)^2 \right\} + (p-1) \cdot d \\ &= R_1 cL + rLC_1 - d + pR_1 C_1 + pd + \frac{0.5rcL^2}{p} = \alpha + p(R_1 C_1 + d) + \frac{0.5rcL^2}{p} \\ (\alpha &\text{ is a constant})\end{aligned}$$

If we insert  $k - 1$  inverters, the delay becomes

$$\tau(k-1) = \alpha + k(R_1 C_1 + d) + \frac{0.5rcL^2}{k}$$

and if we insert  $k + 1$  inverters, the delay becomes

$$\tau(k+1) = \alpha + (k+2)(R_1 C_1 + d) + \frac{0.5rcL^2}{k+2}$$

$$\text{Thus, } \Delta = \tau(k+1) - \tau(k-1) = 2(R_1C_1 + d) + 0.5rcL^2 \left( \frac{1}{k+2} - \frac{1}{k} \right)$$

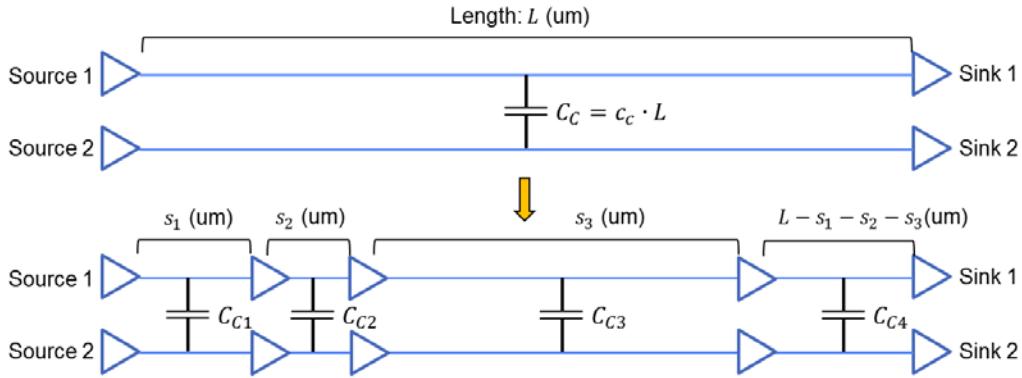
$$= 2(R_1C_1 + d) - \frac{2rcL^2}{k^2 + 2k}$$

$$\Delta > 0 \rightarrow k > -1 + \sqrt{1 + \frac{rcL^2}{(R_1C_1+d)}}$$

Thus, if  $k > -1 + \sqrt{1 + \frac{rcL^2}{(R_1C_1+d)}}$ , inserting  $k-1$  inverters is better.

If  $k < -1 + \sqrt{1 + \frac{rcL^2}{(R_1C_1+d)}}$ , inserting  $k+1$  inverters is better.

## Problem #5 (Interconnect Optimization, 10 points).



Two nets are routed as shown above. The coupling capacitance between them is  $C_c$ . You are supposed to insert buffers into the nets identically as shown above. All the drivers, sinks, and buffers are of the same type. The following list shows all the parameters and variables you should use:

- Output resistance:  $R$
- Input capacitance:  $C$
- Total length:  $L$  (um)
- Unit wire resistance:  $r_w/\mu m$
- Unit wire capacitance:  $c_w/\mu m$
- Unit coupling capacitance:  $c_c/\mu m$
- Buffer delay:  $d$

Insert buffers into the nets optimally, i.e., find the number of buffers to insert and their locations to minimize the delay of the nets.

In the worst case (i.e., the signal pattern of Source 1 is 0101... and that of Source 2 is 1010...), the total capacitance of a segment whose length is  $k(\mu m)$  is  $k(c_w + 2c_c)$ . Suppose we insert  $p - 1$  buffers and the length of the  $m$ -th segment is  $s_m$ . Then, the total delay is

$$\begin{aligned} \tau &= (p - 1) \cdot d + \sum_{i=1}^p [R \cdot \{s_i \cdot (c_w + 2c_c) + C\} + s_i \cdot r_w \cdot C + 0.5r_w(c_w + 2c_c)s_i^2] \\ &= (p - 1) \cdot d + R \cdot (c_w + 2c_c) \cdot L + pRC + r_w \cdot L \cdot C + 0.5r_w(c_w + 2c_c) \cdot (s_1^2 + \dots + s_p^2) \\ \frac{\partial \tau}{\partial s_i} &= 0.5r_w(c_w + 2c_c)\{2s_i - 2s_p\} = 0 \rightarrow s_1 = s_2 = \dots = s_p \end{aligned}$$

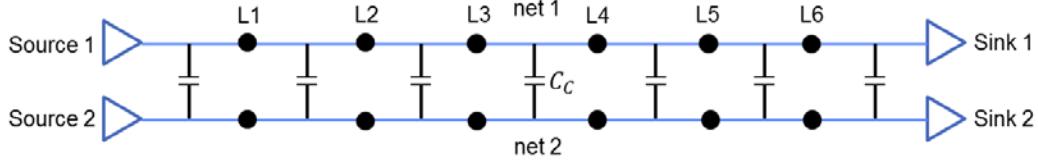
$$\therefore \tau = (p - 1) \cdot d + R \cdot (c_w + 2c_c) \cdot L + pRC + r_w \cdot L \cdot C + 0.5r_w(c_w + 2c_c) \cdot \frac{L^2}{p}$$

$$\frac{d\tau}{dp} = d + RC - \frac{r_w(c_w + 2c_c)L^2}{2p^2} = 0 \rightarrow p = \sqrt{\frac{r_w(c_w + 2c_c)L^2}{2(RC + d)}}$$

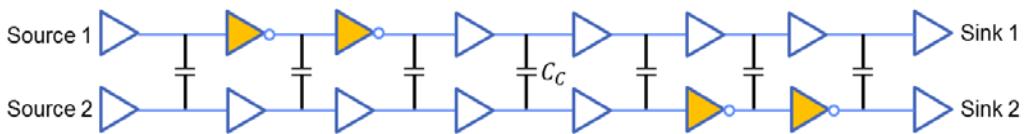
$$\therefore \# buffers = \sqrt{\frac{r_w(c_w + 2c_c)L^2}{2(RC + d)}} - 1$$

(and we evenly distribute the buffers).

## Problem #6 (Interconnect Optimization, 10 points).



Two nets (net 1 and net 2) are coupled as shown above and you are supposed to insert a repeater (inverter or buffer) into each of the designated locations (L1 ~ L6). For instance, the following figure shows a repeater insertion solution (its delay and area are  $12d$  and  $20S$ , respectively):



The following list shows the parameters used in this problem:

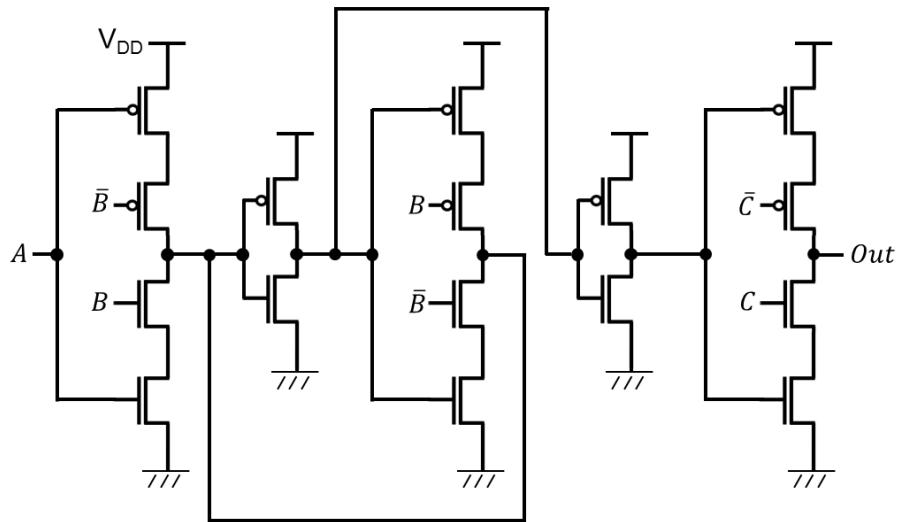
- Area of an inverter:  $S$
- Area of a buffer:  $2S$
- Delay of a net segment whose total capacitance is  $C_g$ :  $d$
- Delay of a net segment whose total capacitance is  $C_g + C_c$ :  $1.5d$
- Delay of a net segment whose total capacitance is  $C_g + 2C_c$ :  $2d$
- Input pattern of net 1: 010101...
- Input pattern of net 2: 101010...
- Signal inversion at the sinks is not allowed.

The goal is to optimally insert repeaters to minimize the sum of the delay values of the nets. However, you should also minimize the total area. Find an optimal solution that minimizes the sum of the delay values and the total area. Notice that minimizing the total delay has a higher priority. Thus, if there exists only one solution that minimizes the total delay, find it. If there exist multiple solutions that minimize the total delay, find the smallest-area solution among them.

Let the k-th repeater of net 1 be  $R_{1,k}$  and the k-th repeater of net 2 be  $R_{2,k}$ .

Minimization of the sum of the delays requires aligning the polarities of the signals. Thus, let  $R_{1,1}$  be an inverter and  $R_{2,1}$  be a buffer. Then, the signals are aligned after the first repeater. Now, we insert inverters after them to reduce the total area, i.e.,  $R_{1,2}, R_{1,3}, R_{1,4}, R_{1,5}, R_{2,2}, R_{2,3}, R_{2,4}, R_{2,5}$  are inverters. To satisfy the signal inversion constraint,  $R_{1,6}$  should be an inverter and  $R_{2,6}$  should be a buffer. Then, the sum of the delays of net 1 is  $2d + 5 * d + 2d = 9d$  and the total delay is  $14S$ .

**Problem #7 (CMOS Gates, 10 points).**



What does the above circuit do? Describe the function of the circuit in as much detail as possible (A, B, C: input, Out: output).

If  $C = 0$ , the output is floating (tri-state).

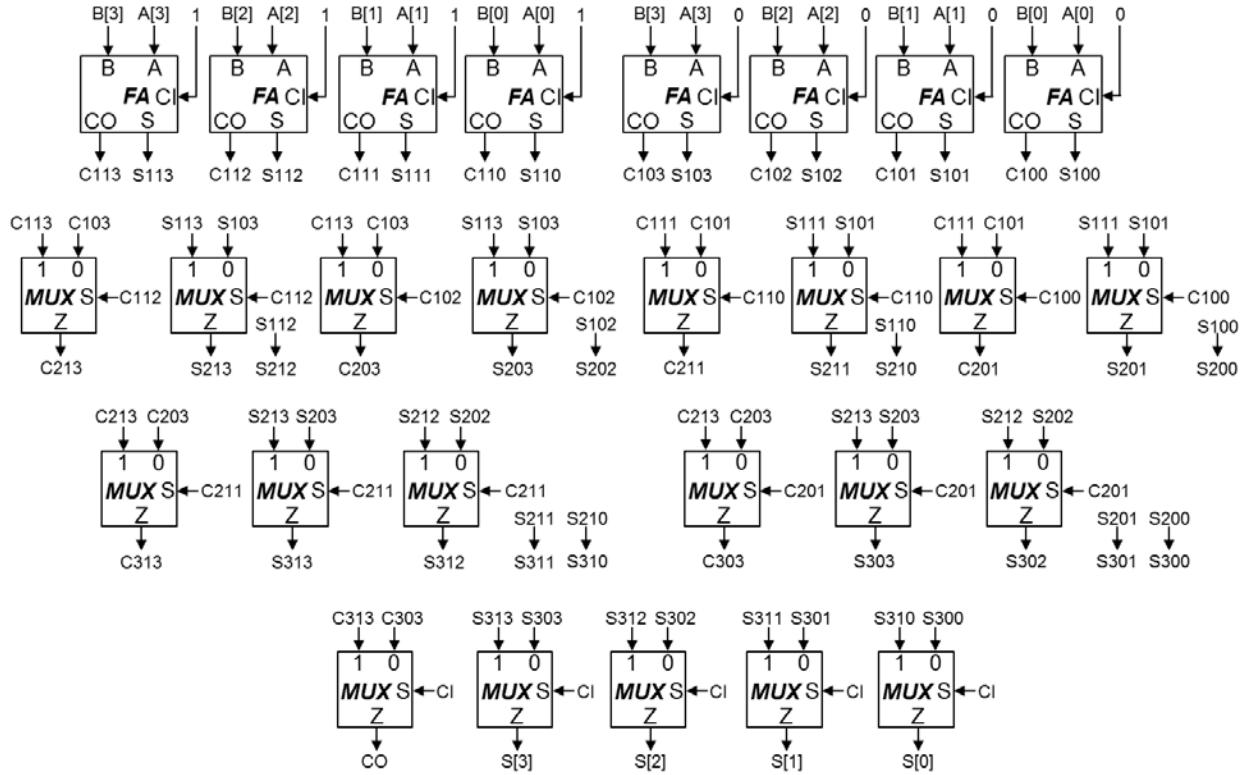
Suppose  $C = 1$ . If  $B = 0$ , input A is blocked and C holds the last value of input A.

If  $C = 1$  and  $B = 1$ , input A is transferred to the output.

Thus, this is a tri-state active-low D latch.

## Problem #8 (Adder, 10 points).

Draw a gate-level schematic of a four-bit conditional sum adder (use full adders and muxes). Input: A[3:0], B[3:0], Cl. Output: S[3:0], CO.



**EE434**  
**ASIC and Digital Systems**

**Midterm Exam 1**

**March 7, 2015. (5:10pm – 6pm)**

**Instructor: Dae Hyun Kim ([daehyun@eeecs.wsu.edu](mailto:daehyun@eeecs.wsu.edu))**

**Name:**

**WSU ID:**

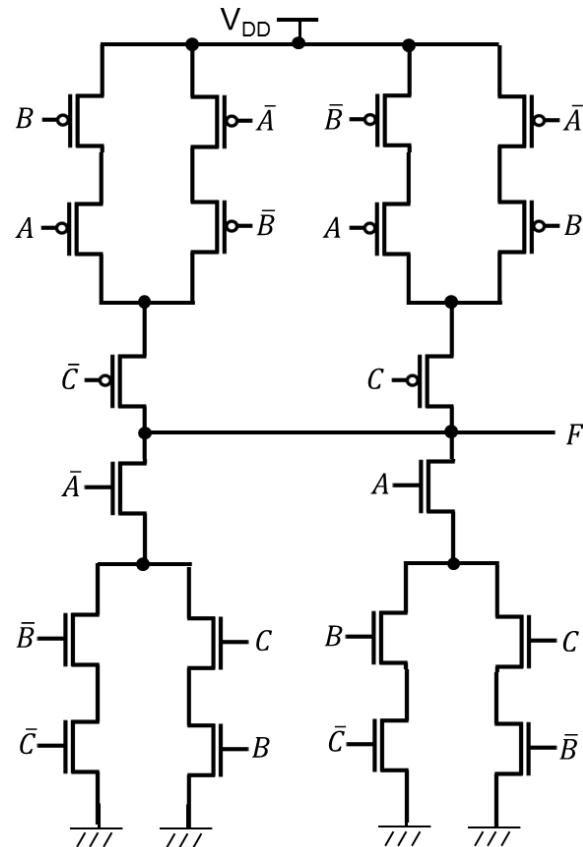
Problem	Points	
1	10	
2	10	
3	10	
4	10	
5	10	
6	10	
7	10	
8	10	
Total	80	

\* Allowed: Textbooks, cheat sheets, class notes, notebooks, calculators, watches

\* Not allowed: Electronic devices (smart phones, tablet PCs, laptops, etc.) except calculators and watches

**Problem #1 (Static CMOS gates, 10 points).**

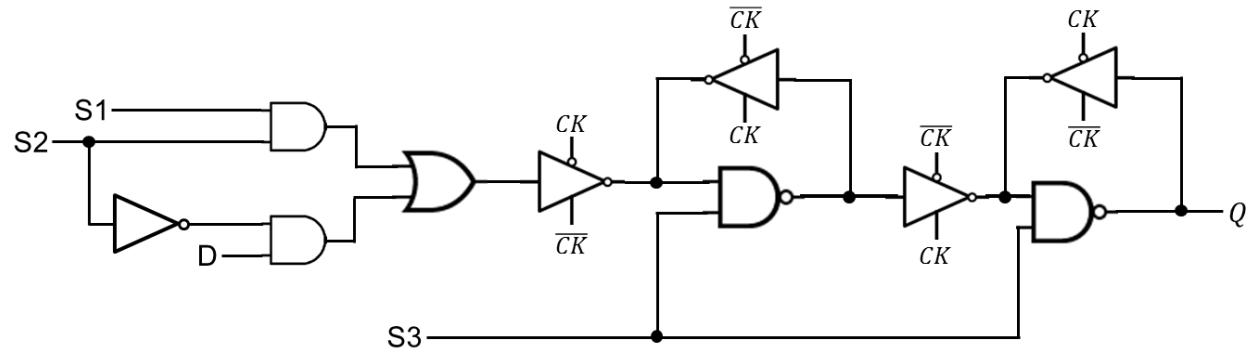
Represent  $F$  as a Boolean function of  $A, B$ , and  $C$ .



$$\begin{aligned}
 F &= \bar{A} \cdot (\bar{B} \cdot \bar{C} + B \cdot C) + A \cdot (B \cdot \bar{C} + \bar{B} \cdot C) = \bar{A} \cdot (\bar{B} \oplus \bar{C}) + A \cdot (B \oplus C) = \overline{\bar{A} \oplus (B \oplus C)} \\
 &= A \oplus B \oplus C
 \end{aligned}$$

## Problem #2 (Static CMOS gates, 10 points).

What does the following circuit do? Describe the function of the circuit in as much detail as possible (CK is a clock signal).



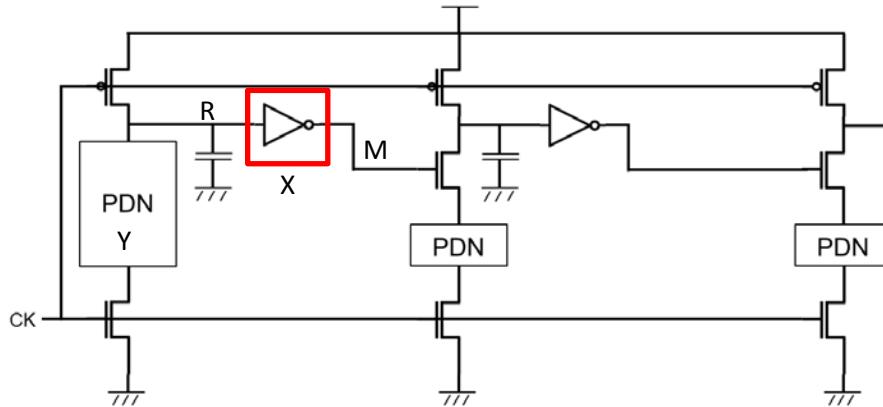
- 1) When  $S_3=0$ :  $Q=1$  regardless of all other signals.  $\Rightarrow S_3$  is an active-low set signal.
- 2) When  $S_3=1$ ,  $S_2=0$ :  $S_1 \cdot S_2 = 0$ , so the output of the OR gate is D. The rest of the circuit is just a positive edge-triggered D-FF in this case ( $Q$  captures D at each clock rising edge).
- 3) When  $S_3=1$ ,  $S_2=1$ : The output of the OR gate is  $S_1$ . The rest of the circuit is just a positive edge-triggered D-FF ( $Q$  captures  $S_1$  at each clock rising edge).

From this, we can conclude that this is a positive edge-triggered D-FF with an asynchronous active-low set and two inputs (D is selected when  $S_2=0$  and  $S_1$  is selected when  $S_2=1$ ).

For your information, this is a positive edge-triggered D-FF with an asynchronous active-low set, a scan input (S1), and an active-high scan enable (S2).

### Problem #3 (Domino Logic and DC Characteristics, 10 points).

The following shows a general three-stage domino logic. When CK is 0, the logic precharges all internal nodes, so node R is 1 and node M is 0. In general, the PMOS transistors for precharging are properly upsized. When CK is 1, the logic evaluates the PDNs. When the PDN Y is true, it discharges node R, so R becomes 0 and M becomes 1. However, the PDNs in the logic are cascaded, so the PDNs should switch quite fast. Thus, the NMOS transistors in the PDNs are also properly upsized.



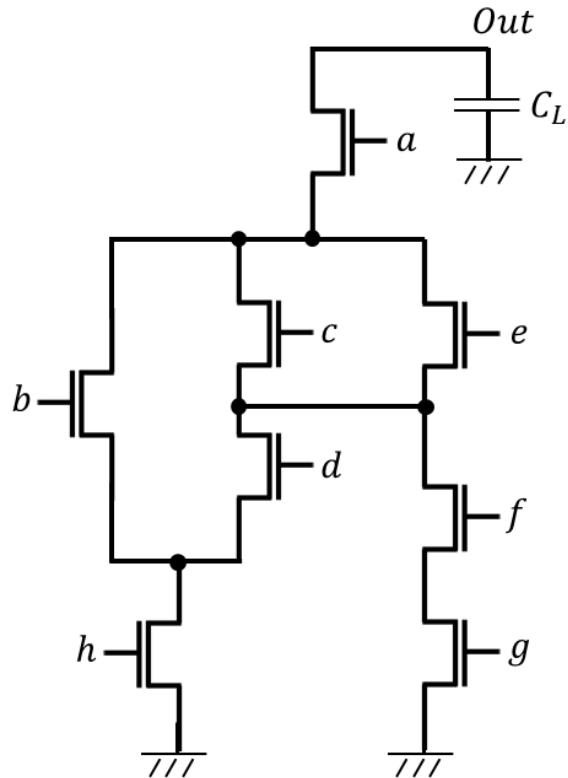
The NMOS transistor of inverter X can be small because it is used to discharge node M, which has a sufficiently small capacitance, during precharging. However, the PMOS transistor of inverter X should be properly upsized (e.g., by 16X) because it is used to charge node M during evaluation, so its switching time should be very short.

Question) Does this size imbalance (e.g., NMOS: 1X, PMOS: 16x) of inverter X cause any DC characteristics (noise margin) problems for inverter X? Explain why it causes (or does not cause any) DC characteristics (noise margin) problems for inverter X.

Yes, it causes some problems. When the PMOS transistor of inverter X is much larger than the NMOS transistor of inverter X, the DC characteristic curve of inverter X shifts to the right. In this case,  $NM_H = |V_{OH} - V_{IH}|$  significantly decreases (because  $V_{IH}$  goes up), but  $NM_L = |V_{IL} - V_{OL}|$  significantly increases (because  $V_{OL}$  goes up). Thus, we have some problems in the high noise margin. For instance, if PDN Y is false, R is 1, so M should be 0. However, if some noise signals affect node R or small charge sharing happens between R and an internal node in PDN Y, R will be slightly less than  $V_{DD}$ , which could invert the output of inverter X when PDN Y is false.

### Problem #4 (Transistor Sizing, 10 points).

Size the transistors in the following pull-down network.  $R_n$  is the resistance of a 1X NMOS transistor. Ignore parasitic capacitances. Target time constant:  $\tau_{target} = R_n \cdot C_L$ . Try to minimize the total area.



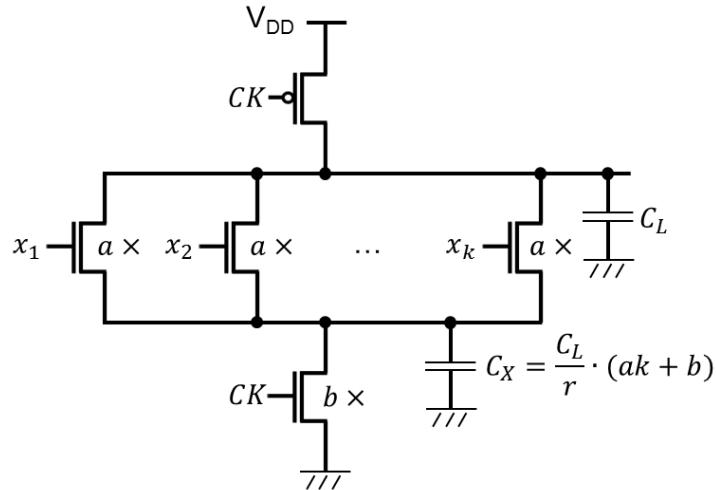
The worst-case path: (a-e-f-g) or (a-e-d-h) or (a-c-f-g) or (a-c-d-h), so we upsize all these transistors by 4X. Path (a-b-h) needs to be  $R_n$ , so b can be 2X.

a, c, d, e, f, g, h: 4X

b: 2X

## Problem #5 (Transistor Sizing, 10 points).

We want to design a  $k$ -input NOR gate. However, the static CMOS gate design methodology is not suitable for the design of the  $k$ -input NOR gate. Thus, we are going to design it using the dynamic CMOS design methodology. The following shows a schematic of the  $k$ -input NOR gate.



$R_n$  is the resistance of a 1X NMOS transistor. Target time constant:  $\tau_{target} = R_n \cdot C_L$ . All the transistors for  $x_1 \sim x_k$  are upsized to  $aX$  and the NMOS transistor for  $CK$  is upsized to  $bX$  ( $a$  and  $b$  are **real** numbers). Unfortunately, the parasitic capacitance at the internal node shown above is proportional to the sum of the width of the transistors connected to the node. Thus, the parasitic capacitance  $C_X$  at the internal node is  $C_X = \frac{C_L}{r} \cdot (ak + b)$  where  $C_L$  is the load cap and  $r$  is a constant ( $r > 1$ ). We minimize the total width

$$Width = a \cdot k + b.$$

Find  $a$  minimizing the total width (i.e., represent  $a$  as a function of  $k$  and  $r$ ).

$$\text{Constraints: } \left(\frac{R_n}{a} + \frac{R_n}{b}\right) \cdot C_L + \frac{R_n}{b} \cdot C_X = R_n \cdot C_L \Rightarrow \left(\frac{1}{a} + \frac{1}{b}\right) + \frac{ak+b}{br} = 1 \Rightarrow$$

$$\begin{aligned} \frac{1}{a} + \frac{1}{b} + \frac{1}{r} + \frac{ak}{br} = 1 &\Rightarrow \frac{1}{b} \left(1 + \frac{ak}{r}\right) = 1 - \frac{1}{a} - \frac{1}{r} = \frac{ar - a - r}{ar} \Rightarrow \frac{1}{b} \\ &= \frac{ar - a - r}{ar} \cdot \frac{r}{ak + r} = \frac{ar - a - r}{ka^2 + ra} \Rightarrow b = \frac{ka^2 + ra}{(r-1)a - r} \end{aligned}$$

$$\text{Minimize } W = ak + b = ak + \frac{ka^2 + ra}{(r-1)a - r}$$

$$W' = k + \frac{(2ka + r)((r-1)a - r) - (ka^2 + ra)(r-1)}{(r-1)a - r)^2}$$

$$\begin{aligned}
&= k + \frac{\{2k(r-1)a^2 - 2kra + r(r-1)a - r^2\} - \{k(r-1)a^2 + r(r-1)a\}}{(r-1)a - r} \\
&= k + \frac{k(r-1)a^2 - 2kra - r^2}{(r-1)a - r} \\
&= \frac{k(r-1)^2a^2 - 2kr(r-1)a + kr^2 + k(r-1)a^2 - 2kra - r^2}{(r-1)a - r} \\
&= \frac{kr(r-1)a^2 - 2kr^2a + kr^2 - r^2}{(r-1)a - r} = \frac{kr(r-1)a^2 - 2kr^2a + (k-1)r^2}{(r-1)a - r} \\
W' = 0 \Rightarrow a &= \frac{kr^2 \pm \sqrt{k^2r^4 - k(k-1)(r-1)r^3}}{kr(r-1)} \\
&= \frac{kr^2 \pm \sqrt{k^2r^4 - (k^2r^4 - k^2r^3 - kr^4 + kr^3)}}{kr(r-1)} = \frac{kr^2 \pm \sqrt{k^2r^3 + kr^4 - kr^3}}{kr(r-1)}
\end{aligned}$$

$W$  is minimized when

$$a = \frac{kr^2 + \sqrt{k^2r^3 + kr^4 - kr^3}}{kr(r-1)}$$

Quick double check:

- 1) When  $r \rightarrow \infty$  (i.e.,  $C_X$  is negligible),  $a \rightarrow \frac{k+\sqrt{k}}{k} = 1 + \frac{1}{\sqrt{k}}$   
 $\frac{1}{a} + \frac{1}{b} = 1$ , so  $\frac{1}{b} = 1 - \frac{k}{k+\sqrt{k}} = \frac{\sqrt{k}}{k+\sqrt{k}} \Rightarrow b \rightarrow 1 + \sqrt{k}$  (which is an answer we obtain when we ignore  $C_X$ ).

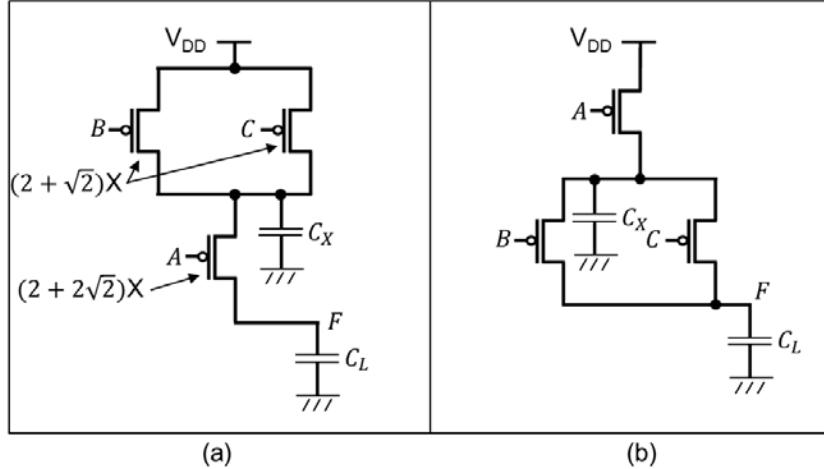
For example, if  $k = 1$  (just a two-input NOR ignoring  $C_X$ ),  $a = b = 2$ .

- 2) When  $r \rightarrow 1$  (i.e.,  $C_X \gg C_L$ ),  $a = \frac{k+\sqrt{k^2}}{k(r-1)} = \frac{2}{r-1}$   
 $\frac{1}{a} + \frac{1}{b} + \frac{ak}{b} = \frac{r-1}{r}$ , so  $\frac{1}{b}(1+ak) = \frac{r-1}{r} - \frac{1}{a} \Rightarrow \frac{1}{b} = \frac{r-1}{2r(1+ak)} \Rightarrow b = \frac{2r}{r-1} \left(1 + \frac{2k}{r-1}\right)$

In this case, we are supposed to upsize  $a$  to  $(2 + \delta)X$  where  $\delta$  is a very small positive number and  $b$  to a large number. This makes sense because the fall time is dominated by the internal cap, so we need to upsize the CK NMOS transistor. (However,  $x_i$  should be upsized by  $(2 + \delta)X$ , otherwise you won't be able to achieve the target time constant).

## Problem #6 (Switching Characteristics, 10 points).

We design  $F = \overline{A + B \cdot C}$  using the static CMOS gate design style. The following shows a transistor-level schematic of the pull-up network of the design:



Both (a) and (b) are functionally equal. The internal node in the pull-up network has parasitic capacitance  $C_X$ . Since the parasitic capacitance of an internal node is generally determined by the size of the transistors connected to the node, both (a) and (b) have the same parasitic capacitance  $C_X$ .  $R_n$  is the resistance of a 1X NMOS transistor.  $\mu_n = 2\mu_p$ . Target time constant:  $\tau_{target} = R_n \cdot C_L$ . If we optimally size the pull-up network to minimize the total width without considering  $C_X$ , the pMOS A is upsized to  $(2 + 2\sqrt{2})X$  and the pMOS B and C are upsized to  $(2 + \sqrt{2})X$  as shown in the figure.

Compute the worst-case rise time in (a) and in (b) considering  $C_X$ , i.e., represent the worst-case rise time as a function of  $R_n$ ,  $C_L$ , and  $C_X$ . Which design has shorter rise time?

$$(a) \text{Worst cases: } (A, B, C) = (0, 0, 1) \text{ or } (0, 1, 0). R_B = R_C = \frac{R_p}{2+\sqrt{2}} = \frac{2R_n}{2+\sqrt{2}}. R_A = \frac{R_p}{2+2\sqrt{2}} = \frac{2R_n}{2+2\sqrt{2}} = \frac{R_n}{1+\sqrt{2}}$$

$$\tau_a = R_B \cdot C_X + (R_B + R_A) \cdot C_L = \frac{2R_n}{2+\sqrt{2}} \cdot C_X + \left( \frac{2R_n}{2+\sqrt{2}} + \frac{R_n}{1+\sqrt{2}} \right) \cdot C_L$$

$$t_r = 2.2\tau_a$$

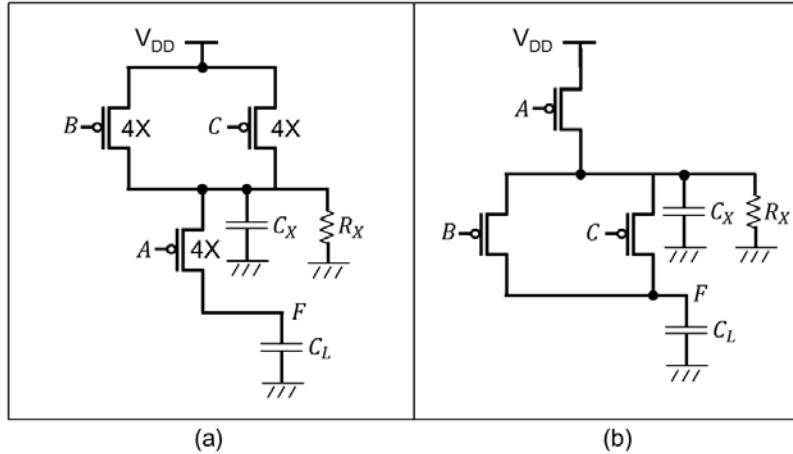
$$(b) \tau_b = R_A \cdot C_X + (R_A + R_B) \cdot C_L = \frac{R_n}{1+\sqrt{2}} \cdot C_X + \left( \frac{2R_n}{2+\sqrt{2}} + \frac{R_n}{1+\sqrt{2}} \right) \cdot C_L$$

$$t_r = 2.2\tau_b$$

I prefer design (b) because it has shorter rise time than (a).

### Problem #7 (Power Consumption, 10 points).

We design  $F = \overline{A + B \cdot C}$  using the static CMOS gate design style. The following shows a transistor-level schematic of the pull-up network of the design:



Both (a) and (b) are functionally equal. The internal node in the pull-up network has parasitic capacitance  $C_X$  and also a parasitic resistance  $R_X$  connected to the ground (this is a leaky path, i.e., even if  $A=B=C=1$  and  $C_X = V_{DD}$ ,  $C_X$  will be slowly discharged through  $R_X$ ). All the transistors are upsized by 4X as shown in the figure. The following table shows the probability that each input signal is 0 or 1:

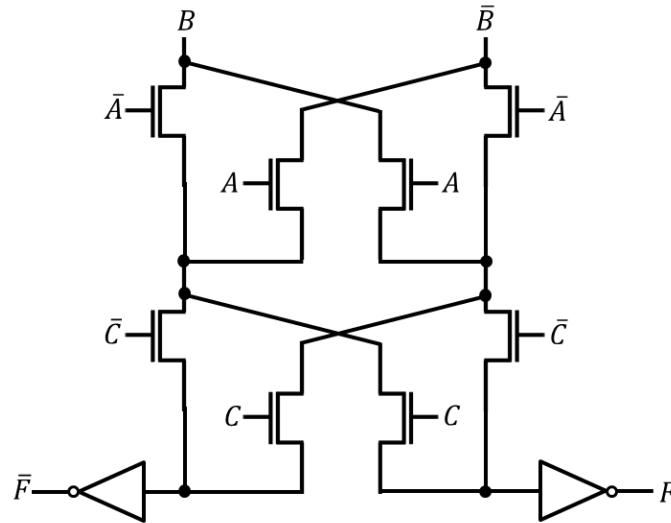
	0	1
A	0.8	0.2
B	0.2	0.8
C	0.2	0.8

We want to minimize power consumption (both dynamic and leakage power) of the gate. Which design do you prefer? (a) or (b)? Select one of them and explain why you prefer the design to minimize power consumption. You can qualitatively and intuitively explain it (without performing accurate probability and power computation).

I would prefer design (a). In (b), A is frequently set to 0, so  $C_X$  will be frequently charged and leak through  $R_X$ . In (a), however, B and C are less frequently set to 0, so it will charge  $C_X$  less frequently than (b). Thus, we can minimize leakage power. However, both (a) and (b) consume the same amount of dynamic power.

**Problem #8 (Pass-Transistor Logic, 10 points).**

Represent  $F$  as a Boolean function of  $A, B$ , and  $C$ .



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

**EE434**  
**ASIC and Digital Systems**

**Midterm Exam 2**

**April 8, 2016. (5:10pm – 6pm)**

**Instructor: Dae Hyun Kim ([daehyun@eecs.wsu.edu](mailto:daehyun@eecs.wsu.edu))**

**Name:**

**WSU ID:**

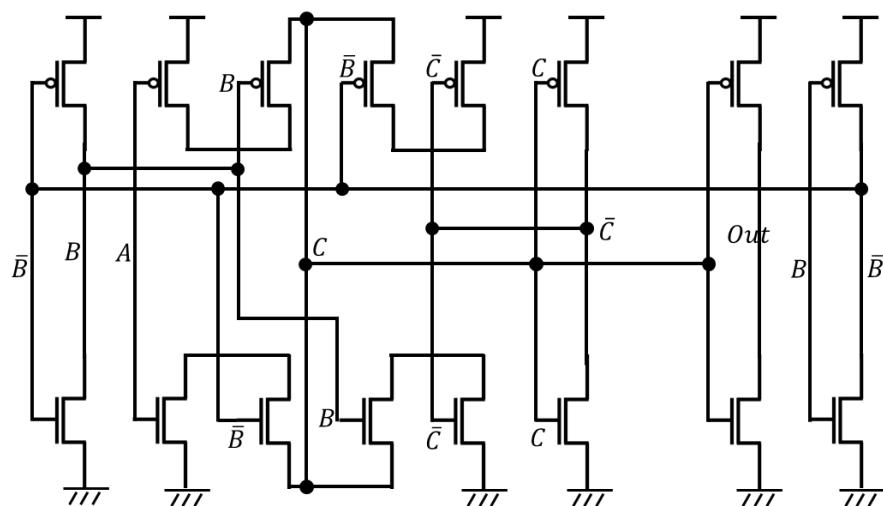
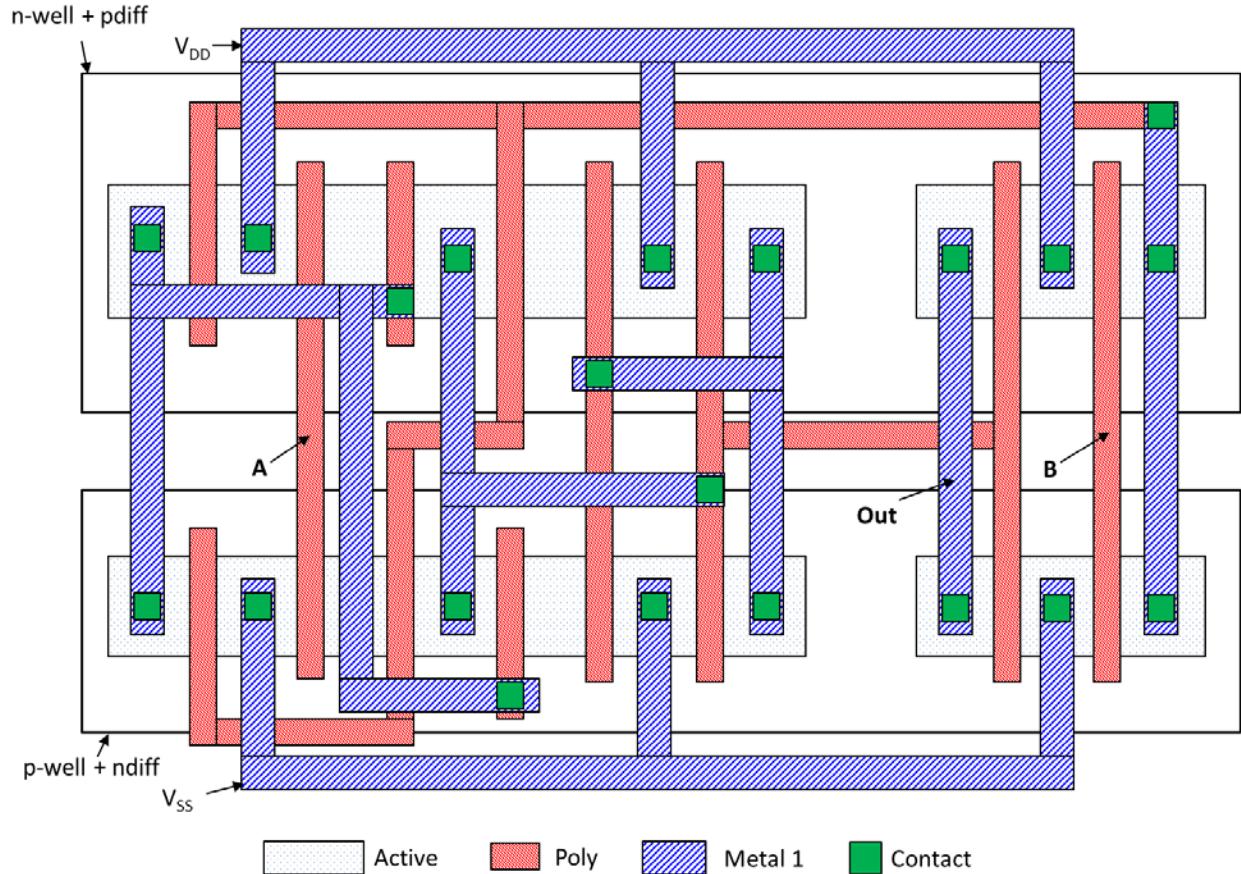
Problem	Points	
1	15	
2	15	
3	15	
4	10	
5	13	
6	12	
Total	80	

\* Allowed: Textbooks, cheat sheets, class notes, notebooks, calculators, watches

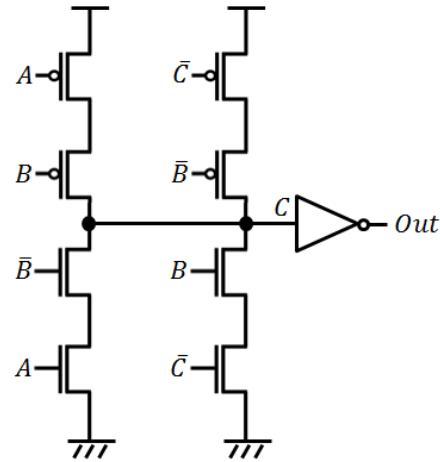
\* Not allowed: Electronic devices (smart phones, tablet PCs, laptops, etc.) except calculators and watches

### Problem #1 (Layout, 15 points).

Represent  $Out$  as a Boolean function of  $A$  and  $B$  or describe the function of the following layout in as much detail as possible (Primary inputs:  $A$ ,  $B$ . Primary output:  $Out$ ).



After further simplification, we get the following:



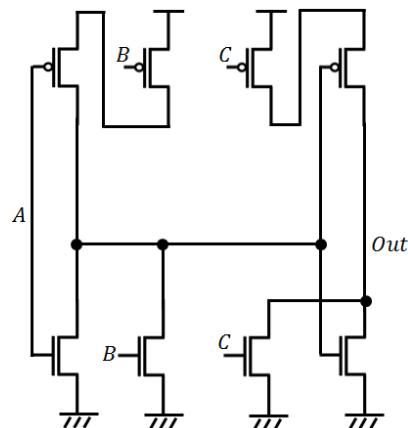
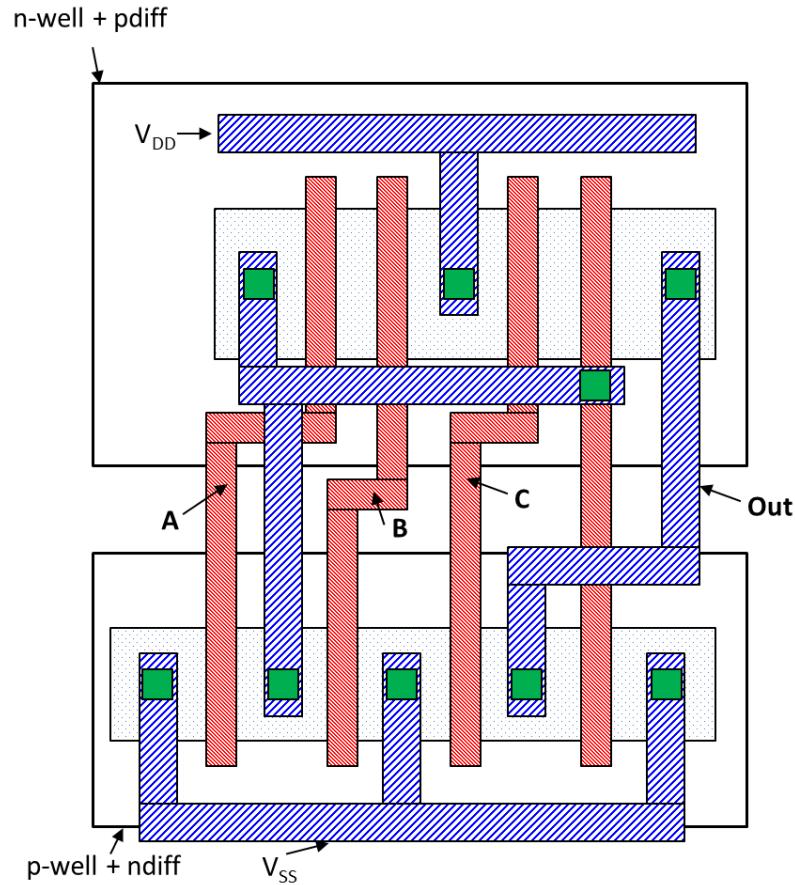
If  $B=1$ , the output holds the current value.

If  $B=0$ , the output is  $\bar{C}$ , which is  $A$ , so the output follows  $A$ .

$=>$  Thus, this is an active-low D-latch (A is a signal input and B is an enable or CK signal).

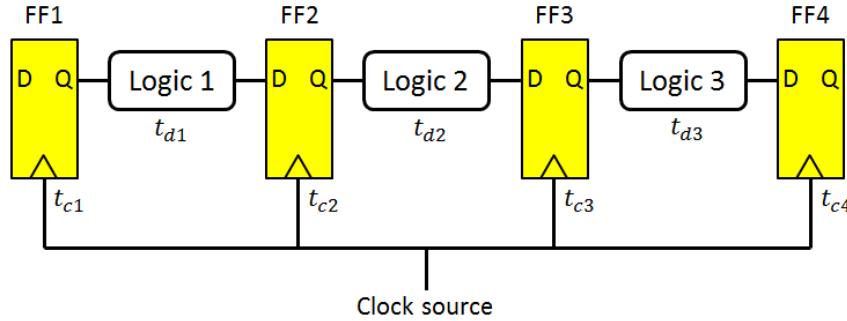
## Problem #2 (Layout, 15 points).

Represent *Out* as a Boolean function of *A*, *B* and *C* or describe the function of the following layout in as much detail as possible (Primary inputs: *A*, *B*, *C*. Primary output: *Out*).



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

### Problem #3 (Timing Analysis, 15 points).



Four flip-flops are connected as shown above.

- Clock period:  $t_{CLK} = 500\text{ps}$
- F/F internal delay:  $t_{CQ} = 50\text{ps}$
- Setup time of each F/F:  $t_s = 50\text{ps}$
- Delay of Logic 1:  $t_{d1} = 450\text{ps}$
- Delay of Logic 2:  $t_{d2} = 480\text{ps}$
- Delay of Logic 3:  $t_{d3} = 250\text{ps}$
- Delay of the clock from the clock source to FF1:  $t_{c1} = 400\text{ps}$
- Delay of the clock from the clock source to FF#:  $t_{c\#}$  ( $\#=2, 3, 4$ )

Since the delays of Logic 1 and Logic 2 are too large, Logic 1 and Logic 2 will violate the setup time constraint if the clock skew is zero (i.e.,  $t_{c1} = t_{c2} = t_{c3} = t_{c4}$ ). To resolve this issue, we want to intentionally use the clock skew. However, we also want to minimize the total clock delay ( $t_{c1} + t_{c2} + t_{c3} + t_{c4}$ ) to minimize the clock power consumption. Find  $t_{c2}$ ,  $t_{c3}$ , and  $t_{c4}$  that satisfies the setup time constraint and minimizes the total clock delay ( $t_{c1} + t_{c2} + t_{c3} + t_{c4}$ ).

1) To satisfy the setup time constraint of FF2, the following inequality should hold:

$$t_{CQ} + t_{d1} \leq t_{CLK} - t_s + (t_{c2} - t_{c1}) \Rightarrow 450\text{ps} \leq t_{c2} \Rightarrow t_{c2} = 450\text{ps} + \delta_2 \quad (\delta_2 > 0)$$

2) To satisfy the setup time constraint of FF3, the following inequality should hold:

$$t_{CQ} + t_{d2} \leq t_{CLK} - t_s + (t_{c3} - t_{c2}) \Rightarrow t_{c2} + 80\text{ps} \leq t_{c3} \Rightarrow$$

$$t_{c3} = (450\text{ps} + \delta_2) + 80\text{ps} + \delta_3 = 530\text{ps} + \delta_2 + \delta_3 \quad (\delta_3 > 0)$$

3) To satisfy the setup time constraint of FF4, the following inequality should hold:

$$t_{CQ} + t_{d3} \leq t_{CLK} - t_s + (t_{c4} - t_{c3}) \Rightarrow t_{c3} - 150\text{ps} \leq t_{c4} \Rightarrow$$

$$t_{c4} = (530\text{ps} + \delta_2 + \delta_3) - 150\text{ps} + \delta_4 = 380\text{ps} + \delta_2 + \delta_3 + \delta_4 \quad (\delta_4 > 0)$$

The total clock delay =  $t_{c1} + t_{c2} + t_{c3} + t_{c4} = (400ps) + (450ps + \delta_2) + (530ps + \delta_2 + \delta_3) + (380ps + \delta_2 + \delta_3 + \delta_4)$

To minimize the total clock delay, we set  $\delta_2$ ,  $\delta_3$ , and  $\delta_4$  to zero. Then, we obtain

$$t_{c2} = 450ps, \quad t_{c3} = 530ps, \quad t_{c4} = 380ps$$

## Problem #4 (Timing Analysis, 10 points).

Can a path have both setup and hold time violations at the same time? Yes/No. Explain why.

A setup time constraint for a path is as follows:

$$t_{LOGIC} \leq t_{CLK} - t_s - t_{CQ} + t_{skew}$$

A hold time constraint for the same path is as follows:

$$t_{LOGIC} \geq t_h - t_{CQ} + t_{skew}$$

Combining both results in:

$$t_h - t_{CQ} + t_{skew} \leq t_{LOGIC} \leq t_{CLK} - t_s - t_{CQ} + t_{skew}$$

Thus, violating both means:

$$1) t_{LOGIC} < t_h - t_{CQ} + t_{skew}$$

$$2) t_{LOGIC} > t_{CLK} - t_s - t_{CQ} + t_{skew}$$

which leads to:

$$t_{CLK} - t_s - t_{CQ} + t_{skew} < t_h - t_{CQ} + t_{skew}$$

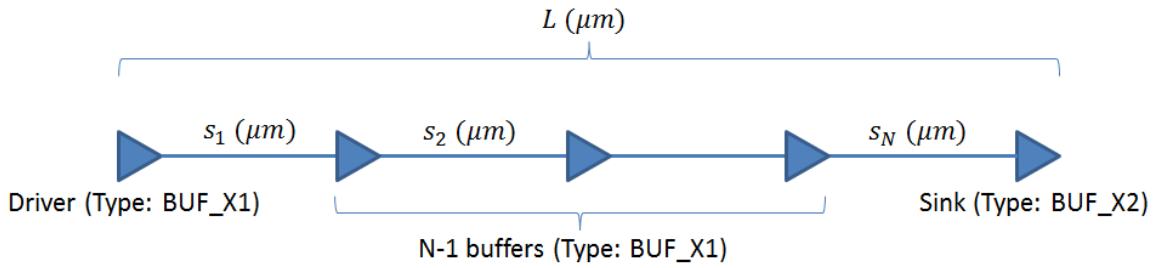
$$\Rightarrow t_{CLK} - t_s < t_h$$

$$\Rightarrow t_{CLK} < t_h + t_s$$

Thus, it is THEORETICALLY possible if the clock period is less than the sum of the hold and setup times.

(Of course, this won't happen in reality because the clock period is much greater than  $t_h + t_s$ ).

## Problem #5 (Interconnect, 13 points).



A source (type: BUF\_X1) drives a sink (type: BUF\_X2) through a net and you are supposed to optimally insert  $N-1$  buffers (type: BUF\_X1) between them. Find the number of buffers to insert (i.e., find  $N$ ) and the optimal locations of the buffers (i.e., find  $s_1, s_2, \dots, s_N$ ) minimizing the total delay.

- Output resistance of BUF\_X1:  $R_1$
- Input capacitance of BUF\_X1:  $C_1$
- Input capacitance of BUF\_X2:  $C_2$
- Total length of the net:  $L (\mu m)$
- Unit wire resistance:  $r_w$
- Unit wire capacitance:  $c_w$

Suppose  $s_N$  is known. Then, the  $N-2$  buffers between the driver and the  $(N-1)$ -th buffer should be evenly distributed to minimize the delay. Thus, we simply get

$$s_1 = s_2 = \dots = s_{N-1} = k$$

The sum of the delays of the first  $(N-1)$  segments is

$$\tau_L = (N-1)\{R_1(k \cdot c_w + C_1) + k \cdot r_w \cdot C_1 + 0.5 \cdot r_w \cdot c_w \cdot k^2\}$$

and the delay of the rightmost segment is

$$\tau_R = R_1((L - (N-1)k) \cdot c_w + C_2) + (L - (N-1)k) \cdot r_w \cdot C_2 + 0.5 \cdot r_w \cdot c_w \cdot (L - (N-1)k)^2$$

so the total delay is

$$\tau = R_1\{c_w L + C_1(N-1) + C_2\} + r_w\{C_2 L + k(N-1)(C_1 - C_2)\} + 0.5 \cdot r_w \cdot c_w \{(N-1)k^2 + (L - (N-1)k)^2\}$$

and

$$\frac{d\tau}{dk} = 0 = r_w(C_1 - C_2)(N-1) + r_w \cdot c_w \cdot \{k(N-1) - (N-1)L + k(N-1)^2\}$$

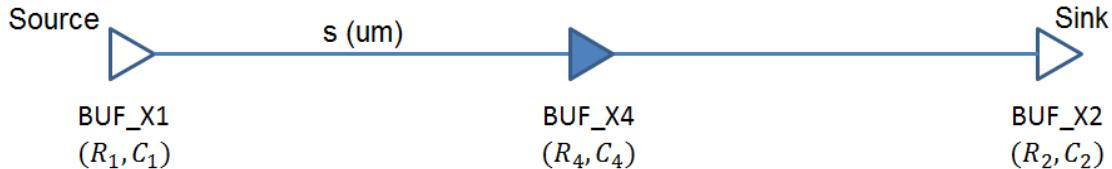
$$\Rightarrow k = \frac{(C_2 - C_1) + c_w L}{c_w \cdot N}$$

$$\text{Thus, } s_1 = s_2 = \dots = s_{N-1} = \frac{(C_2 - C_1) + c_w L}{c_w \cdot N}, \quad s_N = L - k \cdot (N - 1) = \frac{c_w \cdot L - (N - 1) \cdot (C_2 - C_1)}{c_w \cdot N}$$

## Problem #6 (Interconnect, 12 points).

A source (type: BUF\_X1) drives a sink (type: BUF\_X2) through a net and we want to optimally insert a buffer (type: BUF\_X4) between them to minimize the total signal delay as shown in the figure below.  $s$  is the distance between the source and the buffer.

Answer the following questions.



- Output resistance of each cell:  $R_{\#}$  ( $R_1 > R_2 > R_4$ )
- Input capacitance of each cell:  $C_{\#}$  ( $C_1 < C_2 < C_4$ )
- Unit wire resistance and capacitance:  $r_w, c_w$

1) If  $R_4$  goes up,  $s$  should go up. (True/False)

If  $R_4$  goes up, the delay of the right segment goes up. To reduce it, we should move BUF\_X4 to the right.

2) If  $C_4$  goes up,  $s$  should go up. (True/False)

If  $C_4$  goes up, the delay of the left segment goes up. To reduce it, we should move BUF\_X4 to the left.

3) If  $R_1$  goes up,  $s$  should go up. (True/False)

If  $R_1$  goes up, the delay of the left segment goes up. To reduce it, we should move BUF\_X4 to the left.

4) If  $C_2$  goes up,  $s$  should go up. (True/False)

If  $C_2$  goes up, the delay of the right segment goes up. To reduce it, we should move BUF\_X4 to the right.

5) If  $r_w$  goes up,  $s$  should go up. (True/False)

If  $r_w$  goes up, the impact of  $R_1$  and  $R_4$  goes down. Thus, it is better to move the buffer to the center point, so should move BUF\_X4 to the right.

6) If  $c_w$  goes up,  $s$  should go up. (True/False)

If  $c_w$  goes up, the impact of  $C_4$  and  $C_2$  goes down. Thus, it is better to move the buffer to the center point, so we should move BUF\_X4 to the right.

We can analyze it quantitatively too. The delay of the left segment is

$$\tau_1 = R_1(sc_w + C_4) + sr_wC_4 + \frac{1}{2}r_w c_w s^2$$

and that of the right segment is

$$\tau_2 = R_4((L - s)c_w + C_2) + (L - s)r_wC_2 + \frac{1}{2}r_w c_w(L - s)^2$$

so the total delay is  $\tau = \tau_1 + \tau_2$ . To minimize  $\tau$ ,

$$\begin{aligned}\frac{d\tau}{ds} &= 0 = (R_1c_w + r_wC_4 + r_w c_w s) + (-R_4c_w - r_wC_2 - r_w c_w(L - s)) \\ s &= \frac{1}{2}L - \frac{R_1 - R_4}{2r_w} - \frac{C_4 - C_2}{2c_w}\end{aligned}$$

**EE434**  
**ASIC and Digital Systems**

**Final Exam**

**May 4, 2017. (8am – 10am)**

**Instructor: Dae Hyun Kim ([daehyun@eeecs.wsu.edu](mailto:daehyun@eeecs.wsu.edu))**

**Name:**

**WSU ID:**

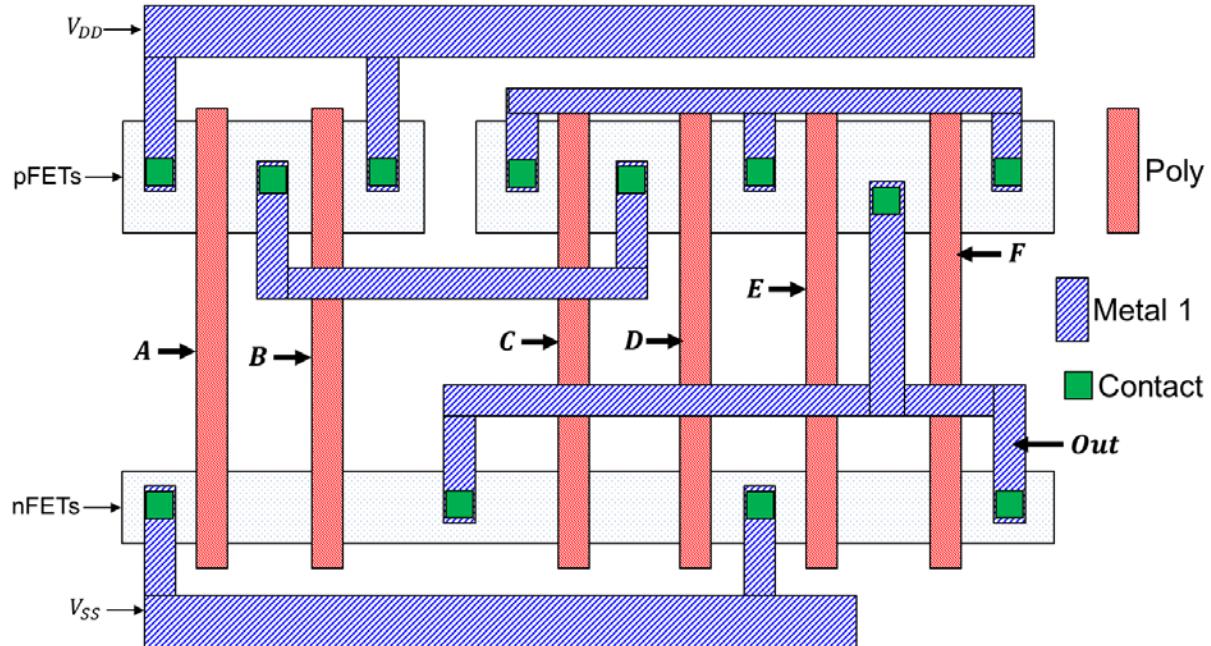
Problem	Points	
1	10	
2	10	
3	10	
4	10	
5	30	
6	10	
7	10	
8	10	
Total	80	

\* Allowed: Textbooks, cheat sheets, class notes, notebooks, calculators, watches

\* Not allowed: Electronic devices (smart phones, tablet PCs, laptops, etc.) except calculators and watches

## Problem #1 (Layout Analysis, 10 points)

The following combinational logic has six primary inputs ( $A, B, C, D, E, F$ ) and a primary output (Out). Find all input vectors that can detect a stuck-at-1 fault at input E.



$$Z = \overline{A \cdot B + C \cdot D + E \cdot F}$$

$$Z_f = \overline{A \cdot B + C \cdot D + F}$$

$$Z \oplus Z_f = 1 \rightarrow E = 0 \rightarrow F = 1 \rightarrow A \cdot B + C \cdot D = 0$$

$$\therefore (A \ B \ C \ D)$$

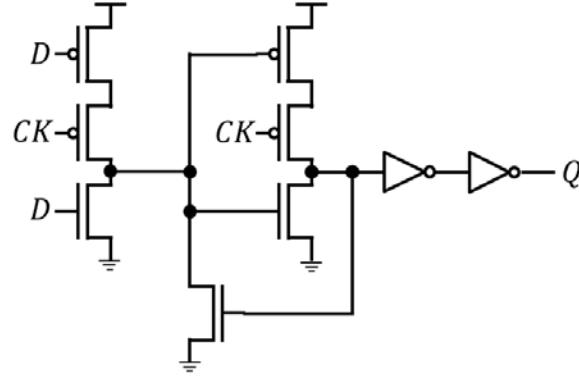
$$= (0 \ 0 \ 0 \ 0), (0 \ 0 \ 0 \ 1), (0 \ 0 \ 1 \ 0), (0 \ 1 \ 0 \ 0), (0 \ 1 \ 0 \ 1), (0 \ 1 \ 1 \ 0), (1 \ 0 \ 0 \ 0), (1 \ 0 \ 0 \ 1), (1 \ 0 \ 1 \ 0)$$

And

$$(E \ F) = (0 \ 1)$$

## Problem #2 (Static CMOS gates, 10 points)

Describe the function of the following circuit in as much detail as possible ( $D$ : data input,  $CK$ : clock).



(Partovi, ISSCC'96)

Suppose X is the node driving the pFET and the nFET in the second stage and Y is the node driving the first inverter.

- When  $CK=0$ .
  - If  $D=0$ , X is 1 due to the pFETs in the first stage.  $\Rightarrow Y=0 \Rightarrow Q=0$ .
  - If  $D=1$ , X is 0 due to the nFET in the first stage.  $\Rightarrow Y=1 \Rightarrow Q=1$ .
  - Thus,  $Q=D$  if  $CK=0$ .
- When  $CK=1$ .
  - Y is floating, i.e., it holds the previous value regardless of D.

Thus, this is an active-low D-latch (i.e., a D-latch in which  $Q=D$  when  $CK=0$ ).

### Problem #3 (Timing Analysis, 10 points)

Answer the following questions.

- WNS can be less than TNS (i.e., “WNS<TNS” can happen). (True/**False**)

$$TNS = \sum \text{Negative slack}(NS) = WNS + \sum_{NS \neq WNS} NS. \sum_{NS \neq WNS} NS \leq 0, \text{ so } TNS \leq WNS.$$

- TNS can be less than WNS (i.e., “TNS<WNS” can happen). (**True**/False)

- WNS can be equal to TNS (i.e., “WNS=TNS” can happen). (**True**/False)

- A design has only two violating paths. In this case, the following can happen.

(True/**False**)

- “WNS of the design is -2ns and TNS of the design is -4.5ns.

In this case,  $TNS = WNS1 + WNS2$  where  $WNS1$  is the negative slack of the first critical path and  $WNS2$  is the negative slack of the second critical path ( $WNS1 \leq WNS2$ ).

$TNS = -4.5ns = WNS1 + WNS2 = -2ns + WNS2$ , so  $WNS2$  is  $-2.5ns$ , but this is a contradiction because  $WNS1$  should be less than or equal to  $WNS2$ . Thus, this cannot happen.

- A design has only four violating paths. In this case, the following can happen.

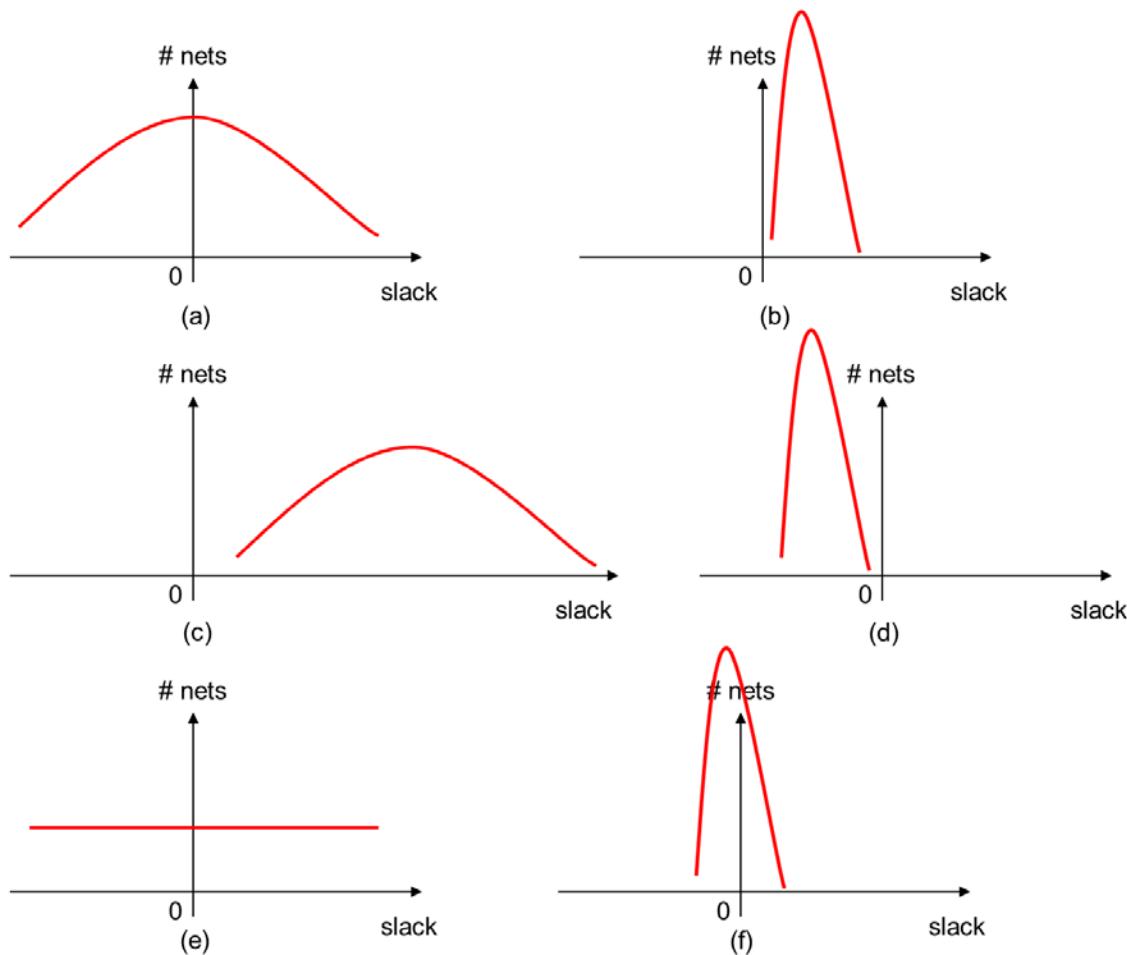
(True/**False**)

- “WNS of the design is -2ns and TNS of the design is -8.2ns.

$WNS1 = -2ns \leq WNS2 \leq WNS3 \leq WNS4$ . Thus,  $TNS = \sum WNS \geq -8ns$ , so  $TNS$  cannot be  $-8.2ns$ .

## Problem #4 (Timing Analysis, 10 points)

You are given six designs, (a), (b), ..., (f). Their timing analysis results are shown below. It is also known that the power consumption and the total layout area of a design are proportional to the total positive slack. You are supposed to choose a design and send it to a foundry for fabrication without any further optimization. Choose one among the six designs and explain why you decided to choose the design for fabrication.

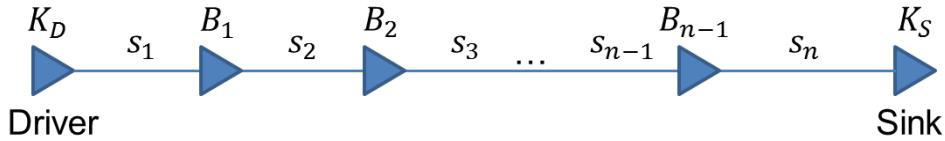


I would choose (b) for the following reasons.

	Timing	Power	Area
(a)	Violated		
(b)		Low	Small
(c)		High	Large
(d)	Violated		
(e)	Violated		
(f)	Violated		

## Problem #5 (Interconnect Optimization, 30 points)

The following figure shows a net optimized by buffer insertion. The driver and the sink are denoted by  $K_D$  and  $K_S$ , respectively, and the inserted buffers are denoted by  $B_i$  ( $1 \leq i \leq n - 1$ ).  $n \geq 2$ , i.e., there is at least one buffer between the driver and the sink.



- Output resistance of  $K_D$ :  $R_D$
- Output resistance of  $B_i$  ( $1 \leq i \leq n - 1$ ):  $R_i$  (e.g.,  $R_1, R_2, \dots$ )
- Input capacitance of  $K_S$ :  $C_S$
- Input capacitance of  $B_i$  ( $1 \leq i \leq n - 1$ ):  $C_i$
- Delay of  $B_i$  ( $1 \leq i \leq n - 1$ ):  $D_i$
- Length of the  $i$ -th net ( $1 \leq i \leq n$ ):  $s_i$  (um)
- $\sum_{i=1}^n s_i = L$  (um)
- Wire unit resistance:  $r$  ( $\Omega/\text{um}$ )
- Wire unit capacitance:  $c$  ( $\text{fF}/\text{um}$ )

We assume that the net is optimized to minimize the delay from the driver to the sink.

(Hint: Derive  $s_1$  and  $s_2$  as functions of the above parameters when  $n = 2$ . You can somehow use the result for the following questions).

Suppose  $R_D = R_0$  and  $C_S = C_n$ . Then, the delay of segment  $s_k$  is

$$\tau_k = R_{k-1}(c \cdot s_k + C_k) + r \cdot C_k \cdot s_k + \frac{1}{2}rcs_k^2$$

Then, the total delay is

$$\begin{aligned} \tau &= \sum_{k=1}^n \tau_k + \sum_{k=1}^{n-1} D_k = c \sum_{k=1}^n R_{k-1}s_k + \sum_{k=1}^n R_{k-1}C_k + r \sum_{k=1}^n C_k s_k + \frac{1}{2}rc \sum_{k=1}^n s_k^2 + \sum_{k=1}^{n-1} D_k \\ \frac{\partial \tau}{\partial s_k} &= c(R_{k-1} - R_{n-1}) + r(C_k - C_n) + rc(s_k - s_n) = 0 \end{aligned}$$

$$\therefore s_k = s_n + \frac{R_{n-1} - R_{k-1}}{r} + \frac{C_n - C_k}{c}$$

From  $\sum_{k=1}^n s_k = L$ ,  $\sum_{k=1}^n s_k = n \cdot s_n + \frac{1}{r} \sum_{k=1}^n (R_{n-1} - R_{k-1}) + \frac{1}{c} \sum_{k=1}^n (C_n - C_k) = n \cdot s_n + \frac{n \cdot R_{n-1} - R_T}{r} + \frac{n \cdot C_n - C_T}{c} = L$  where  $R_T = \sum_{k=0}^{n-1} R_k$  and  $C_T = \sum_{k=1}^n C_k$ .

Thus, we obtain the following:

$$s_k = \frac{L}{n} + \frac{1}{r} \cdot \left( \frac{1}{n} \cdot R_T - R_{k-1} \right) + \frac{1}{c} \cdot \left( \frac{1}{n} \cdot C_T - C_k \right)$$

Answer the following questions for  $n = 10$  (i.e., we insert 9 buffers optimally):

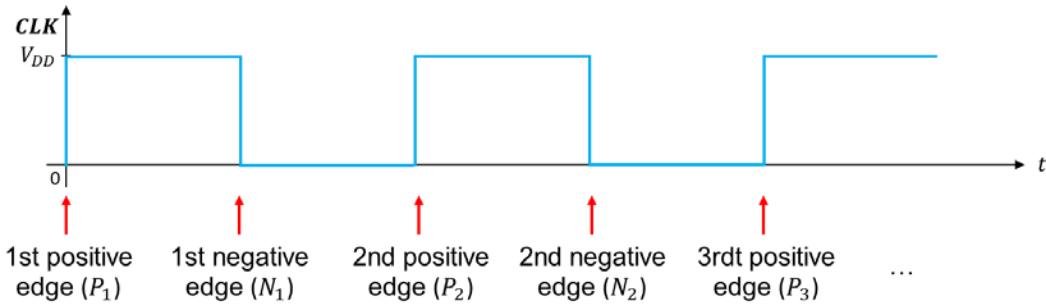
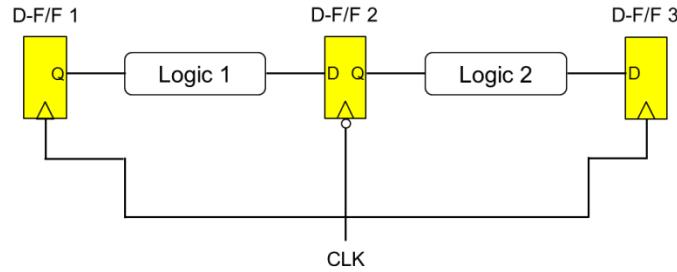
- If  $C_S$  increases, we should increase  $s_1$  to minimize the total delay. (**True/False**)
  - If  $C_S$  increases, the delay of  $s_{10}$  goes up, so we should increase  $s_1, \dots, s_9$ .
- If  $R_9$  increases, we should increase  $s_1$  to minimize the total delay. (**True/False**)
  - If  $R_9$  increases, the delay of  $s_9$  goes up, so we should increase  $s_1, \dots, s_8, s_{10}$ .
- If  $D_9$  increases, we should increase  $s_1$  to minimize the total delay. (**True/False**)
  - Since we always insert 9 buffers,  $D_9$  does not affect the total delay.
- If  $C_5$  increases, we should increase  $s_8$  to minimize the total delay. (**True/False**)
  - True for the same reason as the case of  $C_S \uparrow$ .
- If  $R_5$  increases, we should increase  $s_8$  to minimize the total delay. (**True/False**)
  - True for the same reason as the case of  $R_9 \uparrow$ .
- If  $D_5$  increases, we should increase  $s_8$  to minimize the total delay. (**True/False**)
  - False for the same reason as the case of  $D_9 \uparrow$ .
- If  $R_D$  increases, we should increase  $s_1$  to minimize the total delay. (**True/False**)
  - If  $R_D(R_0)$  increases, the delay of  $s_1$  goes up, so we should increase  $s_2, \dots, s_{10}$ .
- Suppose  $s_1 \approx 0$  because  $R_D \gg R_1, \dots, R_9$ . In this case, if  $r$  and  $c$  increase at the same time, we should increase  $s_1$  to minimize the total delay. (**True/False**)
  - If  $r \rightarrow \infty$  and  $c \rightarrow \infty$ , the impact of output resistance and input capacitance on the delay reduces. In this case, the impact of the wire delay portion ( $\frac{1}{2} r c l^2$ ) goes up, so we should evenly distribute the buffers to minimize the total delay. Since  $s_1$  was almost 0, we should increase  $s_1$ .

Answer the following questions assuming  $n$  is to be determined optimally (i.e., we find # buffers ( $n - 1$ ) and  $s_1 \sim s_n$  optimally) and  $L \gg 0$ , so  $n \gg 1$ :

- If  $R_D$  increases, we should increase  $n$  to minimize the total delay. (**True/False**)
  - In this case,  $s_1$  should be decreased to reduce the total delay, which increases  $s_1, \dots, s_n$ . Thus, we should insert more buffers in general.

- If  $C_2$  increases, we should increase  $n$  to minimize the total delay. (**True/False**)
  - In this case, we should decrease  $s_2$  and increase all the other  $s_k$ . If their lengths go up, we should insert more buffers.
- If  $C_S$  increases, we should increase  $n$  to minimize the total delay. (**True/False**)
  - We should insert more buffers for the same reason as the case  $C_S$  increases.
- If the delay of each buffer is increased, we should generally increase  $n$  to minimize the total delay. (**True/False**)
  - We should decrease the number of buffers because the buffer delay has negative impact on the total delay.
- If  $L$  increases, we should increase  $n$  to minimize the total delay. (**True/False**)
- If  $r$  increases, we should increase  $n$  to minimize the total delay. (**True/False**)
  - The total delay when there is no buffer is
 
$$\tau = R_D(c \cdot L + C_S) + r \cdot L \cdot C_S + \frac{1}{2}rcL^2.$$
  - In this formula,  $r$  can be treated as weighting factors for  $L \cdot C_S$  and  $\frac{1}{2}rcL^2$ . If  $r$  increases, inserting more buffers can reduce  $\frac{1}{2}rcL^2$ . For example, if a buffer is inserted, the sum of the values is  $\frac{1}{2}rc(0.25L^2 + 0.25L^2) = \frac{1}{4}rcL^2$ . Similarly, if three buffers are inserted, the sum of the values is  $\frac{1}{2}rc(L^2 / 16 + L^2 / 16 + L^2 / 16 + L^2 / 16) = \frac{1}{8}rcL^2$ . Although the buffer delays and buffer input capacitance values are delay overheads, if  $r$  increases significantly, inserting more buffers helps reduce the delay as shown above.
- If  $c$  increases, we should increase  $n$  to minimize the total delay. (**True/False**)
  - $r$  and  $c$  basically have similar impacts on the total delay, so if  $c$  goes up, we should insert more buffers.

## Problem #6 (Timing Analysis, 10 points)



- Setup time of a D-FF:  $T_s$
- Hold time of a D-FF:  $T_h$
- D-FF internal delay:  $T_{CQ}$
- Logic 1 delay:  $T_{L1}$
- Logic 2 delay:  $T_{L2}$
- Clock period:  $T_{CK}$  (duty cycle: 50%, i.e., the clock is high for  $T_{CK}/2$  and low for  $T_{CK}/2$ .)
- Delay from CLK to D-FF 1:  $D_1$
- Delay from CLK to D-FF 2:  $D_2$
- Delay from CLK to D-FF 3:  $D_3$
- D-FF 2 is a negative-edge FF (i.e., it captures the input signal at falling edges.)

The above figure shows three FFs connects in series. D-FF 2 is a negative-edge FF, whereas D-FF 1 and 3 are positive-edge FFs. The operation of the circuit is as follows. D-FF 1 captures its input signal at  $k$ -th positive clock edge  $P_k$ . Logic 1 performs computation for the output of D-FF 1. D-FF 2 captures its input signal at  $k$ -th negative clock edge  $N_k$ . Logic 2 performs computation for the output of D-FF2. D-FF 3 captures its input signal at  $(k + 1)$ -th positive clock edge  $P_{k+1}$ .

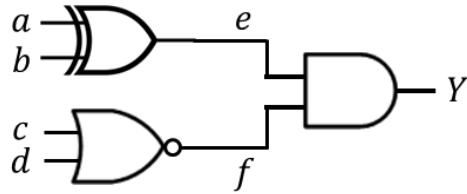
Derive two setup time inequalities (one for Logic 1 and the other for Logic 2).

$$1) \text{ For Logic 1, } D_1 + T_{CQ} + T_{L1} \leq D_2 + \frac{T_{CK}}{2} - T_s \Rightarrow T_{L1} \leq (D_2 - D_1) + \frac{T_{CK}}{2} - T_{CQ} - T_s$$

$$2) \text{ For Logic 2, } D_2 + \frac{T_{CK}}{2} + T_{CQ} + T_{L2} \leq D_3 + T_{CK} - T_s \Rightarrow T_{L2} \leq (D_3 - D_2) + \frac{T_{CK}}{2} - T_{CQ} - T_s$$

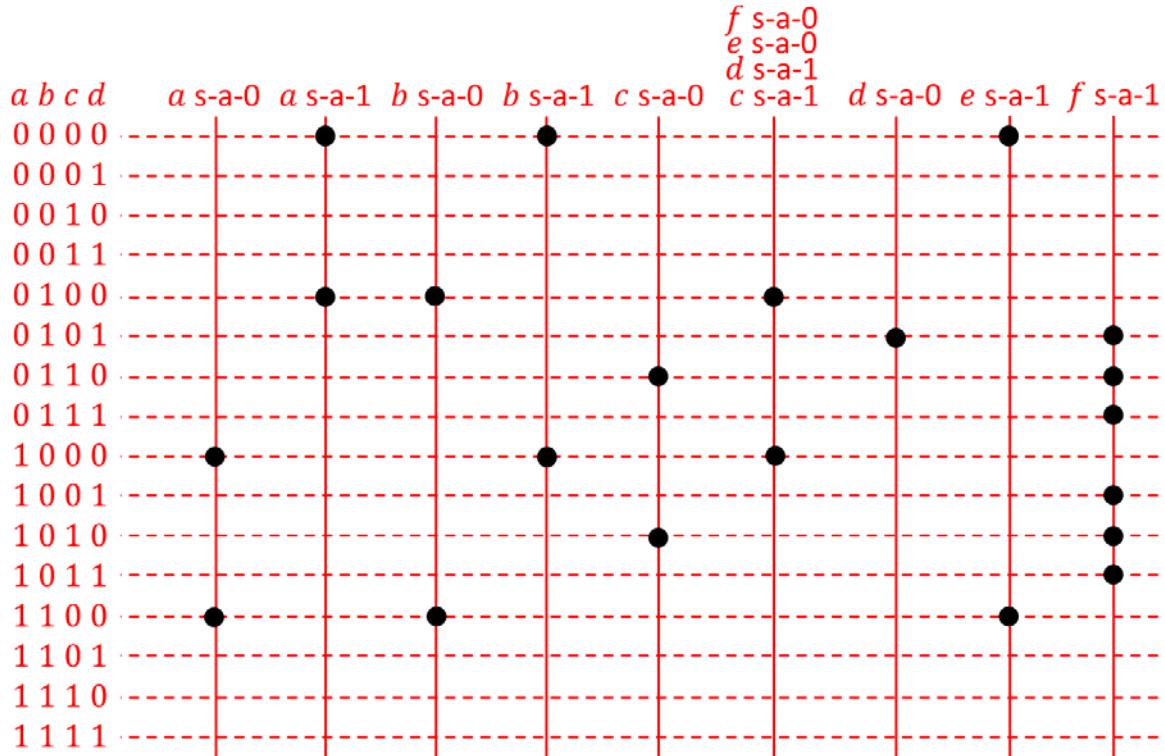
## Problem #7 (Testing, 10 points)

We want to detect stuck-at-0 and stuck-at-1 faults at all the primary inputs,  $a, b, c, d$ , and the two internal nodes,  $e, f$ . Computation of  $Y$  to detect a stuck-at-0/1 fault at an internal node can be done by setting the value of the node to constant 0 (for stuck-at-0 faults) or 1 (for stuck-at-1 faults). Find a minimal set of test vectors that can detect all the s-a-0 and s-a-1 faults at  $a, b, c, d, e$ , and  $f$  for the following logic (Hint: all the minimal sets have five test vectors).



$$Y = (a \oplus b) \cdot \overline{c + d}$$

- s-a-0 at  $a$ :  $Y_f = b \cdot \overline{c + d}$ .  $Y \oplus Y_f = \{(a \oplus b) \cdot \overline{c + d}\} \oplus \{b \cdot \overline{c + d}\} = 1 \Rightarrow (a b c d) = (1 0 0 0) \text{ or } (1 1 0 0)$
- s-a-1 at  $a$ :  $Y_f = \bar{b} \cdot \overline{c + d}$ .  $Y \oplus Y_f = \{(a \oplus b) \cdot \overline{c + d}\} \oplus \{\bar{b} \cdot \overline{c + d}\} = 1 \Rightarrow (a b c d) = (0 0 0 0) \text{ or } (0 1 0 0)$
- s-a-0 at  $b$ :  $(a b c d) = (0 1 0 0) \text{ or } (1 1 0 0)$
- s-a-1 at  $b$ :  $(a b c d) = (0 0 0 0) \text{ or } (1 0 0 0)$
- s-a-0 at  $c$ :  $Y_f = (a \oplus b) \cdot \bar{d}$ .  $Y \oplus Y_f = \{(a \oplus b) \cdot \overline{c + d}\} \oplus \{(a \oplus b) \cdot \bar{d}\} = 1 \Rightarrow (a b c d) = (0 1 1 0) \text{ or } (1 0 1 0)$
- s-a-1 at  $c$ :  $Y_f = 0$ .  $Y \oplus Y_f = \{(a \oplus b) \cdot \overline{c + d}\} \oplus 0 = 1 \Rightarrow (a b c d) = (0 1 0 0) \text{ or } (1 0 0 0)$
- s-a-0 at  $d$ :  $(a b c d) = (0 1 0 1) \text{ or } (1 0 0 1)$
- s-a-1 at  $d$ :  $(a b c d) = (0 1 0 0) \text{ or } (1 0 0 0)$
- s-a-0 at  $e$ :  $Y_f = 0$ .  $Y \oplus Y_f = \{(a \oplus b) \cdot \overline{c + d}\} \oplus 0 = 1 \Rightarrow (a b c d) = (0 1 0 0) \text{ or } (1 0 0 0)$
- s-a-1 at  $e$ :  $Y_f = \overline{c + d}$ .  $Y \oplus Y_f = \{(a \oplus b) \cdot \overline{c + d}\} \oplus \{\overline{c + d}\} = 1 \Rightarrow (a b c d) = (0 0 0 0) \text{ or } (1 1 0 0)$
- s-a-0 at  $f$ :  $Y_f = 0$ .  $Y \oplus Y_f = \{(a \oplus b) \cdot \overline{c + d}\} \oplus 0 = 1 \Rightarrow (a b c d) = (0 1 0 0) \text{ or } (1 0 0 0)$
- s-a-1 at  $f$ :  $Y_f = (a \oplus b)$ .  $Y \oplus Y_f = \{(a \oplus b) \cdot \overline{c + d}\} \oplus \{(a \oplus b)\} = 1 \Rightarrow (a b c d) = (0 1 0 1) \text{ or } (0 1 1 0) \text{ or } (0 1 1 1) \text{ or } (1 0 0 1) \text{ or } (1 0 1 0) \text{ or } (1 0 1 1)$



- 1) Need (0 1 0 1) to cover the 7<sup>th</sup> column. It also covers the 9<sup>th</sup> column.
- 2) Covering the 1<sup>st</sup> column requires either (1 0 0 0) or (1 1 0 0).
- 3) Covering the 2<sup>nd</sup> column requires either (0 0 0 0) or (0 1 0 0).
- 4) For (1 0 0 0) and (0 0 0 0), we should cover the 3<sup>rd</sup> and the 5<sup>th</sup> columns.
- 5) If we proceed this way, we get the following test vectors.

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
Set 1	■					■										
Set 2		■					■									
Set 3	■					■								■		
Set 4	■					■					■					
Set 5				■	■		■			■				■		
Set 6					■	■				■		■				
Set 7	■					■						■				
Set 8	■						■					■				

## Problem #8 (Testing, 10 points)

A combinational logic  $G$  is given. It has  $n$  inputs and one output. The inputs are  $x_1, x_2, \dots, x_n$  ( $n \geq 2$ ) and the output is  $y$ , i.e.,  $y = G(x_1, \dots, x_n)$ . To find an input vector that can detect a stuck-at- $v$  fault ( $v = 0$  or  $1$ ) at  $x_i$  ( $1 \leq i \leq n$ ), we solve  $G(x_1, \dots, x_n) \oplus G_f(x_1, \dots, x_i = v, \dots, x_n) = 1$ . Let  $S_i$  be the set of all input vectors that can detect a stuck-at- $v_i$  ( $v_i = 0$  or  $1$ ) fault at  $x_i$  and  $S_k$  be the set of all input vectors that can detect a stuck-at- $v_k$  ( $v_k = 0$  or  $1$ ) fault at  $x_k$  ( $i \neq k$ ).

If we assume that two stuck-at- $v$  faults occur at the same time, we can find an input vector that can detect the faults. For example, we solve  $G(x_1, \dots, x_n) \oplus G_f(x_1, \dots, x_i = v_i, \dots, x_k = v_k, \dots, x_n) = 1$  to find an input vector that can detect a stuck-at- $v_i$  fault at  $x_i$  and a stuck-at- $v_k$  fault at  $x_k$  occurring at the same time ( $i \neq k$ ). Let  $S_{i,k}$  be the set of all input vectors that can detect a stuck-at- $v_i$  ( $v_i = 0$  or  $1$ ) fault at  $x_i$  and a stuck-at- $v_k$  ( $v_k = 0$  or  $1$ ) fault at  $x_k$  ( $i \neq k$ ).

Prove or disprove the following statement:

$$S_{i,k} = S_i \cap S_k$$

(If you want to disprove it, you can just show a counterexample.)

Counterexample: A two-input AND gate (inputs:  $a, b$ , output:  $Z$ ).

$$Z = a \cdot b$$

Let a s-a-0 fault at input  $a$  be  $f_a$  and a s-a-1 fault at input  $b$  be  $f_b$ . For  $f_a$ ,  $Z_f = 0$ , so  $S_a = \{(a, b) = (1 1)\}$ . For  $f_b$ ,  $Z_f = a$ , so  $S_b = \{(a, b) = (1 0)\}$ .  $S_a \cap S_b = \emptyset$ .

When the two faults occur at the same time,  $Z_f = 0$ . In this case,  $S_{a,b} = \{(a, b) = (1 1)\}$ . Thus,  $S_{a,b} \neq S_a \cap S_b$ .

**EE434**  
**ASIC and Digital Systems**

**Midterm Exam 1**

**Feb. 22, 2017. (4:10pm – 5pm)**

**Instructor: Dae Hyun Kim ([daehyun@eeecs.wsu.edu](mailto:daehyun@eeecs.wsu.edu))**

**Name:**

**WSU ID:**

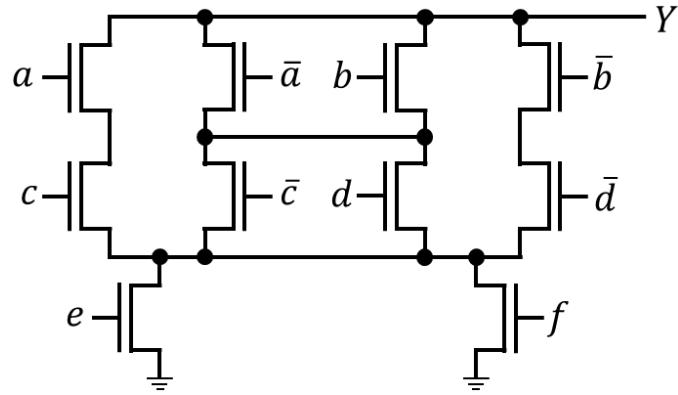
Problem	Points	
1	10	
2	10	
3	10	
4	20	
5	10	
Total	60	

\* Allowed: Textbooks, cheat sheets, class notes, notebooks, calculators, watches

\* Not allowed: Electronic devices (smart phones, tablet PCs, laptops, etc.) except calculators and watches

**Problem #1 (Static CMOS gates, 10 points).**

The pFET network of the following nFET network is designed by the dual network of the nFET network. Represent  $Y$  as a Boolean function of  $a, b, c, d, e$  and  $f$ . (Try to simplify the Boolean function).



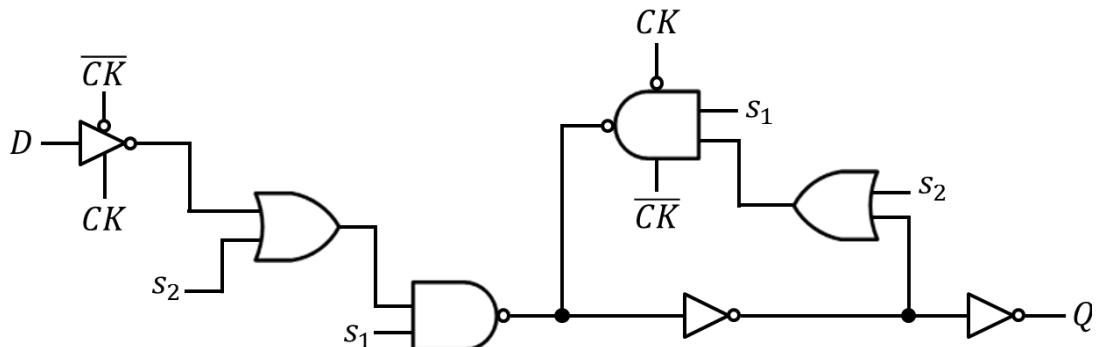
$$\begin{aligned}
 Y &= \overline{ace + acf + \bar{a}\bar{c}e + \bar{a}\bar{c}f + \bar{a}de + \bar{a}df + b\bar{c}e + b\bar{c}f + bde + bd\bar{f} + \bar{b}\bar{d}e + \bar{b}\bar{d}f} \\
 &= \overline{ac(e + f) + \bar{a}\bar{c}(e + f) + \bar{a}d(e + f) + b\bar{c}(e + f) + bd(e + f) + \bar{b}\bar{d}(e + f)} \\
 &= \overline{(e + f) \cdot \{ac + \bar{a}\bar{c} + \bar{a}d + b\bar{c} + bd + \bar{b}\bar{d}\}} \\
 &= \overline{(e + f) \cdot \{\overline{a \oplus c} + \bar{a}d + b\bar{c} + \overline{b \oplus d}\}}
 \end{aligned}$$

## Problem #2 (Analysis of CMOS gates, 10 points).

The following circuit is a sequential logic. Describe the function of the circuit in as much detail as possible ( $D$ : data input,  $CK$ : clock,  $s_1, s_2$ : control inputs).

Note:

- Logic
  - F/F vs. Latch
  - Positive-edge triggered vs. Negative-edge triggered (for F/F)
  - Active-high vs. Active-low (for Latch)
- Control inputs
  - Asynchronous vs. Synchronous
  - Active-high vs. Active-low
  - Set, Reset, Enable, Select, Additional input
  - Dominance (e.g., Set dominates Reset, Enable dominates Set, etc.)

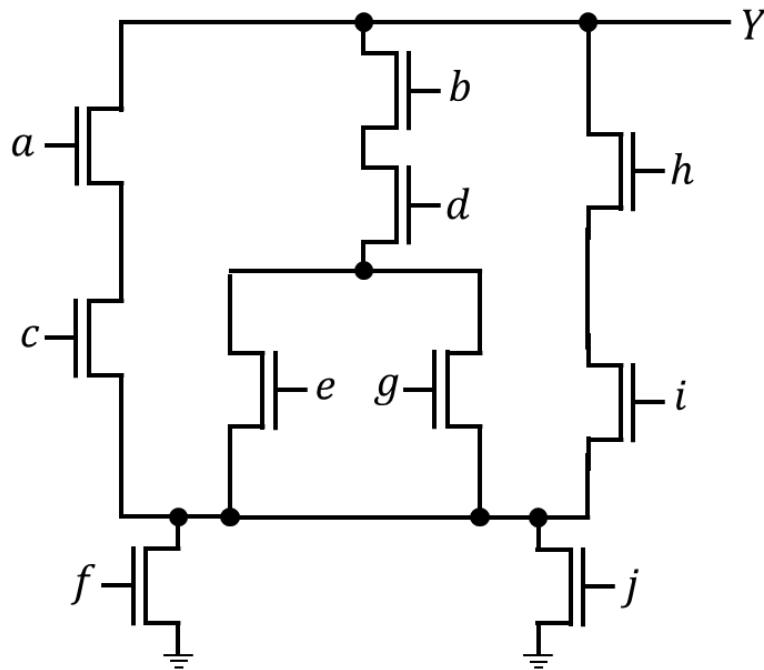


- If  $s_1 = 0, Q = 1 \rightarrow s_1$  is an asynchronous active-low set.
- If  $s_1 = 1$  and  $s_2 = 1, Q = 0 \rightarrow s_2$  is an asynchronous active-high reset.
  - Set dominates Reset.
- When  $s_1 = 1$  and  $s_2 = 0$ 
  - If  $CK = 0, Q = \text{hold}$ .
  - If  $CK = 1, Q = D$ .

Thus, this is an active-high Latch with an asynchronous active-low set and an asynchronous active-high reset (Set dominates Reset).

### Problem #3 (Transistor Sizing, 10 points).

Size the transistors in the following pull-down network.  $R_n$  is the resistance of a 1X NMOS transistor. Ignore parasitic capacitances. Target time constant:  $\tau_{target} = R_n \cdot C_L$ . Try to minimize the total area.



The longest path is b-d-e-f. Each of them is upsized to 4X. Then, we get  $a=c=h=i=8/3$ .

a: 8/3X

b: 4X

c: 8/3X

d: 4X

e: 4X

f: 4X

g: 4X

h: 8/3X

i: 8/3X

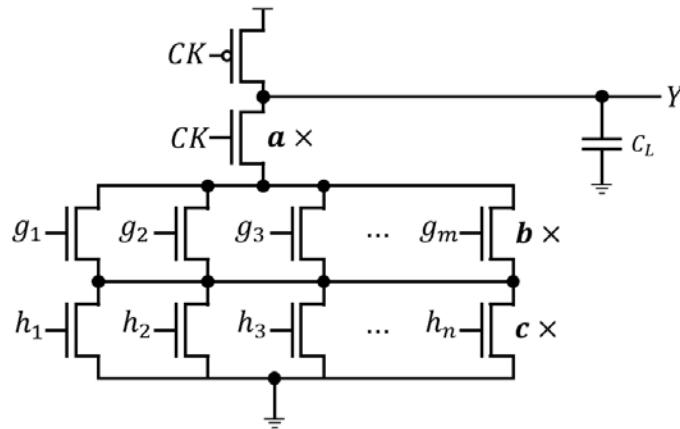
j: 4X

Total width: (34+2/3)X

(If we upsize a-c-f first,  $a=c=h=i=f=j=3X$ . Then,  $b=d=e=g=4.5X$ . Total width = 36X, which is worse than the above one.)

### Problem #4 (Transistor Sizing, 20 points).

We want to design the following logic:  $Y = \overline{(g_1 + g_2 + \dots + g_m)} \cdot (h_1 + h_2 + \dots + h_n)$  where  $g_1 \sim g_m$  and  $h_1 \sim h_n$  are the inputs of the gate. However, the static CMOS gate design methodology is not suitable for the design of the gate. Thus, we are going to design it using the dynamic CMOS design methodology. The following shows a schematic of the gate.



$R_n$  is the resistance of a 1X nFET. Target time constant:  $\tau_{target} = R_n \cdot C_L$ . The nFET connected to  $CK$  is upsized to  $a \times$ , all the nFETs connected to  $g_1 \sim g_m$  are upsized to  $b \times$ , and all the nFETs connected to  $h_1 \sim h_n$  are upsized to  $c \times$  to satisfy the timing constraint ( $a, b, c$  are real numbers). Ignore all the parasitic capacitance. We minimize the total width

$$Width = a + m \cdot b + n \cdot c.$$

Problem 4-1 (8 points): Find  $a, b$ , and  $c$  minimizing the total width (i.e., represent each of  $a, b$ , and  $c$  as a function of  $n$  and  $m$ ).

Timing constraint:  $\frac{R_n}{a} + \frac{R_n}{b} + \frac{R_n}{c} = R_n \rightarrow \frac{1}{a} + \frac{1}{b} + \frac{1}{c} = 1$ . Set  $a = \frac{1}{x}$ ,  $b = \frac{1}{y}$ ,  $c = \frac{1}{z}$ . Then we minimize  $F = \frac{1}{x} + \frac{m}{y} + \frac{n}{z}$  with constraint  $x + y + z = 1$ .

$$\frac{\partial F}{\partial x} = -\frac{1}{x^2} + \frac{n}{z^2} = 0 \rightarrow x = \frac{z}{\sqrt{n}}$$

$$\frac{\partial F}{\partial y} = -\frac{m}{y^2} + \frac{n}{z^2} = 0 \rightarrow y = \frac{\sqrt{m}z}{\sqrt{n}}$$

$$z \left( \frac{1}{\sqrt{n}} + \frac{\sqrt{m}}{\sqrt{n}} + 1 \right) = 1 \rightarrow z = \frac{\sqrt{n}}{1 + \sqrt{m} + \sqrt{n}}$$

$$x = \frac{1}{1 + \sqrt{m} + \sqrt{n}}$$

$$y = \frac{\sqrt{m}}{1 + \sqrt{m} + \sqrt{n}}$$

Thus, we obtain the following:

$$a = (1 + \sqrt{m} + \sqrt{n})X$$

$$b = \frac{1+\sqrt{m}+\sqrt{n}}{\sqrt{m}} = \left(1 + \frac{1+\sqrt{n}}{\sqrt{m}}\right)X$$

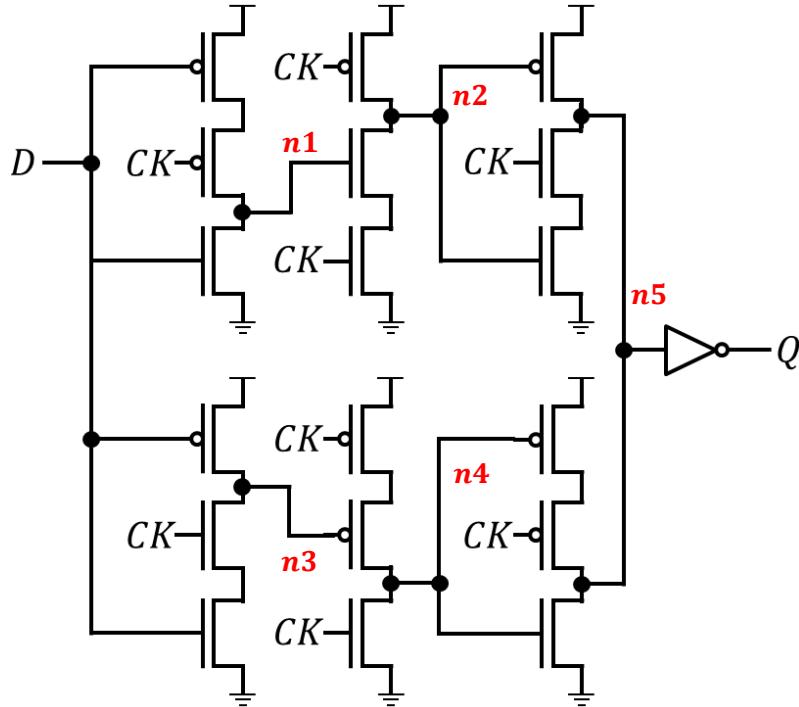
$$c = \frac{1+\sqrt{m}+\sqrt{n}}{\sqrt{n}} = \left(1 + \frac{1+\sqrt{m}}{\sqrt{n}}\right)X$$

Problem 4-2 (12 points, True/False. Use your intuition or the formulas you derived in the above problem).)

- If we increase  $n$ ,  $a$  increases. (True / False)
  - If  $n$  goes up, increasing  $a$  can significantly reduce the size overhead.
- If we increase  $m$ ,  $a$  increases. (True / False)
  - $n$  and  $m$  are interchangeable, so if  $m$  goes up, increasing  $a$  can significantly reduce the size overhead.
- If we increase  $n$ ,  $b$  increases. (True / False)
  - In this case ( $n$  increases),  $a$  and  $b$  can be treated the same way. Thus,  $b$  should go up to minimize the total area.
- If we increase  $m$ ,  $b$  increases. (True / False)
  - Increasing  $m$  increases  $a$  and  $n$  or decrease  $b$ .
- If we increase  $n$ ,  $c$  increases. (True / False)
- If we increase  $m$ ,  $c$  increases. (True / False)

## Problem #5 (Logic Analysis, 10 points).

What does the following circuit do? Describe the function of the circuit in as much detail as possible (CK is a clock signal, D is an input, Q is an output). Note: if a node is floating, it holds its previous value.



(M. Afghahi, JSSC'91)

- 1) CK=0, D=0:  $n_3=1$ , so  $n_4$  is floating. If  $n_4$  was 0,  $Q=0$  (hold). If  $n_4$  was 1,  $Q=1$  (hold). In this case,  $n_2=1$ , but CK=0, so  $n_5$  is driven by  $n_4$ .  $Q=\text{hold}$ .
- 2) CK=0, D=0→1:  $n_3$  is floating. If  $n_3$  was 0,  $n_4$  was 1, so  $Q=1$  (hold). If  $n_3$  was 1,  $n_4$  was floating, so  $Q$  is hold.  $n_1$  is 0, but  $n_2$  is 1 because CK=0, so  $n_5$  is driven by  $n_4$ .  $Q=\text{hold}$ .
- 3) CK=0, D=1 or D=1→0:  $Q=\text{hold}$ .
- 4) CK=0→1:  $n_4$  is 0, so  $n_5$  is not driven by  $n_4$  anymore. If D was 1 right before the clock rising edge,  $n_1=0$ .  $n_2=\text{floating}$  (but it was 1 because CK was 0). Thus,  $n_5$  is 0, so  $Q=1$ . If D was 0 right before the clock rising edge,  $n_1$  was 1. After the rising edge,  $n_2 = 0$ . Thus,  $n_5=1$ , so  $Q=0$ . In other words, it captures the D value at the clock rising edge (positive-edge D-FF).
- 5) CK=1→0:  $n_2=1$ , so  $n_5$  is not driven by the upper logic.  $n_4$  was 0 right before the clock falling edge. If D was 0 right before the clock falling edge,  $n_3=1$ , so  $n_4$  is floating

( $n_4$  holds the previous value 0), so  $n_5=1$ , so  $Q=0$ . If D was 1 right before the clock falling edge,  $n_3$  was 0. After the clock falling edge,  $n_3$  is floating (holds the previous value 0). Now,  $CK=0$ , so  $n_4=1$ , so  $n_5=0$ , so  $Q=1$ . In other words, it captures the D value at the clock falling edge (negative-edge D-FF).

This is a dual-edge D-FF.

**EE434**  
**ASIC and Digital Systems**

**Midterm Exam 2**

**Mar. 31, 2017. (4:10pm – 5pm)**

**Instructor: Dae Hyun Kim ([daehyun@eeecs.wsu.edu](mailto:daehyun@eeecs.wsu.edu))**

**Name:**

**WSU ID:**

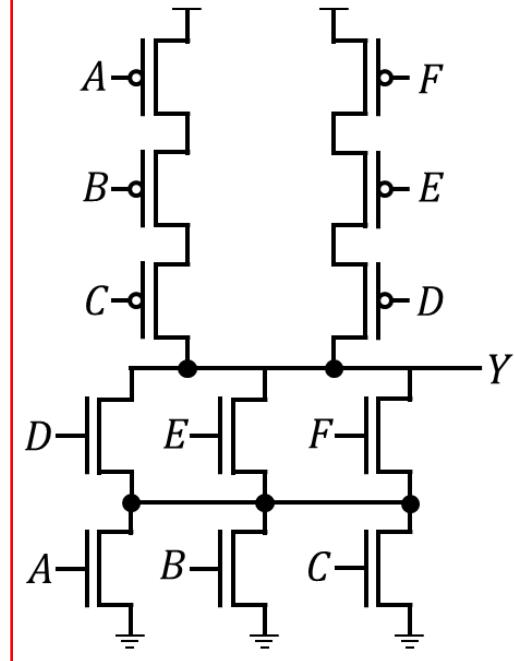
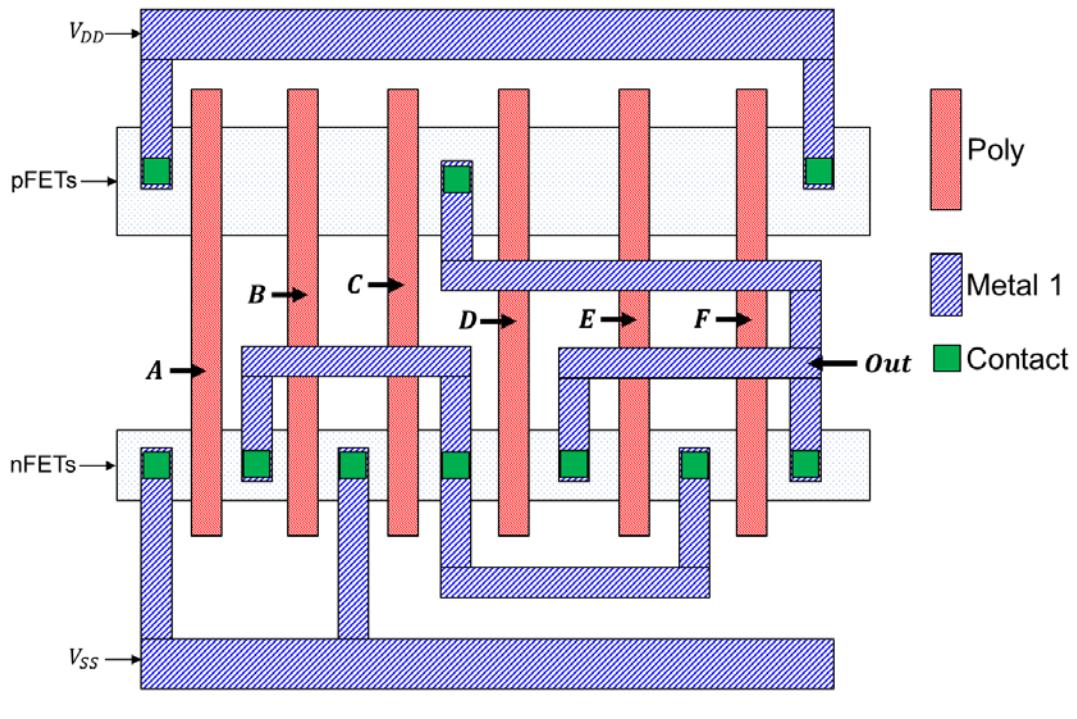
Problem	Points	
1	10	
2	10	
3	10	
4	10	
5	10	
Total	50	

\* Allowed: Textbooks, cheat sheets, class notes, notebooks, calculators, watches

\* Not allowed: Electronic devices (smart phones, tablet PCs, laptops, etc.) except calculators and watches

## Problem #1 (Layout, 10 points).

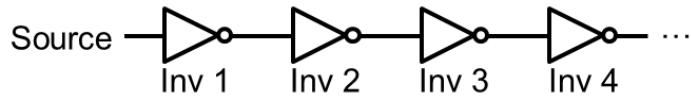
Represent *Out* as a Boolean function of  $A, B, C, D, E, F$ .



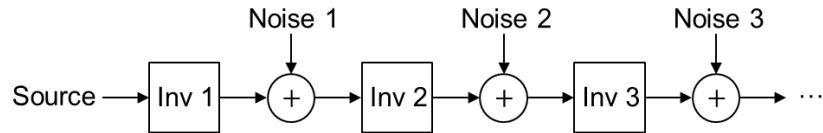
$$Y = \overline{(A + B + C)} \cdot (D + E + F)$$

## Problem #2 (DC Characteristics, 10 points).

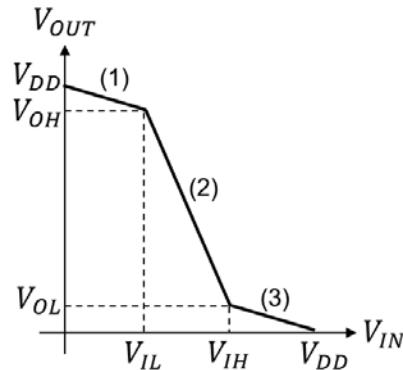
An infinite chain of inverters is defined as follows:



All the inverters are identical, i.e., have the same characteristics. The above chain is modeled as a block diagram as follows:



where *Noise k* is the  $k$ -th noise and *Source* is a signal generator and  $V_{Source} = V_{DD} \cdot u(t)$  (i.e., 0 if  $t < 0$  and  $V_{DD}$  if  $t \geq 0$ ).  $V_{DD} = 1V$ . The DC characteristic of an inverter is approximated using three segments as follows (If  $V_{in} \geq V_{DD}$ ,  $V_{out} = 0$ . If  $V_{in} \leq 0$ ,  $V_{out} = V_{DD}$ ):



- 1)  $V_{out} = \frac{V_{OH} - V_{DD}}{V_{IL}} \cdot V_{in} + V_{DD}$
- 2)  $V_{out} = \frac{V_{OL} - V_{OH}}{V_{IH} - V_{IL}} \cdot (V_{in} - V_{IH}) + V_{OL}$
- 3)  $V_{out} = -\frac{V_{OL}}{V_{DD} - V_{IH}} \cdot (V_{in} - V_{DD})$

Each noise source is an independent voltage signal and its range is as follows ( $V_N > 0$ ):

- $|V_{noise}| \leq V_N$

Compute the max. value of  $V_N$  that does not cause signal inversion for the following two cases (Note: Signal inversion occurs if a signal reaches 0.5V when it should be 0 or 1):

- Case 1)  $V_{OL} = 0V, V_{OH} = 1V, V_{IL} = 0.4V, V_{IH} = 0.6V$
- Case 2)  $V_{OL} = 0V, V_{OH} = 1V, V_{IL} = 0.4V, V_{IH} = 0.8V$

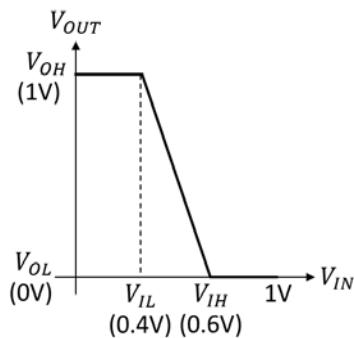
Case 1) In this case,  $y = -5x + 3$  in Region 2. The high noise margin is 0.4V, so is the low noise margin. Suppose  $V_N$  is  $(0.4 + \delta)V$  where  $\delta > 0$ . Let's take a look at the worst cases.

- Input of inverter m: 1V
- Output of inverter m: 0V
- Input of inverter (m+1):  $(0.4 + \delta)V \text{ // } a(n) = 0.4 + b(n)$
- Output of inverter (m+1):  $(1.0 - 5\delta)V \text{ // } -5a(n) + 3 = 1 - 5b(n)$
- Input of inverter (m+2):  $(0.6 - 6\delta)V \text{ // } -5a(n) + 2.6 - \delta = 0.6 - 5b(n) - \delta$
- Output of inverter (m+2):  $(30\delta)V \text{ // } 25a(n) - 10 + 5\delta = 25b(n) + 5\delta$
- Input of inverter (m+3):  $(0.4 + 31\delta)V \text{ // } a(n+1) = 25a(n) - 9.6 + 6\delta = 0.4 + 25b(n) + 6\delta \rightarrow b(n+1) = 25b(n) + 6\delta, a(n+1) = 0.4 + b(n+1)$
- ...

$a(n+1) = 0.4 + b(n+1)$  and  $b(n+1) = 25b(n) + 6\delta$ . Since  $b(n)$  goes to infinity as  $n$  increases, so signal inversion happens.

If  $V_N$  is 0.4V

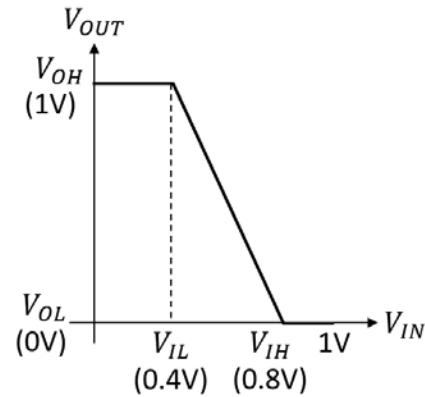
- Input of inverter m: 1V
- Output of inverter m: 0V
- Input of inverter (m+1): **0.4V**
- Output of inverter (m+1): 1V
- Input of inverter (m+2): 0.6V
- Output of inverter (m+2): 0V
- Input of inverter (m+3): **0.4V**
- ...



Thus, if  $\delta > 0$ , signal inversion will eventually occur no matter how small  $\delta$  is. Thus, the maximum  $V_N$  is **0.4V** in this case.

Case 2) In this case,  $y = -2.5x + 2$  in Region 2. When the output is 0.4V, the input voltage is 0.64V. Suppose  $V_N$  is  $X(V)$  where  $X$  is between 0.2V and 0.4V. Let's take a look at the worst cases.

- Input of inverter m: 1V
- Output of inverter m: 0V
- Input of inverter (m+1):  $X(V)$
- Output of inverter (m+1): 1V
- Input of inverter (m+2):  $(1 - X)V$
- Output of inverter (m+2):  $(-0.5 + 2.5X)V$
- Input of inverter (m+3):  $(-0.5 + 3.5X)V$



If  $(-0.5 + 3.5X)V$  is greater than 0.4V, the output voltage is less than 1V, then a positive feedback loop is formed, so we solve the following inequality:

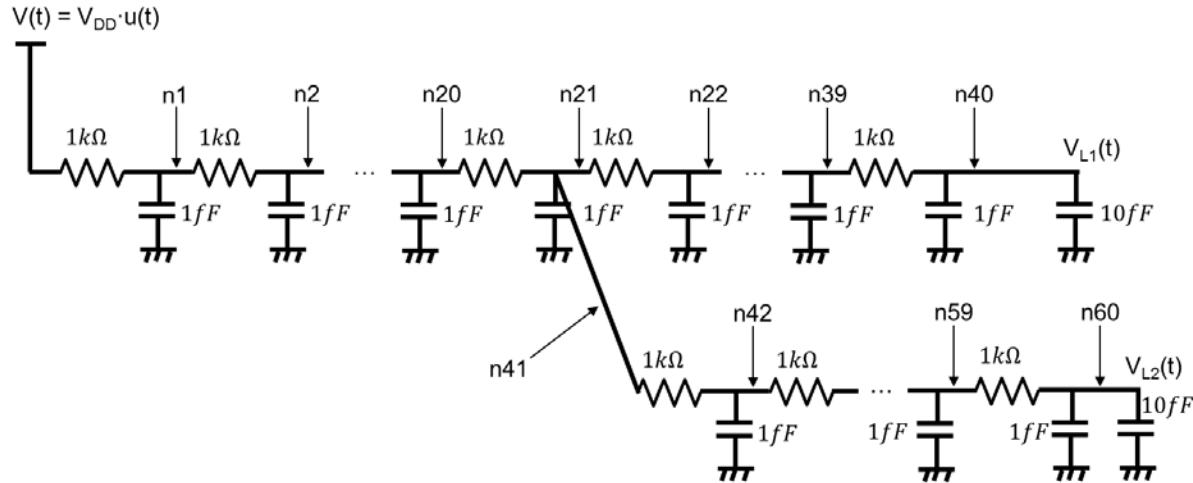
$$(-0.5 + 3.5X)V \leq 0.4V$$

Thus,  $V_N \approx 0.257V$ .

### Problem #3 (Elmore Delay, 10 points).

The RC tree shown below has two delay constraints as follows:

- The delay from the driver to  $V_{L1}$  should be less than or equal to 900ps.
- The delay from the driver to  $V_{L2}$  should be less than or equal to 6823ps.

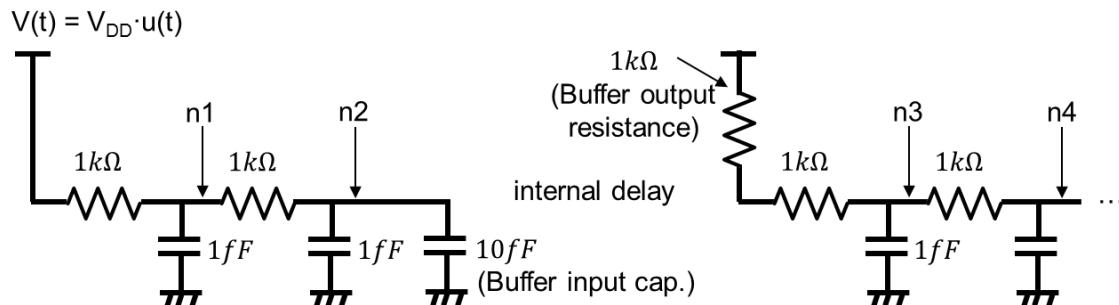


Currently, the delay at  $V_{L1}$  is  $1k * 11f + 1k * 12f + \dots + 1k * 29f + 1k * 59f + 1k * 60f + \dots + 1k * 79f = 1829ps$ , so is the delay at  $V_{L2}$ .

You are supposed to insert only one buffer in the RC tree to satisfy the delay constraints at both  $V_{L1}$  and  $V_{L2}$ . You can insert the buffer only into one of the designated nodes ( $n1 \sim n60$ ). The buffer has the following characteristics:

- Input capacitance:  $10fF$
- Internal delay:  $20ps$
- Output resistance:  $1k\Omega$

When you insert a buffer into a node, the RC tree before and after the buffer are separated as follows (assuming the buffer is inserted into n2 in the figure above):



In this case (inserting a buffer into n2), the delay at  $V_{L1}$  is  $[1k * 11f + 1k * 12f + \dots + 1k * 29f + 1k * 59f + 1k * 60f + \dots + 1k * 77f + 1k * 77f] + [20ps] + [1k * 11f + 1k * 12f] = (1749 + 20 + 23)ps = 1792ps$ , so is the delay at  $V_{L2}$ . As you see, the delay is reduced from 1829ps to 1792ps.

Insert a buffer into one of the nodes ( $n1 \sim n60$ ) so that it satisfies the delay constraints at both  $V_{L1}$  and  $V_{L2}$ . Then, compute the Elmore delay at  $V_{L1}$ .

(Help: The sum of  $a, a + 1, a + 2, \dots, n$  is  $\frac{(n+a)(n-a+1)}{2}$ . For example,  $5 + 6 + \dots + 10 = \frac{(10+5)(10-5+1)}{2} = 45$ .)

If I insert a buffer into node n21, the delay at  $V_{L1}$  is

$$\begin{aligned} & (1k * 11f + 1k * 12f + \dots + 1k * 29f + 1k * 29f) + (20ps) \\ & + (1k * 40f + 1k * 41f + \dots + 1k * 60f) \\ & = (380ps + 29ps) + (20ps) + (1050ps) = 1479ps \end{aligned}$$

If I insert a buffer into node n41, the delay at  $V_{L1}$  is

$$\begin{aligned} & (1k * 11f + 1k * 12f + \dots + 1k * 29f) + (1k * 40f + 1k * 41f + \dots + 1k * 60f) \\ & = (380ps) + (1050ps) = 1430ps \end{aligned}$$

If I insert a buffer into node n20, the delay at  $V_{L1}$  is

$$\begin{aligned} & (1k * 11f + 1k * 12f + \dots + 1k * 29f) + (1k * 59f) + (1k * 59f) + (20ps) \\ & + (1k * 11f + 1k * 12f + \dots + 1k * 30f) \\ & = (380ps) + (59ps) + (59ps) + (20ps) + (410ps) = 928ps \end{aligned}$$

If I insert a buffer into node n19, the delay at  $V_{L1}$  is

$$\begin{aligned} & (1k * 11f + 1k * 12f + \dots + 1k * 29f) + (1k * 59f) + (1k * 60f) + (1k * 60f) + (20ps) \\ & + (1k * 11f + 1k * 12f + \dots + 1k * 29f) \\ & = (380ps) + (59ps) + (60ps) + (20ps) + (380ps) = 899ps \end{aligned}$$

Node: n19

Elmore delay at  $V_{L1}$ : 899ps

### Problem #4 (Pseudo-nMOS, 10 points).

Draw a pseudo-nMOS schematic for  $Y = \overline{A \cdot B \cdot C} + (D + E) \cdot \overline{F}$  and properly size the nFETs to achieve the following output level for logic output 0:

- $V_{out} \leq 0.1V_{DD}$

Use the following parameters:

- Resistance of a  $1 \times$  nFET = Resistance of a  $2 \times$  pFET
- The size of the pFET:  $4 \times$
- Do not use the transistor-mode-based computation for sizing. You can just use the resistance values for sizing.
- Do not oversize the nFETs.

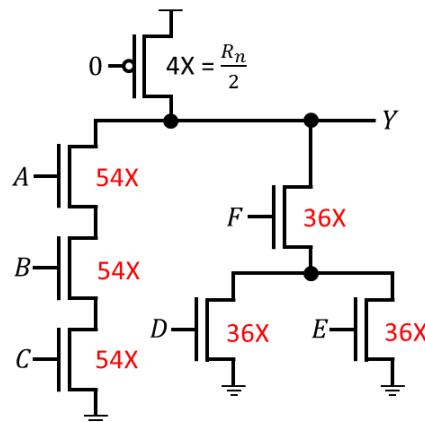
The resistance of the pFET is  $R_n/2$  where the resistance of a  $1X$  nFET is  $R_n$ . Suppose the resistance of an nFET path is  $R_k$ . Then,  $\frac{R_k}{(\frac{R_n}{2} + R_k)} \leq 0.1$ , so  $R_k \leq \frac{1}{18}R_n$ .

For A=B=C=1:

$$3 \cdot R = \frac{1}{18}R_n \Rightarrow R = \frac{1}{54}R_n \Rightarrow A = B = C = 54 \times$$

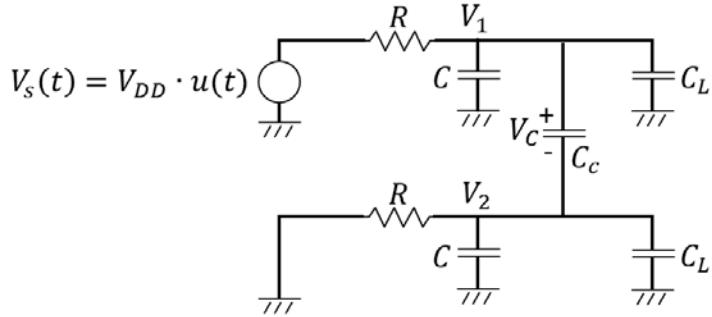
For F=1 and (D or E=1):

$$2 \cdot R = \frac{1}{18}R_n \Rightarrow R = \frac{1}{36}R_n \Rightarrow D = E = F = 36 \times$$



## Problem #5 (Capacitive Coupling, 10 points).

The following figure models the coupling effect between two adjacent wires.



$V_1(t)$  is an aggressor and  $V_2(t)$  is a victim.  $V_2(t)$  is as follows:

$$V_2(t) = \frac{V_{DD}}{2} \left[ e^{-\frac{t}{\tau_1}} - e^{-\frac{t}{\tau_2}} \right] u(t)$$

where

- $\tau_1 = R(C + C_L + 2C_c)$
- $\tau_2 = R(C + C_L)$

In this case, the max. value of  $V_2(t)$  is found by differentiating  $V_2(t)$  with respect to  $t$ . The max. value occurs when  $t$  is

$$t_{max} = \frac{R(C + C_L)(C + C_L + 2C_c)}{2} \cdot \ln \frac{C + C_L + 2C_c}{C + C_L}$$

and the max. value of  $V_2(t)$  is

$$V_{2,max} = \frac{V_{DD}}{2} \cdot \left( 1 - \frac{\tau_2}{\tau_1} \right) \cdot \left( \frac{\tau_2}{\tau_1} \right)^{\frac{\tau_2}{2RC_c}}$$

Answer the following questions (Hint: Use the above formula or your intuition to solve this problem):

- If  $C_c$  increases,  $V_{2,max}$  increases. (**True/False**)
- If  $C$  increases,  $V_{2,max}$  increases. (**True/False**)
- If  $C_L$  increases,  $V_{2,max}$  increases. (**True/False**)
- If  $R$  increases,  $V_{2,max}$  increases. (**True/False**)
- The max. value of  $V_2(t)$  can be greater than  $V_{DD}/2$ . (**True/False**)

(Hint:  $\lim_{x \rightarrow \infty} \left(\frac{1}{x}\right)^{\frac{1}{x}} = 1$ .  $\lim_{x \rightarrow \infty} \left(\frac{x}{x+c}\right)^{\frac{x}{c}} = e$ .  $\left(\frac{x}{x+c}\right)^{\frac{x}{c}} > 1$ . (when  $c > 0$ ))

$$V_{2,max} = \frac{V_{DD}}{2} \cdot \left(1 - \frac{C + C_L}{C + C_L + 2C_C}\right) \cdot \left(\frac{C + C_L}{C + C_L + 2C_C}\right)^{\frac{C+C_L}{2C_C}}$$

$$\left(1 - \frac{C + C_L}{C + C_L + 2C_C}\right) = a, \frac{C + C_L}{C + C_L + 2C_C} = b, \text{ and } \frac{C + C_L}{2C_C} = d.$$

- If  $C_C$  increases,  $a$  approaches 1 and  $b$  and  $d$  approach 0, so  $b^d$  approaches 1. Thus,  $V_{2,max}$  increases (approaches  $\frac{V_{DD}}{2}$ ).
- If  $C$  increases,  $a$  approaches 0 and  $b^d$  approaches  $e$ , so  $V_{2,max}$  decreases (approaches 0).
- $C_L$  and  $C$  are interchangeable, so if  $C_L$  increases,  $V_{2,max}$  decreases (approaches 0).
- $V_{2,max}$  does not include  $R$ , so it does not increase even if  $R$  increases.
- $a$  is less than 1.  $b^d$  is less than 1. Thus,  $V_{2,max}$  is less than  $V_{DD}/2$ .

**EE434**  
**ASIC and Digital Systems**

**Midterm Exam 1**

**Feb. 16, 2018. (4:10pm – 5pm)**

**Instructor: Dae Hyun Kim ([daehyun@eecs.wsu.edu](mailto:daehyun@eecs.wsu.edu))**

**Name:**

**WSU ID:**

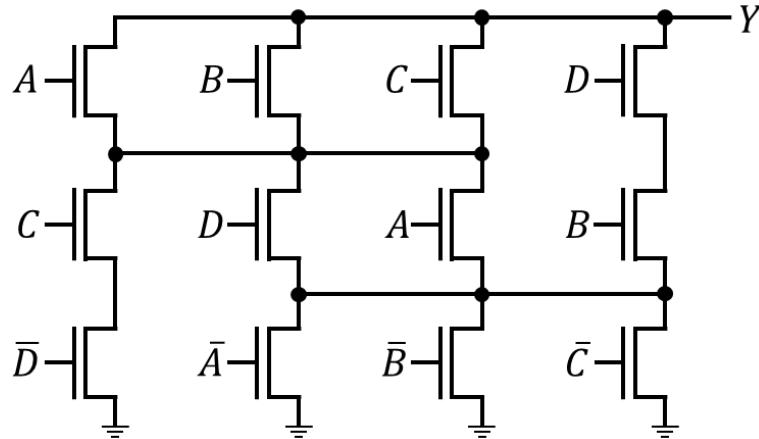
Problem	Points	
1	10	
2	10	
3	10	
4	10	
5	10	
6	10	
Total	60	

\* Allowed: Textbooks, cheat sheets, class notes, notebooks, calculators, watches

\* Not allowed: Electronic devices (smart phones, tablet PCs, laptops, etc.) except calculators and watches

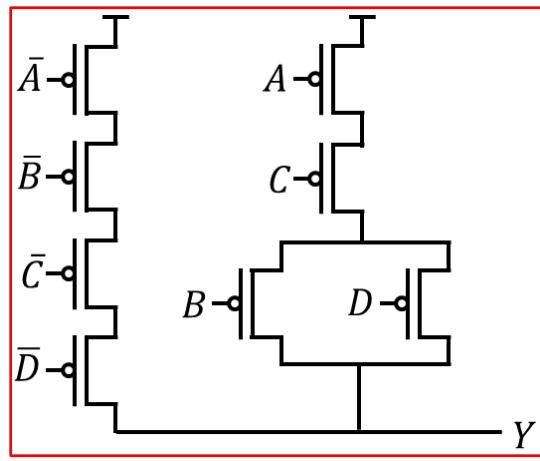
### Problem #1 (Static CMOS gates, 10 points)

The following shows the NFET network of a static CMOS gate. Draw a PFET network for the gate. Available inputs:  $A, B, C, D, \bar{A}, \bar{B}, \bar{C}, \bar{D}$ . You should minimize the total number of PFETs in your PFET network.



A	B	C	D	Y
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	1	0	0
1	1	1	1	1

Thus,  $Y$  is 1 in the following four cases:  $Y = \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}\bar{C}D + \bar{A}B\bar{C}\bar{D} + ABCD = ABCD + \bar{A}\bar{B}\bar{C} + \bar{A}\bar{C}\bar{D} = ABCD + \bar{A}\bar{C}(\bar{B} + \bar{D})$ .

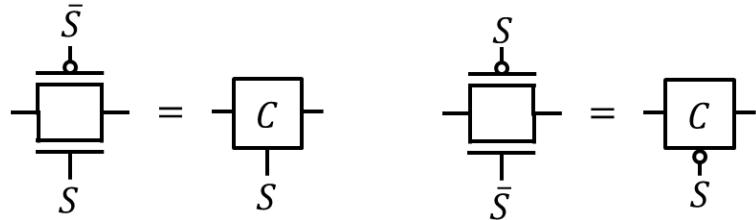


## Problem #2 (Transmission Gates, 10 points)

Design (draw a schematic) the following Boolean function using transmission gates only.

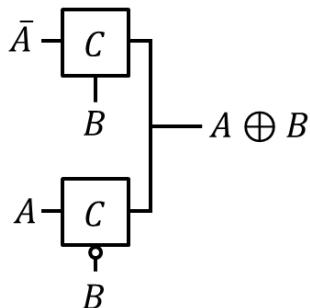
$$Y = A \oplus B \oplus C$$

Available inputs:  $A, B, C, \bar{A}, \bar{B}, \bar{C}$ . You cannot use Power ( $V_{DD}$ ) and Ground ( $V_{SS}$ ). Use the following symbols for the transmission gates.

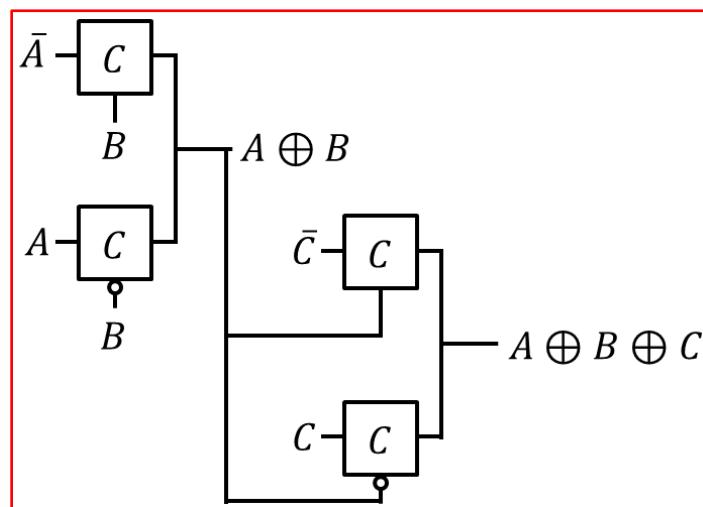


Design constraint: The total # transmission gates should be less than or equal to 10.

$$F = A \oplus B = A\bar{B} + \bar{A}B$$



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

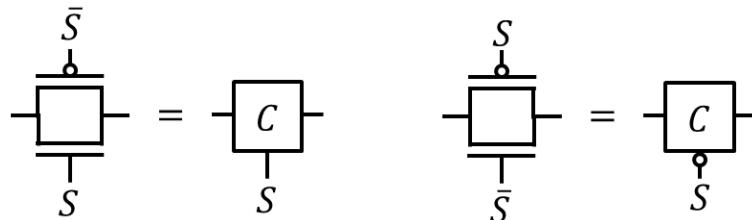


### Problem #3 (Design, 10 points)

Design (draw a schematic) a two-bit comparator using transmission gates only. The following shows a truth table for the comparator:

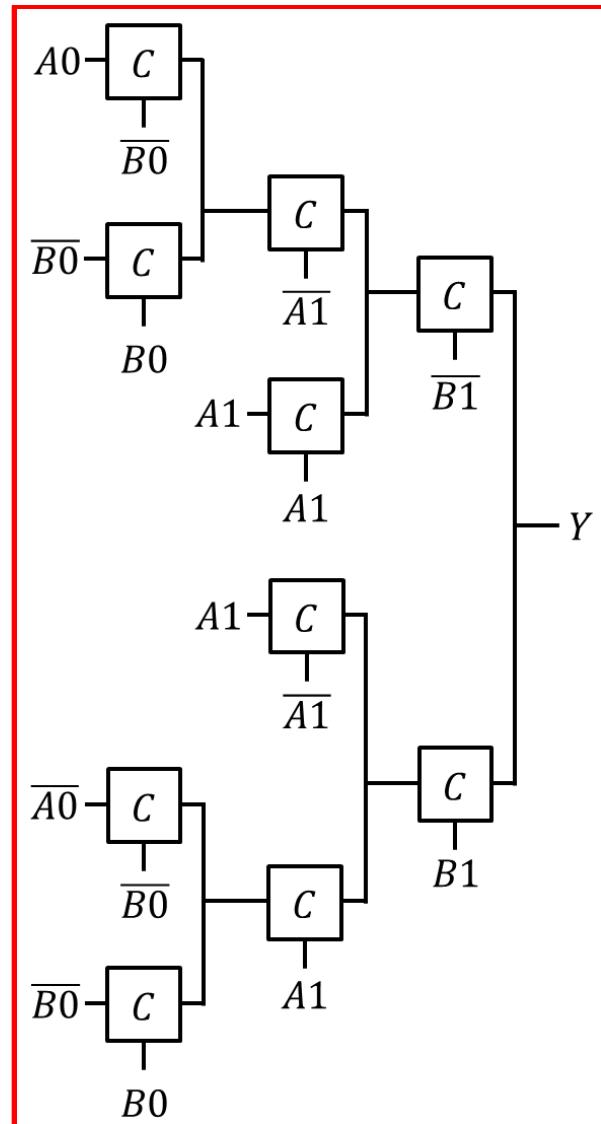
A1	A0	B1	B0	Y1
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

Available inputs:  $A1, A0, B1, B0, \bar{A1}, \bar{A0}, \bar{B1}, \bar{B0}$ . You cannot use Power ( $V_{DD}$ ) and Ground ( $V_{SS}$ ). Use the following symbols for the transmission gates.



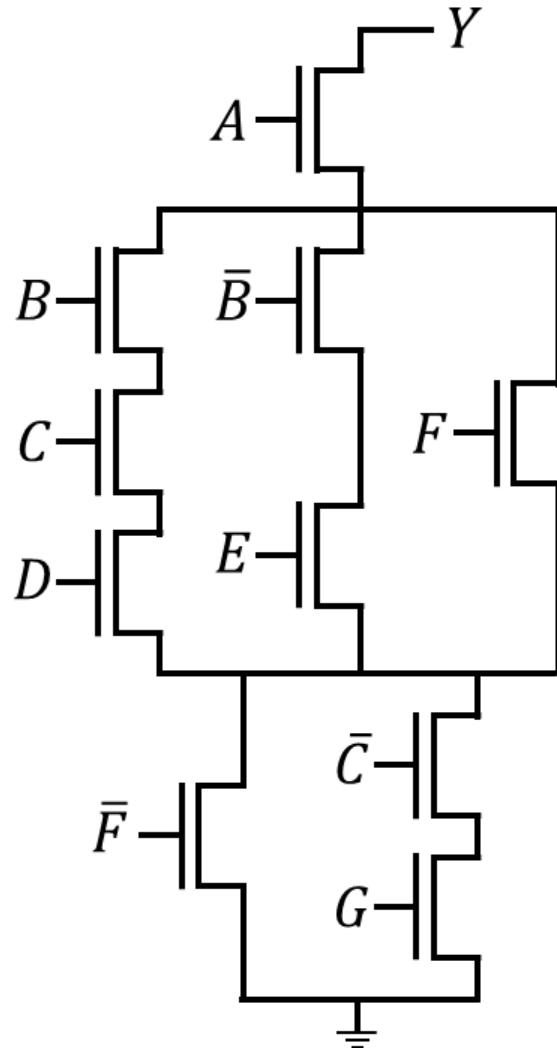
Design constraint: The total # transmission gates should be less than or equal to 12.

$$Y = A1 \cdot \bar{B1} + A0 \cdot \bar{B1} \cdot \bar{B0} + A1 \cdot A0 \cdot \bar{B0}$$



### Problem #4 (Transistor Sizing, 10 points).

Size the transistors in the following pull-down network.  $R_n$  is the resistance of a 1X NMOS transistor. Ignore parasitic capacitances. Target time constant:  $\tau_{target} \leq R_n \cdot C_L$ . Try to minimize the total area.



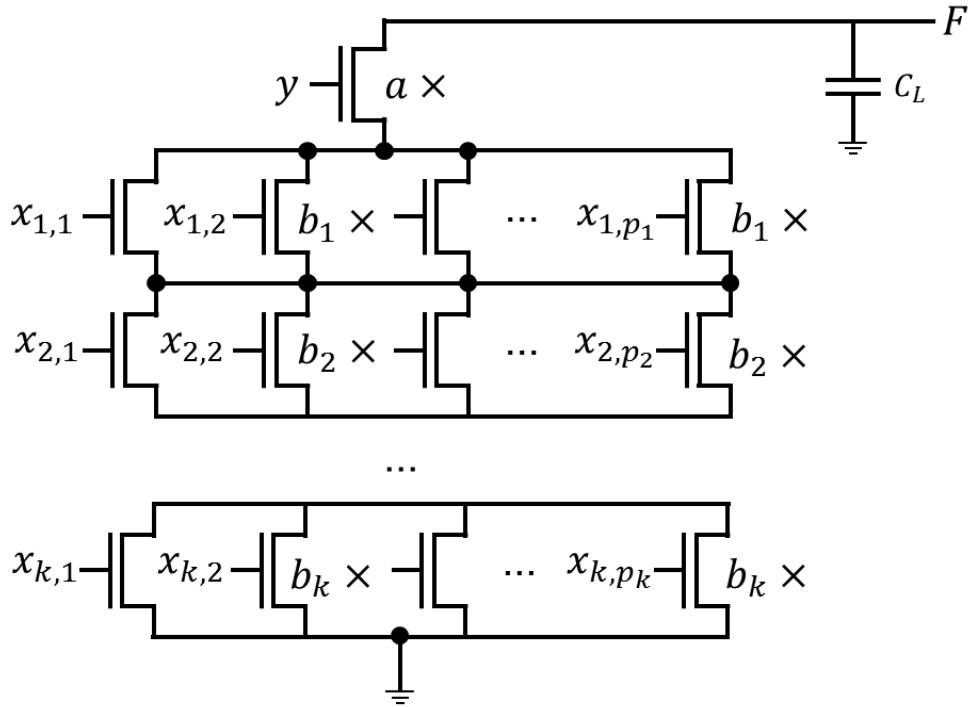
Paths  $A - B - C - D - \bar{F}$  and  $A - \bar{B} - E - \bar{C} - G$  are the two critical paths. Thus, we set the sizes of the NFETs on the path to 5X. To satisfy the timing constraint for the path  $A - F - \bar{C} - G$ , we set the size of the NFET  $F$  to 2.5X.

Total width: 47.5X

### Problem #5 (Transistor Sizing, 10 points).

The following shows an NFET network of

$$F = \overline{y \cdot (x_{1,1} + \dots + x_{1,p_1}) \cdot (x_{2,1} + \dots + x_{2,p_2}) \cdot \dots \cdot (x_{k,1} + \dots + x_{k,p_k})}$$



Notice that  $k, p_1, \dots, p_k$  are constants. The fall delay should be less than or equal to  $R_n C_L$  where  $R_n$  is the resistance of a  $1 \times$  NFET. The NFET  $y$  is upsized to  $a \times$  and each NFET connected to  $x_{i,j}$  is upsized to  $b_i \times$ . We minimize the total width

$$W = a + b_1 \cdot p_1 + b_2 \cdot p_2 + \dots + b_k \cdot p_k.$$

Find  $a$  (the size of the NFET connected to input  $y$ ) that minimizes the total width (i.e., represent  $a$  as a function of  $p_1, p_2, \dots, p_k$ .

Timing constraint:  $C_L \left( \frac{R_n}{a} + \frac{R_n}{b_1} + \frac{R_n}{b_2} + \dots + \frac{R_n}{b_k} \right) \leq R_n C_L$ . We take  $=$ , so the constraint is

$$\frac{1}{a} + \frac{1}{b_1} + \dots + \frac{1}{b_k} = 1.$$

Substitution:  $\frac{1}{a} = A, \frac{1}{b_i} = B_i$

$$A + \sum B_i = 1$$

$$W = a + \sum p_i b_i = \frac{1}{A} + \frac{p_1}{B_1} + \dots + \frac{p_k}{B_k} = \frac{1}{1 - \sum B_i} + \sum \frac{p_i}{B_i}$$

$$\frac{\partial W}{\partial B_i} = \frac{1}{(1 - \sum B_i)^2} - \frac{p_i}{B_i^2} = 0$$

Substitution:  $\sum B_i = S$

$$\frac{\partial W}{\partial B_i} = \frac{1}{(1-S)^2} - \frac{p_i}{B_i^2} = 0, \text{ so } B_i = \sqrt{p_i}(1-S).$$

$$\sum B_i = S = (1-S) \sum \sqrt{p_i}$$

Substitution:  $\sum \sqrt{p_i} = r$

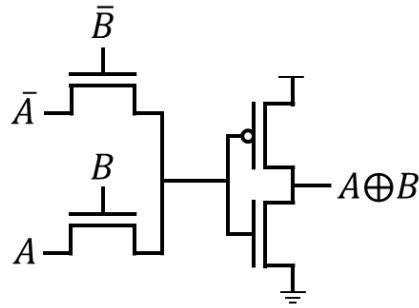
$$S = (1-S)r, \text{ so } S = \frac{r}{r+1}.$$

$$\therefore B_i = \frac{\sqrt{p_i}}{r+1} = \frac{\sqrt{p_i}}{1+\sum \sqrt{p_i}}, \quad b_i = \frac{1+\sum \sqrt{p_i}}{\sqrt{p_i}}$$

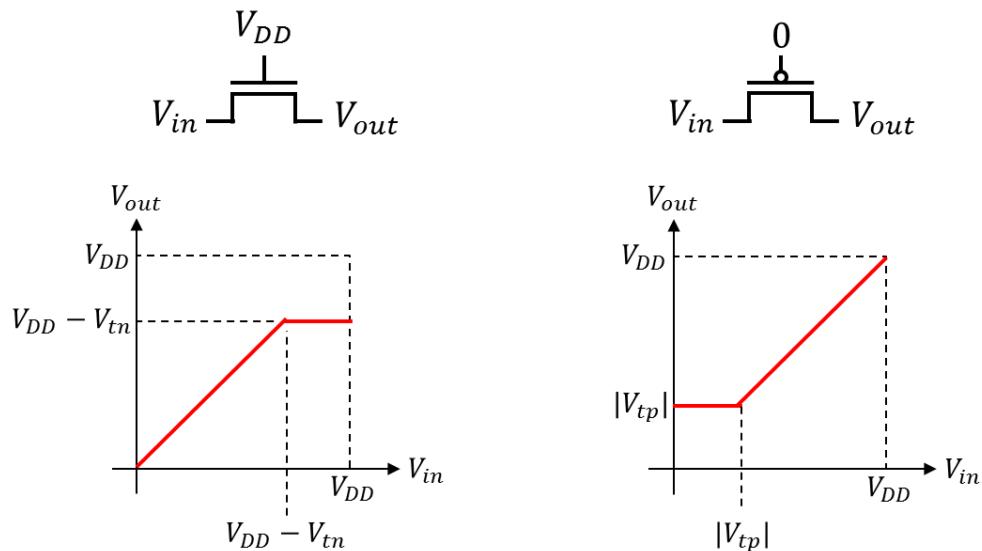
$$A = 1 - S = \frac{1}{r+1}, \quad \boxed{\therefore a = 1 + \sum \sqrt{p_i} = 1 + \sqrt{p_1} + \sqrt{p_2} + \dots + \sqrt{p_k}}$$

## Problem #6 (Pass Transistor, 10 points)

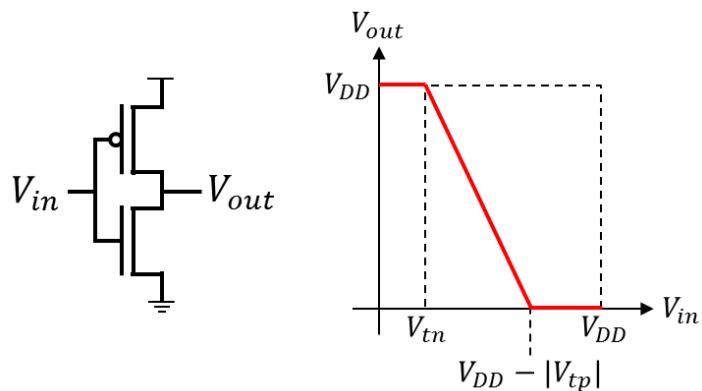
The following schematic implements an XOR gate using pass transistors.



The inverter at the output restores the signal so that the output swing can be  $[0, V_{DD}]$ .  
 The following plots show the  $V_{DS}$  characteristic of the NFET when its gate voltage is  $V_{DD}$  and the  $V_{DS}$  characteristic of the PFET when its gate voltage is 0.



The following plot shows the input-output characteristic of the inverter.



$V_{tn}$  and  $|V_{tp}|$  are the threshold voltages of the NFET and the PFET, respectively.  $A, B, \bar{A}, \bar{B}$  are either 0V (for 0) or  $V_{DD}$  (for 1). The swing of the final output  $A \oplus B$  should be  $[0, V_{DD}]$ , i.e., 0V if  $A \oplus B = 0$  and  $V_{DD}$  if  $A \oplus B = 1$ . Find all inequalities for the full output swing. (Hint: The inequalities might consist of  $V_{DD}$ ,  $V_{tn}$ , and  $|V_{tp}|$ .)

A	B	<u>Input of the inverter (logical)</u>
0	0	$V_{DD} - V_{tn}$
0	1	0
1	0	0
1	1	$V_{DD} - V_{tn}$

Thus, if the input of the inverter is 0, the output voltage is  $V_{DD}$ . If the input is  $V_{DD} - V_{tn}$ , the output should be 0V. Thus, this should be greater than or equal to  $V_{DD} - |V_{tp}|$ . Thus,  $V_{tn} \leq |V_{tp}|$  is the only inequality it should satisfy.

**EE434**  
**ASIC and Digital Systems**

**Midterm Exam 2**

**Mar. 28, 2018. (4:10pm – 5pm)**

**Instructor: Dae Hyun Kim ([daehyun@eecs.wsu.edu](mailto:daehyun@eecs.wsu.edu))**

**Name:**

**WSU ID:**

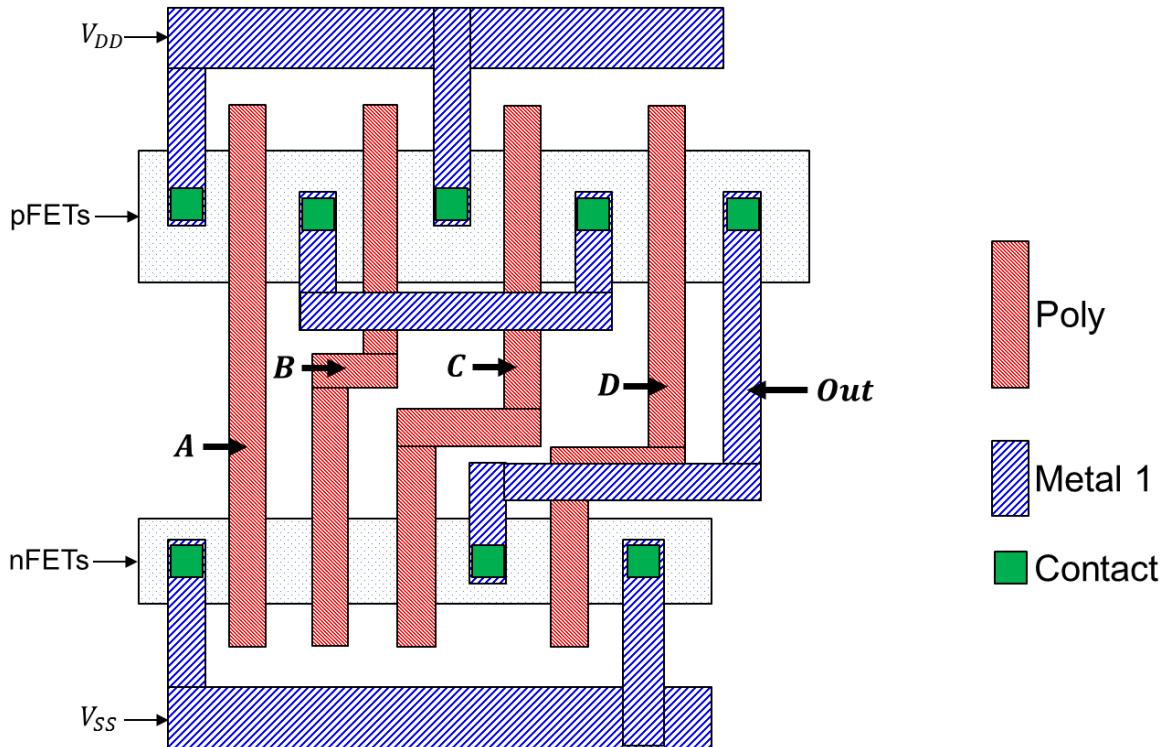
Problem	Points	
1	10	
2	10	
3	10	
4	20	
5	5	
6	15	
Total	70	

\* Allowed: Textbooks, cheat sheets, class notes, notebooks, calculators, watches

\* Not allowed: Electronic devices (smart phones, tablet PCs, laptops, etc.) except calculators and watches

## Problem #1 (Layout, 10 points)

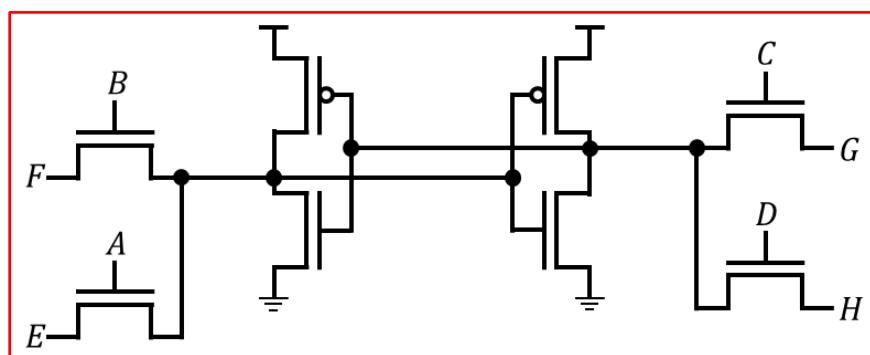
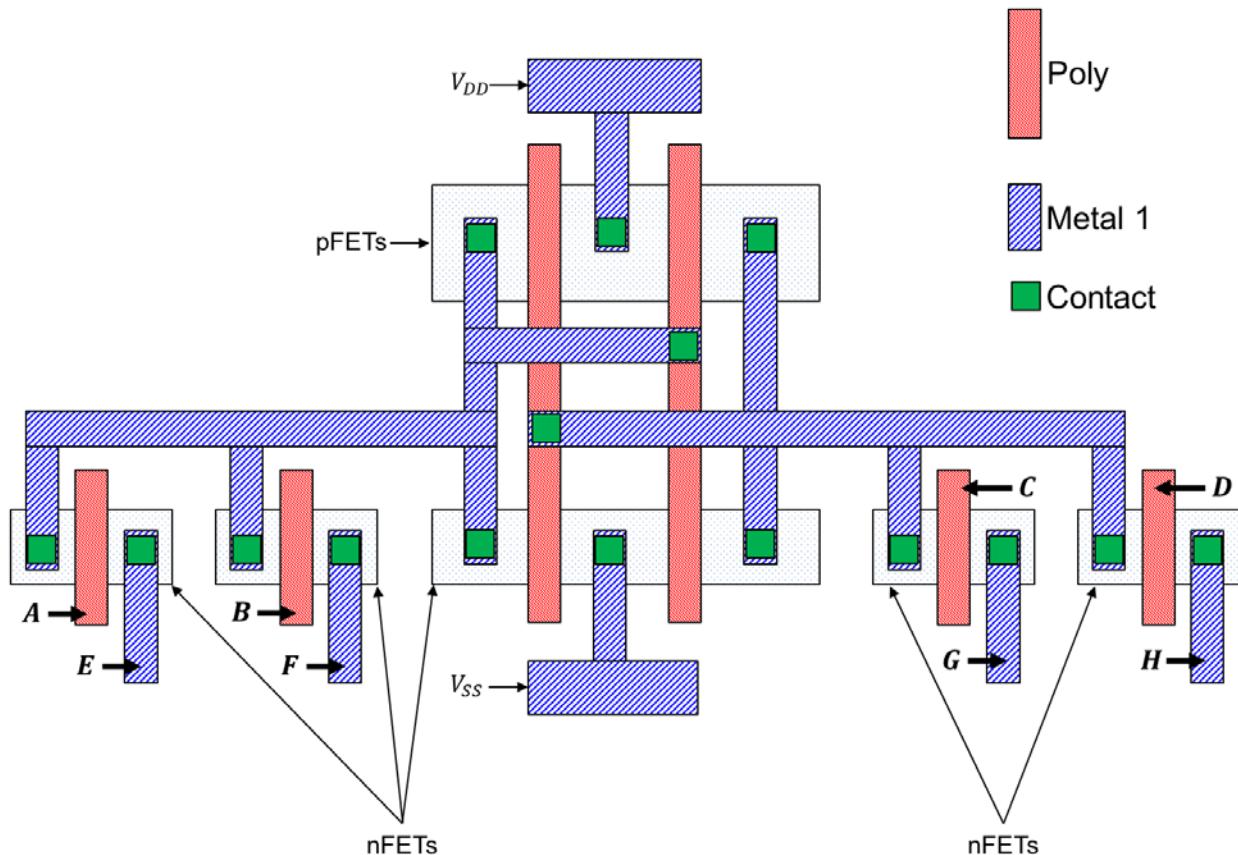
Represent *Out* as a Boolean function of  $A, B, C, D$ .



$$Out = \overline{ABC} + D$$

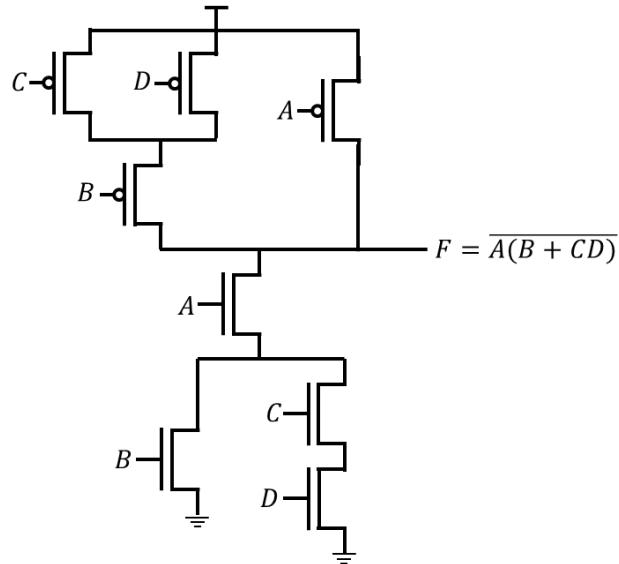
## Problem #2 (Layout, 10 points)

Draw a transistor-level schematic (netlist) for the following layout. Input ports: A, B, C, D. Inout (input/output) ports: E, F, G, H.

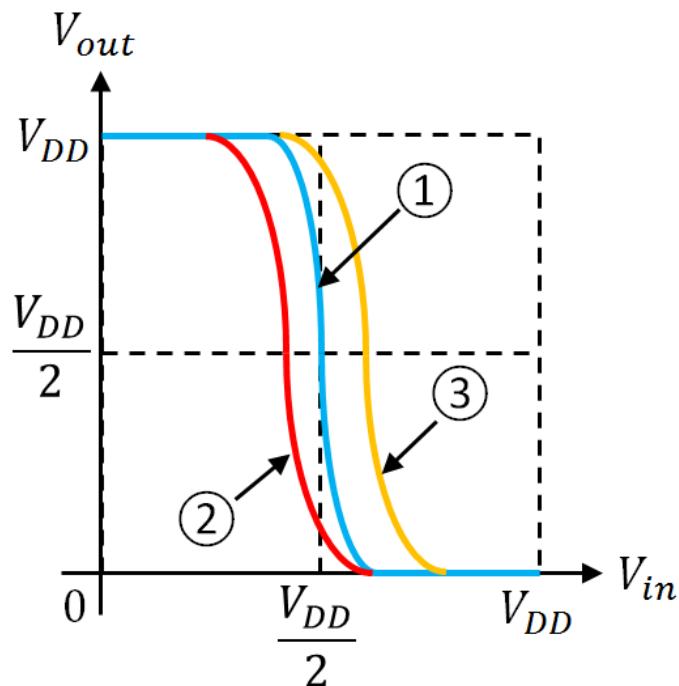


### Problem #3 (DC Analysis, 10 points)

The following schematic implements  $F = \overline{A(B + CD)}$ .



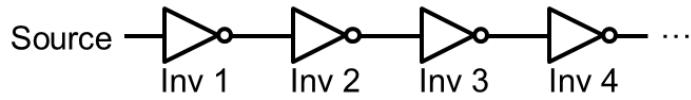
The following shows a DC characteristic graph of the logic above. Currently, the DC characteristic of the above logic follows the curve ①.



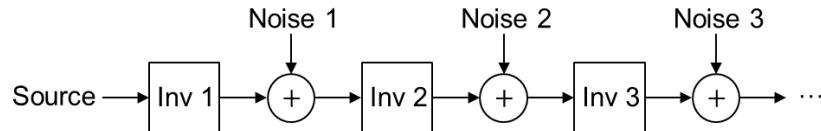
- 1) If  $\mu_n$  (the electron mobility) increases, the DC characteristic of the logic will move from ① to ②. (True/False)
- 2) If  $\beta_p$  of all the PFETs increases, the DC characteristic of the logic will move from ① to ③. (True/False)
- 3) If  $\beta_n$  of the NFET connected to input C increases, the DC characteristic of the logic will move from ① to ③. (True/False)
- 4) If the threshold voltages of all the NFETs increase (due to the body-bias effect), the DC characteristic of the logic will move from ① to ②. (True/False)
- 5) If the length of the PFET connected to input B increases, the DC characteristic of the logic will move from ① to ②. (True/False)

## Problem #4 (DC Analysis, 20 points)

An infinite chain of inverters is defined as follows:

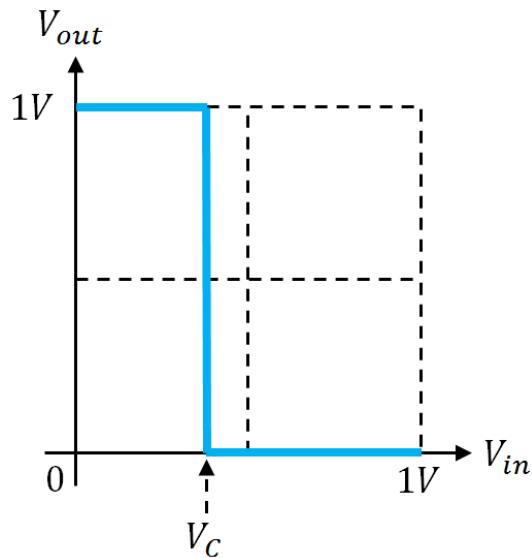


All the inverters are identical, i.e., have the same characteristics. The above chain is modeled as a block diagram as follows:



where *Noise k* is the *k*-th noise and *Source* is a signal generator and  $V_{Source} = V_{DD} \cdot u(t)$  (i.e., 0 if  $t < 0$  and  $V_{DD}$  if  $t \geq 0$ ).  $V_{DD} = 1V$ . The signal source is either 0V (for logic 0) or 1V (for logic 1). All the noise sources are independent. For example, if the range of the value of each noise source is [-0.1V, 0.1V], the value of noise source 1 could be 0.05V while the value of noise source 2 is 0.07V and the value of noise source 3 is -0.03V.

- 1) The following shows the DC characteristics of the inverters.  $V_C$  is between 0V and 1V. If the range of the value of each noise source is [0,0.3V] (i.e,  $0V \leq noise \leq 0.3V$ ), what is the minimum value of  $V_C$  that does not lead to signal inversion? (5 points)



If the output of an inverter is  $t$ , it becomes  $[t, t + 0.3]$  at the input of its next inverter. If the logical value of  $t$  is 1,  $[t, t + 0.3]$  should be greater than  $V_C$ . If the logical value of  $t$  is 0,  $[t, t + 0.3]$  should be less than  $V_C$ . Thus,  $t > V_C$  and  $t + 0.3 < V_C$  should be satisfied. If

$t$  is 1V for the signal source, it is always greater than  $V_C$ . If  $t$  is 0V for the signal source,  $t + 0.3V = 0.3V$  should always be less than  $V_C$ . Thus, the minimum value of  $V_C$  is 0.3V.

- 2) Assume that all the inverters follow the DC characteristic curve shown above. If the range of the value of each noise source is  $[-0.1V, 0.2V]$  (i.e,  $-0.1V \leq \text{noise} \leq 0.2V$ ), what is the minimum value of  $V_C$  that does not lead to signal inversion? (5 points) What is the maximum value of  $V_C$  that does not lead to signal inversion? (5 points)

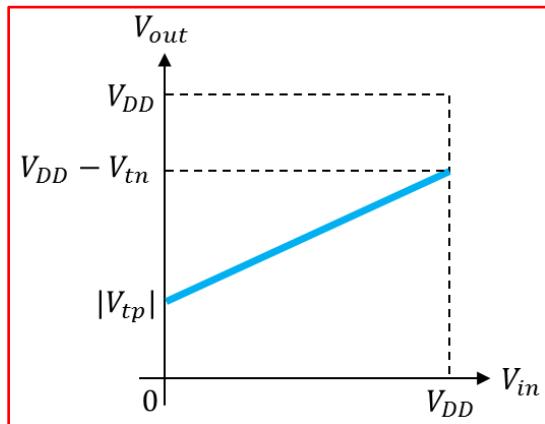
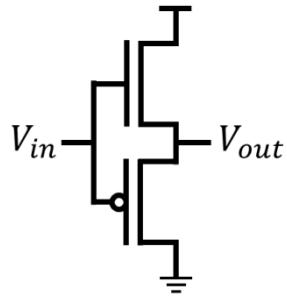
We apply the same analysis to find min. and max. of  $V_C$ . If the output of an inverter is  $t$ , it becomes  $[t - 0.1, t + 0.2]$  at the input of its next inverter. If the logical value of  $t$  is 1,  $[t - 0.1, t + 0.2] = [0.9V, 1.2V]$  should be greater than  $V_C$ . If the logical value of  $t$  is 0,  $[t - 0.1, t + 0.2] = [-0.1V, 0.2V]$  should be less than  $V_C$ . Thus,  $V_C$  should be less than 0.9V and greater than 0.2V.

Min: 0.2V

Max: 0.9V

3) Draw a DC characteristic curve for the following buffer circuit. (5 points)

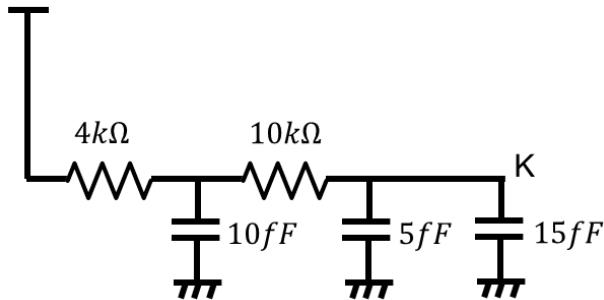
- $V_{tn}$ : Threshold voltage of the NFET.
- $V_{tp}$ : Threshold voltage of the PFET.
- You don't need to derive equations or formulas to draw it. Just a rough sketch will be accepted.
- However, you should show the output values for  $V_{in} = 0$  and  $V_{in} = V_{DD}$ .



### Problem #5 (Elmore Delay, 5 points)

Compute the Elmore delay at node K.

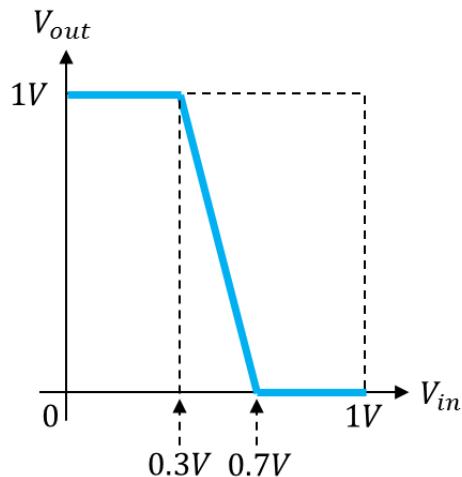
$$V(t) = V_{DD} \cdot u(t)$$



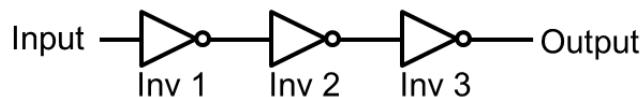
$$\tau = 10k * 20f + 4k * 30f = 200p + 120p = 320ps$$

## Problem #6 (DC Analysis, 15 points)

The following shows the DC characteristic curve of an inverter.



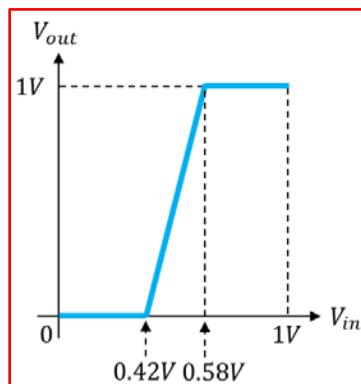
Draw a DC characteristic curve between the input and the output of the following inverter chain composed of three inverters whose DC characteristics follow the graph shown above. Note: The DC curve you draw should be very accurate, i.e., show all the important values such as x-intercepts, y-intercepts, etc.



The equation of the straight line in the middle is as follows:

$$y = -2.5x + 1.75$$

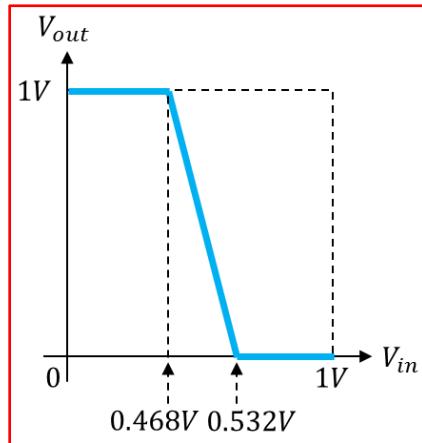
Thus, the output is 0.7V when the input is 0.42V. Similarly, the output is 0.3V when the input is 0.58V. Thus, the combined DC characteristic of the first two inverters is as follows:



The equation of the straight line in the middle is as follows:

$$y = 6.25x - 2.625$$

Thus, the output is 0.7V when the input is 0.532V. Similarly, the output is 0.3V when the input is 0.468V. Thus, the DC characteristic of the whole circuit is as follows:



**EE434**  
**ASIC and Digital Systems**

**Final Exam**

**May 1, 2019. (8am – 10am)**

**Instructor: Dae Hyun Kim ([daehyun@eecs.wsu.edu](mailto:daehyun@eecs.wsu.edu))**

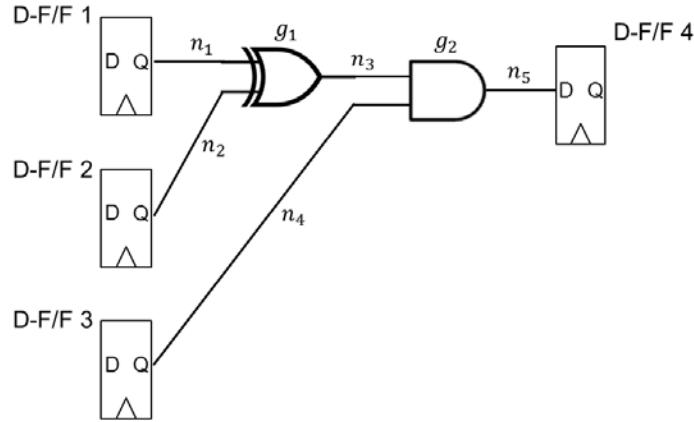
**Name:**

**WSU ID:**

Problem	Points	
1	20	
2	20	
3	30	
4	20	
5	20	
6	20	
7	20	
8	20	
Total	170	

## Problem #1 (Static Timing Analysis, 20 points)

The following shows a pipeline stage. D-FF1, D-FF2, and D-FF3 are stage-k D-FFs and D-FF4 is a stage-(k+1) D-FF.  $n_i$  and  $g_m$  are net and gate delays, respectively.



The properties of D-FF $p$  ( $p = 1 \sim 4$ ) are as follows:

- Setup time:  $s_p$  (for example,  $s_3$  is the setup time of D-FF3.)
- Hold time:  $h_p$  (for example,  $h_2$  is the hold time of D-FF2.)
- C-Q delay:  $c_p$  (for example,  $c_1$  is the C-Q delay of D-FF1.)
- Delay from the clock source to the clock pin of D-FF $p$ :  $d_p$  (e.g.,  $d_4$  is the delay from the clock source to the clock pin of D-FF4.)

$T_{CLK}$  is the clock period. You can also use the “MAX” and “MIN” operators.

$\text{MAX}(a, b) = a$  (if  $a > b$ ) or  $b$  (otherwise).  $\text{MIN}(a, b) = a$  (if  $a < b$ ) or  $b$  (otherwise).

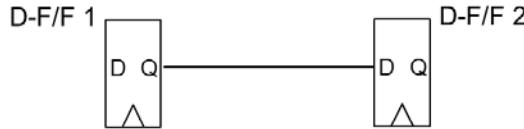
(1) Find all inequalities for the setup time constraints of the system shown above.

$$\text{MAX}\{\text{MAX}(d_1 + c_1 + n_1, d_2 + c_2 + n_2) + g_1 + n_3, d_3 + c_3 + n_4\} + g_2 + n_5 \leq d_4 + T_{CLK} - s_4$$

(2) Find all inequalities for the hold time constraints of the system shown above.

$$\text{MIN}\{\text{MIN}(d_1 + c_1 + n_1, d_2 + c_2 + n_2) + g_1 + n_3, d_3 + c_3 + n_4\} + g_2 + n_5 \geq d_4 + h_4$$

## Problem #2 (STA & Power, 20 points)



The output Q of D-FF 1 is directly connected to the input D of D-FF 2. The length of the net is negligible, so the net delay is zero.

- $T_{skew}$ : 0ps
- $T_{h2}$  (the hold time of D-FF 2): 40ps
- $T_{CQ1}$  (the clock-to-Q delay of D-FF 1): 10ps
- Available buffers: BUF\_X1, BUF\_X2, BUF\_X4
- The input and output capacitance of a buffer: negligible (0 fF)
- The internal delay of a buffer BUF\_Xs:  $\frac{12}{s}$  ps (for example, the internal delay of a buffer BUF\_X2 is  $12/2=6$ ps.)
- The power consumption of a buffer BUF\_Xs:  $10 \cdot s + 20$  (nW) (for example, the power consumption of a BUF\_X2 is 40 nW.)

You are supposed to insert buffers into the net so that you can satisfy the hold time constraint and minimize the total power consumption. Find how many buffers you should insert into the net.

Hold time constraint:  $d_1 + T_{CQ1} + T_{logic} \geq d_2 + T_{h2} \Leftrightarrow T_{logic} \geq 30$ ps

Suppose we insert #a BUF\_X1, #b BUF\_X2, #c BUF\_X4 buffers. Then, the total logic delay is  $12a + 6b + 4c$ .  $12a + 6b + 4c \geq 30$ , so  $6a + 3b + 2c \geq 15$ .

The total power consumption is  $(30a + 40b + 60c)$  nW.

Let's enumerate all the possibilities for  $(a, b, c)$  and get their power values.

$(0, 0, 8)$ : 480nW,  $(0, 1, 6)$ : 400nW,  $(0, 2, 5)$ : 380nW,  $(0, 3, 3)$ : 300nW,  $(0, 4, 2)$ : 280nW,  $(0, 5, 0)$ : 200nW.

$(1, 0, 5)$ : 330nW,  $(1, 1, 3)$ : 250nW,  $(1, 2, 2)$ : 230nW,  $(1, 3, 0)$ : 150nW

$(2, 0, 2)$ : 180nW,  $(2, 1, 0)$ : 100nW,  $(3, 0, 0)$ : 90nW

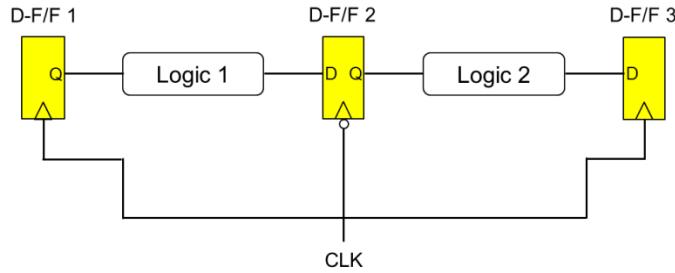
# BUF\_X1: 3

# BUF\_X2: 0

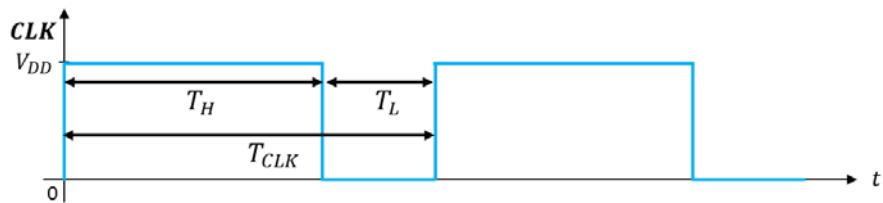
# BUF\_X4: 0

### Problem #3 (STA, 30 points)

The following figure shows two pipeline stages. Notice that D-FF 1 and D-FF 3 are positive-edge-triggered FFs, whereas D-FF 2 is a negative-edge-triggered FF.



The following shows the waveform of the clock.



Notice that the duty cycle of the clock ( $= T_H / T_{CLK}$ ) is not 50%.

- Delay from the clock source to D-FF 1, 2, 3:  $d_1, d_2, d_3$
- C-Q delay of D-FF 1, 2, 3:  $c_1, c_2, c_3$
- Delay of Logic 1 and Logic 2:  $T_1, T_2$
- Setup time of D-FF 1, 2, 3:  $s_1, s_2, s_3$
- Hold time of D-FF 1, 2, 3:  $h_1, h_2, h_3$

(1) Find all inequalities for the setup time constraints of the system shown above.

$$d_1 + c_1 + T_1 \leq d_2 + T_H - s_2$$

$$d_2 + c_2 + T_2 + T_H \leq d_3 + T_{CLK} - s_3 \Leftrightarrow d_2 + c_2 + T_2 \leq d_3 + T_L - s_3$$

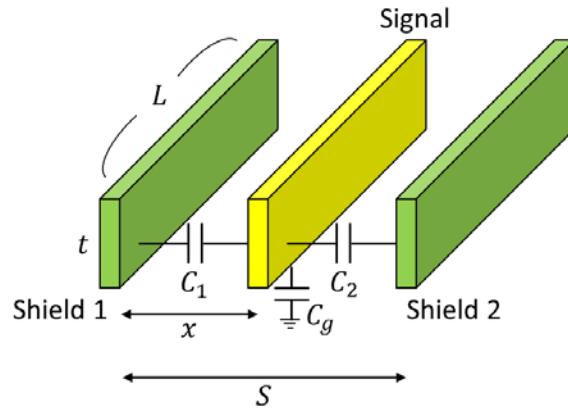
(2) Find all inequalities for the hold time constraints of the system shown above.

$$d_1 + T_{CLK} + c_1 + T_1 \geq d_2 + T_{CLK} - T_L + h_2 \Leftrightarrow d_1 + c_1 + T_1 \geq d_2 - T_L + h_2$$

$$d_2 + T_{CLK} + T_H + c_2 + T_2 \geq d_3 + T_{CLK} + h_3 \Leftrightarrow d_2 + T_H + c_2 + T_2 \geq d_3 + h_3$$

## Problem #4 (Coupling, 20 points)

The following figure shows a signal net surrounded by two shield nets (Shield 1 and Shield 2) that are grounded. The coupling caps between Shield 1 and the signal net and between the signal net and Shield 2 are  $C_1$  and  $C_2$ , respectively.



- Delay of the signal net:  $(R + 0.5rL) \cdot (C_g + C_1 + C_2)$
- $R$  (constant): The output resistance of the driver driving the signal net
- $r$  (constant): Unit wire resistance
- $L$  (constant): The length of the wires
- $t$  (constant): The thickness of the wires
- $S$  (constant): The distance between Shield 1 and Shield 2
- $C_g$  (constant): The ground cap of the signal net
- $\epsilon$  (constant): Permittivity of the insulator material
- $x$  (variable): The distance between Shield 1 and the signal net
- $C_1 = \frac{\epsilon t L}{x}, C_2 = \frac{\epsilon t L}{S-x}$

Find  $x$  that minimizes the delay of the signal net (you should express the optimal value of  $x$  as a function of some of the constants given above).

$$\text{Delay } D = (R + 0.5rL) \cdot (C_g + \frac{\epsilon t L}{x} + \frac{\epsilon t L}{(S-x)})$$

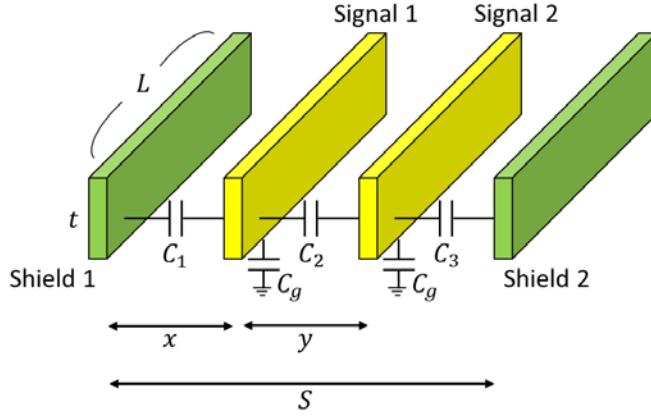
$$\frac{dD}{dx} = (R + 0.5rL) \cdot \left( -\frac{\epsilon t L}{x^2} + \frac{\epsilon t L}{(S-x)^2} \right) = 0$$

$$x = S - x$$

$$x = \frac{S}{2}$$

## Problem #5 (Coupling, 20 points)

This problem is similar to Problem #4, but now we will consider two signal nets as follows.



- Delay of Signal 1:  $(R + 0.5rL) \cdot (C_g + C_1 + 2C_2)$
- Delay of Signal 2:  $(R + 0.5rL) \cdot (C_g + C_3 + 2C_2)$
- $x$  (variable): The distance between Shield 1 and Signal 1
- $y$  (variable): The distance between Signal 1 and Signal 2
- $C_1 = \frac{\epsilon t L}{x}, C_2 = \frac{\epsilon t L}{y}, C_3 = \frac{\epsilon t L}{S-(x+y)}$

Find  $x$  and  $y$  that minimize the sum of the delays of the two signal nets (you should express the optimal values of  $x$  and  $y$  as functions of some of the constants given above).

$$\text{Sum of the delays } D = (R + 0.5rL) \cdot (2C_g + \frac{\epsilon t L}{x} + \frac{\epsilon t L}{S-(x+y)} + 4 \frac{\epsilon t L}{y})$$

1) Since the structure is symmetric, we can use  $= S - (x + y)$ .  $y = S - 2x$ .

$$\frac{dD}{dx} = (R + 0.5rL) \cdot \left( -2 \frac{\epsilon t L}{x^2} + \frac{8\epsilon t L}{(S-2x)^2} \right) = 0.$$

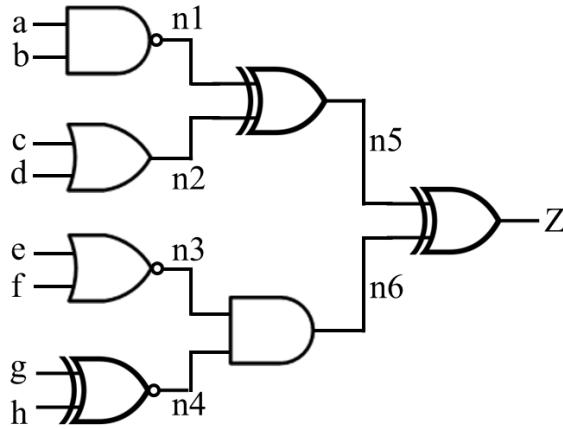
$$4x^2 = (S-2x)^2 \cdot x = \frac{S}{4} \cdot y = \frac{S}{2}.$$

$$2) \frac{\partial D}{\partial x} = (R + 0.5rL) \cdot \left( -\frac{\epsilon t L}{x^2} + \frac{\epsilon t L}{(S-(x+y))^2} \right) = 0. \text{ From this, we get } x = S - (x + y).$$

$$\frac{\partial D}{\partial y} = (R + 0.5rL) \cdot \left( \frac{\epsilon t L}{(S-(x+y))^2} - 4 \frac{\epsilon t L}{y^2} \right) = 0. \text{ From this, we get } S - (x + y) = \frac{y}{2}.$$

$$\text{From them, we get } x = \frac{S}{4}, y = \frac{S}{2}.$$

## Problem #6 (Testing, 20 points)



- 1) Find all test vectors that can detect a s-a-0 fault at input  $f$ . (You can use  $X$  for don't-cares).

$$X \oplus X_F = 1.$$

$$X = \{\overline{a \cdot b} \oplus (c + d)\} \oplus \{\overline{e + f} \cdot \overline{g \oplus h}\}$$

$$X_F = \{\overline{a \cdot b} \oplus (c + d)\} \oplus \{\bar{e} \cdot \overline{g \oplus h}\}$$

$f = 1$ . Then,  $\overline{e + f} = 0$ . Thus,  $e = 0$ . Then,  $= \{\overline{a \cdot b} \oplus (c + d)\} \oplus \{0\}$ .  $X_F = \{\overline{a \cdot b} \oplus (c + d)\} \oplus \{\overline{g \oplus h}\}$ . Therefore,  $\overline{g \oplus h} = 1$ , i.e.,  $(g, h) = (0,0)$  or  $(1,1)$ .

Now,  $X = \{\overline{a \cdot b} \oplus (c + d)\}$ .  $X_F = \{\overline{a \cdot b} \oplus (c + d)\}$ . Thus,  $X \oplus X_F = 1$  for any  $a, b, c, d$ .

Answer: (XXXX0100) or (XXXX0111).

- 2) Find all test vectors that can detect a s-a-1 fault at node  $n_3$ . (You can use  $X$  for don't-cares).

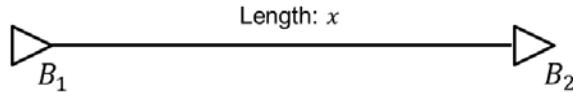
$$X_F = \{\overline{a \cdot b} \oplus (c + d)\} \oplus \{1 \cdot \overline{g \oplus h}\} = \{\overline{a \cdot b} \oplus (c + d)\} \oplus \{\overline{g \oplus h}\}$$

Thus,  $\overline{g \oplus h} = 1$  and  $\overline{e + f} = 0$ . From the former, we get  $(g, h) = (0,0)$  or  $(1,1)$ . From the latter, we get  $(e, f) = (0,1)$  or  $(1,0)$  or  $(1,1)$ .

Answer: (XXXX0100) or (XXXX0111) or (XXXX1000) or (XXXX1011) or (XXXX1100) or (XXXX1111).

## Problem #7 (Interconnects, 20 points)

The following figure shows a buffer (B1) driving a buffer (B2). We can route the net through a lower metal layer such as metal layer 1 or an upper metal layer such as metal layer 6. The following shows the properties of the two metal layers.



	Metal layer 1	Metal layer 6
Resistivity	$\rho$	
Permittivity	$\epsilon$	
Thickness	$t$	$k_1 \cdot t$
Width	$w$	$k_2 \cdot w$
Spacing between two metal wires	$s$	$k_3 \cdot s$

The wire is sufficiently long, so you can ignore the input capacitance of B2. You can also assume that the output resistance of B1 is very small (almost zero ohm).

Problem: Suppose  $\tau_1$  and  $\tau_2$  are the delays of the net routed in metal layer 1 and metal layer 6, respectively. If  $k_1 = 2$ ,  $k_2 = 4$ ,  $k_3 = 2$ , what is the ratio between  $\tau_1$  and  $\tau_2$ ?

$$\tau_1 = \frac{1}{2} r_1 c_1 x^2 \text{ where } c_1 = \epsilon \frac{t}{s}, r_1 = \rho \frac{1}{wt}.$$

$$\tau_2 = \frac{1}{2} r_2 c_2 x^2 \text{ where } c_2 = \epsilon \frac{k_1 t}{k_3 s}, r_2 = \rho \frac{1}{k_1 k_2 wt}.$$

$$\text{Thus, } \frac{\tau_1}{\tau_2} = \frac{r_1 c_1}{r_2 c_2} = k_2 k_3.$$

## Problem #8 (Interconnects, 20 points)

The following figure shows a driver ( $K_D$ ), a sink ( $K_S$ ), and evenly-distributed  $\#4n$  buffers.



Since the buffers are evenly distributed,  $s_1 = s_2 = \dots = s_{4n+1}$  where  $s_j$  is the distance between  $B_j$  and  $B_{j+1}$  (you can think of  $K_D$  as  $B_0$  and  $K_S$  as  $B_{4n+1}$ ).

However, there is one constraint. A half of the buffers must be BUF\_X1 and the other half must be BUF\_X2. The following shows the characteristics of the buffers:

- BUF\_X1: Output resistance ( $R_0$ ), input capacitance ( $C_0$ ), output capacitance ( $C_m$ ), internal delay ( $d_0$ )
- BUF\_X2: Output resistance ( $\frac{R_0}{2}$ ), input capacitance ( $2C_0$ ), output capacitance ( $2C_m$ ), internal delay ( $2d_0$ )

The driver and the sink are BUF\_X1 buffers.

Problem: Determine the size of each buffer (from  $B_1$  to  $B_{4n}$ ) to minimize the total delay from the driver to the sink.

There are four cases as follows:

- 1) A 1X buffer drives a 1X buffer: Delay  $\tau_{1,1} = R_0(C_m + C_w + C_0) + R_w C_0 + \frac{1}{2}R_w C_w$
- 2) A 1X buffer drives a 2X buffer: Delay  $\tau_{1,2} = R_0(C_m + C_w + 2C_0) + R_w 2C_0 + \frac{1}{2}R_w C_w$
- 3) A 2X buffer drives a 1X buffer: Delay  $\tau_{2,1} = \frac{R_0}{2}(2C_m + C_w + C_0) + R_w C_0 + \frac{1}{2}R_w C_w$
- 4) A 2X buffer drives a 2X buffer: Delay  $\tau_{2,2} = \frac{R_0}{2}(2C_m + C_w + 2C_0) + R_w 2C_0 + \frac{1}{2}R_w C_w$

Whenever a buffer drives a buffer, there is a minimum delay  $R_0 C_m + R_w C_0 + \frac{1}{2}R_w C_w$ . In addition, the buffers should anyway drive the wires. Thus, we do not need to think about them. We can only care about the additional terms as follows:

- 1) A 1X buffer drives a 1X buffer: Delay  $\tau_{1,1} = R_0 C_0$
- 2) A 1X buffer drives a 2X buffer: Delay  $\tau_{1,2} = 2R_0 C_0$
- 3) A 2X buffer drives a 1X buffer: Delay  $\tau_{2,1} = \frac{R_0 C_0}{2}$
- 4) A 2X buffer drives a 2X buffer: Delay  $\tau_{2,2} = R_0 C_0$

Notice that if a 1X buffer drives a 1X buffer, there should be a 2X buffer driving a 2X buffer. The delay of these pairs is  $\tau_{1,1} + \tau_{2,2} = 2R_0C_0$ . Similarly, if a 1X buffer drives a 2X buffer, there should be a 2X buffer driving a 1X buffer. The delay of these pairs is  $\tau_{1,2} + \tau_{2,1} = 2.5R_0C_0$ . Comparing these pairs, the former has a shorter delay. Thus, the answer is to place 2X buffers consecutively. For example,

$$K_D - 1X - 1X - \dots - 1X - 2X - 2X - \dots - 2X - 1X - 1X - \dots - 1X - K_S$$

**EE434**  
**ASIC and Digital Systems**

**Midterm Exam 1**

**Feb. 27, 2019. (4:10pm – 5pm)**

**Instructor: Dae Hyun Kim ([daehyun@eeecs.wsu.edu](mailto:daehyun@eeecs.wsu.edu))**

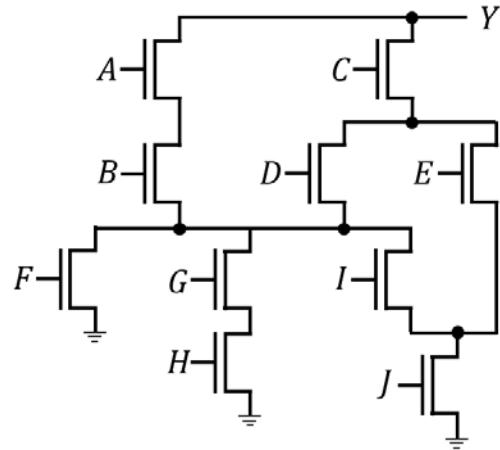
**Name:**

**WSU ID:**

Problem	Points	
1	10	
2	10	
3	10	
4	20	
5	20	
6	10	
Total	80	

### Problem #1 (Static CMOS gates, 10 points)

The following shows the NFET network of a static CMOS gate. Express the output  $Y$  as a Boolean function of the inputs ( $A \sim J$ ). (You don't need to simplify the expression.)



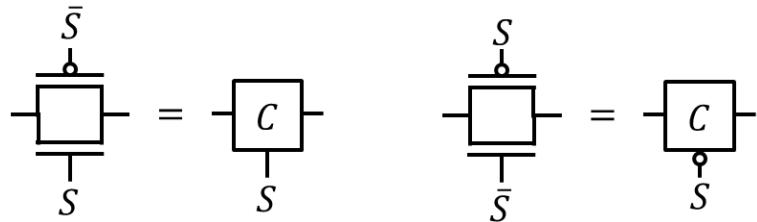
$$Y = \overline{AB(F + GH + IJ + DEJ)} + CD(F + GH + IJ) + CE(J + IF + IGH)$$

## Problem #2 (Transmission Gates, 10 points)

Design (draw a schematic) the following Boolean function using transmission gates only.

$$Y = (A \oplus B) + \overline{((A \cdot B) \oplus C)}$$

Available inputs:  $A, B, C, \bar{A}, \bar{B}, \bar{C}$ . You cannot use Power ( $V_{DD}$ ) and Ground ( $V_{SS}$ ). Use the following symbols for the transmission gates.



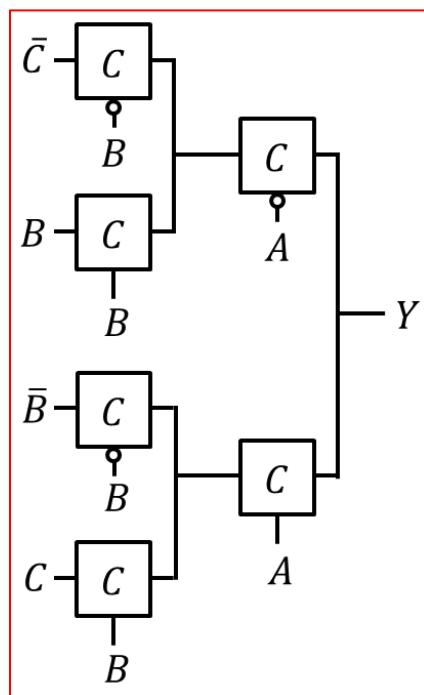
(# TGs  $\leq$  6: 10 points. 7  $\leq$  # TGs  $\leq$  8: 7 points. 9  $\leq$  # TGs  $\leq$  10: 5 points. # TGs > 10: 0 points)

$$AB = 00: Y = \bar{C}$$

$$AB = 01: Y = 1$$

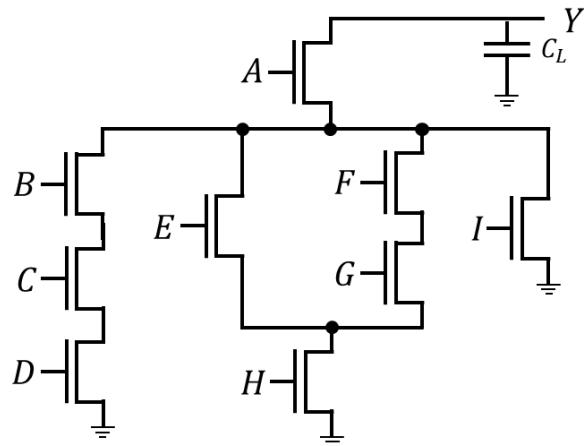
$$AB = 10: Y = 1$$

$$AB = 11: Y = C$$



### Problem #3 (Transistor Sizing, 10 points)

Size the transistors in the following pull-down network.  $R_n$  is the resistance of a 1X NMOS transistor.  $C_L$  is the load capacitance. Ignore parasitic capacitances. Target delay:  $\tau_T \leq R_n \cdot C_L$ . Try to minimize the total area.



(Total width  $W \leq 31X$ : 10 points.  $31X < W \leq 32X$ : 8 points.  $32X < W \leq 34X$ : 5 points.  $34X < W$ : 3 points)

Longest paths: A-B-C-D, A-F-G-H, so they are upsized to 4X.

A: 4X

B: 4X

C: 4X

D: 4X

E: 2X

F: 4X

G: 4X

H: 4X

I: 4/3X

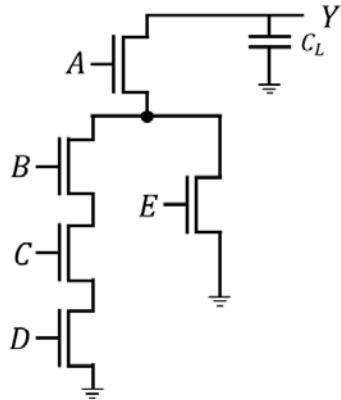
Total: 94/3X (31.33X)

Transistor A is a bottleneck, so if we upsize it to 5X, B,C,D,F,G,H can be upsized to 15/4X. In this case, E becomes 15/8X and I becomes 5/4X. The total width in this case is 245/8X, which is 30.625X.

## Problem #4 (Transistor Sizing, 20 points)

Solve either 4-(1) or 4-(2). You don't need to solve both.

- (1) (20 points) Size the transistors in the following pull-down network.  $R_n$  is the resistance of a 1X NMOS transistor.  $C_L$  is the load capacitance. Ignore parasitic capacitances. Target delay:  $\tau_T \leq R_n \cdot C_L$ . Minimize the total area (i.e., size the transistors optimally).



A: aX, B,C,D: bX, E: cX

$$\frac{1}{a} + \frac{3}{b} = 1, \frac{1}{a} + \frac{1}{c} = 1$$

$$b = \frac{3a}{a-1}, c = \frac{a}{a-1}$$

$$W = a + 3b + c = a + \frac{9a}{a-1} + \frac{a}{a-1} = a + 10 \frac{a}{a-1}$$

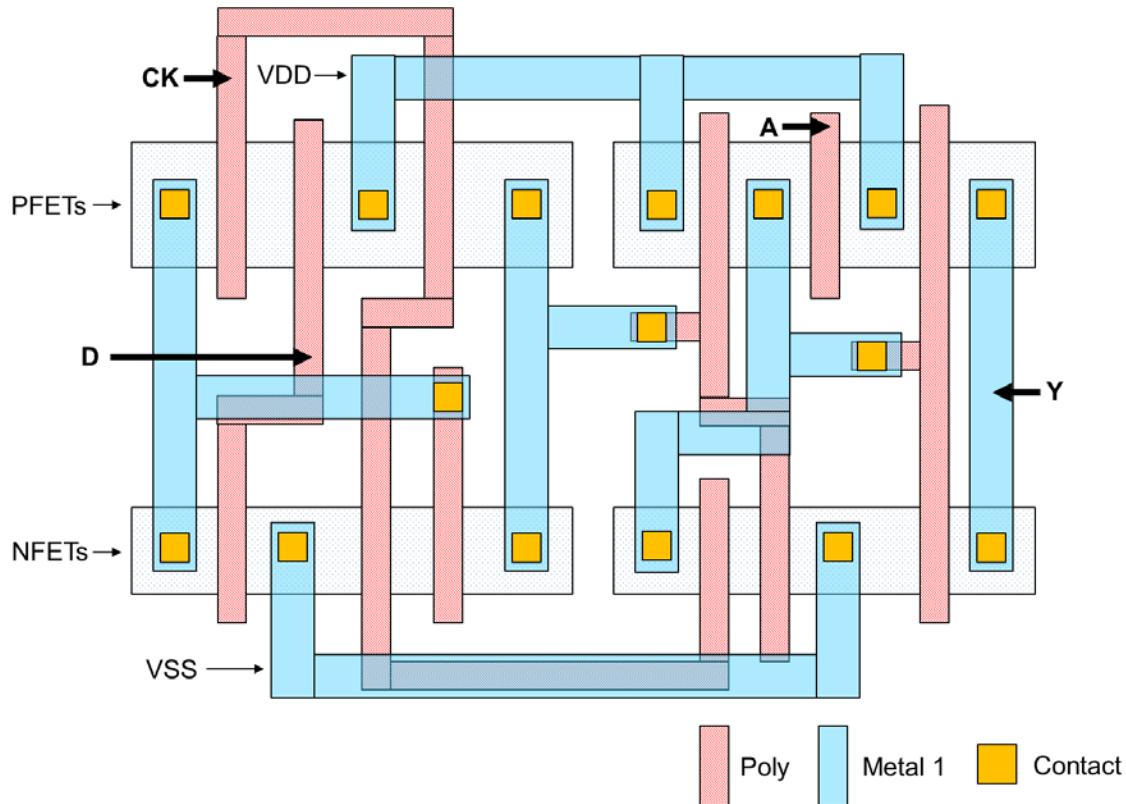
$$W' = 1 + \frac{10(a-1) - 10a}{(a-1)^2} = 0$$

$$a = 1 + \sqrt{10}, b = 3 + \frac{3}{\sqrt{10}}, c = 1 + \frac{1}{\sqrt{10}}, W = 11 + 2\sqrt{10}$$

(2) (12 points) Answer the following questions.

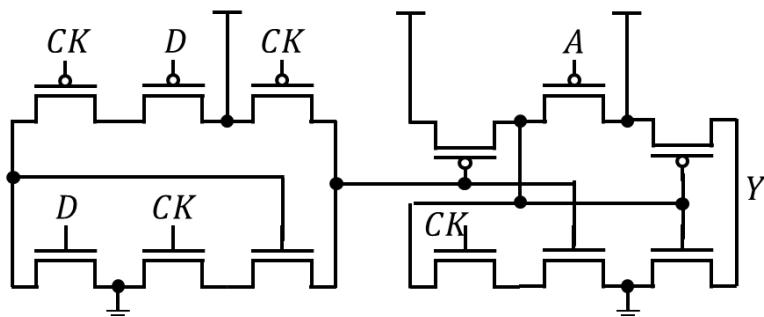
- (a) The optimal size of transistor A is greater than 4X  True /  False).
- (b) The optimal size of transistor B is greater than 4X ( T /  F).
- (c) The optimal size of transistor B is equal to the optimal size of transistor D  T /  F).
- (d) The optimal size of transistor E is greater than 2X ( T /  F).
- (e) The optimal size of transistor C is  $3 \times$  the optimal size of transistor E  T /  F).
- (f) The sum of the optimal widths of all the transistors is greater than or equal to 18X ( T /  F).

## Problem #5 (Layout, 20 points)



Signal input: A, D. Signal output: Y. Clock: CK

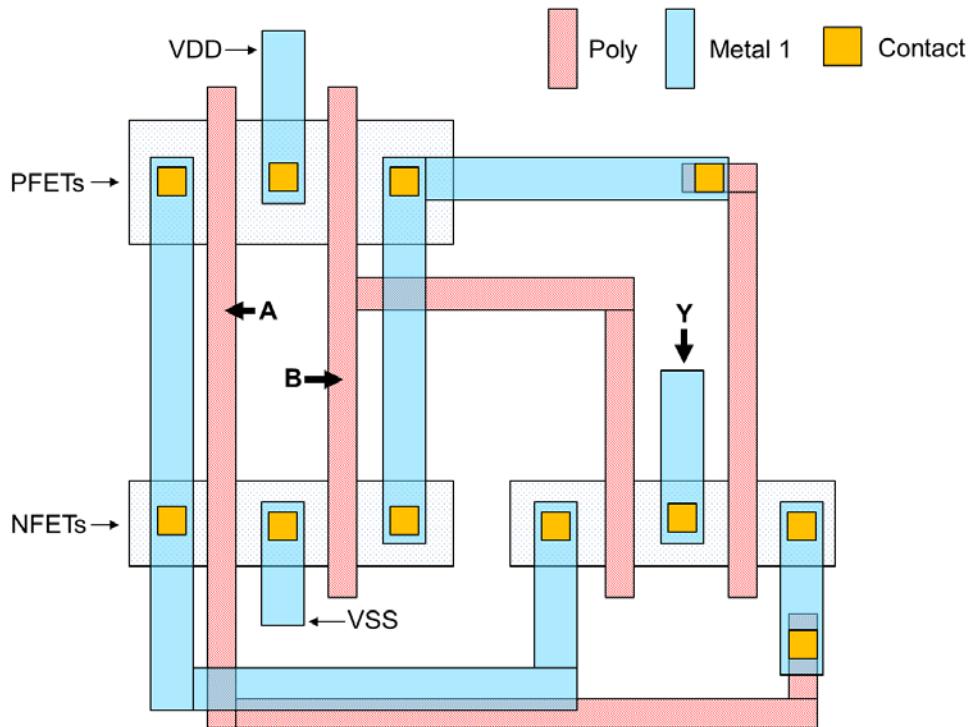
- 1) (10 points) Convert the layout into a transistor-level schematic.



- 2) (10 points) What is the function of the circuit?

This is a positive-edge-triggered D-F/F with an asynchronous active-low reset signal A.

## Problem #6 (Layout, 10 points)



Input: A, B

Output: Y

What is the function of this circuit?

It is an XOR gate.  $Y = A \oplus B$ .

**EE434**  
**ASIC and Digital Systems**

**Midterm Exam 2**

**Mar. 27, 2019. (4:10pm – 5pm)**

**Instructor: Dae Hyun Kim ([daehyun@eecs.wsu.edu](mailto:daehyun@eecs.wsu.edu))**

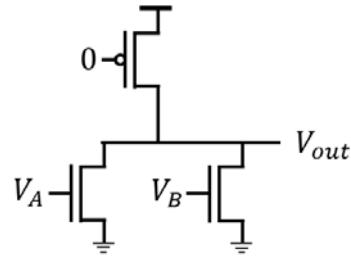
**Name:**

**WSU ID:**

Problem	Points	
1	10	
2	15	
3	20	
4	25	
5	20	
6	20	
Total	110	

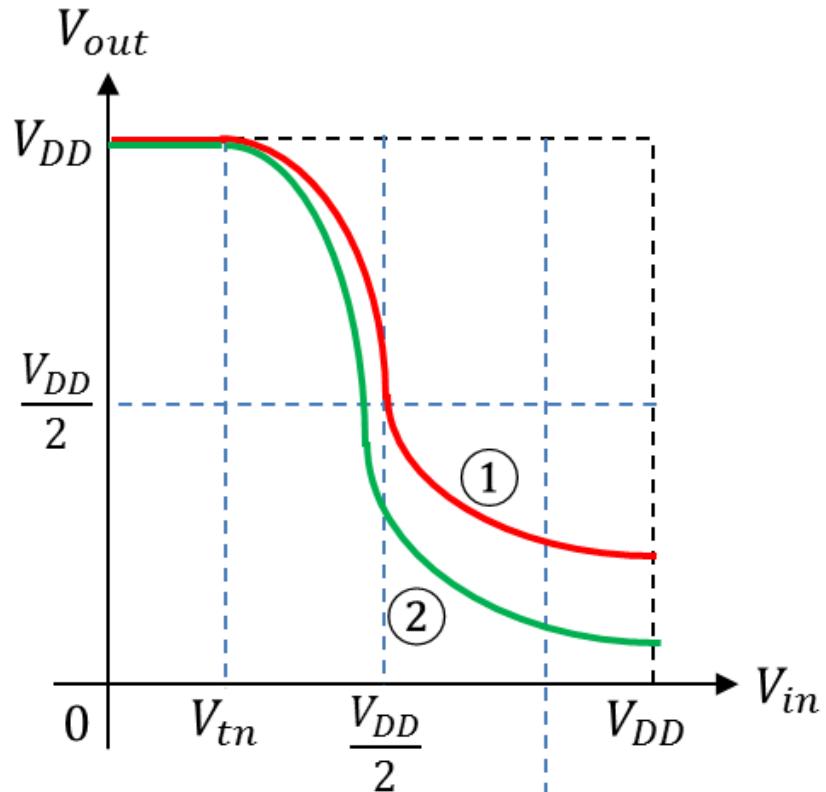
### Problem #1 (DC Analysis, 10 points)

Draw a DC characteristic curve for the following pseudo-NMOS two-input NOR gate. Just a rough sketch will be accepted.



(1) For input AB: 00 → 10

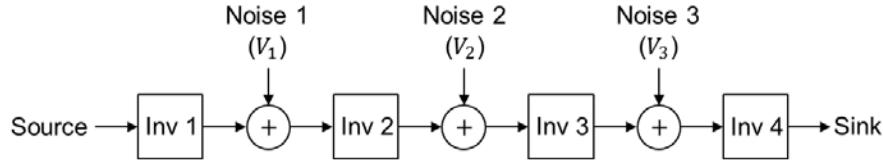
(2) For input AB: 00 → 11



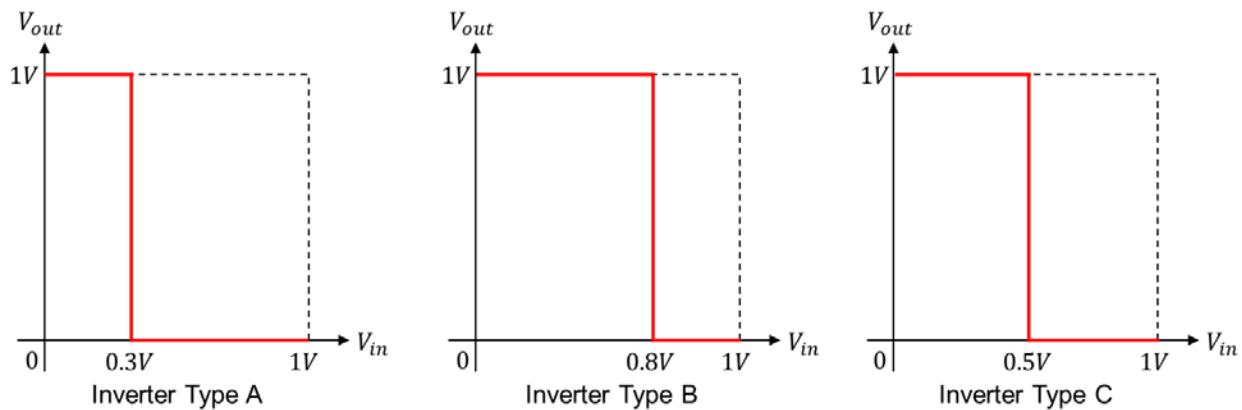
$$V_{DD} - |V_{tp}|$$

## Problem #2 (DC Analysis, 15 points)

The following figure shows a chain of four inverters with three noise sources. The value at the source node is 0V (for logic 0) or 1V (for logic 1).



The following shows the DC characteristic curves for three types of inverters (Type A, B, C).



(1) (5 points)  $|V_1| \leq 0.28V, |V_2| \leq 0.16V, |V_3| \leq 0.37V$ . If Inv 1: Type A, Inv 2: Type B, Inv 3: Type C, Inv 4: Type C, will the inverter chain work without logic inversion at the sink node? (Yes / No)

Output: Source (0V) → Inv 1 (1V) → Noise 1 (0.72V) → Inv 2 (1V) → Noise 2 (0.84V) → Inv 3 (0V) → Noise 3 (0.37V) → Inv 4 (1V), so logic inversion occurs.

(2) (5 points)  $|V_1| \leq 0.15V, |V_2| \leq 0.45V, |V_3| \leq 0.25V$ . If Inv 1: Type A, Inv 2: Type B, Inv 3: Type C, Inv 4: Type A, will the inverter chain work without logic inversion at the sink node? (Yes / No)

Output: Source (0V) → Inv 1 (1V) → Noise 1 (0.85V) → Inv 2 (0V) → Noise 2 (0.45V) → Inv 3 (1V) → Noise 3 (0.75V) → Inv 4 (0V)

Source (1V) → Inv 1 (0V) → Noise 1 (0.15V) → Inv 2 (1V) → Noise 2 (0.55V) → Inv 3 (0V) → Noise 3 (0.25V) → Inv 4 (1V)

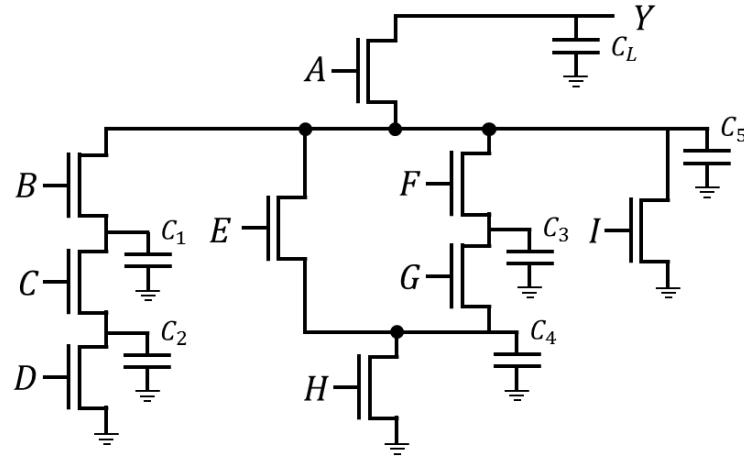
(3) (5 points)  $|V_1| \leq 0.12V$ ,  $|V_2| \leq 0.14V$ ,  $|V_3| \leq 0.34V$ . If Inv 1: Type C, Inv 2: Type A, Inv 3: Type B, Inv 4: Type A, will the inverter chain work without logic inversion at the sink node? (Yes / No)

Output: Source (0V) → Inv 1 (1V) → Noise 1 (0.88V) → Inv 2 (0V) → Noise 2 (0.14V) → Inv 3 (1V) → Noise 3 (0.66V) → Inv 4 (0V)

Source (1V) → Inv 1 (0V) → Noise 1 (0.12V) → Inv 2 (1V) → Noise 2 (0.86V) → Inv 3 (0V) → Noise 3 (0.34V) → Inv 4 (0V), so logic inversion occurs.

### Problem #3 (Delay, 20 points)

Calculate the Elmore delay of the circuit shown below for the following inputs.  $R_n$  is the resistance of an NFET.  $C_L$  is the load capacitance. All the other capacitances are parasitic capacitances.



1) ABCDEFGHI = 111100000

$$\tau = R_n C_L + R_n(C_5 + C_L) + R_n(C_1 + C_5 + C_L) + R_n(C_1 + C_2 + C_5 + C_L)$$

2) ABCDEFGHI = 111010010

$$\tau = R_n C_L + R_n(C_1 + C_2 + C_5 + C_L) + R_n(C_1 + C_2 + C_4 + C_5 + C_L)$$

3) ABCDEFGHI = 101101101

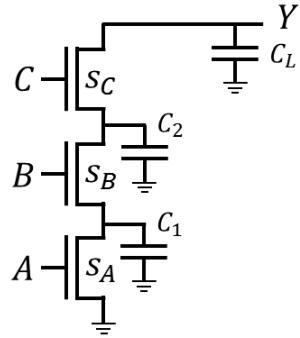
$$\tau = R_n C_L + R_n(C_3 + C_4 + C_5 + C_L)$$

4) ABCDEFGHI = 111010110

$$\tau = R_n C_L + R_n(C_1 + C_2 + C_5 + C_L) + R_n(C_1 + C_2 + C_3 + C_4 + C_5 + C_L)$$

## Problem #4 (Transistor Sizing, 25 points)

The following figure shows the NFET network of a three-input NAND gate.  $C_L$ : Load capacitance.  $C_1, C_2$ : Parasitic capacitance.  $R_n$ : The resistance of a 1X NFET. Assume that  $C_1, C_2, C_L$  are independent of the widths of the NFETs.  $s_A, s_B, s_C$ : The width of NFET A, B, C, respectively. “Optimal transistor sizes” means the widths of the transistors that satisfy a given timing constraint and minimize the sum of the widths.



(1) (7 points) Express the Elmore delay of the NAND gate as a function of  $R_n, s_A, s_B, s_C, C_1, C_2, C_L$ .

$$\tau = \frac{R_n}{s_C} C_L + \frac{R_n}{s_B} (C_2 + C_L) + \frac{R_n}{s_A} (C_1 + C_2 + C_L)$$

(2) (18 points) Answer the following questions.

(a) If  $C_1$  increases, we should increase the width of transistor A for optimal transistor sizing (True/ False).

(b) If  $C_2$  increases, we should increase the width of transistor B for optimal transistor sizing (True/ False).

(c) If  $C_L$  increases, we should increase the width of transistor C for optimal transistor sizing (True/ False).

(d) If  $C_2$  increases, we should increase the width of transistor  $A$  for optimal transistor sizing (True/ False).

(e) If  $C_2$  increases, we should increase the width of transistor  $B$  for optimal transistor sizing (True/ False).

(f) If  $C_2$  increases, we should increase the width of transistor  $C$  for optimal transistor sizing (True/ False).

(g) If  $C_1$  increases, we should increase the width of transistor  $A$  for optimal transistor sizing (True/ False).

(h) If  $C_1$  increases, we should increase the width of transistor  $B$  for optimal transistor sizing (True/ False).

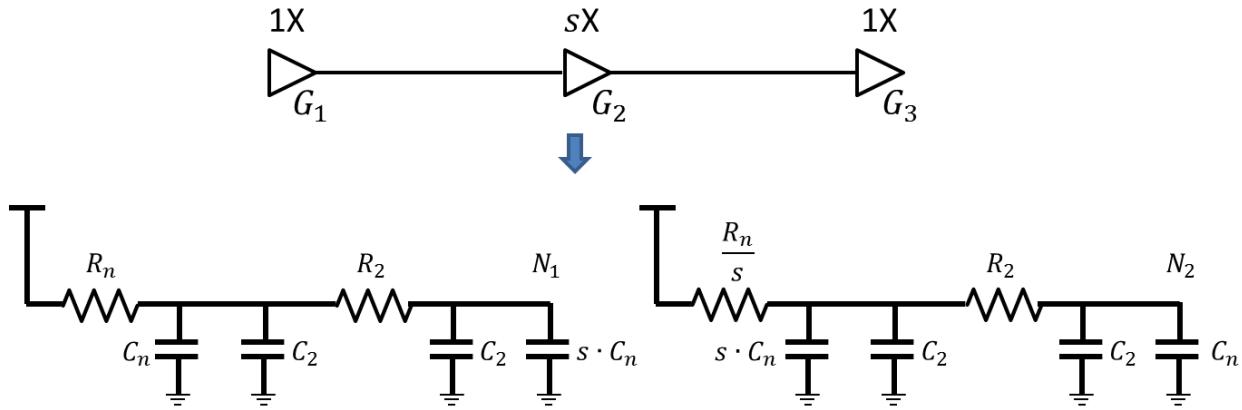
(i) If  $C_L$  increases, we should increase the width of transistor  $C$  for optimal transistor sizing (True/ False).

We can answer the above questions by deeply investigating the Elmore delay equation. However, we can also optimally get the result as follows.

Minimize  $s_A + s_B + s_C$ . Let  $x_i = 1/s_i$ . Then, minimize  $W = \frac{1}{x_A} + \frac{1}{x_B} + \frac{1}{x_C}$  for  $\tau = R_n C_L x_C + R_n(C_2 + C_L)x_B + R_n(C_1 + C_2 + C_L)x_A = t$  where  $t$  is a given delay constraint. Assuming  $x_A$  and  $x_B$  are two independent variables,  $\frac{\partial \tau}{\partial x_A} = R_n(C_1 + C_2 + C_L) + R_n C_L \frac{\partial x_C}{\partial x_A} = 0$ ,  $\frac{\partial \tau}{\partial x_B} = R_n(C_2 + C_L) + R_n C_L \frac{\partial x_C}{\partial x_B} = 0$ . Now,  $\frac{\partial W}{\partial x_A} = -\frac{1}{x_A^2} - \frac{1}{x_C^2} \cdot \frac{\partial x_C}{\partial x_A} = 0$ ,  $\frac{\partial W}{\partial x_B} = -\frac{1}{x_B^2} - \frac{1}{x_C^2} \cdot \frac{\partial x_C}{\partial x_B} = 0$ . Thus,  $x_C = x_A \sqrt{\frac{C_1+C_2+C_L}{C_L}}$ ,  $x_C = x_B \sqrt{\frac{C_2+C_L}{C_L}}$ . Thus,  $R_n C_L x_C + R_n(C_2 + C_L)x_C \sqrt{\frac{C_L}{C_2+C_L}} + R_n(C_1 + C_2 + C_L)x_C \sqrt{\frac{C_L}{C_1+C_2+C_L}} = t$ , so  $x_C = \frac{t}{R_n(C_L + \sqrt{C_L(C_2+C_L)} + \sqrt{C_L(C_1+C_2+C_L)})}$ .  $x_A = \frac{t}{R_n((C_1+C_2+C_L)+\sqrt{(C_2+C_L)(C_1+C_2+C_L)}+\sqrt{C_L(C_1+C_2+C_L)})}$ .  $x_B = \frac{t}{R_n((C_2+C_L)+\sqrt{(C_2+C_L)(C_1+C_2+C_L)}+\sqrt{C_L(C_2+C_L)})}$ .

Thus,  $s_A = \frac{1}{t} \cdot R_n \cdot \left[ (C_1 + C_2 + C_L) + \sqrt{(C_2 + C_L)(C_1 + C_2 + C_L)} + \sqrt{C_L(C_1 + C_2 + C_L)} \right]$ ,  
 $s_B = \frac{1}{t} \cdot R_n \cdot \left[ (C_2 + C_L) + \sqrt{(C_2 + C_L)(C_1 + C_2 + C_L)} + \sqrt{C_L(C_2 + C_L)} \right]$ ,  $s_C = \frac{1}{t} \cdot R_n \cdot \left[ C_L + \sqrt{C_L(C_2 + C_L)} + \sqrt{C_L(C_1 + C_2 + C_L)} \right]$ .

## Problem #5 (Interconnect, 20 points)



The figure shown above simulates the delay of the signal path from gate \$G\_1\$ to gate \$G\_3\$. For the delay calculation, we use the two RC trees. The left-hand one is for the first half (from \$G\_1\$ to \$G\_2\$) of the path and the right-hand one is for the second half (from \$G\_2\$ to \$G\_3\$). The total delay is estimated by the sum of the delays at node \$N\_1\$ and \$N\_2\$.

- \$G\_1\$ and \$G\_3\$ are 1X cells, so their output resistances, output capacitances, and input capacitances are \$R\_n\$, \$C\_n\$, and \$C\_n\$, respectively.
- \$G\_2\$ is upsized to \$sX\$, so its output resistance, output capacitance, and input capacitance are \$\frac{R\_n}{s}\$, \$sC\_n\$, and \$sC\_n\$, respectively.
- \$R\_2\$ and \$C\_2\$ model the wires.
- \$R\_n, C\_n, C\_2, R\_2\$ are constants. \$s\$ is the only variable in this problem.

(1) (7 points) Express the total delay as a function of \$s, R\_n, C\_n, R\_2\$, and \$C\_2\$.

$$\tau = R_2(C_2 + sC_n) + R_n(C_n + 2C_2 + sC_n) + R_2(C_2 + C_n) + \frac{R_n}{s}(sC_n + 2C_2 + C_n)$$

(2) (5 points) Express optimal \$s\$ minimizing the total delay as a function of \$R\_n, C\_n, R\_2\$, and \$C\_2\$.

$$\frac{d\tau}{ds} = R_2C_n + R_nC_n - \frac{R_n(2C_2+C_n)}{s^2} = 0, \text{ so } s = \sqrt{\frac{R_n(2C_2+C_n)}{C_n(R_2+R_n)}} = \sqrt{\frac{1+2\frac{C_2}{C_n}}{1+\frac{R_2}{R_n}}}$$

(3) (8 points) Answer the following questions.

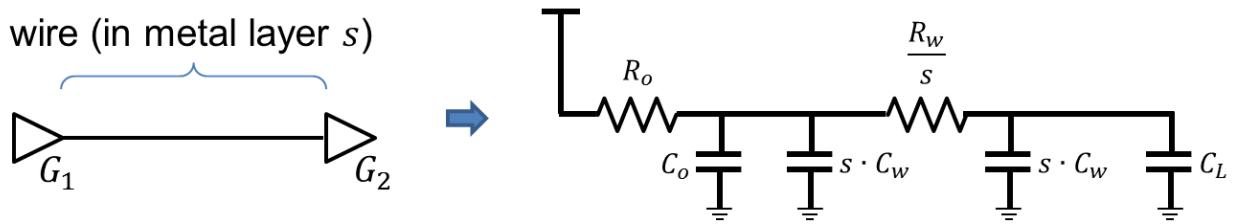
(a) If  $R_n$  increases, we should increase  $s$  for minimum delay. (True / False)

(b) If  $C_n$  increases, we should increase  $s$  for minimum delay. (True / False)

(c) If  $R_2$  increases, we should increase  $s$  for minimum delay. (True / False)

(d) If  $C_2$  increases, we should increase  $s$  for minimum delay. (True / False)

## Problem #6 (Interconnect, 20 points)



The figure shown above simulates the delay of a net routed in metal layer  $s$ .

- $R_o, C_o$ : Output resistance and capacitance of  $G_1$ , respectively (constants).
- $C_L$ : Load capacitance (constant)
- $R_w, C_w$ : Resistance and capacitance of the wire, respectively (constants).
- $s$ : Metal layer (variable)

When we route the net in a layout, we can use any metal layer. If we select an upper layer (e.g., Metal layer 10) for the routing of the net, the resistance of the wire goes down, but the capacitance of the wire goes up. Similarly, if we select a lower layer (e.g., Metal layer 3), the resistance of the wire goes up, but the capacitance of the wire goes down. The RC tree provides a very simple model for the delay estimation.

(1) (5 points) Express the delay from  $G_1$  to  $G_2$  as a function of  $R_o, C_o, R_w, C_w, C_L$ , and  $s$ .

$$\tau = \frac{R_w}{s} \cdot (s \cdot C_w + C_L) + R_o(C_o + 2sC_w + C_L) = (R_w C_w + R_o C_o + R_o C_L) + 2R_o C_w s + \frac{R_w C_L}{s}$$

(2) (5 points) Find  $s$  minimizing the delay (express  $s$  as a function of  $R_o, C_o, R_w, C_w$ , and  $C_L$ ).

$$\frac{d\tau}{ds} = 2R_o C_w - \frac{R_w C_L}{s^2} = 0, \text{ so } s = \sqrt{\frac{R_w C_L}{2R_o C_w}}$$

(3) (8 points) Answer the following questions.

(a) If  $R_w$  increases, we should increase  $s$  to minimize the delay. (True / False)

(b) If  $R_o$  increases, we should increase  $s$  to minimize the delay. (True / False)

(c) If  $C_w$  increases, we should increase  $s$  to minimize the delay. (True / False)

(d) If  $C_o$  increases, we should increase  $s$  to minimize the delay. (True / False)

(e) If  $C_L$  increases, we should increase  $s$  to minimize the delay. (True / False)

**EE434**  
**ASIC and Digital Systems**

**Final Exam**

**May 5, 2020. (8am – 10am)**

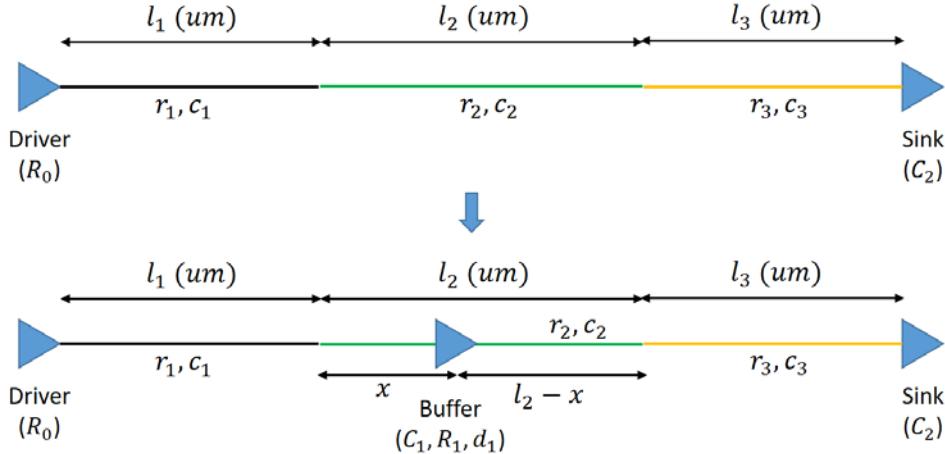
**Instructor: Dae Hyun Kim ([daehyun@eecs.wsu.edu](mailto:daehyun@eecs.wsu.edu))**

**Name:**

**WSU ID:**

Problem	Points	
1	30	
2	40	
3	30	
4	50	
5	20	
6	50	
7	40	
Total	260	

## Problem #1 (Interconnects, 30 points)



The figure shows a net composed of three wire segments. The length and unit wire R and C of section  $k$  ( $k = 1, 2, 3$ ) are  $l_k$ ,  $r_k$ , and  $c_k$ , respectively as shown in the figure. The output resistance of the driver is  $R_0$ , the input capacitance of the sink is  $C_2$ , and the output resistance, input capacitance, and the delay of a buffer is  $R_1$ ,  $C_1$ , and  $d_1$ , respectively. We are going to insert a buffer into the second section. The first and third sections are not bufferable.

(1) Find the optimal location for the buffer insertion (express the location as a function of the constants). (10 points)

Let's assume that the distance from the left side of the second section to the buffer is  $x$  (um) (so,  $0 \leq x \leq l_2$ ). Then, the delay of the left side of the buffer is

$$\tau_1 = R_0(c_1 l_1 + c_2 x + C_1) + r_1 l_1(c_2 x + C_1) + \frac{1}{2}r_1 c_1 l_1^2 + r_2 x C_1 + \frac{1}{2}r_2 c_2 x^2$$

The delay of the right side is

$$\begin{aligned} \tau_2 &= R_1(c_2(l_2 - x) + c_3 l_3 + C_2) + r_2(l_2 - x)(c_3 l_3 + C_2) + \frac{1}{2}r_2 c_2(l_2 - x)^2 + r_3 l_3 C_2 \\ &\quad + \frac{1}{2}r_3 c_3 l_3^2 \end{aligned}$$

and the total delay is

$$\tau = \tau_1 + d_1 + \tau_2$$

$$\frac{d\tau}{dx} = R_0 c_2 + r_1 l_1 c_2 + r_2 C_1 + r_2 c_2 x - R_1 c_2 - r_2(c_3 l_3 + C_2) - r_2 c_2(l_2 - x) = 0$$

$$x(2r_2 c_2) = (R_1 - R_0)c_2 + (C_2 - C_1)r_2 - r_1 c_2 l_1 + r_2 c_3 l_3 + r_2 c_2 l_2$$

$$\therefore x = \frac{1}{2} \left\{ \frac{R_1 - R_0}{r_2} + \frac{C_2 - C_1}{c_2} - \frac{r_1}{r_2} l_1 + l_2 + \frac{c_3}{c_2} l_3 \right\}$$

(2) Answer the following questions. (Correct: +2 points. Wrong: -1 point. No answer: 0.)

(1) If  $R_0 = R_1, C_1 = C_2, r_1 = r_2 = r_3, c_1 = c_2 = c_3$ , and  $l_1 = l_2 = l_3$ , then  $x = \frac{1}{2}l_2$ . (**True** / False)

(2) If  $R_0 < R_1, C_1 = C_2, r_1 = r_2 = r_3, c_1 = c_2 = c_3$ , and  $l_1 = l_2 = l_3$ , then  $x < \frac{1}{2}l_2$ . (True / **False**)

(3) If  $R_0 = R_1, C_1 < C_2, r_1 = r_2 = r_3, c_1 = c_2 = c_3$ , and  $l_1 = l_2 = l_3$ , then  $x < \frac{1}{2}l_2$ . (True / **False**)

(4) If  $R_0 = R_1, C_1 = C_2, r_1 < r_2 = r_3, c_1 = c_2 = c_3$ , and  $l_1 = l_2 = l_3$ , then  $x < \frac{1}{2}l_2$ . (True / **False**)

(5) If  $R_0 = R_1, C_1 = C_2, r_1 = r_2 > r_3, c_1 = c_2 = c_3$ , and  $l_1 = l_2 = l_3$ , then  $x < \frac{1}{2}l_2$ . (True / **False**)

(6) If  $R_0 = R_1, C_1 = C_2, r_1 = r_2 = r_3, c_1 < c_2 = c_3$ , and  $l_1 = l_2 = l_3$ , then  $x < \frac{1}{2}l_2$ . (True / **False**)

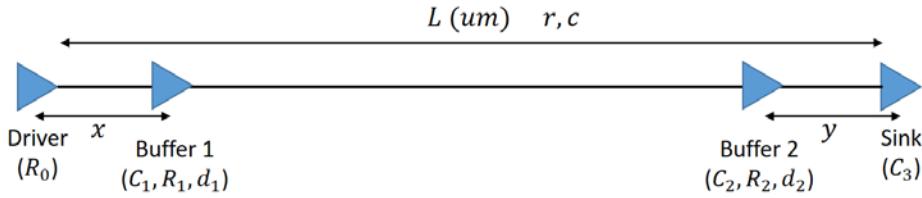
(7) If  $R_0 = R_1, C_1 = C_2, r_1 = r_2 = r_3, c_1 = c_2 > c_3$ , and  $l_1 = l_2 = l_3$ , then  $x < \frac{1}{2}l_2$ . (**True** / False)

(8) If  $R_0 = R_1, C_1 = C_2, r_1 = r_2 = r_3, c_1 = c_2 = c_3$ , and  $l_1 < l_2 = l_3$ , then  $x < \frac{1}{2}l_2$ . (True / **False**)

(9) If  $R_0 = R_1, C_1 = C_2, r_1 = r_2 = r_3, c_1 = c_2 = c_3$ , and  $l_1 = l_2 > l_3$ , then  $x < \frac{1}{2}l_2$ . (**True** / False)

(10) If  $r_3$  increases,  $x$  increases (i.e., the optimal buffer location is shifted to the right.) (True / **False**)

## Problem #2 (Interconnects, 40 points)



The figure shows a net composed of a long wire whose unit resistance and capacitance are  $r$  and  $c$ , respectively. We want to insert two buffers, Buffer 1 and Buffer 2 ( $0 \leq x$ ,  $0 \leq y$ ,  $x + y \leq L$ ). The output resistance ( $R_{\#}$ ), input capacitance ( $C_{\#}$ ), and internal delay ( $d_{\#}$ ) of each cell are shown above.

(a) Find the optimal locations for Buffer 1 and Buffer 2 (express  $x$  (and also  $y$ ) as a function of the constants). Hint: Express the total delay as a function of  $x$  and  $y$ . Then, differentiate it w.r.t.  $x$  and set it to zero. Differentiate it w.r.t.  $y$  and set it to zero. Then, you get two equations for two variables,  $x$  and  $y$ . Solve it. (12 points)

$$\tau_1 = R_0(C_1 + cx) + rxC_1 + \frac{1}{2}rcx^2$$

$$\tau_2 = R_1\{C_2 + c(L - x - y)\} + r(L - x - y)C_2 + \frac{1}{2}rc(L - x - y)^2$$

$$\tau_3 = R_2(C_3 + cy) + ryC_3 + \frac{1}{2}rcy^2$$

$$\text{Total delay } \tau = \tau_1 + d_1 + \tau_2 + d_2 + \tau_3$$

$$\frac{\partial \tau}{\partial x} = R_0c + rC_1 + rcx - R_1c - rC_2 - rc(L - x - y) = 0$$

$$\frac{\partial \tau}{\partial y} = R_2c + rC_3 + rcy - R_1c - rC_2 - rc(L - x - y) = 0$$

$$2rcx + rcy = rcL + (R_1 - R_0)c + (C_2 - C_1)r$$

$$rcx + 2rcy = rcL + (R_1 - R_2)c + (C_2 - C_3)r$$

$$\therefore x = \frac{1}{3}\left\{L + \frac{R_1 + R_2 - 2R_0}{r} + \frac{C_2 + C_3 - 2C_1}{c}\right\}$$

$$y = \frac{1}{3}\left\{L + \frac{R_0 + R_1 - 2R_2}{r} + \frac{C_1 + C_2 - 2C_3}{c}\right\}$$

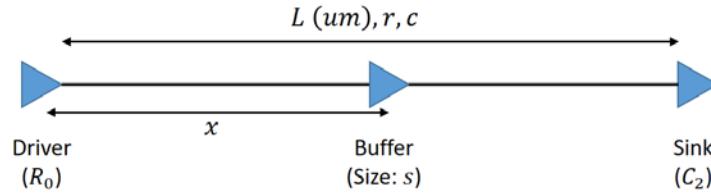
(b) Answer the following questions. (Correct: +2 points. Wrong: -1 point. No answer: 0.)

Notice that if  $x$  increases, Buffer 1 is shifted to the right.

However, if  $y$  increases, Buffer 2 is shifted to the left.

- (1) If  $R_0 = R_1 = R_2$  and  $C_1 = C_2 = C_3$ , then  $x = \frac{1}{3}L$ . (**True** / False)
- (2) If  $R_0 = R_1 = R_2$  and  $C_1 = C_2 = C_3$ , then  $y = \frac{1}{3}L$ . (**True** / False)
- (3) If  $R_0$  increases,  $x$  increases. (True / **False**)
- (4) If  $R_0$  increases,  $y$  increases. (**True** / False)
- (5) If  $R_1$  increases,  $x$  increases. (**True** / False)
- (6) If  $R_1$  increases,  $y$  increases. (**True** / False)
- (7) If  $R_2$  increases,  $x$  increases. (**True** / False)
- (8) If  $R_2$  increases,  $y$  increases. (True / **False**)
- (9) If  $C_1$  increases,  $x$  increases. (True / **False**)
- (10) If  $C_1$  increases,  $y$  increases. (**True** / False)
- (11) If  $C_2$  increases,  $x$  increases. (**True** / False)
- (12) If  $C_2$  increases,  $y$  increases. (**True** / False)
- (13) If  $C_3$  increases,  $x$  increases. (**True** / False)
- (14) If  $C_3$  increases,  $y$  increases. (True / **False**)

### Problem #3 (Interconnects, 30 points)



The figure shows a long wire whose length, unit resistance and capacitance are  $L$ ,  $r$ , and  $c$ , respectively. We want to insert a buffer ( $0 \leq x \leq L$ ). The size of the buffer is  $s$  and its output resistance, input capacitance, and internal delay are as follows:

- Output resistance:  $\frac{R}{s}$  (where  $R$  is a constant)
- Input capacitance:  $sP$  (where  $P$  is a constant)
- Internal delay:  $sD$  (where  $D$  is a constant)

Notice that there are two variables,  $x$  and  $s$ .

(a) Express the total delay as a function of the variables and the constants. (7 points)

$$\tau = R_0(cx + sP) + rx(sP) + \frac{1}{2}rcx^2 + sD + \frac{R}{s}\{c(L - x) + C_2\} + r(L - x)C_2 + \frac{1}{2}rc(L - x)^2$$

(b) Assuming the location ( $x$ ) is given (i.e., you can treat it as a constant.), find the optimal size ( $s$ ) of the buffer, i.e., express the optimal value of  $s$  as a function of the constants and  $x$ . (7 points)

$$\frac{\partial \tau}{\partial s} = R_0P + rxP + D - \frac{R\{c(L - x) + C_2\}}{s^2} = 0$$

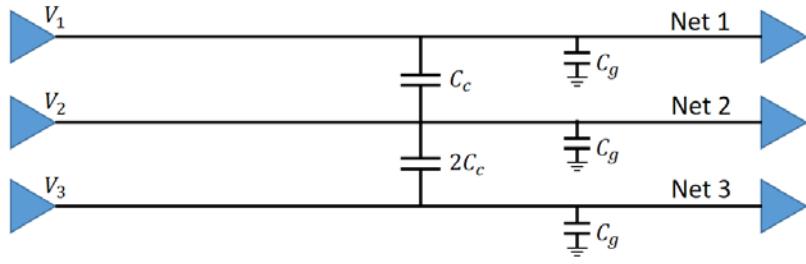
$$s = \sqrt{\frac{R\{c(L - x) + C_2\}}{(R_0 + rx)P + D}}$$

(c) Answer the following questions. (Correct: +2 points. Wrong: -1 point. No answer: 0.)

- (1) If  $x$  increases,  $s$  increases. (True / False)
- (2) If  $R$  increases,  $s$  increases. (True / False)
- (3) If  $c$  increases,  $s$  increases. (True / False)
- (4) If  $C_2$  increases,  $s$  increases. (True / False)
- (5) If  $R_0$  increases,  $s$  increases. (True / False)
- (6) If  $r$  increases,  $s$  increases. (True / False)
- (7) If  $P$  increases,  $s$  increases. (True / False)

(8) If  $D$  increases,  $s$  increases. (True / **False**)

## Problem #4 (Coupling, 50 points)



The figure shows three parallel nets. The coupling capacitance between Net 1 and Net 2 is  $C_c$ . The coupling capacitance between Net 2 and Net 3 is  $2C_c$ . The following shows a three-bit binary encoding technique for four decimal numbers (0, 1, 2, 3).

Value	Encoding
0	000
1	010
2	101
3	111

(1) Calculate the total effective capacitance of Net 1 for all possible transitions ( $0 \rightarrow 1$ ,  $0 \rightarrow 2$ ,  $0 \rightarrow 3$ ,  $1 \rightarrow 0$ ,  $1 \rightarrow 2$ ,  $1 \rightarrow 3$ ,  $2 \rightarrow 0$ ,  $2 \rightarrow 1$ ,  $2 \rightarrow 3$ ,  $3 \rightarrow 0$ ,  $3 \rightarrow 1$ ,  $3 \rightarrow 2$ ) for the encoding technique, i.e., calculate the sum of the effective capacitances of Net 1 for all possible transitions. (10 points)

$0 \rightarrow 1: 0, 0 \rightarrow 2: C_g + C_c, 0 \rightarrow 3: C_g, 1 \rightarrow 0: 0, 1 \rightarrow 2: C_g + 2C_c, 1 \rightarrow 3: C_g + C_c, 2 \rightarrow 0: C_g + C_c, 2 \rightarrow 1: C_g + 2C_c, 2 \rightarrow 3: 0, 3 \rightarrow 0: C_g, 3 \rightarrow 1: C_g + C_c, 3 \rightarrow 2: 0.$  (Notice that  $a \rightarrow b$  and  $b \rightarrow a$  have the same effective capacitance.)

Total effective capacitance:  $8C_g + 8C_c$

(2) Repeat it for Net 2. (10 points)

$0 \rightarrow 1: C_g + 3C_c, 0 \rightarrow 2: 0, 0 \rightarrow 3: C_g, 1 \rightarrow 2: C_g + 6C_c, 1 \rightarrow 3: 0, 2 \rightarrow 3: C_g + 3C_c.$

Total effective capacitance:  $8C_g + 24C_c$

(3) Repeat it for Net 3. (10 points)

$0 \rightarrow 1: 0, 0 \rightarrow 2: C_g + C_c, 0 \rightarrow 3: C_g, 1 \rightarrow 2: C_g + 4C_c, 1 \rightarrow 3: C_g + 2C_c, 2 \rightarrow 3: 0$

Total effective capacitance:  $8C_g + 14C_c$

Thus, the total effective capacitance for all the nets for all the possible transitions is

$$24C_g + 46C_c$$

- (4) Find a new three-bit encoding of the four values (0, 1, 2, 3) that minimizes the total effective capacitance of the three nets for all possible transitions (Assume  $C_g = C_c$ ). (20 points)

Value	Encoding
0	000
1	100
2	110
3	111

In this case,

$$0 \rightarrow 1: C_g + C_c \text{ (Net 1)} + 0 \text{ (Net 2)} + 0 \text{ (Net 3)} = C_g + C_c$$

$$0 \rightarrow 2: C_g \text{ (Net 1)} + C_g + 2C_c \text{ (Net 2)} + 0 \text{ (Net 3)} = 2C_g + 2C_c$$

$$0 \rightarrow 3: C_g \text{ (Net 1)} + C_g \text{ (Net 2)} + C_g \text{ (Net 3)} = 3C_g$$

$$1 \rightarrow 2: 0 \text{ (Net 1)} + C_g + 3C_c \text{ (Net 2)} + 0 \text{ (Net 3)} = C_g + 3C_c$$

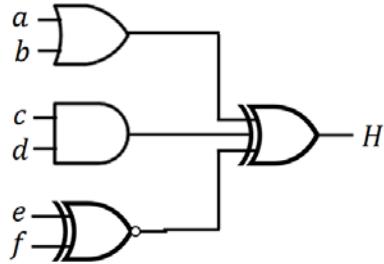
$$1 \rightarrow 3: 0 \text{ (Net 1)} + C_g + C_c \text{ (Net 2)} + C_g \text{ (Net 3)} = 2C_g + C_c$$

$$2 \rightarrow 3: 0 \text{ (Net 1)} + 0 \text{ (Net 2)} + C_g + 2C_c \text{ (Net 3)} = C_g + 2C_c$$

Total effective capacitance (for Net 1, 2, and 3):  $20C_g + 18C_c = 38C_g$

## Problem #5 (Testing, 20 points)

The following shows  $H = (a + b) \oplus (c \cdot d) \oplus (\overline{e \oplus f})$ . Answer the following questions.



(1) Find all input vectors that can detect a s-a-0 fault at input  $a$ . (10 points)

$$H \oplus H_f = \{(a + b) \oplus (c \cdot d) \oplus (\overline{e \oplus f})\} \oplus \{(b) \oplus (c \cdot d) \oplus (\overline{e \oplus f})\} = 1$$

Thus,  $a$  should be 1. Then,

$$H(a=1) \oplus H_f = \{1 \oplus (c \cdot d) \oplus (\overline{e \oplus f})\} \oplus \{(b) \oplus (c \cdot d) \oplus (\overline{e \oplus f})\} = 1$$

Thus,  $b$  should be 0. Then,

$$H(ab=10) \oplus H_f = \{1 \oplus (c \cdot d) \oplus (\overline{e \oplus f})\} \oplus \{(c \cdot d) \oplus (\overline{e \oplus f})\} = 1$$

Now, it is always true regardless of  $c, d, e, f$ .

Answer:  $(abcdef) = (10XXXX)$  (where X is a don't care).

(2) Find all input vectors that can detect a s-a-1 fault at input  $e$ . (10 points)

$$H \oplus H_f = \{(a + b) \oplus (c \cdot d) \oplus (\overline{e \oplus f})\} \oplus \{(a + b) \oplus (c \cdot d) \oplus (\overline{1 \oplus f})\} = 1$$

$$H \oplus H_f = \{(a + b) \oplus (c \cdot d) \oplus (\overline{e \oplus f})\} \oplus \{(a + b) \oplus (c \cdot d) \oplus f\} = 1$$

Thus,  $e$  should be 0. Then,

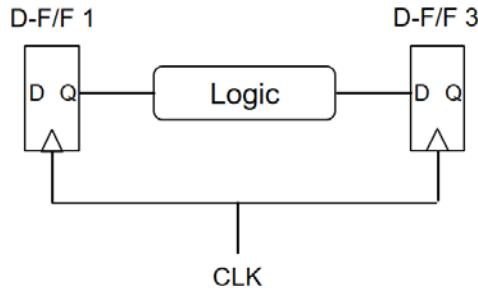
$$H(e=0) \oplus H_f = \{(a + b) \oplus (c \cdot d) \oplus (\bar{f})\} \oplus \{(a + b) \oplus (c \cdot d) \oplus f\} = 1$$

Now, this is always true for any  $(a, b, c, d, f)$ .

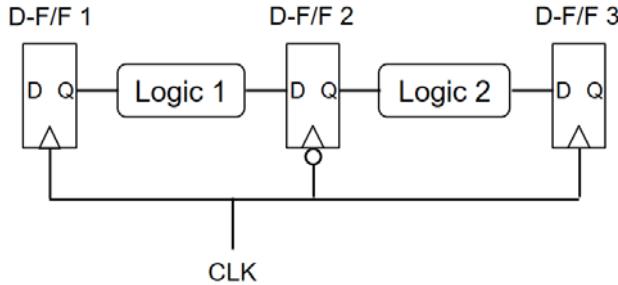
Answer:  $(abcdef) = (XXXX0X)$

## Problem #6 (Static Timing Analysis, 50 points)

The following shows two pipelining methodologies. Figure (a) shows a logic in a single pipeline stage. Figure (b) shows a partitioned version in which two pipeline stages are cascaded and the flip-flop in the middle is negative-edge-triggered.



(a)



(b)

### Parameters

- $d_1, d_2, d_3$ : The delay from the clock source to Flip-Flop 1, 2, 3
- $c_1, c_2, c_3$ : Clock-to-Q delay ( $T_{CQ}$ ) of Flip-Flop 1, 2, 3
- $s_1, s_2, s_3$ : Setup time of Flip-Flop 1, 2, 3
- $T_A, T_B$  : Clock period for Figure (a) and (b)
- $N (\gg 1)$ : # instructions to execute
- $T_L$ : The logic delay in Figure (a). In Figure (b), the delay of each logic is  $\frac{T_L}{2}$ .
- Execution time of the system in Figure (a):  $(N + 1) \cdot T_A$
- Execution time of the system in Figure (b):  $(N + 3) \cdot T_B$

For the clock periods, we use the minimum clock periods that satisfy all the setup time constraints. Notice that the clock duty cycle is 50%.

- (1) The system in Figure (a) has one setup time constraint. Show the inequality. (10 points)

$$d_1 + c_1 + T_L \leq d_3 + T_A - s_3$$

(2) The system in Figure (b) has two setup time constraints. Show the inequalities. (16 points)

$$d_1 + c_1 + \frac{T_L}{2} \leq d_2 + \frac{T_B}{2} - s_2$$

$$d_2 + \frac{T_B}{2} + c_2 + \frac{T_L}{2} \leq d_3 + T_B - s_3$$

(3) Now, you have execution times for Figure (a) and (b). Answer the following questions. (24 points)

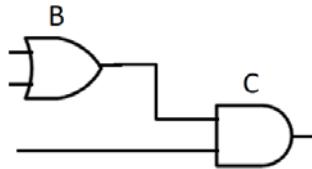
For Figure (a),  $T_A = d_1 + c_1 + T_L - d_3 + s_3$

For Figure (b),  $T_B = d_1 - d_3 + c_1 + c_2 + T_L + s_2 + s_3$

- (a) For Figure (a): If  $d_1$  increases, the execution time increases (**True** / False).
- (b) For Figure (a): If  $c_1$  increases, the execution time increases (**True** / False).
- (c) For Figure (a): If  $T_L$  increases, the execution time increases (**True** / False).
- (d) For Figure (a): If  $d_3$  increases, the execution time increases (**True** / **False**).
- (e) For Figure (a): If  $s_3$  increases, the execution time increases (**True** / False).
- (f) For Figure (b): If  $d_1$  increases, the execution time increases (**True** / False).
- (g) For Figure (b): If  $c_1$  increases, the execution time increases (**True** / False).
- (h) For Figure (b): If  $T_L$  increases, the execution time increases (**True** / False).
- (i) For Figure (b): If  $d_3$  increases, the execution time increases (**True** / **False**).
- (j) For Figure (b): If  $s_2$  increases, the execution time increases (**True** / False).
- (k) For Figure (b): If  $s_3$  increases, the execution time increases (**True** / False).
- (l) For Figure (b): If  $c_2$  increases, the execution time increases (**True** / False).

## Problem #7 (Static Timing Analysis, 40 points)

The following shows two gates B and C, which are in the middle of a circuit.



"A gate has a (setup or hold) time violation" means that at least one of the paths going through the gate violates given (setup or hold) time constraints.

- (1) Is it possible that gate B has a setup-time violation and gate C has a setup-time violation? Explain why.

Yes. The violation path goes through both B and C.

- (2) Is it possible that gate B has a setup-time violation and gate C has a hold-time violation? Explain why.

Yes. gate B has a large delay, so it has a setup-time violation, but gate C has a small delay, so it has a hold-time violation (the hold-time violation path comes from the lower path of gate C).

- (3) Is it possible that gate B has a hold-time violation and gate C has a setup-time violation? Explain why.

Yes. Both gate B and gate C have short delays, but a gate connected to the output of gate C could have a long delay.

- (4) Is it possible that gate B has a hold-time violation and gate C has a hold-time violation? Explain why.

Yes. Both gate B and gate C have very small delays, so a path going through gate B and gate C could violate a hold-time constraint.

**EE434**  
**ASIC and Digital Systems**

**Midterm Exam 1**

**Mar. 4, 2020. (2:10pm – 3pm)**

**Instructor: Dae Hyun Kim ([daehyun@eecs.wsu.edu](mailto:daehyun@eecs.wsu.edu))**

**Name:**

**WSU ID:**

Problem	Points	
1	10	
2	10	
3	10	
4	10	
5	10	
6	10	
7	10	
8	10	
Total	80	

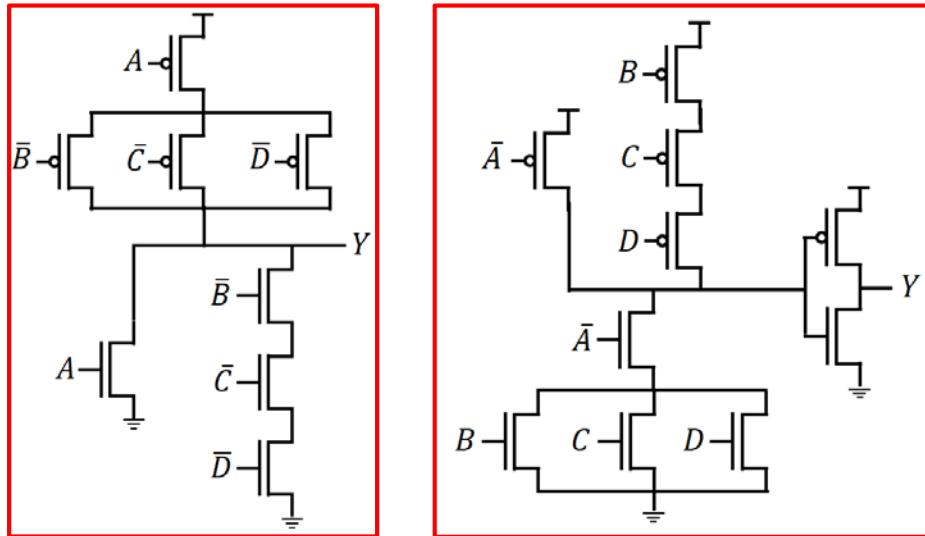
## Problem #1 (Static CMOS gates, 10 points)

Design the following logic using the static CMOS design methodology. Try to minimize the # transistors. Available input:  $A, B, C, D$ .

$$Y = \overline{A + \bar{B} \cdot \bar{C} \cdot \bar{D}}$$

If we design it directly, we will need eight TRs + six TRs (for three inverters) = 14 TRs.

However, if we design  $Y = \overline{\overline{A + \bar{B} \cdot \bar{C} \cdot \bar{D}}} = \overline{\bar{A} \cdot (B + C + D)}$ , we will need eight TRs + four TRs (two for the inverter for  $\bar{A}$  and two for the outer (last-level) inverter).



Left:  $8 + 2 + 2 + 2 = 14$  TRs (7 points)

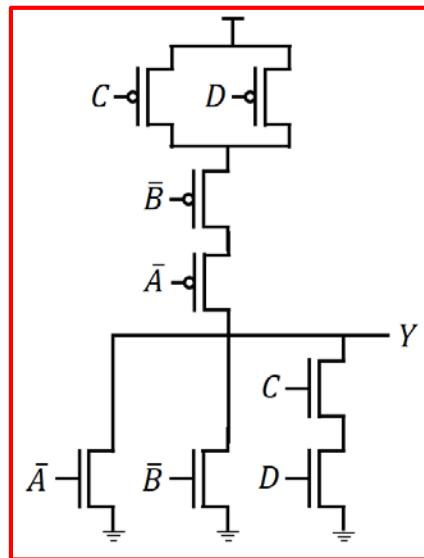
Right:  $8 + 2 + 2 = 12$  TRs (10 points)

## Problem #2 (Static CMOS gates, 10 points)

Design the following logic using the static CMOS design methodology. Try to minimize the # transistors. Available input:  $A, B, C, D$ .

$$Y = A \cdot B \cdot (\bar{C} + \bar{D})$$

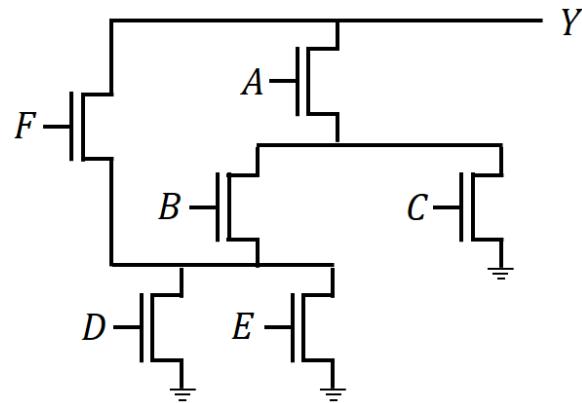
$$Y = \overline{\bar{A} + \bar{B} + C \cdot D}$$



$$8 + 4 = 12 \text{ TRs (10 points)}$$

### Problem #3 (Static CMOS gates, 10 points)

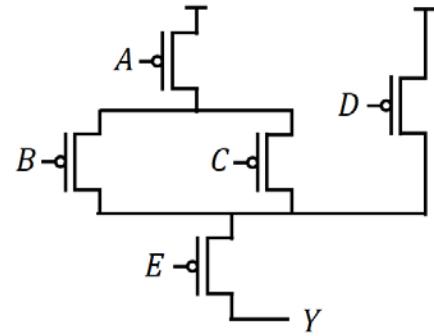
The following shows the NFET network of a static CMOS gate. Express the output  $Y$  as a Boolean function of the inputs ( $A \sim F$ ). (You don't need to simplify the expression.)



$$Y = \overline{F(D + E + BC) + A(C + B(D + E))}$$

### Problem #4 (Static CMOS gates, 10 points)

The following shows the PFET network of a static CMOS gate. Express the output  $Y$  as a Boolean function of the inputs ( $A \sim E$ ). (You don't need to simplify the expression.)



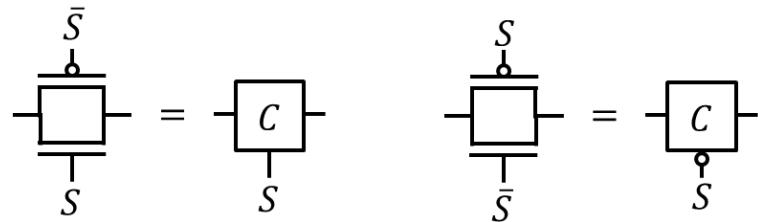
$$Y = (\bar{A} \cdot (\bar{B} + \bar{C}) + \bar{D}) \cdot \bar{E} = \overline{(A + BC) \cdot D + E}$$

## Problem #5 (Transmission Gates, 10 points)

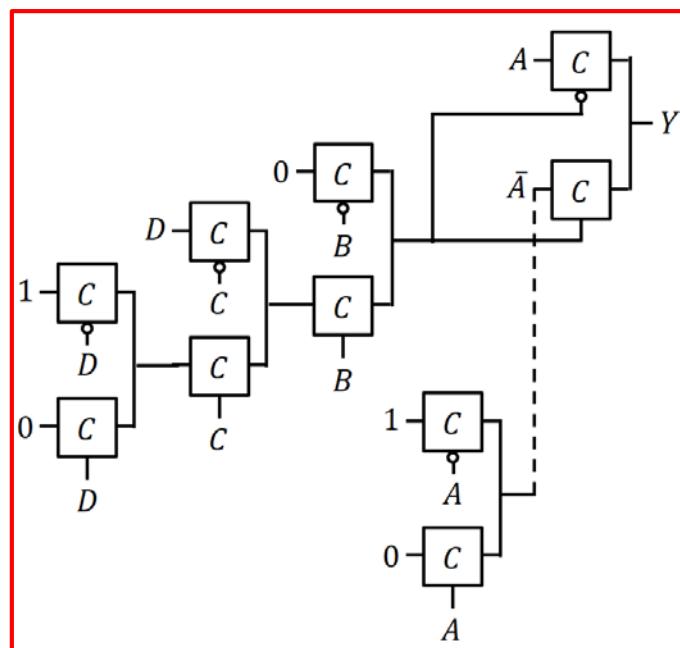
Design (draw a schematic) the following Boolean function using transmission gates only.

$$Y = A \oplus (B \cdot (C \oplus D))$$

Available inputs:  $A, B, C, D, 0, 1$ . Use the following symbols for the transmission gates.



(# TGs $\leq$ 12: 10 points. 13 $\leq$ # TGs $\leq$ 15: 7 points. 16 $\leq$ # TGs $\leq$ 18: 5 points. # TGs $>$ 18: 3 points)



## Problem #6 (Sequential Logic, 10 points)

The following truth table shows the function of a sequential logic. CK is the clock signal. A, B, D, E, and F are data or control (e.g., reset) signals. What does the gate do?? Explain the function in detail.

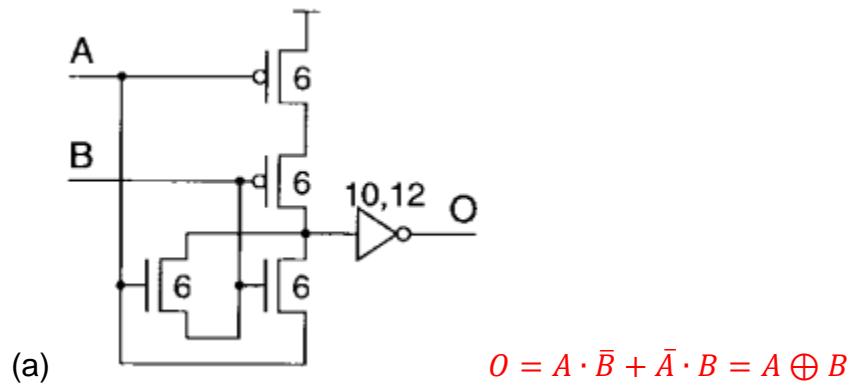
A	B	D	E	F	CK		Q <sup>+</sup>
0	X	0	E	X	↓		E
0	X	1	X	F	↓		F
1	B	X	X	X	↓		B
X	X	X	X	X	↑		Q
X	X	X	X	X	0 or 1		Q

It is a negative-edge-triggered D-FF with the following features:

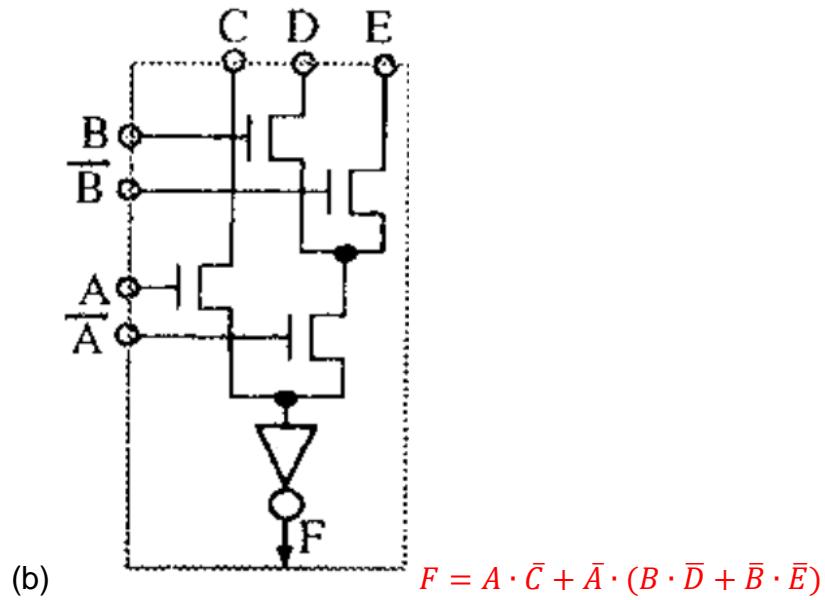
- A and D are used as input signal selectors as follows:
  - A=0, D=0: E is the data input.
  - A=0, D=1: F is the data input.
  - A=1: B is the data input.

### Problem #7 (Analysis, 10 points)

What do the following circuits do? (You can ignore the numbers in the schematics.) For each schematic, you can draw a truth table or express the output as a function of the inputs.



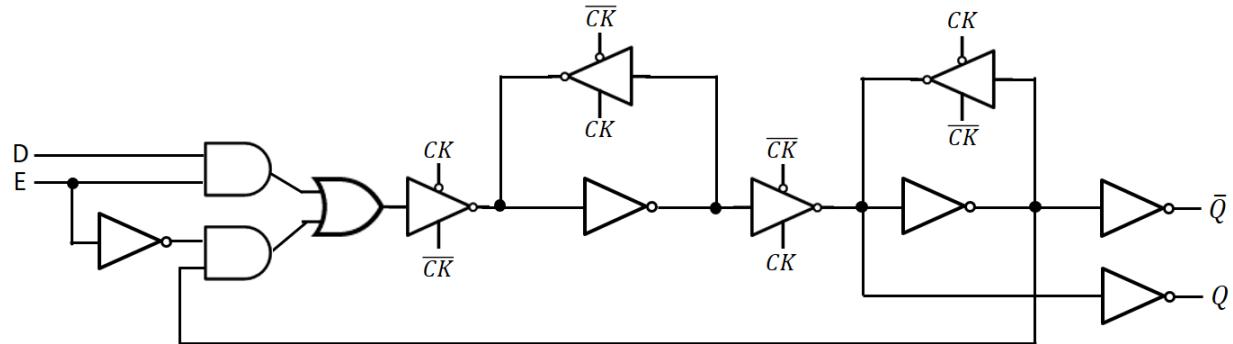
(JSSC'97)



(JSSC'96)

## Problem #8 (Sequential Logic, 10 points)

Explain the function of the following logic. CK: Clock. D, E: Data and/or control input.



If E=0, Q=hold regardless of the clock signal.

If E=1, it is a normal positive-edge-triggered D-F/F (in this case, D is the data input).

**EE434**  
**ASIC and Digital Systems**

**Midterm Exam 2**

**Apr. 8, 2020. (2:10pm – 3pm)**

**Instructor: Dae Hyun Kim ([daehyun@eecs.wsu.edu](mailto:daehyun@eecs.wsu.edu))**

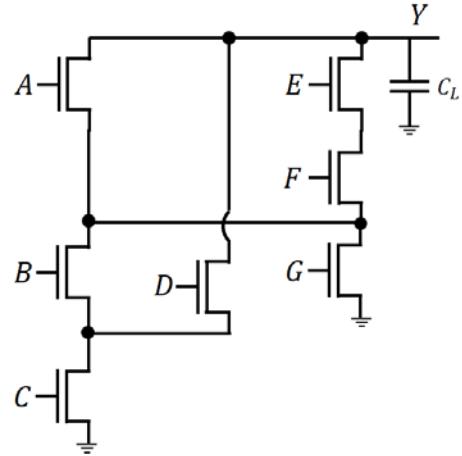
**Name:**

**WSU ID:**

Problem	Points	
1	10	
2	10	
3	20	
4	10	
5	10	
6	10	
7	10	
Total	80	

## Problem #1 (Transistor Sizing, 10 points)

Size the transistors in the following NFET network. Timing constraint:  $\tau \leq R_n C_L \cdot \frac{\mu_n}{\mu_p} = 2$ .  $R_n$  is the resistance of a 1X NFET. Try to minimize the total TR width heuristically.

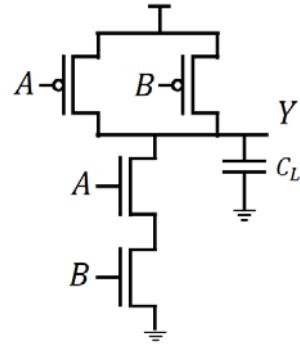


The longest path is EFBC, so we upsize them to 4X. The next longest path is ABC or DBG. For ABC, A becomes 2X. For DBG, D and G becomes  $\frac{8}{3}$ X.

- A: 2X
- B: 4X
- C: 4X
- D:  $\frac{8}{3}$ X
- E: 4X
- F: 4X
- G:  $\frac{8}{3}$ X

## Problem #2 (Transistor Sizing + Timing Analysis, 10 points)

In this problem, we will size the transistors of a gate for more complex timing constraints.  $\frac{\mu_n}{\mu_p} = 2$ .  $R_n$  is the resistance of a 1X NFET. Timing constraint:  $\tau \leq R_n C_L$ .



We usually assume that  $A$  and  $B$  are available at time 0, so the output  $Y$  should be ready by time  $R_n C_L$ . In this problem,  $A$  is available at time  $t = 0$  (i.e., the arrival time of signal  $A$  is 0), but  $B$  is available at time  $t = \frac{R_n C_L}{2}$  (i.e., the arrival time of signal  $B$  is  $\frac{R_n C_L}{2}$ ). The output  $Y$  should be ready by time  $t = R_n C_L$ . Size the transistors to satisfy the timing constraint (and you should try to minimize the total TR width).

NFETs: Now, the timing constraint is  $\frac{R_n C_L}{2}$ . Thus, A and B should be 4X.

PFETs: For signal  $A$ , the timing constraint is  $R_n C_L$ , so A should be 2X. For signal  $B$ , the timing constraint is  $\frac{R_n C_L}{2}$ , so B should be 4X.

NFETs

A: 4X

B: 4X

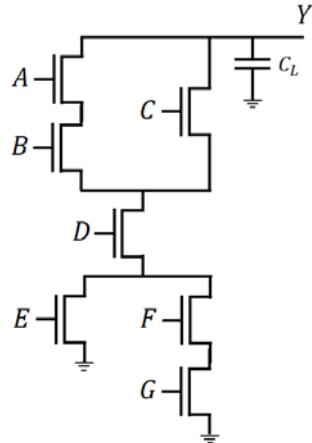
PFETs

A: 2X

B: 4X

### Problem #3 (Transistor Sizing + Timing Analysis, 20 points)

This problem is similar to Problem #2.  $\frac{\mu_n}{\mu_p} = 2$ .  $R_n$  is the resistance of a 1X NFET. Timing constraint:  $\tau \leq R_n C_L$ . The following shows the NFET network of  $Y = \overline{(AB + C)D(E + FG)}$ .



The following shows the arrival times of the input signals. Size the transistors properly to satisfy the timing constraint (and you should try to minimize the total TR width).

- $A: t = \frac{R_n C_L}{6}$
- $B, D, E, G: t = \frac{R_n C_L}{8}$
- $C: t = \frac{R_n C_L}{4}$
- $F: t = \frac{R_n C_L}{16}$

For path ABDFG: delay should be  $\leq R_n C_L - \frac{R_n C_L}{6} = \frac{5R_n C_L}{6}$

For path ABDE: delay should be  $\leq R_n C_L - \frac{R_n C_L}{6} = \frac{5R_n C_L}{6}$

For path CDFG: delay should be  $\leq R_n C_L - \frac{R_n C_L}{4} = \frac{3R_n C_L}{4}$

For path CDE: delay should be  $\leq R_n C_L - \frac{R_n C_L}{4} = \frac{3R_n C_L}{4}$

Optimize ABDFG first: Size all of them to  $aX$ . Then,  $\frac{R_n}{a} * 5 * C_L \leq \frac{5R_n C_L}{6}$ , so  $a = 6X$ .

Then, optimize CDFG: Size C to  $cX$ . Then,  $(\frac{R_n}{c} + \frac{R_n}{6} * 3) * C_L \leq \frac{3R_n C_L}{4}$ , so  $c = 4X$ .

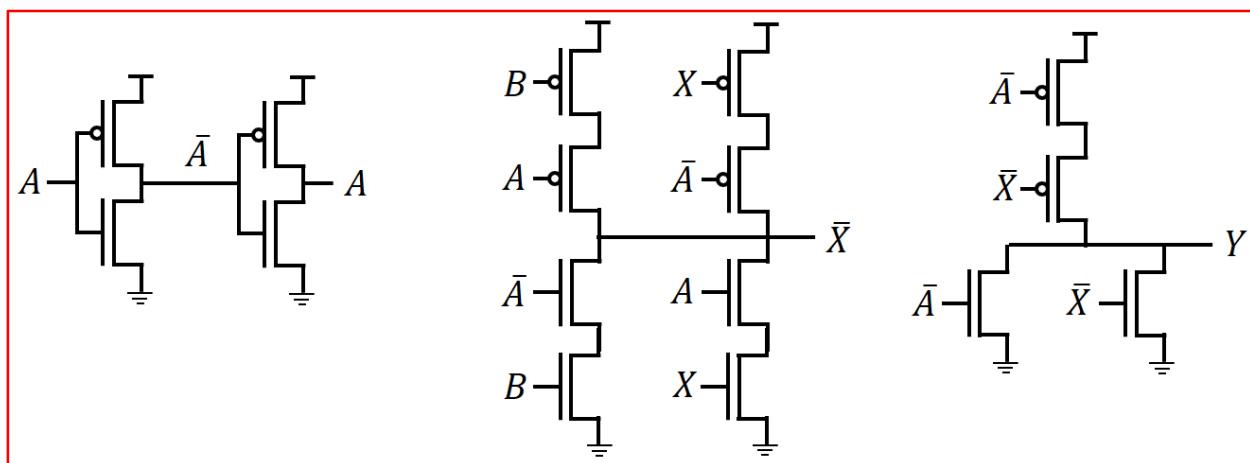
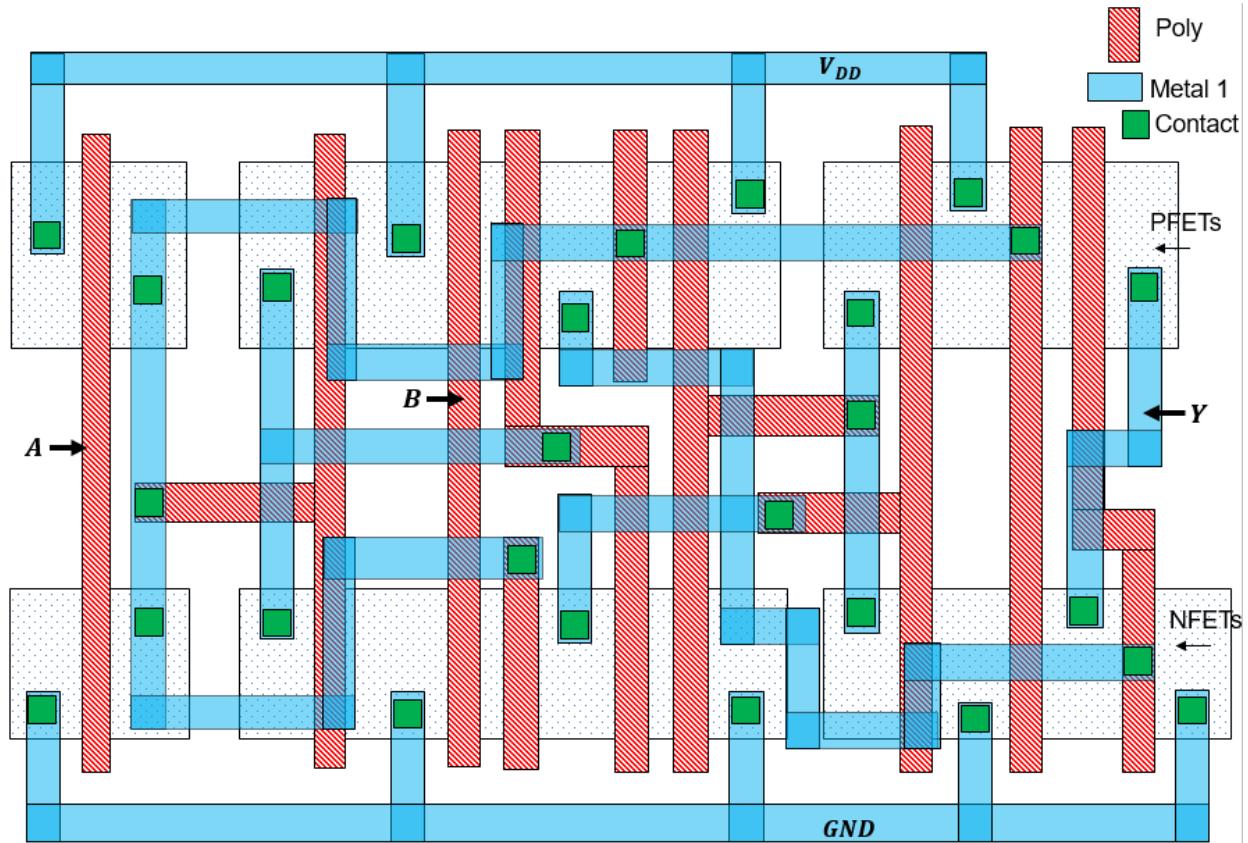
Then, optimize CDE: Size E to  $eX$ . Then,  $(\frac{R_n}{e} + \frac{R_n}{4} + \frac{R_n}{6}) * C_L \leq \frac{3R_n C_L}{4}$ , so  $e = 3X$ .

For ABDE: Delay =  $(\frac{R_n}{6} * 3 + \frac{R_n}{3}) * C_L = \frac{5R_n C_L}{6} \leq \frac{5R_n C_L}{6}$  (satisfies the timing constraint).

Answer) A, B, D, F, G: 6X. C: 4X. E: 3X.

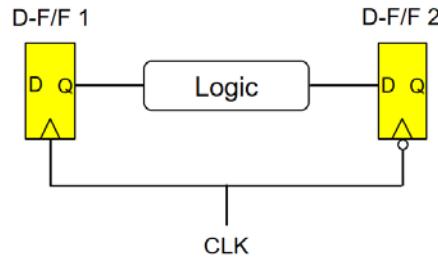
## Problem #4 (Layout, 20 points)

Draw a transistor-level schematic for the following layout.



## Problem #5 (Static Timing Analysis, 10 points)

If the clock skew for the following logic is too negative or too positive (i.e., too large), it won't work correctly (the signals won't be captured correctly). Derive two inequalities that the clock skew should satisfy.



Use the following constants:

- Setup time of D-FF 1, 2:  $s_1, s_2$
- Hold time of D-FF 1, 2:  $h_1, h_2$
- Clock period:  $T_{CLK}$
- Logic delay:  $T_{logic}$
- C-Q delay of D-FF 1, 2:  $c_1, c_2$
- Delay from the CLK source to D-FF 1, 2:  $d_1, d_2$
- The clock skew is defined by  $T_{skew} = d_2 - d_1$

If the skew is too large (i.e.,  $d_2 \gg d_1$ ), the hold time violation at D-FF2 will be a problem. If the skew is too small (i.e.,  $d_1 \gg d_2$ ), the setup time violation at D-FF2 will be a problem. Thus, the result of the logic ( $d_1 + c_1 + T_{logic}$ ) should be available after D-FF2 captures its input correctly ( $d_2 + h_2$ ), but before D-FF2 captures the result of the logic ( $d_2 + T_{CLK} - s_2$ ).

$$d_2 + h_2 \leq d_1 + c_1 + T_{logic} \leq d_2 + T_{CLK} - s_2$$

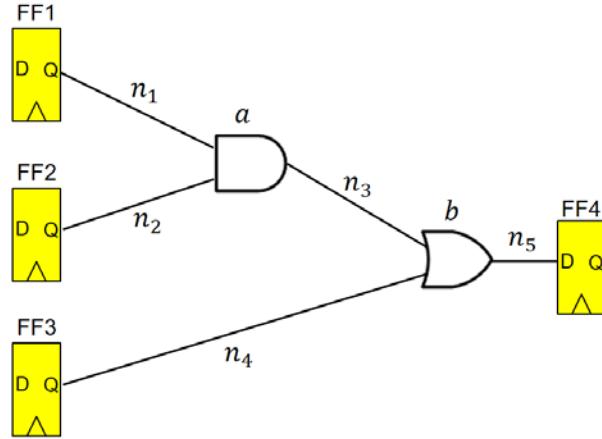
Thus,

$$\text{Answer: } c_1 + T_{logic} - T_{CLK} + s_2 \leq T_{skew} \leq c_1 + T_{logic} - h_2$$

(Notice that this is just a rearrangement of the setup- and hold-time inequalities we studied.)

## Problem #6 (Static Timing Analysis, 10 points)

Find setup and hold time inequalities that the following logic has to satisfy.



Use the following constants:

- Setup time of D-FF 1, 2, 3, 4:  $s_1, s_2, s_3, s_4$
- Hold time of D-FF 1, 2, 3, 4:  $h_1, h_2, h_3, h_4$
- Clock period:  $T_{CLK}$
- C-Q delay of D-FF 1, 2, 3, 4:  $c_1, c_2, c_3, c_4$
- Delay from the CLK source to D-FF 1, 2, 3, 4:  $d_1, d_2, d_3, d_4$
- Net and gate delays:  $n_1, n_2, n_3, n_4, a, b$

You can also use MAX and MIN operators.

**Setup:**

$$\text{MAX}[\text{MAX}(d_1 + c_1 + n_1, d_2 + c_2 + n_2) + a + n_3, d_3 + c_3 + n_4] + b + n_5 \leq d_4 + T_{CLK} - s_4$$

**Hold:**

$$\text{MIN}[\text{MIN}(d_1 + c_1 + n_1, d_2 + c_2 + n_2) + a + n_3, d_3 + c_3 + n_4] + b + n_5 \geq d_4 + h_4$$

## Problem #7 (Static Timing Analysis + Pipelining, 10 points)

Suppose an (ideally-partitionable) logic is given. Its delay is  $d$ . If we partition it into  $p$  equally-distributed pipeline stages, the delay of the sub-logic in each pipeline stage becomes  $\frac{d}{p}$  (Notice that  $p \geq 1$ ).

If we run  $N$  instructions sequentially in the pipeline, it takes total  $\#(N + p - 1)$  clock cycles to execute all the instructions. The total execution time is  $(N + p - 1) \cdot T_p$  where  $T_p$  is the clock period for the  $p$ -pipelined system.

All the flip-flops have the same characteristics:

- C-Q delay:  $c$
- Setup time:  $s$
- Hold time:  $h$

Answer the following questions.

The system should satisfy the typical setup and hold time constraints.

$$h - c \leq \frac{d}{p} \leq T_p - (c + s)$$

(1) Assume  $c > 0, s > 0, h > 0, h > c$ , and the clock skew is zero. Find the maximum value of  $p$  that does not lead to hold-time violations.

From the inequality above, the maximum value of  $p$  is  $\frac{d}{h-c}$ .

(2) Assume  $c > 0, s > 0, h > 0$ , and the clock skew is zero. Find the minimum value of  $T_p$  that does not lead to setup-time violations.

From the inequality above, the minimum value of  $T_p$  is  $\frac{d}{p} + c + s$ .

(3) If  $c = 0, s = 0, h = 0$ , we should always increase  $p$  to reduce the execution time (**True** / **False**).

In this case,  $T_p = \frac{d}{p}$ , so the execution time is  $\frac{d}{p} \cdot (N + p - 1) = d + \frac{d(N-1)}{p}$ , so we should increase  $p$  as much as we can.

(4) If  $c > 0, s = 0, h = 0$ , we should always increase  $p$  to reduce the execution time (**True** / **False**).

$T_p = \frac{d}{p} + c$ , so the execution time is  $\left(\frac{d}{p} + c\right) \cdot (N + p - 1) = d + \frac{d(N-1)}{p} + c(N - 1) + cp$ . Thus,  $p$  has a certain optimal value. (If you want to compute, you can. From  $-\frac{d(N-1)}{p^2} + c = 0$ , we get  $p = \sqrt{\frac{d(N-1)}{c}}$ )

(5) If  $c = 0, s > 0, h = 0$ , we should always increase  $p$  to reduce the execution time (**True** / **False**).

$T_p = \frac{d}{p} + s$ , which is similar to case (2).

(6) If  $c = 0, s = 0, h > 0$ , we should always increase  $p$  to reduce the execution time (True / False).

In this case,  $T_p = \frac{d}{p}$ . However, since  $h \leq \frac{d}{p}$ ,  $p$  cannot be greater than  $\frac{d}{h}$ .

**EE434**  
**ASIC and Digital Systems**

**Final Exam**

**May 4, 2021. (1pm – 4pm)**

**Instructor: Dae Hyun Kim ([daehyun@eecs.wsu.edu](mailto:daehyun@eecs.wsu.edu))**

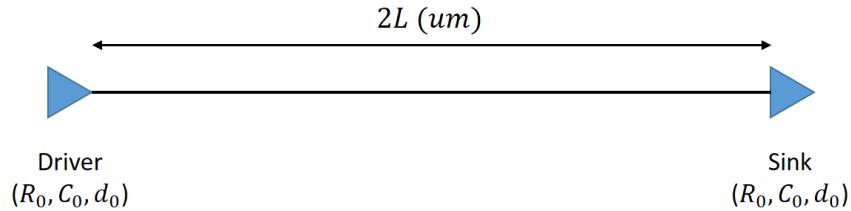
**Name:**

**WSU ID:**

Problem	Points	
1	40	
2	40	
3	20	
4	20	
5	10	
6	20	
7	20	
8	20	
Total	190	

## Problem #1 (Interconnect Optimization, 40 points)

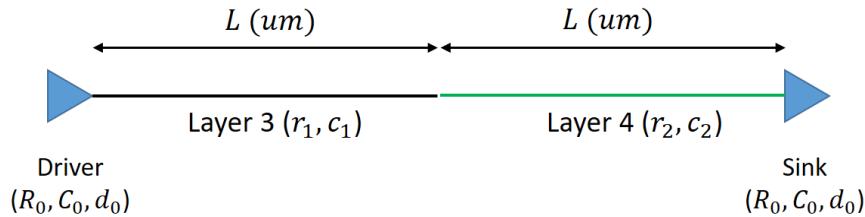
In the figure below, the driver is driving the sink through a long wire. The wire length is  $2L$  ( $\mu m$ ).



- $R_0, C_0, d_0$ : The output resistance, the input capacitance, and the internal delay of the gates, respectively.
- $r_1, c_1$ : Unit wire resistance and capacitance of metal layer 3
- $r_2, c_2$ : Unit wire resistance and capacitance of metal layer 4

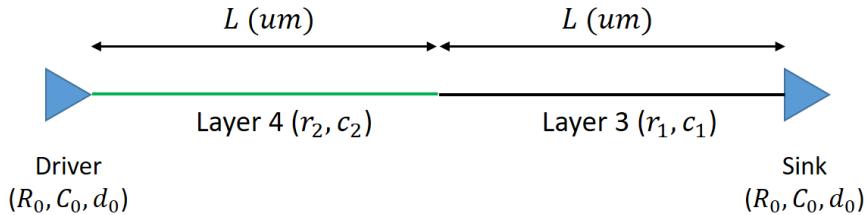
Answer the following questions.

(a) (5 points) The first wire segment is routed on the metal layer 3 and the second wire segment is routed on the metal layer 4 as shown below. Express the total delay  $d_A$  from the output of the driver to the input of the sink as a function of the parameters above.



$$d_A = R_0(c_1L + c_2L + C_0) + r_1L(c_2L + C_0) + \frac{1}{2}r_1c_1L^2 + r_2LC_0 + \frac{1}{2}r_2c_2L^2$$

(b) (5 points) The first wire segment is routed on the metal layer 4 and the second wire segment is routed on the metal layer 3 as shown below. Express the total delay  $d_B$  from the output of the driver to the input of the sink as a function of the parameters above.



$$d_B = R_0(c_2L + c_1L + C_0) + r_2L(c_1L + C_0) + \frac{1}{2}r_2c_2L^2 + r_1LC_0 + \frac{1}{2}r_1c_1L^2$$

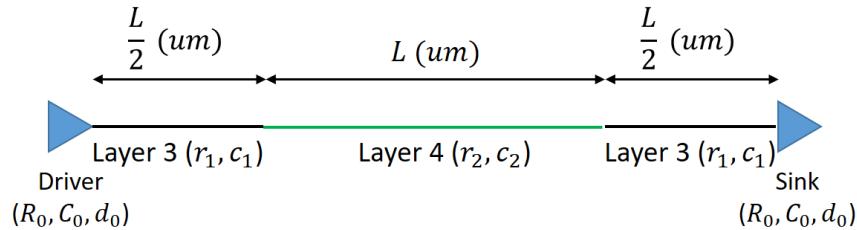
(c) (6 points) When are they ( $d_A$  and  $d_B$ ) equal? Show the condition for  $d_A = d_B$ .

$$d_A - d_B = (r_1 c_2 - r_2 c_1)L^2, \text{ so } d_A = d_B \Leftrightarrow \frac{r_2}{r_1} = \frac{c_2}{c_1}.$$

(d) (2 points) If  $r_1 = r_2$  and  $c_1 > c_2$ ,  $d_A > d_B$ . (True / False)

(e) (2 points) If  $c_1 = c_2$  and  $r_1 > r_2$ ,  $d_A > d_B$ . (True / False)

Now, you are supposed to route the driver and the sink using metal layer 3 and 4. However, the total wire length in the metal layer 3 should  $L$  (um). Similarly, the total wire length in the metal layer 4 should also be  $L$  (um). The following shows an example.



Answer the following questions.

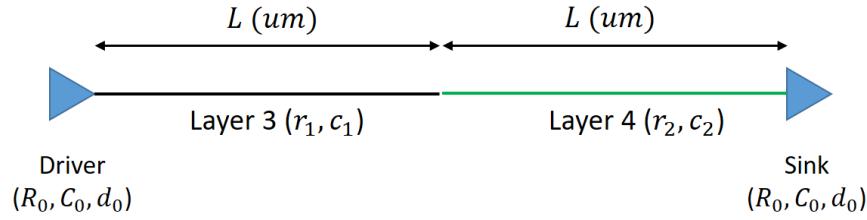
In general, suppose you have a metal 3 wire ( $w_{3,1}$ ), then a metal 4 wire ( $w_{4,1}$ ), then a metal 3 wire ( $w_{3,2}$ ), etc. Thus, you have  $w_{3,1}, w_{4,1}, w_{3,2}, w_{4,2}, \dots, w_{3,n}, w_{4,n}$ . Suppose  $w_{i,j}$  is the length of the  $j$ -th segment in the metal  $i$  layer ( $w_{i,j} \geq 0, \sum w_{3,j} = L, \sum w_{4,j} = L$ ). Then, the total delay is

$$\begin{aligned} d = & R_0(c_1 L + c_2 L + C_0) + r_1 L C_0 + r_2 L C_0 + \sum_{j=1}^n \left( r_1 w_{3,j}(C_{3,j}) \right) + \sum_{j=1}^n \left( r_2 w_{4,j}(C_{4,j}) \right) \\ & + \sum_{j=1}^n \frac{1}{2} r_1 c_1 w_{3,j}^2 + \sum_{j=1}^n \frac{1}{2} r_2 c_2 w_{4,j}^2 \end{aligned}$$

where  $C_{i,j}$  is the downstream capacitance of  $w_{i,j}$ . We can differentiate it w.r.t.  $w_{3,j}$  and  $w_{4,j}$  and set them to 0 to find optimal  $w_{3,j}$  and  $w_{4,j}$ . However, it is quite complicated, so we can solve the problem intuitively.

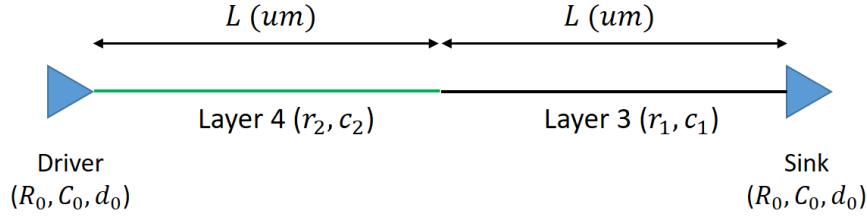
(f) (10 points) If  $r_1 = r_2$  and  $c_1 > c_2$ , how would you route it to minimize the total delay?

If  $c_1 > c_2$ , it is better to place the metal layer 3 wires in the left half. A wire segment has to drive its downstream capacitance, so if the left half of the net is routed in the metal layer 3, the wire segments on the right half can drive smaller cap. Thus, the answer is



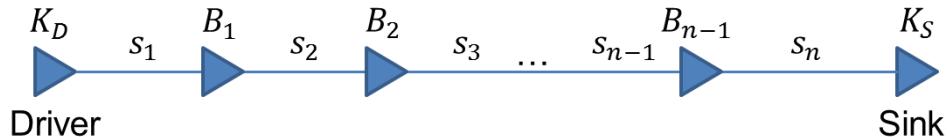
(g) (10 points) If  $c_1 = c_2$  and  $r_1 > r_2$ , how would you route it to minimize the total delay?

If  $r_1 > r_2$ , it is better to place the metal layer 3 wires in the right half so that the wires on the left half (which have smaller wire resistance) drive larger cap and the wires on the right half (which have larger wire resistance) drive smaller cap. Thus, the answer is



## Problem #2 (Interconnect Optimization, 40 points)

The following figure shows a net optimized by buffer insertion. The driver and the sink are denoted by  $K_D$  and  $K_S$ , respectively, and the inserted buffers are denoted by  $B_i$  ( $1 \leq i \leq n - 1$ ).  $n \geq 2$ , i.e., there is at least one buffer between the driver and the sink.



- Output resistance of  $K_D$ :  $R_D$
- Output resistance of  $B_i$  ( $1 \leq i \leq n - 1$ ):  $R_i$  (e.g.,  $R_1, R_2, \dots$ )
- Input capacitance of  $K_S$ :  $C_S$
- Input capacitance of  $B_i$  ( $1 \leq i \leq n - 1$ ):  $C_i$
- Delay of  $B_i$  ( $1 \leq i \leq n - 1$ ):  $D_i$
- Length of the  $i$ -th net ( $1 \leq i \leq n$ ):  $s_i$  (um)
- $\sum_{i=1}^n s_i = L$  (um)
- Unit wire resistance of the  $i$ -th net ( $1 \leq i \leq n$ ):  $r_i$  ( $\Omega/\text{um}$ )
- Unit wire capacitance of the  $i$ -th net ( $1 \leq i \leq n$ ):  $c_i$  (fF/um)

We assume that the net is optimized to minimize the delay from the driver to the sink.  
Resizing a buffer means upsizing or downsizing the buffer.

Answer the following questions for  $n = 10$  (i.e., we have inserted 9 buffers optimally):

(Correct: +4 points, No answer: 0 point, Wrong: -2 points)

- If  $r_1$  increases and we don't want to resize the buffers, we should increase  $s_1$  to minimize the total delay. (True / False)
- If  $r_1$  increases and we don't want to resize the buffers, we should increase  $s_n$  to minimize the total delay. (True / False)
- If  $c_1$  increases and we don't want to resize the buffers, we should increase  $s_1$  to minimize the total delay. (True / False)
- If  $c_1$  increases and we don't want to resize the buffers, we should increase  $s_n$  to minimize the total delay. (True / False)
- If  $r_1$  increases and we don't want to move the buffers, we should upsize  $B_1$  to minimize the total delay. (True / False)
- If  $r_1$  increases and we don't want to move the buffers, we should upsize  $B_{n-1}$  to minimize the total delay. (True / False)
- If  $c_1$  increases and we don't want to move the buffers, we should upsize  $B_1$  to minimize the total delay. (True / False)

- If  $c_1$  increases and we don't want to move the buffers, we should upsize  $B_{n-1}$  to minimize the total delay. (True / **False**)
- If  $r_1$  and  $c_1$  increase at the same time and we don't want to resize the buffers, we should increase  $s_1$  to minimize the total delay. (True / **False**)
- If  $r_1$  and  $c_1$  increase at the same time and we don't want to move the buffers, we should upsize  $B_1$  to minimize the total delay. (True / **False**)

Suppose  $R_D = R_0$  and  $C_S = C_n$ . Then, the delay of segment  $s_k$  is

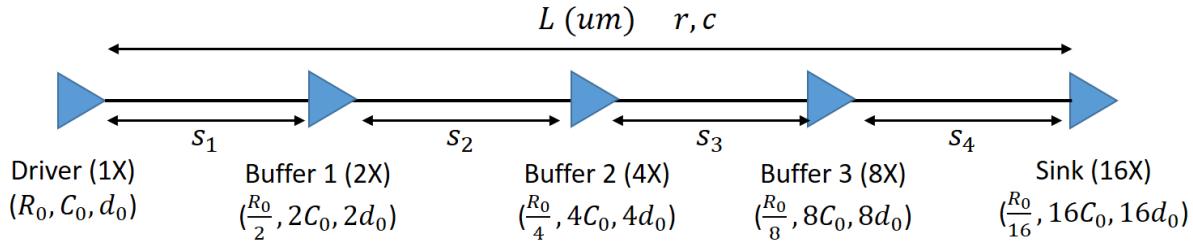
$$\tau_k = R_{k-1}(c_k \cdot s_k + C_k) + r_k \cdot s_k \cdot C_k + \frac{1}{2}r_k c_k s_k^2$$

Then, the total delay is

$$\tau = \sum_{k=1}^n \tau_k + \sum_{k=1}^{n-1} D_k = \sum_{k=1}^n R_{k-1} c_k s_k + \sum_{k=1}^n R_{k-1} C_k + \sum_{k=1}^n r_k s_k C_k + \frac{1}{2} \sum_{k=1}^n r_k c_k s_k^2 + \sum_{k=1}^{n-1} D_k$$

### Problem #3 (Interconnect Optimization, 20 points)

The following shows a buffer insertion problem. The triplet  $(R, C, D)$  of a gate is the (output resistance, input capacitance, internal delay) of the gate.



- $L$ : Net length
- $r, c$ : Unit wire resistance and capacitance

You insert three buffers as shown above (2X, 4X, 8X) to minimize the delay. Find their locations (i.e., find  $s_1, s_2, s_3$ ).

$$d = R_0(cs_1 + 2C_0) + rs_1 2C_0 + \frac{1}{2}rcs_1^2 + \frac{R_0}{2}(cs_2 + 4C_0) + rs_2 4C_0 + \frac{1}{2}rcs_2^2 + \frac{R_0}{4}(cs_3 + 8C_0) + rs_3 8C_0 + \frac{1}{2}rcs_3^2 + \frac{R_0}{16}(cs_4 + 16C_0) + rs_4 16C_0 + \frac{1}{2}rcs_4^2 + D \text{ where } D \text{ is the sum of the internal delays of the buffers.}$$

$$\frac{\partial d}{\partial s_1} = R_0c + 2rC_0 + rcs_1 - \frac{R_0}{16}c - 16rC_0 - rcs_4 = 0, \text{ so } s_1 = s_4 - \frac{15R_0}{16r} + 14\frac{C_0}{c}$$

$$\frac{\partial d}{\partial s_2} = \frac{R_0c}{2} + 4rC_0 + rcs_2 - \frac{R_0}{16}c - 16rC_0 - rcs_4 = 0, \text{ so } s_2 = s_4 - \frac{7R_0}{16r} + 12\frac{C_0}{c}$$

$$\frac{\partial d}{\partial s_3} = \frac{R_0c}{4} + 8rC_0 + rcs_3 - \frac{R_0}{16}c - 16rC_0 - rcs_4 = 0, \text{ so } s_3 = s_4 - \frac{3R_0}{16r} + 8\frac{C_0}{c}$$

$$\text{From } s_1 + s_2 + s_3 + s_4 = L, \text{ we get } 4s_4 - \frac{25R_0}{16r} + 34\frac{C_0}{c} = L.$$

$$s_1 = \frac{L}{4} - \frac{35R_0}{64r} + \frac{11C_0}{2c}$$

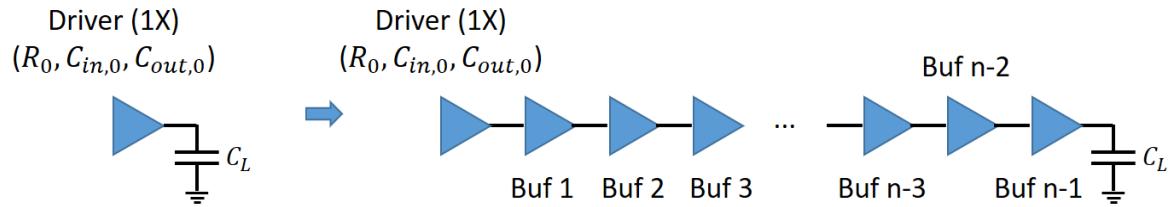
$$s_2 = \frac{L}{4} - \frac{3R_0}{64r} + \frac{7C_0}{2c}$$

$$s_3 = \frac{L}{4} + \frac{13R_0}{64r} - \frac{C_0}{2c}$$

$$s_4 = \frac{L}{4} + \frac{25R_0}{64r} - \frac{17C_0}{2c}$$

## Problem #4 (Interconnect Optimization, 20 points)

In the following figure (left), the driver is driving a very large load capacitor,  $C_L$ . The triplet  $(R, C, T)$  is (output resistance, input capacitance, output capacitance) of the gate. Since it is driving a large load cap, Michael inserts buffers as shown in the figure (right).



The size of Buf  $S$  is  $k^S \times$ , so the triplet of Buf  $S$  is  $\left(\frac{R_0}{k^S}, k^S \cdot C_{in,0}, k^S \cdot C_{out,0}\right)$ . In other words, the size of Buf 1 is  $k \times$ , the size of Buf 2 is  $k^2 \times$ , the size of Buf 3 is  $k^3 \times$ , and so on. Michael found an optimal solution, so both  $k$  and  $n$  are optimal.

Answer the following questions. Assume  $n = 10$ .

(Correct: +5 points, No answer: 0 point, Wrong: -3 points)

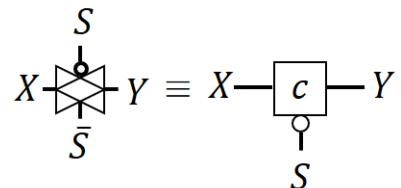
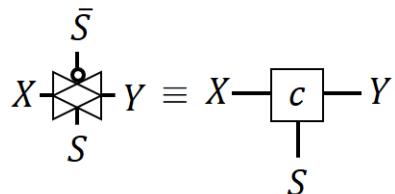
- If I upsize Buf 5 from  $k^5 \times$  to  $B \times$  ( $B > k^5$ ), I should upsize Buf 3 too for optimal buffer insertion. (True / False)
- If I upsize Buf 5 from  $k^5 \times$  to  $B \times$  ( $B > k^5$ ), I should upsize Buf 7 too for optimal buffer insertion. (True / False)
- If  $C_L$  goes down and  $n$  is still 10, I should upsize Buf 5 for optimal buffer insertion. (True / False)
- If  $C_L$  goes down and I don't want to size the buffers, I should remove some of the buffers (i.e., reduce  $n$ ) for optimal buffer insertion. (True / False)

## Problem #5 (Transmission Gates, 10 points)

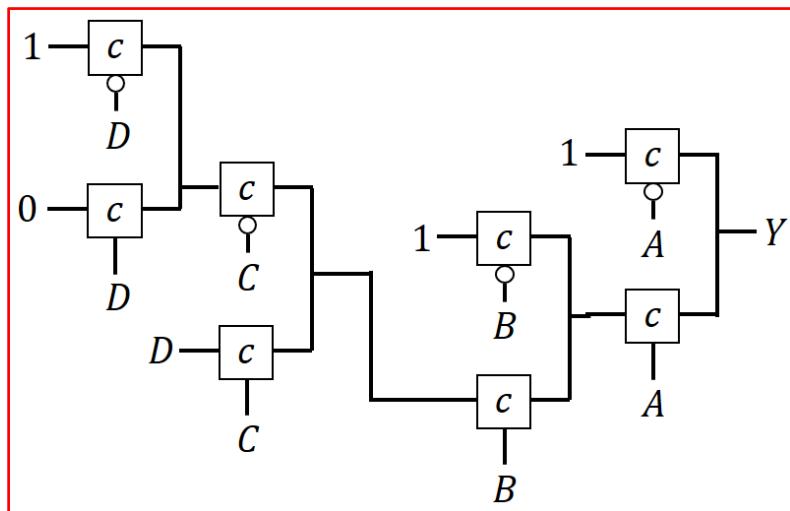
Design the following function (draw a transmission-gate-level schematic) using transmission gates only. Available input:  $A, B, C, D, 0 (GND), 1 (VDD)$ .

$$Y = \overline{A \cdot B \cdot (C \oplus D)}$$

You can use the following symbols for transmission gates.

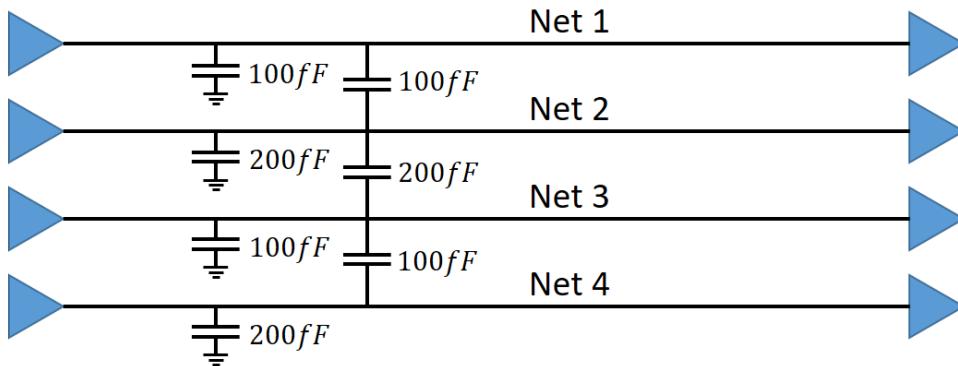


# TGs  $\leq 14$ : 10 points,  $\leq 16$ : 7 points,  $\leq 18$ : 4 points



## Problem #6 (Capacitive Coupling & STA, 20 points)

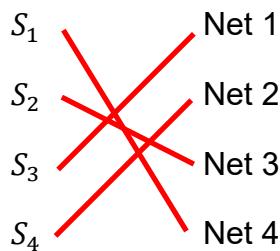
The following figure shows four nets and their ground and coupling capacitances. You can estimate the delay of a net by  $R \cdot C_{eff}$  where  $R$  is the output resistance of the driver driving the net and  $C_{eff}$  is the effective capacitance of the net.



Assign four given signals ( $S_1, S_2, S_3, S_4$ ) to the four nets for signal transmission. You should satisfy the given timing constraints.

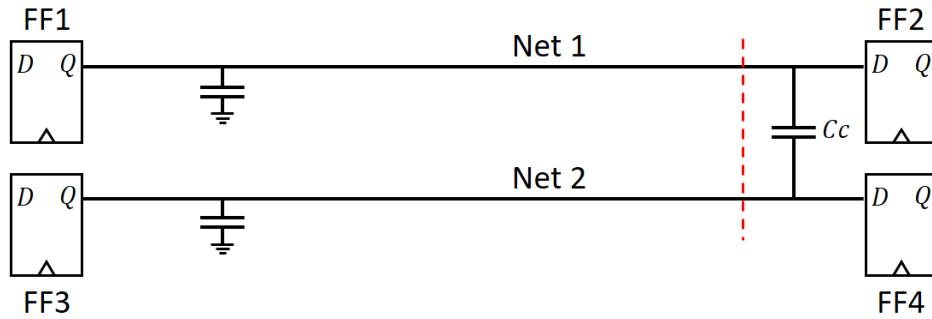
- $R: 1k\Omega$  for all the drivers.
- All the drivers have the same arrival time, AT=0ps.
- $S_1$  and  $S_2$  always switch in the same direction.
- $S_3$  and  $S_4$  always switch in the same direction.
- $S_1$  (or  $S_2$ ) and  $S_3$  (or  $S_4$ ) always switch in opposite directions.

RT (at the sink)
$S_1: 250\text{ps}$
$S_2: 550\text{ps}$
$S_3: 150\text{ps}$
$S_4: 700\text{ps}$

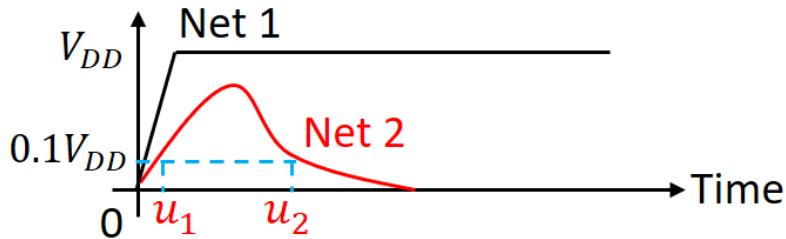


### Problem #7 (Crosstalk & STA, 20 points)

The following figure shows two nets and their coupling (Do not assume that they have the same length.) Notice that the coupling occurs on the right side of the dotted line, which is physically close to FF2 and FF4.



The following figure shows two waveforms when a crosstalk occurs between Net 1 (aggressor) and Net 2 (victim). If the potential of Net 2 is higher than  $0.1V_{DD}$ , signal inversion could happen in Net 2.



- $d_k$ : delay of Net  $k$  ( $k = 1 \sim 2$ )
- $s_k$ : setup time of  $FF_k$  ( $k = 1 \sim 4$ )
- $h_k$ : hold time of  $FF_k$  ( $k = 1 \sim 4$ )
- $x_k$ : delay from the clock source to the clock pin of  $FF_k$  ( $k = 1 \sim 4$ )
- $c_k$ : clk-to-Q delay of  $FF_k$  ( $k = 1 \sim 4$ )
- $T_{CLK}$ : clock period

(a) Derive a setup time constraint (inequality) related to the crosstalk for Net 2.

$$x_1 + c_1 + d_1 + u_2 \leq x_4 + T_{CLK} - s_4$$

(b) Derive a hold time constraint (inequality) related to the crosstalk for Net 2.

$$x_1 + c_1 + d_1 + u_1 \geq x_4 + h_4$$

## Problem #8 (STA, 20 points)

(a) (5 points) For the setup time analysis, we defined “slack” as “slack = required time – arrival time”. In this case, the required time is the time by which the signal should arrive. A positive slack means no timing violation and a negative slack means a timing violation.

Can you define “slack” for the hold time analysis in a similar way? For the hold time analysis, the required time is the time after which the signal should arrive. The goal is to make the slack positive if there is no hold-time violation and negative if there is a hold-time violation.

slack = arrival time – require time.

(b) Answer the following questions for the “hold-time” slack. WNS is the worst negative “hold-time” slack (the smallest slack similar to the definition of WNS for the setup time analysis) and TNS is the total negative “hold-time” slack (the sum of the negative slacks).

(Correct: +3 points, No answer: 0 point, Wrong: -2 points)

- “WNS = TNS” could happen. (**True** / False)
- “WNS < TNS” could happen. (True / **False**)
- “WNS > TNS” could happen. (**True** / False)
- “WNS > 0 and TNS < 0” could happen. (True / **False**)
- “TNS ≪ 0” could happen if many paths have very small delays. (**True** / False)

**EE434**  
**ASIC and Digital Systems**

**Midterm Exam 1**

**Feb. 26, 2021. (2:10pm – 3pm)**

**Instructor: Dae Hyun Kim ([daehyun@eeecs.wsu.edu](mailto:daehyun@eeecs.wsu.edu))**

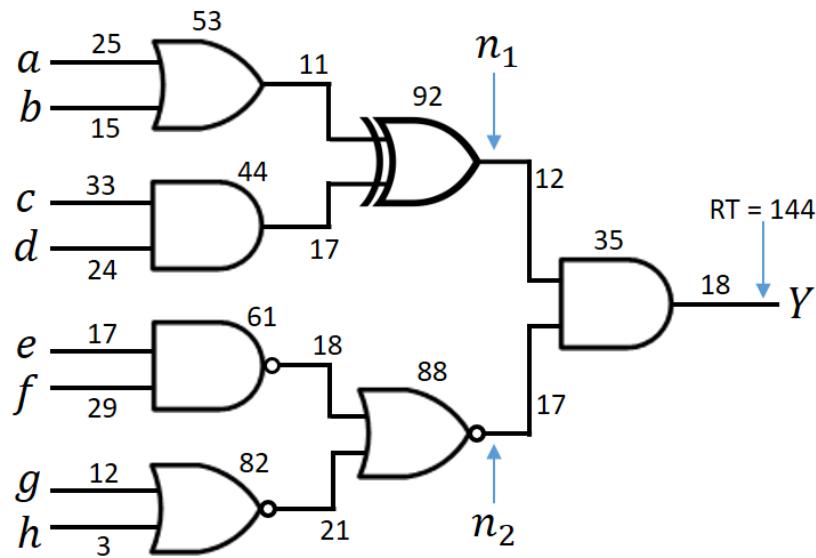
**Name:**

**WSU ID:**

Problem	Points	
1	10	
2	40	
3	40	
4	40	
Total	130	

## Problem #1 (STA, 10 points)

The following figure shows the delays (in ps) of the gates and nets of a logic design. The required time (RT) at the output node  $Y$  is 144 ps. All the inputs have zero arrival time.



- 1) Calculate the slack at node  $n_1$ .

$$\text{AT: } \text{MAX}(\text{MAX}(25, 15) + 53 + 11, \text{MAX}(33, 24) + 44 + 17) + 92 = \text{MAX}(89, 94) + 92 = 186\text{ps}$$

$$\text{RT: } 144 - 18 - 35 - 12 = 79\text{ps}$$

$$\text{Slack: } 79 - 186 = -107\text{ps}$$

- 2) Calculate the slack at node  $n_2$ .

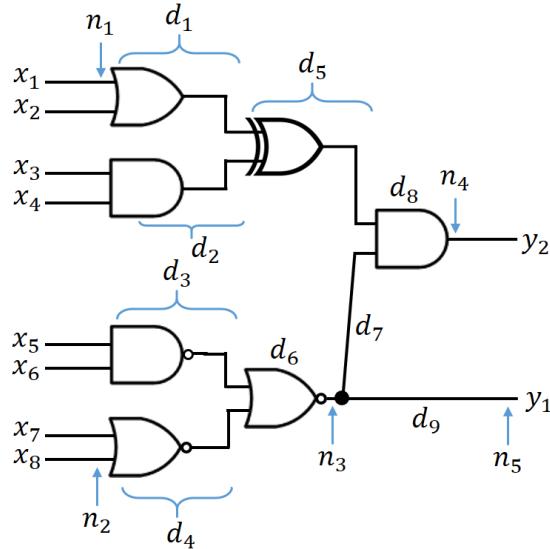
$$\text{AT: } \text{MAX}(\text{MAX}(17, 29) + 61 + 18, \text{MAX}(12, 3) + 82 + 21) + 88 = \text{MAX}(108, 115) + 88 = 203\text{ps}$$

$$\text{RT: } 144 - 18 - 35 - 17 = 74\text{ps}$$

$$\text{Slack: } 74 - 203 = -129\text{ps}$$

## Problem #2 (STA, 40 points)

The following figure shows a logic design.  $x_k$ 's are input signals and  $y_k$ 's are output signals. The slack at node  $n_k$  is denoted by  $s_k$ . Notice that the gates and nets have some delays. The inputs have the same arrival time  $a$  and the outputs have the same required time  $r$ .



- 1) Prove or disprove that  $s_1 \geq s_4$  is always true.

Denote the AT and RT at node  $n_k$  by  $a_k$  and  $r_k$ , respectively.

$$S = s_4 - s_1 = (r_4 - a_4) - (r_1 - a_1) = (r_4 - r_1) - (a_4 - a_1).$$

$$r_1 = r_4 - d_8 - d_5 - d_1, \text{ so } r_4 - r_1 = d_1 + d_5 + d_8.$$

$a_4 = \text{MAX}(a_{top}, a_{bot}) + d_8$  where  $a_{top}$  and  $a_{bot}$  are the ATs of the top and bottom nodes of the AND gate before  $n_4$ . Then,  $a_4 \geq a_{top} + d_8$  is true. Notice that  $a_{top} \geq a_1 + d_1 + d_5$ . Thus,  $a_4 \geq a_1 + d_1 + d_5 + d_8$ , or I can rewrite it as  $a_4 = a_1 + d_1 + d_5 + d_8 + e$  where  $e \geq 0$ . Then,  $a_4 - a_1 = d_1 + d_5 + d_8 + e$ .

$$\text{Now, } S = (d_1 + d_5 + d_8) - (d_1 + d_5 + d_8 + e) = -e, \text{ so } S = -e \leq 0, \text{ so } s_4 \leq s_1.$$

Intuitively speaking, if  $n_1$  has the worst slack than all the other nodes, the slack is propagated to the outputs.

2) Prove or disprove that  $s_2 \geq s_4$  is always true.

If n2-n3-n5 is the critical path,  $s_2 < s_4$ , so it is not true.

3) Prove or disprove that  $s_3 \geq s_4$  is always true.

For the same reason,  $s_3 < s_4$  can happen.

4) Prove or disprove that  $s_4 \geq s_5$  is always true.

$$S = s_5 - s_4 = (r_5 - r_4) - (a_5 - a_4).$$

$r_4 = r_5 - y$  ( $y$  is the delay of the net between  $n_4$  and  $y_2$ ,  $y \geq 0$ ).

$$r_5 - r_4 = y.$$

$$a_5 = a_3 + d_9.$$

$a_4 = \text{MAX}(a_3 + d_7, a_8) + d_8$  where  $a_8$  is the arrival time at the top input of the AND gate whose delay is  $d_8$ .

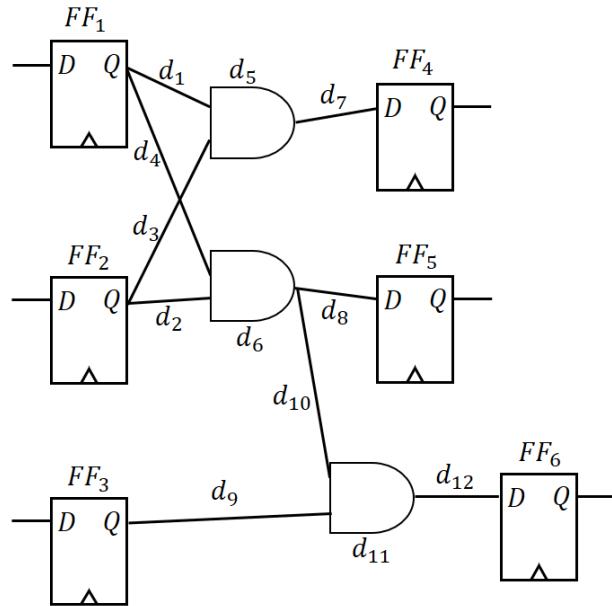
$$a_5 - a_4 = a_3 + d_9 - \text{MAX}(a_3 + d_7, a_8) - d_8.$$

$$\text{Thus, } S = y + d_8 - d_9 + \text{MAX}(a_3 + d_7, a_8) - a_3.$$

Depending on the delay values,  $S$  could be negative, zero, or positive. Thus, it is not true.

### Problem #3 (Setup and Hold Time, 40 points)

The following figure shows a part of a design.



- $d_k$ : delay ( $k = 1 \sim 12$ )
- $s_k$ : setup time of  $FF_k$  ( $k = 1 \sim 6$ )
- $h_k$ : hold time of  $FF_k$  ( $k = 1 \sim 6$ )
- $x_k$ : delay from the clock source to the clock pin of  $FF_k$  ( $k = 1 \sim 6$ )
- $c_k$ : clk-to-Q delay of  $FF_k$  ( $k = 1 \sim 6$ )
- $T_{CLK}$ : clock period
- $MIN, MAX$ : MIN, MAX operators

1) Derive a setup time constraint (inequality) for the signals coming to the input of  $FF_4$ .

$$MAX(x_1 + c_1 + d_1, x_2 + c_2 + d_3) + d_5 + d_7 \leq x_4 + T_{CLK} - s_4$$

2) Derive a hold time constraint (inequality) for the signals coming to the input of  $FF_4$ .

$$MIN(x_1 + c_1 + d_1, x_2 + c_2 + d_3) + d_5 + d_7 \geq x_4 + h_4$$

3) Express the slack at the input pin  $D$  of  $FF_5$  as a function of the constants above.

$$RT = x_5 + T_{CLK} - s_5$$

$$AT = MAX(x_1 + c_1 + d_4, x_2 + c_2 + d_2) + d_6 + d_8$$

$$Slack = RT - AT = x_5 + T_{CLK} - s_5 - MAX(x_1 + c_1 + d_4, x_2 + c_2 + d_2) - d_6 - d_8$$

4) Express the slack at the output pin  $Q$  of  $FF_3$  as a function of the constants above.

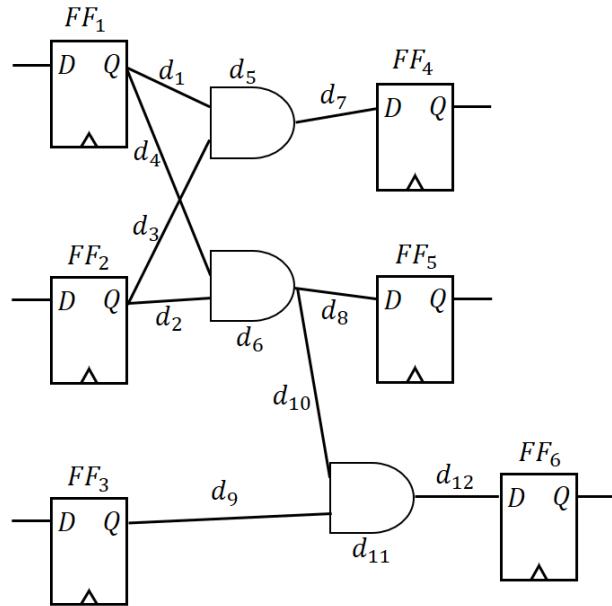
$$RT = x_6 + T_{CLK} - s_6 - d_{12} - d_{11} - d_9$$

$$AT = x_3 + c_3$$

$$Slack = x_6 + T_{CLK} - s_6 - d_{12} - d_{11} - d_9 - x_3 - c_3$$

## Problem #4 (Setup and Hold Time, 40 points)

The following figure shows a part of a design.



- $d_k$ : delay ( $k = 1 \sim 12$ )
- $S(FF_k, D)$ : Slack at the input pin D of  $FF_k$

Assume that all the flip-flops have the same setup time s, the same hold time h, the same clock-to-Q delay c, and the same delay x from the clock source to the clock pins.

Answer the following questions.

(Correct: +4 points, No answer: 0 point, Wrong: -2 points)

1) It is possible that  $FF_5$  satisfies its setup time constraint while  $FF_6$  violates its setup time constraint. (True / False)

The path from  $FF_3$  to  $FF_6$  could violate the setup time constraint.

2) It is possible that  $FF_5$  satisfies its setup time constraint while  $FF_6$  violates its hold time constraint. (True / False)

The path from  $FF_3$  to  $FF_6$  could violate the hold time constraint.

3) It is possible that  $FF_5$  satisfies its hold time constraint while  $FF_6$  violates its setup time constraint. (True / False)

The path from  $FF_3$  to  $FF_6$  could violate the setup time constraint.

- 4) It is possible that  $FF_5$  satisfies its hold time constraint while  $FF_6$  violates its hold time constraint. (**True** / False)

The path from  $FF_3$  to  $FF_6$  could violate the hold time constraint.

- 5) It is possible that  $FF_5$  violates its setup time constraint while  $FF_6$  satisfies its setup time constraint. (**True** / False)

If  $d_8$  is large,  $FF_5$  violates its setup time constraint, but the path to  $FF_6$  can still satisfy the setup time constraint.

- 6) It is possible that  $FF_5$  violates its setup time constraint while  $FF_6$  satisfies its hold time constraint. (**True** / False)

If  $d_8$  is large,  $FF_5$  violates its setup time constraint, but the path to  $FF_6$  can still satisfy the hold time constraint.

- 7) It is possible that  $FF_5$  violates its hold time constraint while  $FF_6$  satisfies its setup time constraint. (**True** / False)

- 8) It is possible that  $FF_5$  violates its hold time constraint while  $FF_6$  satisfies its hold time constraint. (**True** / False)

- 9) If  $d_4$  goes down,  $S(FF_6, D)$  goes up. (True / **False**)

If  $FF_3$  to  $FF_6$  is the critical path,  $S(FF_6, D)$  doesn't go up even if  $d_4$  goes down.

- 10) If  $d_9$  goes down,  $S(FF_6, D)$  goes up. (True / **False**)

**EE434**  
**ASIC and Digital Systems**

**Midterm Exam 2**

**Apr. 7, 2021. (2:10pm – 3pm)**

**Instructor: Dae Hyun Kim ([daehyun@eecs.wsu.edu](mailto:daehyun@eecs.wsu.edu))**

**Name:**

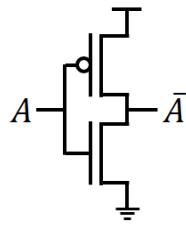
**WSU ID:**

Problem	Points	
1	10	
2	10	
3	15	
4	10	
5	10	
6	15	
Total	70	

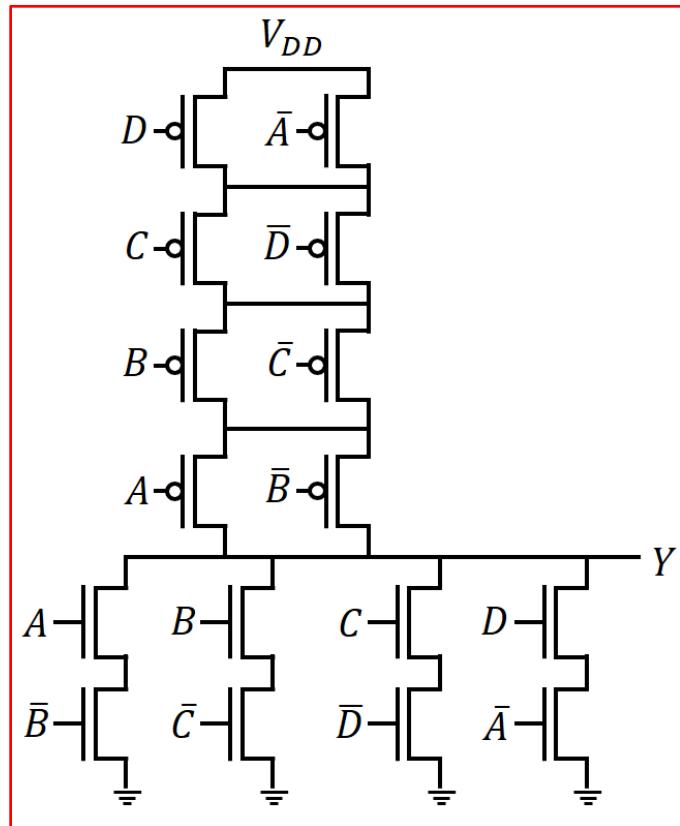
### Problem #1 (Static CMOS, 10 points)

Design the following function (draw a transistor-level schematic) using the static CMOS logic design methodology. Available input:  $A, B, C, D$ . # TRs should be less than or equal to 24.

$$Y = \overline{A \cdot \bar{B} + B \cdot \bar{C} + C \cdot \bar{D} + D \cdot \bar{A}}$$



(same for B, C, D)

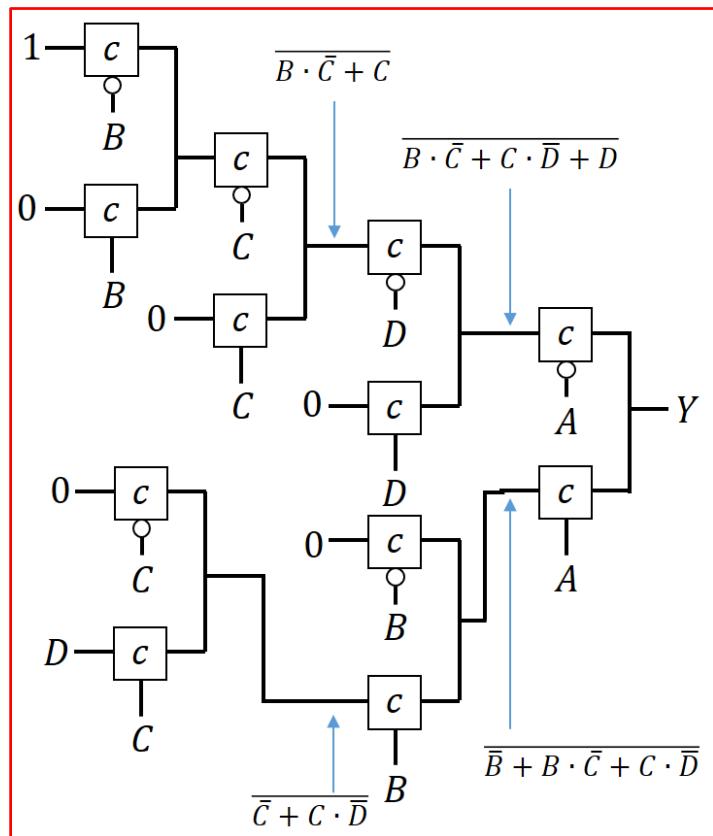
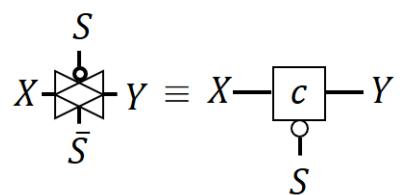
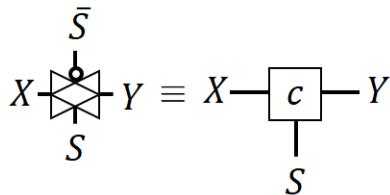


## Problem #2 (Transmission Gates, 10 points)

Design the following function (draw a transmission-gate-level schematic) using transmission gates only. Available input:  $A, B, C, D, 0 (GND), 1 (VDD)$ .

$$Y = \overline{A \cdot \bar{B} + B \cdot \bar{C} + C \cdot \bar{D} + D \cdot \bar{A}}$$

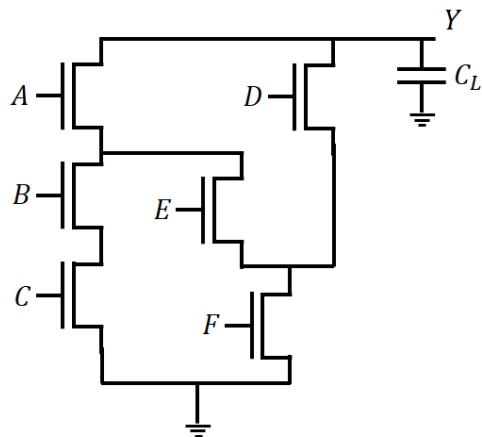
You can use the following symbols for transmission gates.



### Problem #3 (TR Sizing, 15 points)

Size the transistors in the following schematic to satisfy the given timing constraint.

- $R_n$ : The resistance of a 1X NFET.
- The delay should be  $\leq R_n C_L$ .
- Try to minimize the total transistor width.



1) (10 points) Size the transistors. You will get 10 points if the total TR width is  $\leq 22X$ .

- |    |   |   |
|----|---|---|
| A: | 2 | X |
| B: | 4 | X |
| C: | 4 | X |
| D: | 4 | X |
| E: | 4 | X |
| F: | 4 | X |

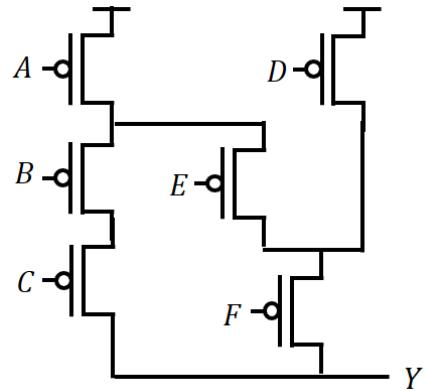
If the total TR width you obtained above is less than  $22X$ , you don't need to solve the second problem. You will get 15 points. If the width is greater than  $22X$ , you don't need to solve the second problem. You won't get the extra points. If the width is  $22X$ , solve the second problem.

2) (5 points) In your solution, you can upsize only one TR (and of course downsize all the other TRs as a result) to reduce the total TR width further. Which TR do you want to upsize?

Transistor E.

### Problem #4 (Logic Analysis, 10 points)

The following shows the PFET network of a static CMOS logic gate. Express the output  $Y$  as a Boolean function of the input variables  $A, B, C, D, E, F$ .



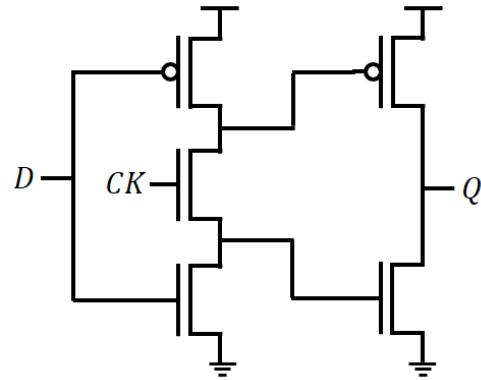
(You don't need to simplify the expression.)

Y becomes 1 when  $A=B=C=0$  or  $A=E=F=0$  or  $B=C=D=E=0$  or  $D=F=0$ .

$$Y = \bar{A} \cdot (\bar{B} \cdot \bar{C} + \bar{E} \cdot \bar{F}) + \bar{D} \cdot (\bar{B} \cdot \bar{C} \cdot \bar{E} + \bar{F})$$

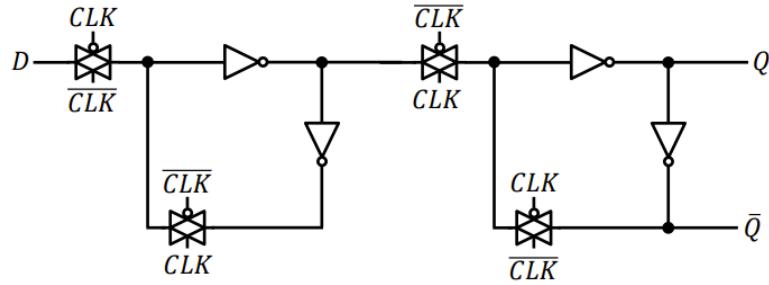
### Problem #5 (Logic Analysis, 10 points)

What does the following circuit do? Describe its functionality in as much detail as possible. (D: data input. CK: clock. Q: data output)



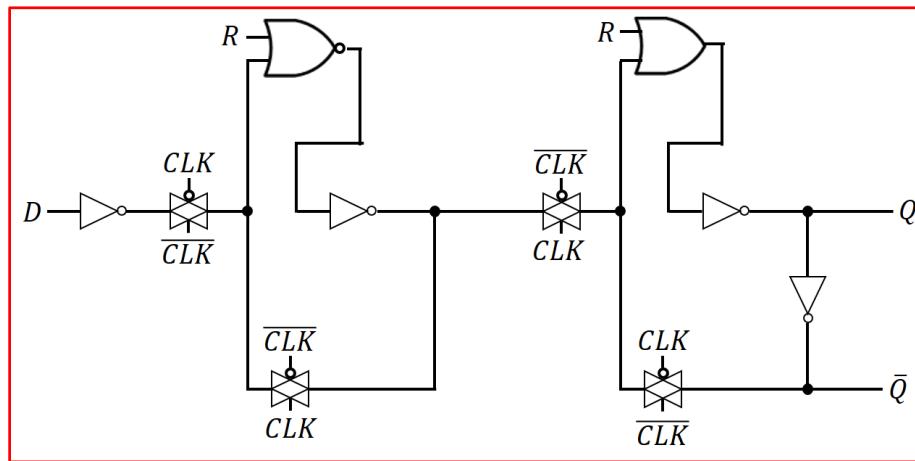
This is a positive-edge-triggered D flip-flop.

## Problem #6 (Logic Design, 15 points)



This is a positive-edge-triggered D flip-flop (FF). Can you modify the design (add some logic gates such as AND and OR to the design and/or remove some gates from the design, etc.) and add a control signal  $R$  to the design so that you can reset the D FF? Here is a more detailed description of the behavior of the logic.

- If  $R = 1$ ,  $Q$  becomes 0 immediately regardless of  $D$  and  $CLK$ .
- If  $R$  switches from 1 to 0,  $Q$  should still be 0 until the next clock rising edge.



**EE434**  
**ASIC and Digital Systems**

**Final Exam**  
**May 5, 2022. (1:30pm – 3:30pm)**  
**Instructor: Dae Hyun Kim**

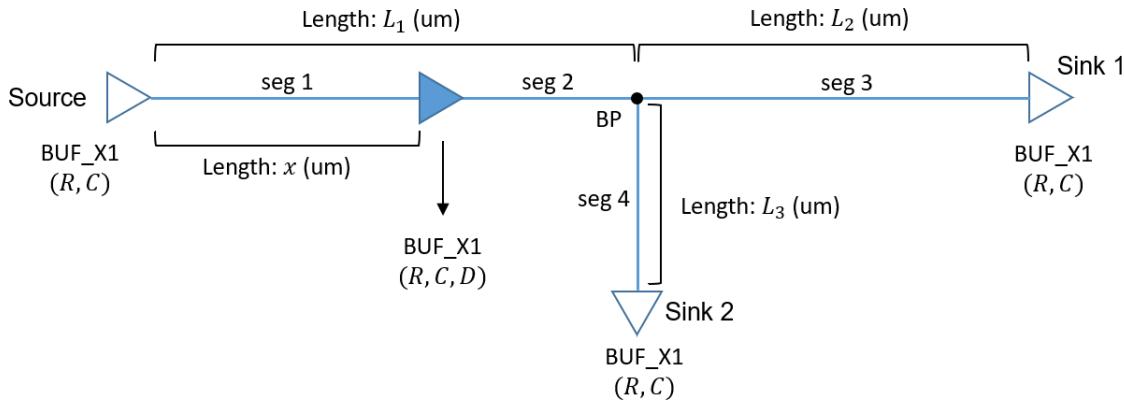
**Name:**

**WSU ID:**

Problem	Points	
1	80	
2	80	
3	50	
4	50	
Total	260	

## Problem #1 (Interconnect Optimization, 80 points)

The following figure shows a net optimized by a buffer.



Constants

- Output resistance of a cell:  $R$
- Input capacitance of a cell:  $C$
- Internal delay of a buffer:  $D$
- Unit wire resistance and capacitance:  $r$  ( $\Omega/\text{um}$ ),  $c$  ( $\text{fF}/\text{um}$ )
- Length of each subnet:  $L_1, L_2, L_3$  ( $\text{um}$ )

Independent variable

- Location of the buffer:  $x$  ( $0 < x < L_1$ )

We want to optimize the delay from the driver to Sink 1 or Sink 2. “BP” is a branch point. Answer the following questions.

(1) Replace each segment (seg 1, 2, 3, 4) by a PI model. Then, express the delay from the source to Sink 1 using the Elmore delay model. (10 points)

$$\begin{aligned}\tau = & rL_2 \left( \frac{cL_2}{2} + C \right) + r(L_1 - x) \left( \frac{c(L_1 - x)}{2} + cL_2 + cL_3 + 2C \right) \\ & + R(c(L_1 - x) + cL_2 + cL_3 + 2C) + D + rx \left( \frac{cx}{2} + C \right) + R(cx + C)\end{aligned}$$

(2) Find the optimal location ( $x$ ) of the buffer minimizing the delay from the source to Sink 1. (10 points)

$$\frac{d\tau}{dx} = Rc + rcx + rC - Rc + rc(x - L_1) - r(cL_2 + cL_3 + 2C) = 0$$

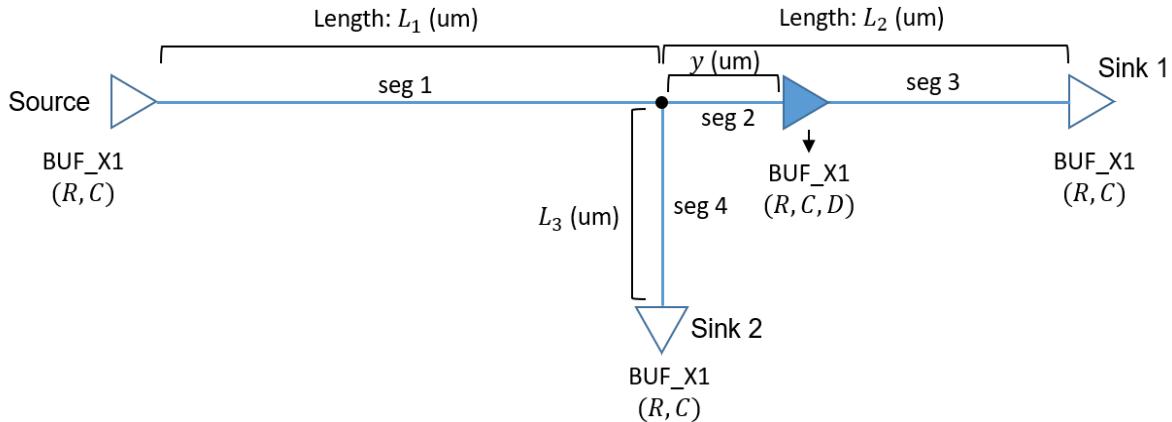
$$2cx = c(L_1 + L_2 + L_3) + C$$

$$\therefore x = \frac{1}{2}(L_1 + L_2 + L_3) + \frac{C}{2c}$$

(3) Answer the following questions for the problems (1)-(2) (Correct: +4 points, Wrong: -4 points, No answer: 0) for the minimization of the delay from the source to Sink 1.

- If  $L_1$  increases,  $x$  increases. (**True** / False)
- If  $L_2$  increases,  $x$  increases. (**True** / False)
- If  $L_3$  increases,  $x$  increases. (**True** / False)
- If  $C$  (the input capacitance of a cell) increases,  $x$  increases. (**True** / False)
- If  $c$  (the unit wire capacitance) increases,  $x$  increases. (**True** / **False**)

Now, we are moving the buffer to the branch connected to Sink 1 as follows:



(4) Replace each segment (seg 1, 2, 3, 4) by a PI model. Then, express the delay from the source to Sink 1 using the Elmore delay model. (10 points)

$$\begin{aligned} \tau = & r(L_2 - y) \left( \frac{c(L_2 - y)}{2} + C \right) + R(c(L_2 - y) + C) + D + ry \left( \frac{cy}{2} + C \right) \\ & + rL_1 \left( \frac{cL_1}{2} + cy + cL_3 + 2C \right) + R(cL_1 + cy + cL_3 + 2C) \end{aligned}$$

(5) Find the optimal location ( $y$ ) of the buffer minimizing the delay from the source to Sink 1. (10 points)

$$\frac{d\tau}{dy} = Rc + rcL_1 + rcy + rC - Rc + rc(y - L_2) - rC = 0$$

$$2cy = c(L_2 - L_1)$$

$$\therefore y = \frac{1}{2}(L_2 - L_1)$$

(6) Answer the following questions for the problems (4)-(5) (Correct: +5 points, Wrong: - 5 points, No answer: 0) for the minimization of the delay from the source to Sink 1.

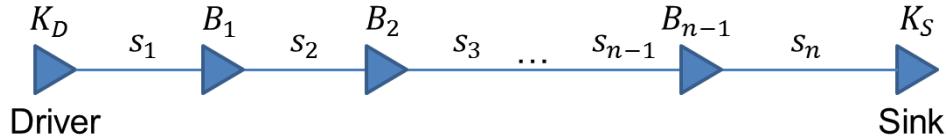
- If  $L_1$  increases,  $y$  increases. (True / **False**)
- If  $L_2$  increases,  $y$  increases. (**True** / False)

(7) Replace each segment (seg 1, 2, 3, 4) by a PI model. Then, express the delay from the source to Sink 2 using the Elmore delay model. (10 points)

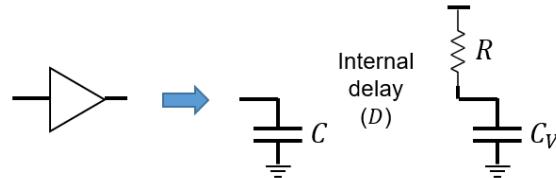
$$\tau = rL_3 \left( \frac{cL_3}{2} + C \right) + rL_1 \left( \frac{cL_1}{2} + cL_3 + cy + 2C \right) + R(cL_1 + cL_3 + y + 2C)$$

## Problem #2 (Interconnect Optimization, 80 points)

The following figure shows a net optimized by buffer insertion. The driver and the sink are denoted by  $K_D$  and  $K_S$ , respectively, and the inserted buffers are denoted by  $B_i$  ( $1 \leq i \leq n - 1$ ).  $n \geq 2$ , i.e., there is at least one buffer between the driver and the sink.



- Output resistance of a cell ( $K_D$  and  $B_i$ ):  $R$
- Input capacitance of a cell ( $K_S$  and  $B_i$ ):  $C$
- Delay of a buffer:  $D$
- Length of the  $i$ -th net ( $1 \leq i \leq n$ ):  $s_i$  (um)
- $\sum_{i=1}^n s_i = L$  (um)
- Unit wire resistance and capacitance:  $r$  ( $\Omega/\text{um}$ ),  $c$  ( $\text{fF}/\text{um}$ )
- Use the PI model to model each segment.
- Use the following model to model a cell (the driver, the sink, and the buffers). (Notice that there is an output capacitor with capacitance  $C_V$ .)



(1) Find the optimal locations of the buffers (i.e., express  $s_i$  as a function for the given parameters for each  $i = 1, 2, \dots, n$ ) minimizing the delay from the driver to the sink. (20 points)

$$\tau_i = R(C_V + cs_i + C) + rs_iC + \frac{1}{2}rcs_i^2$$

$$\tau = \sum \tau_i = nR(C_V + C) + RCL + rCL + \frac{1}{2}rc \left( \sum s_i^2 \right) + (n-1)D$$

$$\frac{\partial \tau}{\partial s_i} = rcs_i + rc(L - (s_1 + \dots + s_{n-1}))(-1) = 0, \therefore s_i = L - (s_1 + \dots + s_{n-1})$$

$$\therefore s_1 = \dots = s_{n-1}, s_i = \frac{L}{n}$$

(2) Then, find the optimal # buffers ( $n$ ) minimizing the delay (express  $n$  as a function of the given parameters). (20 points)

$$\tau = \sum \tau_i = nR(C_V + C) + RcL + rCL + \frac{1}{2}rc\frac{L^2}{n} + (n - 1)D$$

$$\frac{\partial \tau}{\partial n} = R(C_V + C) - \frac{1}{2}rc\frac{L^2}{n^2} + D = 0$$

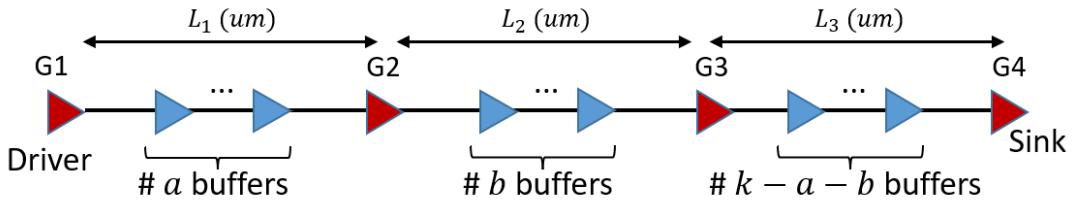
$$n = \sqrt{\frac{rcL^2}{2\{R(C_V + C) + D\}}}$$

(3) Answer the following questions for the problem (1) (Correct: +4 points, Wrong: -4 points, No answer: 0) for the minimization of the delay from the driver to the sink.

- If  $L$  increases,  $n$  increases. (True / False)
- If  $r$  increases,  $n$  increases. (True / False)
- If  $c$  increases,  $n$  increases. (True / False)
- If  $R$  increases,  $n$  increases. (True / False)
- If  $C$  increases,  $n$  increases. (True / False)
- If  $C_V$  increases,  $n$  increases. (True / False)
- If  $D$  increases,  $n$  increases. (True / False)
- If the output capacitance ( $C_V$ ) of  $B_{n-1}$  increases,  $s_n$  increases. (True / False)
- If the output capacitance ( $C_V$ ) of  $B_{n-1}$  increases,  $s_{n-1}$  increases. (True / False)
- If the output capacitance ( $C_V$ ) of  $B_{n-1}$  increases,  $s_1$  increases. (True / False)

### Problem #3 (Interconnect Optimization, 50 points)

The following shows three nets connecting four gates ( $G_1, G_2, G_3, G_4$ ). To reduce the delay, you insert buffers between  $G_1$  and  $G_2$ , between  $G_2$  and  $G_3$ , and between  $G_3$  and  $G_4$ . The total number of buffers you insert is  $k$  (constant). Thus, the total # of buffers you insert between  $G_1$  and  $G_2$  is  $a$ , the total # buffers you insert between  $G_2$  and  $G_3$  is  $b$ , and total # buffers you insert between  $G_3$  and  $G_4$  is  $k - a - b$ .



- All the gates and buffers are of the same type (they have the same output resistance, input capacitance, and internal delay). You can ignore the output capacitance of each gate.

(1) Find  $a$  and  $b$  minimizing the delay from the driver to the sink. (30 points)

First of all, if we insert  $t$  number of buffers into a net, we should evenly distribute them to minimize the delay (notice that all the gates are of the same type). Thus, if we insert  $a$  buffers into the first net, its delay is

$$\tau_1 = (a+1) \left\{ R \left( c \frac{L_1}{a+1} + C \right) + r \frac{L_1}{a+1} C + \frac{1}{2} r c \left( \frac{L_1}{a+1} \right)^2 \right\} + aD$$

Similarly, if we insert  $b$  buffers into the second net, its delay is

$$\tau_2 = (b+1) \left\{ R \left( c \frac{L_2}{b+1} + C \right) + r \frac{L_2}{b+1} C + \frac{1}{2} r c \left( \frac{L_2}{b+1} \right)^2 \right\} + bD$$

The delay of the third net is

$$\begin{aligned} \tau_3 &= (k - a - b + 1) \left\{ R \left( c \frac{L_3}{k - a - b + 1} + C \right) + r \frac{L_3}{k - a - b + 1} C \right. \\ &\quad \left. + \frac{1}{2} r c \left( \frac{L_3}{k - a - b + 1} \right)^2 \right\} + (k - a - b)D \end{aligned}$$

Thus, the total delay is

$$\begin{aligned}
\tau &= \tau_1 + \tau_2 + \tau_3 \\
&= RCL_1 + RC(a+1) + rL_1C + \frac{1}{2}rc \frac{L_1^2}{a+1} + RCL_2 + RC(b+1) + rL_2C \\
&\quad + \frac{1}{2}rc \frac{L_2^2}{b+1} + RCL_3 + RC(k-a-b+1) + rL_3C + \frac{1}{2}rc \frac{L_3^2}{k-a-b+1} + kD \\
&= RCL + RC(k+3) + rcL + kD + \frac{1}{2}rc \left\{ \frac{L_1^2}{a+1} + \frac{L_2^2}{b+1} + \frac{L_3^2}{k-a-b+1} \right\}
\end{aligned}$$

where ( $L = L_1 + L_2 + L_3$ )

$$\frac{\partial \tau}{\partial a} = 0 \Rightarrow -\frac{L_1^2}{(a+1)^2} + \frac{L_3^2}{(k-a-b+1)^2} = 0$$

$$\frac{\partial \tau}{\partial b} = 0 \Rightarrow -\frac{L_2^2}{(b+1)^2} + \frac{L_3^2}{(k-a-b+1)^2} = 0$$

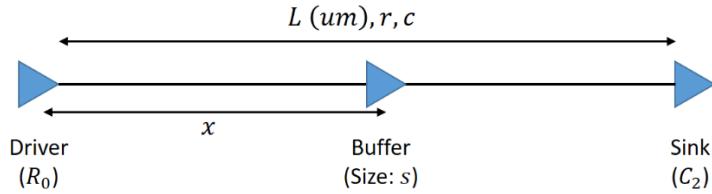
$$a = \frac{(k+2)L_1 - L_2 - L_3}{L_1 + L_2 + L_3}$$

$$b = \frac{(k+2)L_2 - L_1 - L_3}{L_1 + L_2 + L_3}$$

(2) Answer the following questions for the problem (1) (Correct: +5 points, Wrong: -5 points, No answer: 0) for the minimization of the delay from the driver to the sink.

- If  $k$  increases,  $a$  increases. (True / False)
- If  $k$  increases,  $b$  increases. (True / False)
- If the input capacitance of the sink increases,  $a$  increases. (True / False)
- If the input capacitance of the sink increases,  $b$  increases. (True / False)

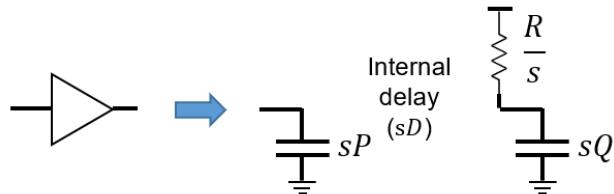
## Problem #4 (Interconnects, 40 points)



The figure shows a long wire whose length, unit resistance and capacitance are  $L$ ,  $r$ , and  $c$ , respectively. We want to insert a buffer ( $0 < x < L$ ). The size of the buffer is  $s$  and its output resistance, input capacitance, output capacitance, and internal delay are as follows:

- Output resistance:  $\frac{R}{s}$  (where  $R$  is a constant)
- Input capacitance:  $sP$  (where  $P$  is a constant)
- Output capacitance:  $sQ$  (where  $Q$  is a constant)
- Internal delay:  $sD$  (where  $D$  is a constant)

$R_0$  is the output resistance of the driver and  $C_2$  is the input capacitance of the sink. Notice that there are two variables,  $x$  and  $s$ . You can model the buffer using the following model.



(1) Express the total delay as a function of the variables and the constants. (10 points)

$$\begin{aligned}\tau = & R_0(cx + sP) + rxsP + \frac{1}{2}rcx^2 + sD + \frac{R}{s}(sQ + c(L - x) + C_2) + r(L - x)C_2 \\ & + \frac{1}{2}rc(L - x)^2\end{aligned}$$

(2) Find the optimal location of the buffer (you can treat  $s$  as a constant.) (10 points)

$$\frac{\partial \tau}{\partial x} = R_0c + rsP + rcx - \frac{RC}{s} - rC_2 + rc(x - L) = 0$$

$$x = \frac{1}{2} \left( L + \frac{C_2}{c} + \frac{R}{rs} - \frac{R_0}{r} - \frac{sP}{c} \right)$$

(3) Answer the following questions for the problem (1) (Correct: +5 points, Wrong: -5 points, No answer: 0) for the minimization of the delay from the driver to the sink.

Assume that  $R_0 > \frac{R}{s}$  and  $C_2 > sP$ .

- If  $C_2$  increases,  $x$  increases. (**True / False**)
- If  $c$  increases,  $x$  increases. (**True / False**)
- If  $R$  increases,  $x$  increases. (**True / False**)
- If  $r$  increases,  $x$  increases. (**True / False**)
- If  $R_0$  increases,  $x$  increases. (**True / False**)
- If  $P$  increases,  $x$  increases. (**True / False**)

**EE434**  
**ASIC and Digital Systems**

**Midterm Exam 1**  
**Feb. 25, 2022. (2:10pm – 3pm)**  
**Instructor: Dae Hyun Kim**

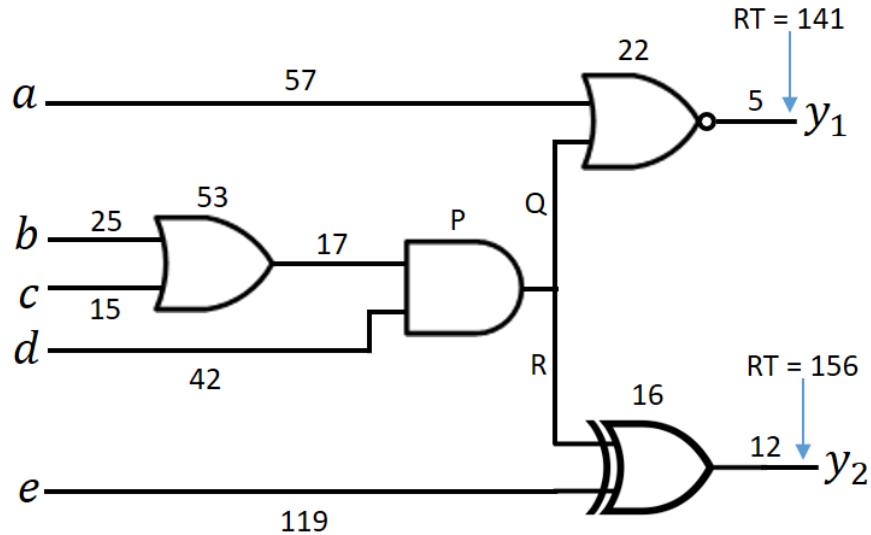
**Name:**

**WSU ID:**

Problem	Points	
1	20	
2	70	
3	60	
Total	150	

### Problem #1 (20 points)

The following figure shows the delays (in ps) of the gates and nets of a logic design. The required times (RTs) at the output nodes are given as shown below. All the inputs have zero arrival time. ( $P, Q, R > 0$ )



- 1) Express the slack at the output  $y_1$  using  $P$  and/or  $Q$  and/or  $R$ . (You can also use MIN, MAX operators.) (5 points)

$$AT = \text{MAX}(57, 95 + P + Q) + 27 = 122 + P + Q$$

$$\text{Slack} = RT - AT = 141 - (122 + P + Q) = 19 - P - Q$$

- 2) Express the slack at the output  $y_2$  using  $P$  and/or  $Q$  and/or  $R$ . (You can also use MIN, MAX operators.) (5 points)

$$AT = \text{MAX}(95 + P + R, 119) + 28 = \text{MAX}(P + R, 24) + 123$$

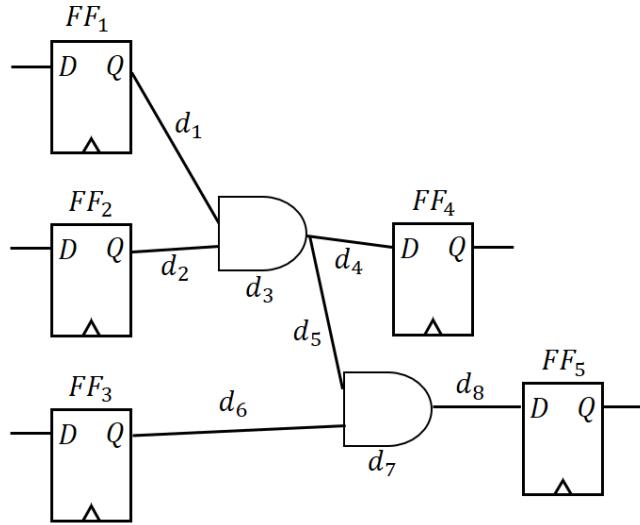
$$\text{Slack} = RT - AT = 128 - \text{MAX}(95 + P + R, 119) = 33 - \text{MAX}(P + R, 24)$$

- 3) Suppose  $P, Q, R$  are positive integers and satisfy  $P + Q + R = 30$ . Find  $P, Q, R$  that make the slacks at both  $y_1$  and  $y_2$  positive or zero. (Note: You can just find one set of  $(P, Q, R)$  because there are many  $(P, Q, R)$  sets satisfying the given condition.) (10 points)

$$P + Q \leq 19 \text{ and } P + R \leq 33.$$

$$(P, Q, R) = (14, 2, 14)$$

## Problem #2 (70 points)



- $d_k$ : gate and net delays ( $k = 1 \sim 8$ )
- $s_k$ : setup time of  $FF_k$  ( $k = 1 \sim 5$ )
- $h_k$ : hold time of  $FF_k$  ( $k = 1 \sim 5$ )
- $x_k$ : delay from the clock source to the clock pin of  $FF_k$  ( $k = 1 \sim 5$ )
- $T_{CLK}$ : clock period
- $\text{MIN}, \text{MAX}$ : MIN, MAX operators

(1) Show all the setup time inequalities in the design (10 points). (You can show all the inequalities without using the MIN, MAX operators, or show them using the MIN, MAX operators.)

$$\text{MAX}(x_1 + c_1 + d_1, x_2 + c_2 + d_2) + d_3 + d_4 \leq x_4 + T_{CLK} - s_4$$

$$\text{MAX}(\text{MAX}(x_1 + c_1 + d_1, x_2 + c_2 + d_2) + d_3 + d_5, x_3 + c_3 + d_6) + d_7 + d_8 \leq x_5 + T_{CLK} - s_5$$

(2) Show all the hold time inequalities in the design (10 points).

$$x_4 + h_4 \leq \text{MIN}(x_1 + c_1 + d_1, x_2 + c_2 + d_2) + d_3 + d_4$$

$$x_5 + h_5 \leq \text{MIN}(\text{MIN}(x_1 + c_1 + d_1, x_2 + c_2 + d_2) + d_3 + d_5, x_3 + c_3 + d_6) + d_7 + d_8$$

Notation: FF<sub>x</sub>-FF<sub>y</sub> is the path from the output of FF<sub>x</sub> to the input of FF<sub>y</sub>.

Answer the following questions. To fix a problem, you can increase or decrease the delay of only one net or gate (i.e., only one of  $d_1 \sim d_8$ ). If there are multiple ways to fix a problem, you should choose the best one. If you cannot fix it, just say “cannot fix it” and explain why.

(3) FF<sub>2</sub> – FF<sub>4</sub> has a hold time violation. How would you fix it? (5 points)

Increase  $d_2$ .

(4) FF<sub>1</sub> – FF<sub>4</sub> and FF<sub>2</sub> – FF<sub>4</sub> have setup time violations. How would you fix both at the same time? (5 points)

Decrease  $d_4$ . (Decreasing  $d_3$  might violate the hold time from FF<sub>1</sub> (or FF<sub>2</sub>) to FF<sub>5</sub>.)

(5) FF<sub>1</sub> – FF<sub>4</sub> and FF<sub>2</sub> – FF<sub>4</sub> have hold time violation. How would you fix both at the same time? (5 points)

Increase  $d_4$ . (Increasing  $d_3$  might violate the setup time from FF<sub>1</sub> (or FF<sub>2</sub>) to FF<sub>5</sub>.)

(6) FF<sub>1</sub> – FF<sub>5</sub> and FF<sub>1</sub> – FF<sub>4</sub> have setup time violations. How would you fix both at the same time? (5 points)

Decrease  $d_1$ .

(7) FF<sub>1</sub> – FF<sub>4</sub> has a setup time violation and FF<sub>1</sub> – FF<sub>5</sub> has a hold time violation. How would you fix both at the same time? (5 points)

Cannot fix both at the same time. Can fix only one of them.

(8) FF<sub>1</sub> – FF<sub>4</sub> and FF<sub>2</sub> – FF<sub>5</sub> have setup time violations. How would you fix both at the same time? (5 points)

Decrease  $d_3$ .

(9)  $FF_1 - FF_4$  has a hold time violation and  $FF_2 - FF_5$  has a setup time violation. How would you fix both at the same time? (5 points)

Cannot fix both at the same time. Can fix only one of them.

(10)  $FF_1 - FF_4$  has a setup time violation and  $FF_2 - FF_5$  has a hold time violation. How would you fix both at the same time? (5 points)

Cannot fix both at the same time. Can fix only one of them.

(11)  $FF_1 - FF_5$  and  $FF_3 - FF_5$  have setup time violations. How would you fix both at the same time? (5 points)

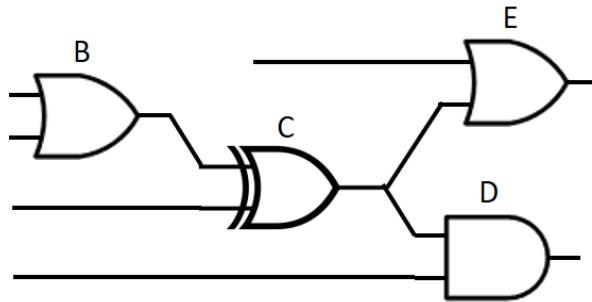
Decrease  $d_7$  or  $d_8$ .

(12)  $FF_1 - FF_5$  and  $FF_2 - FF_5$  hold time violations and  $FF_3 - FF_5$  has a setup time violation. How would you fix both at the same time? (5 points)

Cannot fix both at the same time.

### Problem #3 (60 points)

The following shows three gates B, C, and D, which are in the middle of a circuit.



"A gate has a setup (or hold) time violation" means that at least one of the paths going through the gate violates given setup (or hold) time constraints.

Answer the following questions. Correct: +3 points, Wrong: -3 points, No answer: 0.

(1) It is possible that gate B has a setup-time violation and gate E has a setup-time violation too. (**True** / False)

(2) It is possible that gate B has a setup-time violation and gate E has a hold-time violation. (**True** / False)

(3) It is possible that gate B has a hold-time violation and gate E has a setup-time violation. (**True** / False)

(5) It is possible that gate B has a setup-time violation, but gate E does not have any setup-time violation. (**True** / False)

(6) It is possible that gate B has a setup-time violation, but gate E does not have any hold-time violation. (**True** / False)

(7) It is possible that gate B has a hold-time violation, but gate E does not have any setup-time violation. (**True** / False)

(8) It is possible that gate B has a hold-time violation, but gate E does not have any hold-time violation. (**True** / False)

(9) It is possible that gate D has a setup-time violation and gate E has a setup-time violation too. (**True** / False)

(10) It is possible that gate D has a setup-time violation and gate E has a hold-time violation. (**True** / False)

(11) It is possible that gate D has a hold-time violation and gate E has a setup-time violation. (**True** / False)

(12) It is possible that gate D has a hold-time violation and gate E has a hold-time violation too. (**True** / False)

(13) It is possible that gate D has a setup-time violation, but gate E does not have any setup-time violation. (**True** / False)

(14) It is possible that gate D has a setup-time violation, but gate E does not have any hold-time violation. (**True** / False)

(15) It is possible that gate D has a hold-time violation, but gate E does not have any setup-time violation. (**True** / False)

(16) It is possible that gate D has a hold-time violation, but gate E does not have any hold-time violation. (**True** / False)

(17) It is possible that gates B and D have setup-time violations, but gate C does not have any setup-time violation. (True / **False**)

(18) It is possible that gates B and D have setup-time violations, but gate C does not have any hold-time violation. (**True** / False)

(19) It is possible that gates B and D have hold-time violations, but gate C does not have any setup-time violation. (**True** / False)

(20) It is possible that gates B and D have hold-time violations, but gate C does not have any hold-time violation. (True / **False**)

**EE434**  
**ASIC and Digital Systems**

**Midterm Exam 2**  
**Apr. 13, 2022. (2:10pm – 3pm)**  
**Instructor: Dae Hyun Kim**

**Name:**

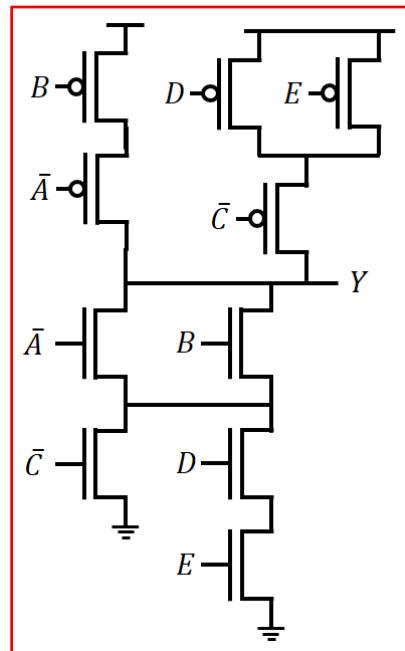
**WSU ID:**

Problem	Points	
1	10	
2	10	
3	10	
4	10	
5	10	
6	20	
7	20	
8	20	
9	10	
Total	120	

### Problem #1 (Static CMOS, 10 points)

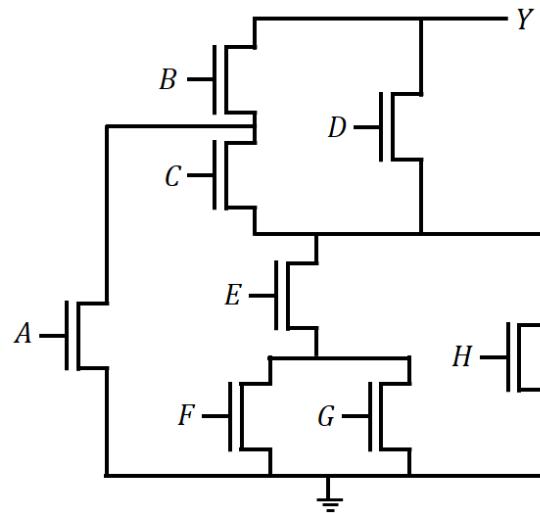
Design the following function (draw a transistor-level schematic) using the static CMOS logic design methodology. Available input:  $A, B, C, D, E$ . Try to minimize the # TRs.

$$Y = A \cdot \bar{B} + C \cdot (\bar{D} + \bar{E})$$



## Problem #2 (Static CMOS, 10 points)

The following schematic shows the NFET network of a static CMOS logic gate.

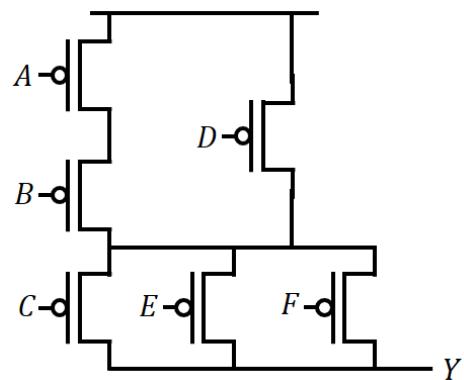


Express the output  $Y$  as a function of the input signals.

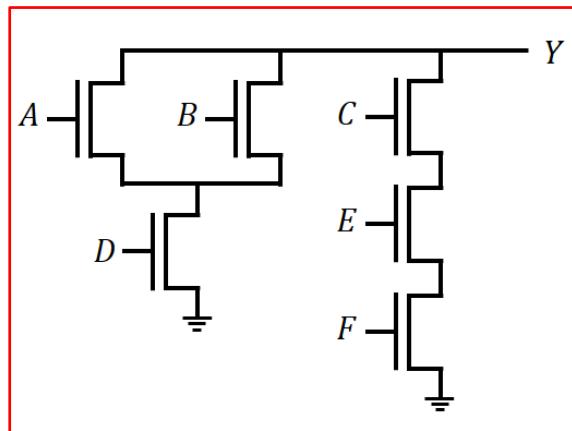
$$Y = \overline{A \cdot (B + C \cdot D) + (B \cdot C + D) \cdot (H + E \cdot (F + G))}$$

### Problem #3 (Static CMOS, 10 points)

The following schematic shows the PFET network of a static CMOS logic gate.



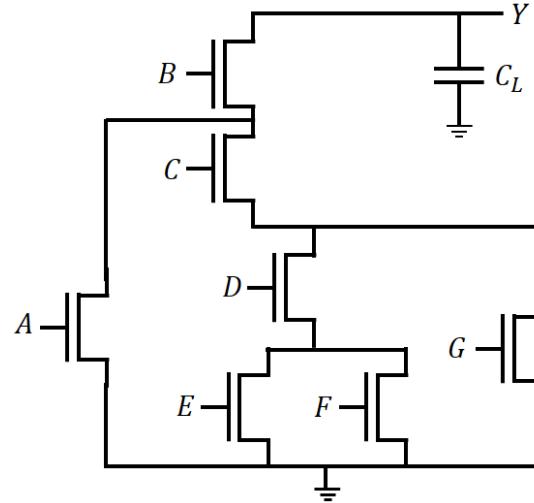
Design an NFET network for the logic gate.



### Problem #4 (TR Sizing, 10 points)

$k = \frac{\mu_n}{\mu_p}$ .  $R_n$  is the resistance of a 1X NFET (whose width is  $w_{min}$ ). “ $h \times$ ” for a TR means that the width of the TR is  $h \cdot w_{min}$ .

The following figure shows the NFET network of a static CMOS logic gate.



Size the transistors in the NFET network (show the size of each TR below). Timing constraint:  $\tau_f \leq R_n C_L$  ( $\tau_f$  is the fall delay). Try to minimize the total transistor width.

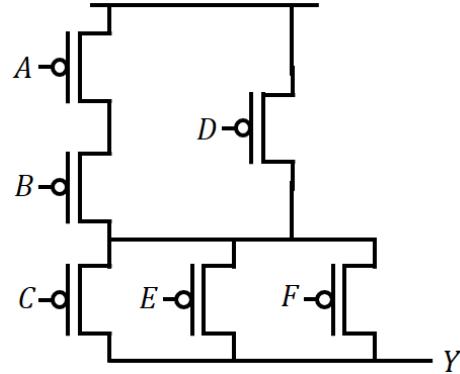
A: 4/3X      B: 4X      C: 4X      D: 4X

E: 4X      F: 4X      G: 2X

## Problem #5 (TR Sizing, 10 points)

$k = \frac{\mu_n}{\mu_p}$ .  $R_n$  is the resistance of a 1X NFET (whose width is  $w_{min}$ ). “ $h \times$ ” for a TR means that the width of the TR is  $h \cdot w_{min}$ .

The following figure shows the PFET network of a static CMOS logic gate.



Size the transistors in the NFET network (show the size of each TR below). Timing constraint:  $\tau_r \leq 0.25R_nC_L$  ( $\tau_r$  is the rise delay). Try to minimize the total transistor width.

A: 12kX

B: 12kX

C: 12kX

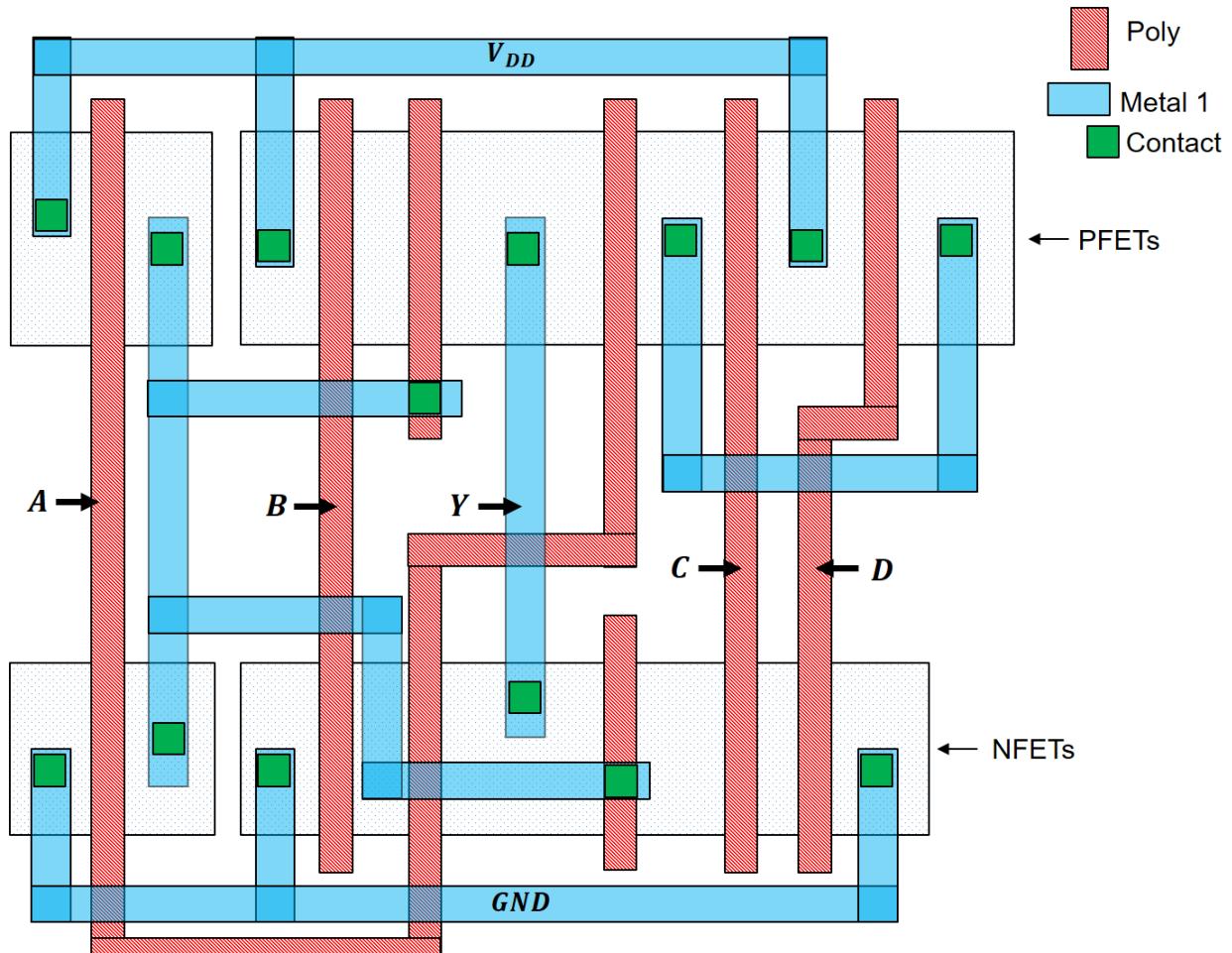
D: 6kX

E: 12kX

F: 12kX

## Problem #6 (Layout, 20 points)

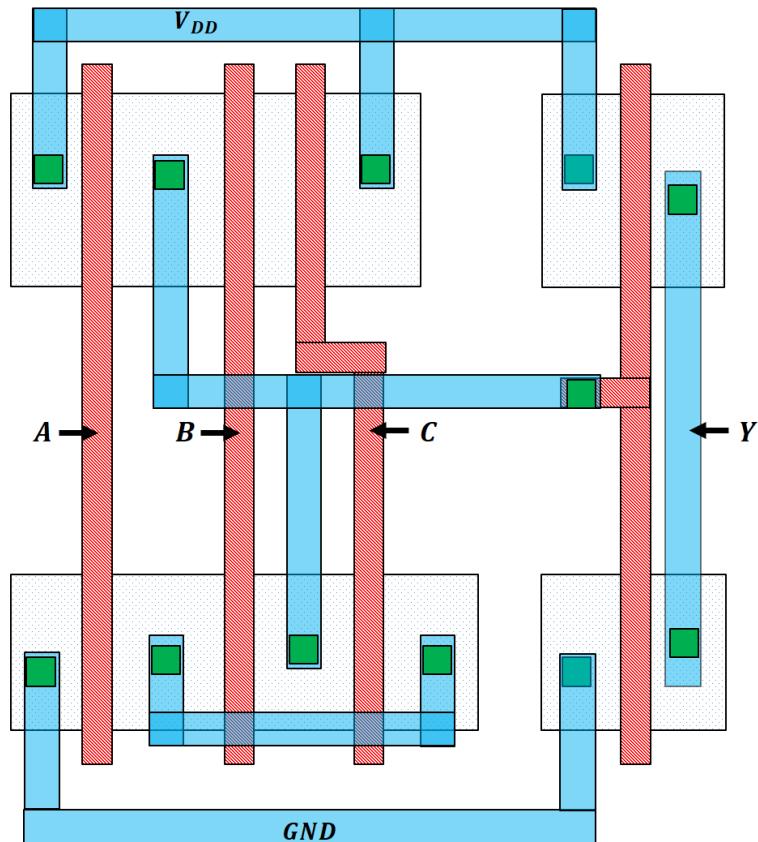
Express the output  $Y$  as a Boolean function of the input signals, A, B, C, and D.



$$Y = A \cdot \bar{B} + \bar{A} \cdot (\bar{C} \cdot \bar{D})$$

### Problem #7 (Layout, 20 points)

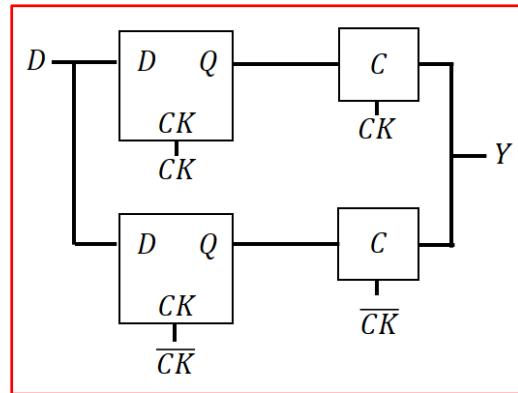
Draw a layout for  $Y = A \cdot (B + C)$ . (Use solid-line rectangles/polygons for Metal 1 wires, dotted rectangles/polygons for the poly (gate), and black-filled rectangles for contacts.)



### Problem #8 (Logic Design, 20 points)

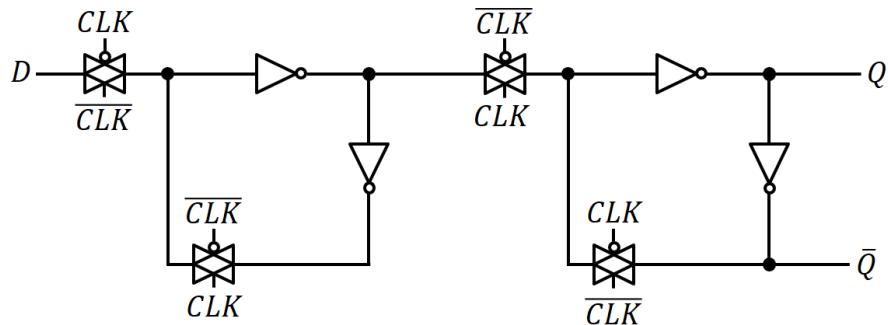
Design a dual-edge D flip flop (it captures the data input D whenever the clock signal goes high or low) using two single-edge D flip flops and some more gates (AND, NAND, NOR, ..., transmission gates, tristate inverters, tristate buffers, ...).

- Both of the two single-edge D flip flops could be positive-edge-triggered (PET) or negative-edge-triggered (NET), or one of them can be PET and the other can be NET.



### Problem #9 (Logic Design, 10 points)

Modify the following D flip flop design so that you can add an “EN(ABLE)” signal.



- If EN is 1, the flip flop works as a D flip flop.
- If EN is 0, the flip flop output holds the current output value (i.e., it does not capture the input data D).

Replace  $D$  by  $EN \cdot D + \overline{EN} \cdot Q$ .

**EE434**  
**ASIC and Digital Systems**

**Final Exam**

**May 2, 2023. (1:30pm – 3:30pm)**

**Instructor: Dae Hyun Kim**

**Name:**

**WSU ID:**

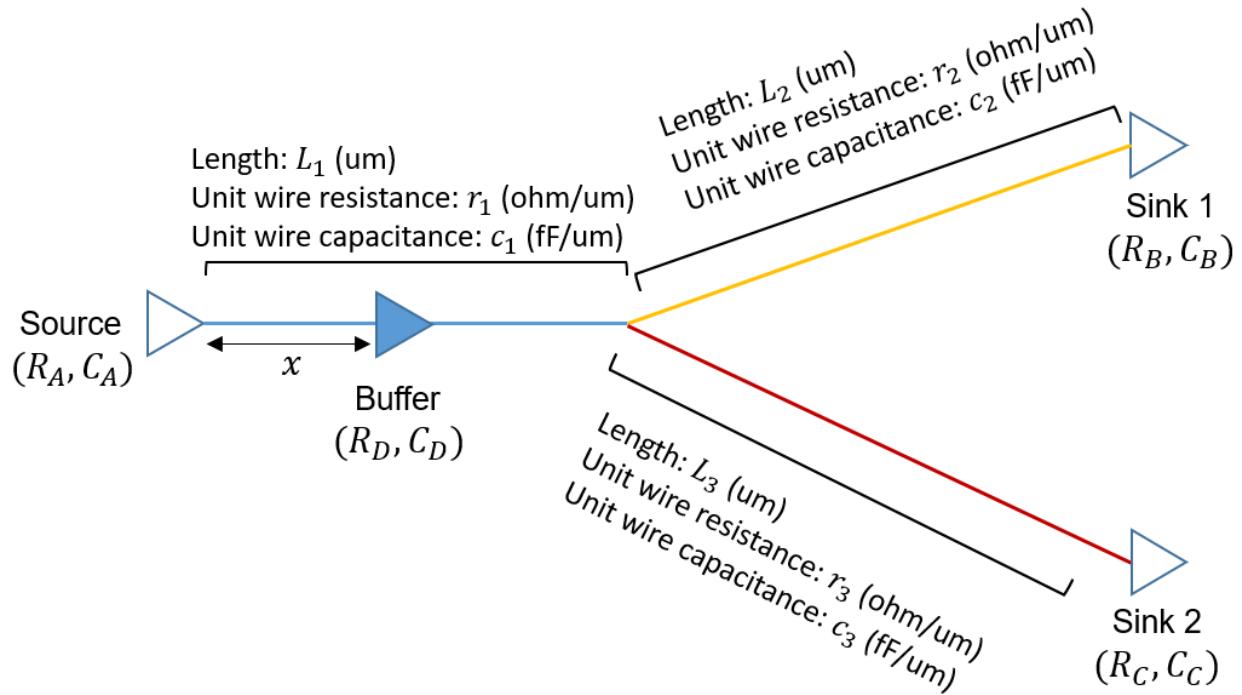
Problem	Points	
1	100	
2	90	
3	100	
4	90	
Total	380	

\* Allowed: Books, cheat sheets, class notes, notebooks, calculators, watches, laptops, smartphones, tablet PCs, etc.

\* Not allowed: Chatting apps, iMessage, etc. (basically, don't talk to anyone)

## Problem #1 (Interconnect Optimization, 100 points)

The following figure shows a net optimized by a buffer.



Notice that  $(R, C)$  is the (output resistance, input capacitance) of the corresponding cell.  $x$  is the optimal location of the buffer ( $0 < x < L_1$ ). We ignore the internal delay of the buffer. Use the PI model for each wire segment for delay estimation. We want to minimize the delay from the source to Sink 1 in this problem.

(1) Express the delay from the source to Sink 1 before the buffer insertion. (10 points)

$$d_{1,before} = r_2 L_2 \left( C_B + \frac{c_2 L_2}{2} \right) + r_1 L_1 \left( C_B + C_C + c_2 L_2 + c_3 L_3 + \frac{c_1 L_1}{2} \right) + R_A (C_B + C_C + c_1 L_1 + c_2 L_2 + c_3 L_3)$$

(2) Express the delay from the source to Sink 1 after the buffer insertion. (10 points)

$$d_{1,after} = r_1 x \left( C_D + \frac{c_1 x}{2} \right) + R_A (C_D + c_1 x) + r_2 L_2 \left( C_B + \frac{c_2 L_2}{2} \right) + r_1 (L_1 - x) \left( C_B + C_C + c_2 L_2 + c_3 L_3 + \frac{c_1 (L_1 - x)}{2} \right) + R_D (C_B + C_C + c_1 (L_1 - x) + c_2 L_2 + c_3 L_3)$$

(3) Assuming the buffer insertion reduces the delay from the source to Sink 1, find the optimal location  $x$  of the buffer. (10 points)

$$\frac{dd_{1,after}}{dx} = r_1 C_D + r_1 c_1 x + R_A c_1 - r_1 (C_B + C_C + c_2 L_2 + c_3 L_3) - r_1 c_1 (L_1 - x) - R_D c_1 = 0.$$

$$2r_1 c_1 x = r_1 (C_B + C_C + c_1 L_1 + c_2 L_2 + c_3 L_3) + R_D c_1 - (r_1 C_D + R_A c_1)$$

$$\therefore x = \frac{L_1}{2} + \frac{1}{2c_1} (C_B + C_C + c_2 L_2 + c_3 L_3) + \frac{R_D - R_A}{2r_1} - \frac{C_D}{2c_1}$$

(4) Answer the following questions. Correct: +5 points. Wrong: -5 points. Min: 0 points.

- If  $L_1$  increases, then  $x$  increases too. (**True** / **False**)
- If  $L_2$  increases, then  $x$  increases too. (**True** / **False**)
- If  $L_3$  increases, then  $x$  increases too. (**True** / **False**)
- If  $R_A$  increases, then  $x$  increases too. (**True** / **False**)
- If  $R_D$  increases, then  $x$  increases too. (**True** / **False**)
- If  $C_B$  increases, then  $x$  increases too. (**True** / **False**)
- If  $C_C$  increases, then  $x$  increases too. (**True** / **False**)
- If  $C_D$  increases, then  $x$  increases too. (**True** / **False**)

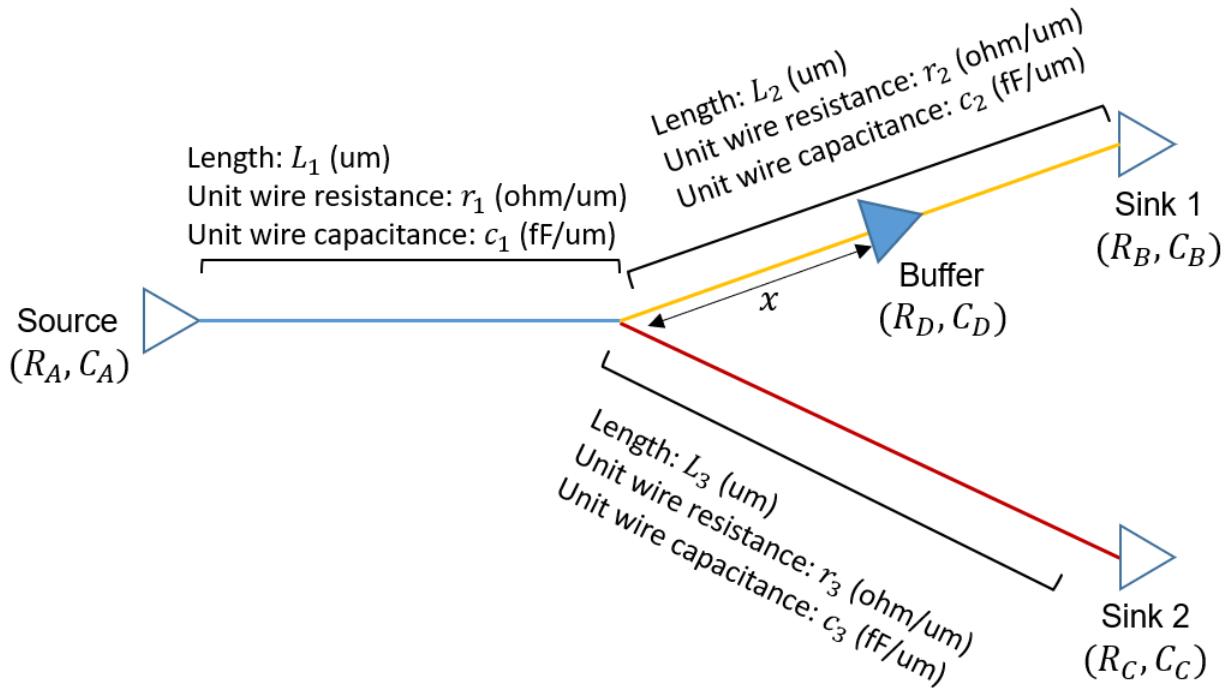
(5) Answer the following questions. Correct: +5 points. Wrong: -5 points. Min: 0 points.

Assume  $R_D < R_A$ . Assume  $C_D$  is negligible.

- If  $r_1$  increases, then  $x$  increases too. (**True** / **False**)
- If  $r_2$  increases, then  $x$  increases too. (**True** / **False**)
- If  $r_3$  increases, then  $x$  increases too. (**True** / **False**)
- If  $c_1$  increases, then  $x$  increases too. (**True** / **False**)
- If  $c_2$  increases, then  $x$  increases too. (**True** / **False**)
- If  $c_3$  increases, then  $x$  increases too. (**True** / **False**)

## Problem #2 (Interconnect Optimization, 100 points)

The following figure shows a net optimized by a buffer.



Notice that  $(R, C)$  is the (output resistance, input capacitance) of the corresponding cell.  $x$  is the optimal location of the buffer ( $0 < x < L_2$ ). We ignore the internal delay of the buffer. Use the PI model for each wire segment for delay estimation. We want to minimize the delay from the source to Sink 1 in this problem.

(1) Express the delay from the source to Sink 1 after the buffer insertion. (10 points)

$$d_{1,\text{after}} = r_2(L_2 - x) \left( C_B + \frac{c_2(L_2 - x)}{2} \right) + R_D(C_B + c_2(L_2 - x)) + r_2x \left( C_D + \frac{c_2x}{2} \right) + r_1L_1 \left( C_C + C_D + c_2x + c_3L_3 + \frac{c_1L_1}{2} \right) + R_A(C_C + C_D + c_1L_1 + c_2x + c_3L_3)$$

(2) Assuming the buffer insertion reduces the delay from the source to Sink 1, find the optimal location  $x$  of the buffer. (10 points)

$$\frac{dd_{1,\text{after}}}{dx} = -r_2C_B - r_2c_2(L_2 - x) - R_DC_D + r_2C_D + r_2c_2x + r_1L_1c_2 + R_AC_2 = 0.$$

$$2r_2c_2x = r_2(C_B - C_D + c_2L_2) + R_DC_2 - r_1L_1c_2 - R_AC_2$$

$$\therefore x = \frac{L_2}{2} + \frac{1}{2c_2}(C_B - C_D) + \frac{R_D - R_A}{2r_2} - \frac{r_1L_1}{2r_2}$$

(3) Answer the following questions. Correct: +5 points. Wrong: -5 points. Min: 0 points.

- If  $L_1$  increases, then  $x$  increases too. (True / False)
- If  $L_2$  increases, then  $x$  increases too. (True / False)
- If  $R_A$  increases, then  $x$  increases too. (True / False)
- If  $R_D$  increases, then  $x$  increases too. (True / False)
- If  $C_B$  increases, then  $x$  increases too. (True / False)
- If  $C_D$  increases, then  $x$  increases too. (True / False)

(4) Answer the following questions. Correct: +10 points. Wrong: -10 points. Min: 0 points.

Notice that the optimal location  $x$  of the buffer found above (in Problem 2-2) is independent of  $L_3$  and  $C_C$  (i.e.,  $x$  does not have  $L_3$  and  $C_C$ ). In fact, however, the optimal location  $x$  is also dependent on  $L_3$  and  $C_C$ . Answer the following questions NOT based on  $x$  you found above, BUT based on your intuition and knowledge considering the real-world factors.

- If  $L_3$  increases, then  $x$  increases too. (True / False)
- If  $C_C$  increases, then  $x$  increases too. (True / False)

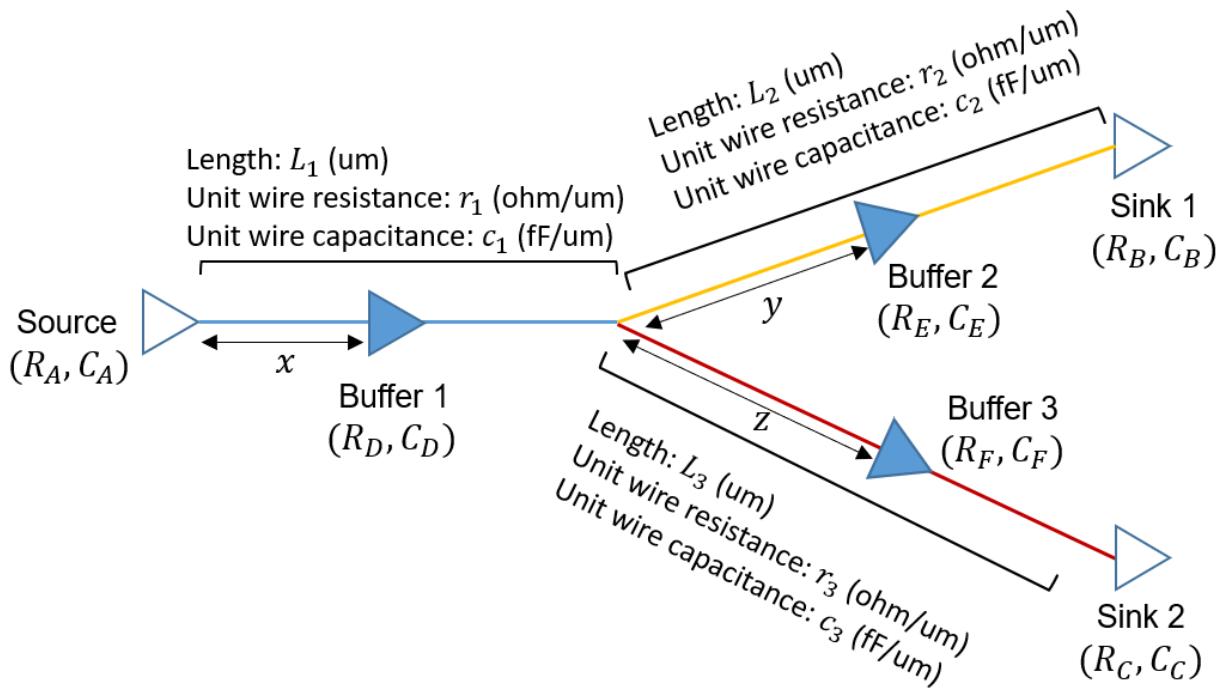
(5) Answer the following questions. Correct: +5 points. Wrong: -5 points. Min: 0 points.

Assume  $R_D < R_A$ . Assume  $C_D$  is negligible.

- If  $r_1$  increases, then  $x$  increases too. (True / False)
- If  $r_2$  increases, then  $x$  increases too. (True / False)
- If  $c_1$  increases, then  $x$  increases too. (True / False)
- If  $c_2$  increases, then  $x$  increases too. (True / False)

### Problem #3 (Interconnect Optimization, 100 points)

The following figure shows a net optimized by three buffers.



Notice that  $(R, C)$  is the (output resistance, input capacitance) of the corresponding cell.  $x, y, z$  are the optimal locations of Buffer 1, 2, 3, respectively ( $0 < x < L_1, 0 < y < L_2, 0 < z < L_3$ ). We ignore the internal delays of the buffers. We want to minimize both the delay from the source to Sink 1 and the delay from the source to Sink 2 in this problem.

Do not use the mathematical models we studied in the class. Instead, use your intuition to answer the following questions.

(1) Answer the following questions. Correct: +5 points. Wrong: -5 points. Min: 0 points.

- If  $L_1$  increases, then  $x$  increases too. (True / False)
- If  $L_1$  increases, then  $y$  increases too. (True / False)
- If  $L_2$  increases, then  $x$  increases too. (True / False)
- If  $L_2$  increases, then  $y$  increases too. (True / False)
- If  $L_2$  increases, then  $z$  increases too. (True / False)

(2) Answer the following questions. Correct: +5 points. Wrong: -5 points. Min: 0 points.

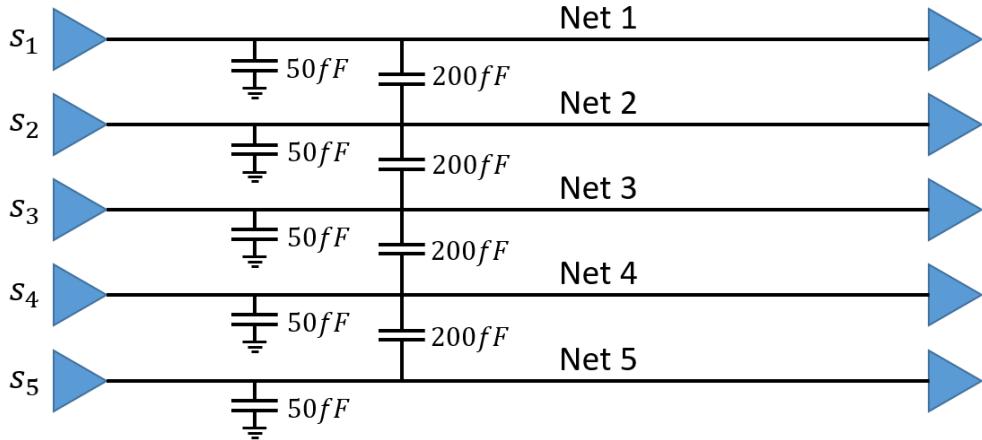
- If  $R_A$  increases, then  $x$  increases too. (True / False)
- If  $R_A$  increases, then  $y$  increases too. (True / False)
- If  $R_D$  increases, then  $x$  increases too. (True / False)
- If  $R_D$  increases, then  $y$  increases too. (True / False)

- If  $R_E$  increases, then  $x$  increases too. (**True** / **False**)
- If  $R_E$  increases, then  $y$  increases too. (**True** / **False**)
- If  $R_E$  increases, then  $z$  increases too. (**True** / **False**)

(3) Answer the following questions. Correct: +5 points. Wrong: -5 points. Min: 0 points.

- If  $C_D$  increases, then  $x$  increases too. (**True** / **False**)
- If  $C_D$  increases, then  $y$  increases too. (**True** / **False**)
- If  $C_E$  increases, then  $x$  increases too. (**True** / **False**)
- If  $C_E$  increases, then  $y$  increases too. (**True** / **False**)
- If  $C_E$  increases, then  $z$  increases too. (**True** / **False**)
- If  $C_B$  increases, then  $x$  increases too. (**True** / **False**)
- If  $C_B$  increases, then  $y$  increases too. (**True** / **False**)
- If  $C_B$  increases, then  $z$  increases too. (**True** / **False**)

## Problem #4 (Coupling, 40 points)



This figure shows a bus composed of five nets. The following shows the data we are going to transfer through the bus:

$$0(\text{initial}) \rightarrow 5 \rightarrow 11 \rightarrow 4$$

We compute the power consumption for a signal transition as follows:

$$P = C_{t,eff} V_{DD}^2$$

where  $V_{DD} = 1V$  and  $C_{t,eff}$  is the sum of the effective capacitances.

(1) Compute the power consumption of the data transfer (three transitions) using the conventional 5-bit binary codes for the data. (20 points)

$$00000 \rightarrow 00101 \rightarrow 01011 \rightarrow 00100$$

- $00000 \rightarrow 00101: C = (50 + 200 + 200) + (50 + 200) = 700fF$
- $00101 \rightarrow 01011: C = (50 + 200 + 400) + (50 + 400 + 400) + (50 + 400) = 1950fF$
- $00101 \rightarrow 00100: C = (50 + 200 + 400) + (50 + 400 + 400) + (50 + 400) + (50) = 2000fF$

Thus,  $P = 4650fW = 4.65pW$

(2) Compute the power consumption of the data transfer using the 5-bit forbidden pattern free crosstalk avoidance code. Ignore the power consumption of the encoder/decoder. (20 points)

$$00000 \rightarrow 01100 \rightarrow 11110 \rightarrow 00111$$

- $00000 \rightarrow 01100: C = (50 + 200) + (50 + 200) = 500fF$
- $01100 \rightarrow 11110: C = (50 + 200) + (50 + 200 + 200) = 700fF$
- $11110 \rightarrow 00111: C = (50) + (50 + 200) + (50 + 200) = 550fF$

Thus,  $P = 1750fW = 1.75pW$

(3) Let's estimate the efficiency of the forbidden pattern free crosstalk avoidance code (FPF-CAC). (50 points)

- The Fibonacci sequence is defined as follows:
  - $f_1 = 1$
  - $f_2 = 1$
  - $f_{k+2} = f_{k+1} + f_k$
- Complete the following. (10 points)
  - $f_3 = 2$
  - $f_4 = 3$
  - $f_5 = 5$
  - $f_6 = 8$
  - $f_7 = 13$
  - $f_8 = 21$
  - $f_9 = 34$
  - $f_{10} = 55$
  - $f_{11} = 89$
  - $f_{12} = 144$
  - $f_{13} = 233$
  - $f_{14} = 377$
- In page 33 of the interconnect lecture note, the # bits we need to represent a number in FPF-CAC is determined by the “MSB stage”. For a given data (number)  $v$ , we should find the max.  $f_{m+1}$  satisfying  $v \geq f_{m+1}$ . For example, if  $v = 19$ ,  $f_1 \leq \dots \leq f_7 \leq 19 < f_8$ , so the max.  $f_{m+1}$  satisfying the above condition is  $f_7$ . Thus,  $m + 1 = 7$ , so  $m = 6$ , so we need 6 bits to represent 19 in FPF-CAC.
- The max. data (number) that an  $m$ -bit FPF-CAC code can represent is as follows (Complete the following from  $m = 7$  to 12) (20 points):
  - $m = 1: 1$
  - $m = 2: 2$
  - $m = 3: 4$
  - $m = 4: 7$
  - $m = 5: 12$
  - $m = 6: 20$
  - $m = 7: 33$
  - $m = 8: 54$
  - $m = 9: 88$
  - $m = 10: 143$
  - $m = 11: 232$
  - $m = 12: 376$
- Similarly, the max. data (number) that an  $m$ -bit conventional binary code (e.g.,  $20 = 10100_2$ ) can represent is as follows (Complete the following from  $m = 7$  to 12) (10 points):
  - $m = 1: 1$
  - $m = 2: 3$

- $m = 3: 7$
  - $m = 4: 15$
  - $m = 5: 31$
  - $m = 6: 63$
  - $m = 7: 127$
  - $m = 8: 255$
  - $m = 9: 511$
  - $m = 10: 1023$
  - $m = 11: 2047$
  - $m = 12: 4095$
- Thus, the efficiency is estimated as follows (Complete the following from  $m = 7$  to 12) (10 points):
- $m = 1: 1/1 = 1.0$
  - $m = 2: 2/3 = 0.67$
  - $m = 3: 4/7 = 0.57$
  - $m = 4: 7/15 = 0.47$
  - $m = 5: 12/31 = 0.39$
  - $m = 6: 20/63 = 0.32$
  - $m = 7: 33/127 = 0.26$
  - $m = 8: 54/255 = 0.21$
  - $m = 9: 88/511 = 0.17$
  - $m = 10: 143/1023 = 0.14$
  - $m = 11: 232/2047 = 0.11$
  - $m = 12: 376/4095 = 0.09$

**EE434**  
**ASIC and Digital Systems**

**Midterm Exam 1**  
**Mar. 1, 2023. (2:10pm – 3pm)**  
**Instructor: Dae Hyun Kim**

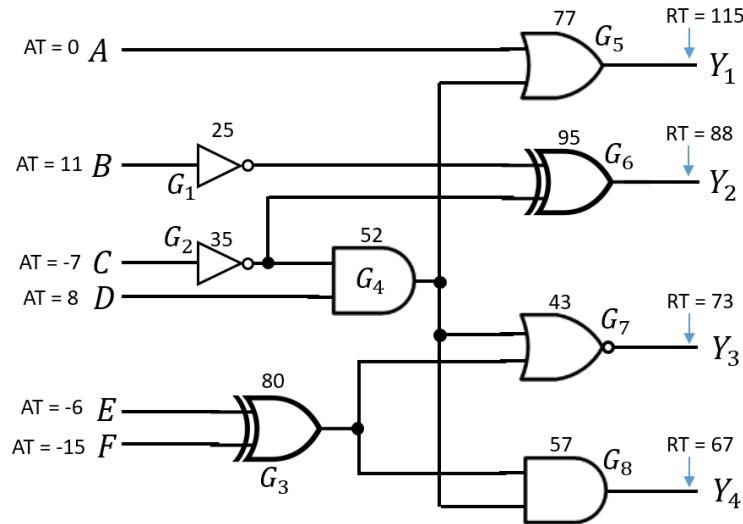
**Name:**

**WSU ID:**

Problem	Points	
1	20	
2	50	
3	90	
Total	160	

## Problem #1 (20 points)

The following figure shows the delays of the gates of a logic design. All the nets have zero delay. Required times (RTs) at the output nodes and Arrival times (ATs) at the input nodes are given as shown below.



(1) (5 points) Express the slack at the output  $Y_1$ .

$$AT = \text{MAX}(0 + 77, -7 + 35 + 52 + 77, 8 + 52 + 77) = 157$$

$$\text{Slack} = RT - AT = 115 - 157 = -42$$

(2) (5 points) Express the slack at the output  $Y_3$ .

$$AT = \text{MAX}(-7 + 35 + 52 + 43, 8 + 52 + 43, -6 + 80 + 43, -15 + 80 + 43) = 123$$

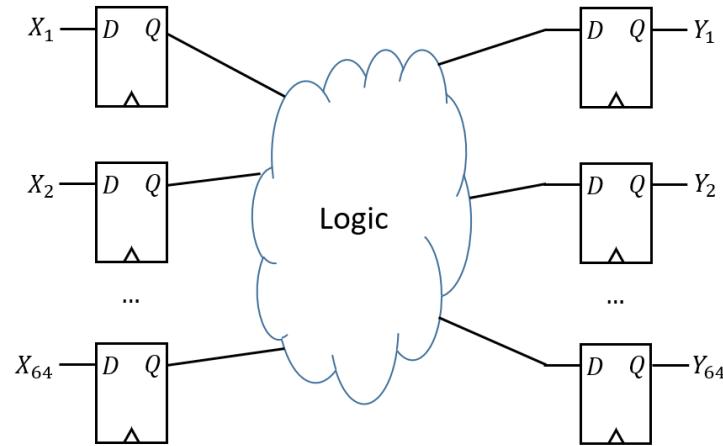
$$\text{Slack} = RT - AT = 73 - 123 = -50$$

(3) (10 points) If the delay of  $G_2$  goes down by 5, does the slack of the path from  $D$  to  $Y_3$  go up? If no, just say no. If yes, calculate the increment of the slack.

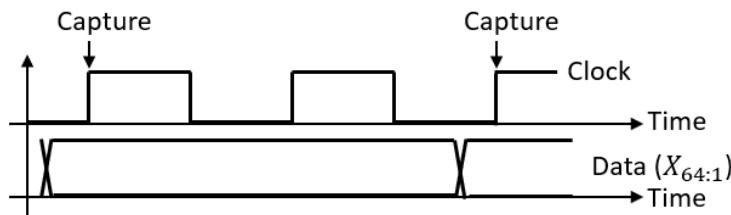
The slack of  $D-Y_3$  is  $RT-AT$  where  $RT=73$  and  $AT$  is the sum of the delays of  $G4$  and  $G7$ . Thus, it is not affected by the delay of  $G2$ . The answer is NO.

## Problem #2 (50 points)

The following figure shows a two-cycle pipeline stage. “Two-cycle” means capturing occurs every two cycles.



The following figure shows two waveforms of the clock and data.



- $L_{i-k}$ : the logic delay from input  $X_i$  to output  $Y_k$  (e.g.,  $L_{64-2}$  means the logic delay from input  $X_{64}$  to output  $Y_2$ )
- $s$ : setup time of a flip-flop
- $h$ : hold time of a flip-flop
- $x$ : delay from the clock source to the clock pin of a flip-flop
- $c$ : clk-to-Q delay of a flip-flop
- $T_{CLK}$ : clock period
- $\text{MIN}, \text{MAX}$ : MIN, MAX operators

(1) Show all the setup time inequalities of the design (10 points). (Use the MIN, MAX operators.)

$$x + c + \text{MAX}(L_{1-1}, \dots, L_{64-64}) \leq x + 2 \cdot T_{CLK} - s$$

(2) Show all the hold time inequalities of the design (10 points). (Use the MIN, MAX operators.)

$$x + h \leq x + c + \text{MIN}(L_{1-1}, \dots, L_{64-64})$$

Answer the following questions for the figure above. Correct: +5 points, Wrong: -5 points, No answer: 0. Min: 0 points.

(3) Assume WNS < 0 and TNS < 0. In this case, if  $s$  goes up, WNS goes down. (**True** / False)

(4) Assume WNS < 0 and TNS < 0. In this case, if  $s$  goes up, TNS goes down. (**True** / False)

(5) Assume WNS < 0 and TNS < 0. In this case, if  $c$  goes up, WNS goes down. (**True** / False)

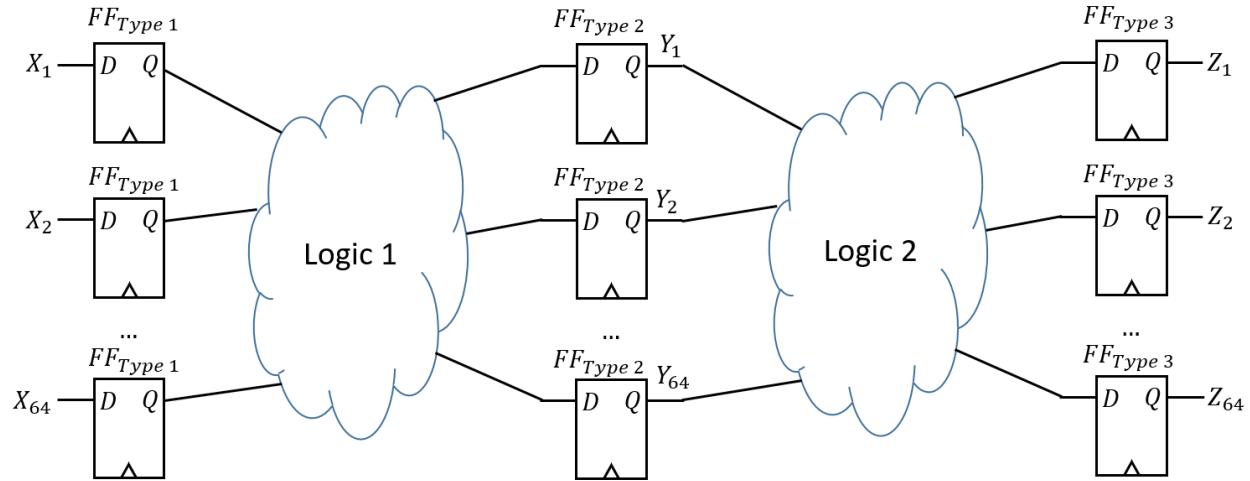
(6) Assume WNS < 0 and TNS < 0. In this case, if  $c$  goes up, TNS goes down. (**True** / False)

(7) Assume WNS < 0 and TNS < 0. In this case, if  $x$  goes up, WNS goes down. (**True** / **False**)

(8) Assume WNS < 0 and TNS < 0. In this case, if  $x$  goes up, TNS goes down. (**True** / **False**)

### Problem #3 (90 points)

The following shows two pipeline stages of a system.



- $L_{i-k}$ : the logic delay from  $X_i$  to  $Y_k$  (e.g.,  $L_{64-2}$  means the logic delay from  $X_{64}$  to  $Y_2$ )
- $M_{i-k}$ : the logic delay from  $Y_i$  to  $Z_k$  (e.g.,  $M_{8-16}$  means the logic delay from  $Y_8$  to  $Z_{16}$ )
- $s_k$ : setup time of a flip-flop of type k
- $h_k$ : hold time of a flip-flop of type k
- $x_k$ : delay from the clock source to the clock pin of a flip-flop of type k
- $c_k$ : clk-to-Q delay of a flip-flop of type k

Answer the following questions. Correct: +5 points, Wrong: -5 points, No answer: 0. Min: 0 points. WNS and TNS are obtained for the whole system (including both the first and second pipeline stages.)

(1) Assume WNS < 0 and TNS < 0. In this case, if  $s_2$  goes up, WNS goes down. (True / False)

(2) Assume WNS < 0 and TNS < 0. In this case, if  $s_2$  goes up, TNS goes down. (True / False)

(3) Assume WNS < 0 and TNS < 0. In this case, if  $s_3$  goes up, WNS goes down. (True / False)

(4) Assume WNS < 0 and TNS < 0. In this case, if  $s_3$  goes up, TNS goes down. (True / False)

(5) Assume WNS < 0 and TNS < 0. In this case, if  $x_2$  goes up, WNS goes down. (True / False)

(6) Assume WNS < 0 and TNS < 0. In this case, if  $x_2$  goes up, TNS goes down. (True / False)

(7) Assume WNS < 0 and TNS < 0. In this case, if  $x_3$  goes up, WNS goes down. (True / False)

(8) Assume WNS < 0 and TNS < 0. In this case, if  $x_3$  goes up, TNS goes down. (True / False)

(9) Assume WNS < 0 and TNS < 0. In this case, if  $c_2$  goes up, WNS goes down. (True / False)

(10) Assume WNS < 0 and TNS < 0. In this case, if  $c_2$  goes up, TNS goes down. (True / False)

(11) Assume WNS < 0 and TNS < 0. In this case, if  $c_3$  goes up, WNS goes down. (True / False)

(12) Assume WNS < 0 and TNS < 0. In this case, if  $c_3$  goes up, TNS goes down. (True / False)

(13) Assume WNS < 0 and TNS < 0. In this case, if both  $s_2$  and  $s_3$  go up, WNS goes down. (True / False)

(14) Assume WNS < 0 and TNS < 0. In this case, if both  $s_2$  and  $s_3$  go up, TNS goes down. (True / False)

(15) Assume WNS < 0 and TNS < 0. In this case, if both  $x_2$  and  $x_3$  go up, WNS goes down. (True / False)

(16) Assume WNS < 0 and TNS < 0. In this case, if both  $x_2$  and  $x_3$  go up, TNS goes down. (True / False)

(17) Assume WNS < 0 and TNS < 0. In this case, if both  $c_2$  and  $c_3$  go up, WNS goes down. (True / False)

(18) Assume WNS < 0 and TNS < 0. In this case, if both  $c_2$  and  $c_3$  go up, TNS goes down. (True / False)

(19) Assume WNS < 0 and TNS < 0. In this case, if both  $s_1$  and  $s_2$  go up, WNS goes down. (True / False)

(20) Assume WNS < 0 and TNS < 0. In this case, if both  $s_1$  and  $s_2$  go up, TNS goes down. (True / False)

(21) Assume WNS < 0 and TNS < 0. In this case, if both  $x_1$  and  $x_2$  go up, WNS goes down. (True / False)

(22) Assume WNS < 0 and TNS < 0. In this case, if both  $x_1$  and  $x_2$  go up, TNS goes down. (True / False)

(23) Assume WNS < 0 and TNS < 0. In this case, if both  $c_1$  and  $c_2$  go up, WNS goes down. (**True** / False)

(24) Assume WNS < 0 and TNS < 0. In this case, if both  $c_1$  and  $c_2$  go up, TNS goes down. (**True** / False)

(25) Assume WNS < 0 and TNS < 0. In this case, if  $s_1$ ,  $s_2$ , and  $s_3$  go up, WNS goes down. (**True** / False)

(26) Assume WNS < 0 and TNS < 0. In this case, if  $s_1$ ,  $s_2$ , and  $s_3$  go up, TNS goes down. (**True** / False)

(27) Assume WNS < 0 and TNS < 0. In this case, if  $x_1$ ,  $x_2$ , and  $x_3$  go up, WNS goes down. (True / **False**)

(28) Assume WNS < 0 and TNS < 0. In this case, if  $x_1$ ,  $x_2$ , and  $x_3$  go up, TNS goes down. (True / **False**)

(29) Assume WNS < 0 and TNS < 0. In this case, if  $c_1$ ,  $c_2$ , and  $c_3$  go up, WNS goes down. (**True** / False)

(30) Assume WNS < 0 and TNS < 0. In this case, if  $c_1$ ,  $c_2$ , and  $c_3$  go up, TNS goes down. (**True** / False)

**EE434**  
**ASIC and Digital Systems**

**Midterm Exam 2**  
**Apr. 7, 2023. (2:10pm – 3pm)**  
**Instructor: Dae Hyun Kim**

**Name:**

**WSU ID:**

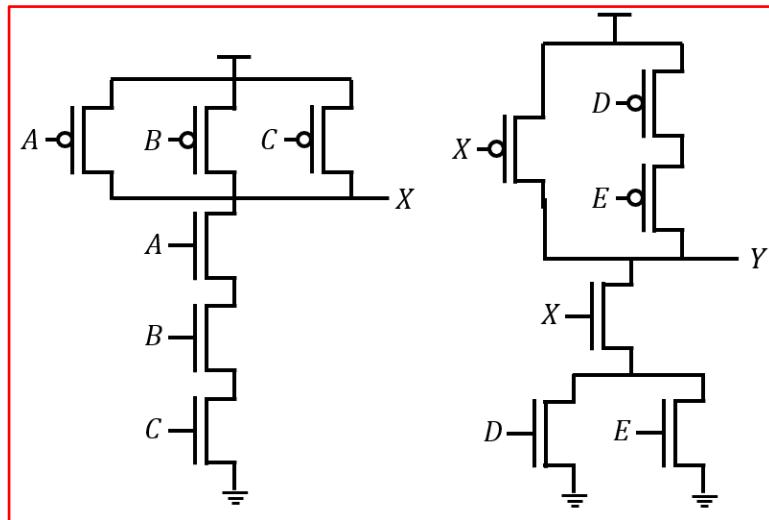
Problem	Points	
1	10	
2	10	
3	10	
4	10	
5	20	
6	20	
7	10	
Total	90	

### Problem #1 (Static CMOS, 10 points)

Design the following function (draw a transistor-level schematic) using the static CMOS logic design methodology. Available input:  $A, B, C, D, E$ . Try to minimize the # TRs.

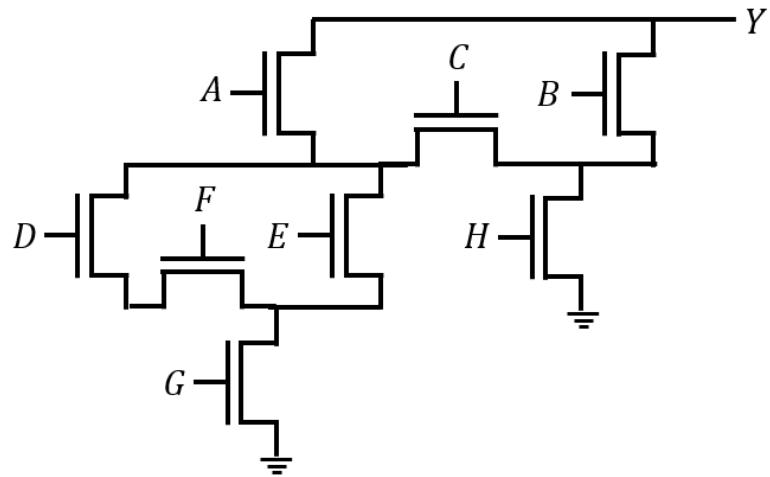
$$Y = A \cdot B \cdot C + \bar{D} \cdot \bar{E}$$

$$Y = \overline{A \cdot B \cdot C + \bar{D} \cdot \bar{E}} = \overline{A \cdot B \cdot \bar{C} \cdot (D + E)}$$



## Problem #2 (Static CMOS, 10 points)

The following schematic shows the NFET network of a static CMOS logic gate.

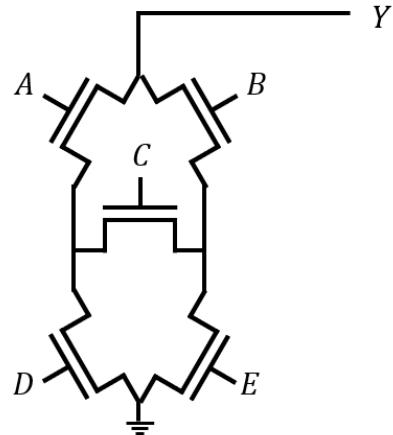


Express the output  $Y$  as a function of the input signals.

$$Y = \overline{A(CH + (DF + E)G)} + \overline{B(H + C(DF + E)G)}$$

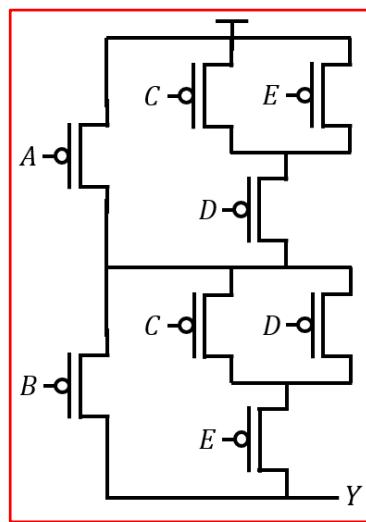
### Problem #3 (Static CMOS, 10 points)

The following schematic shows the NFET network of a static CMOS logic gate.



Design a PFET network for the logic gate. Try to minimize # TRs.

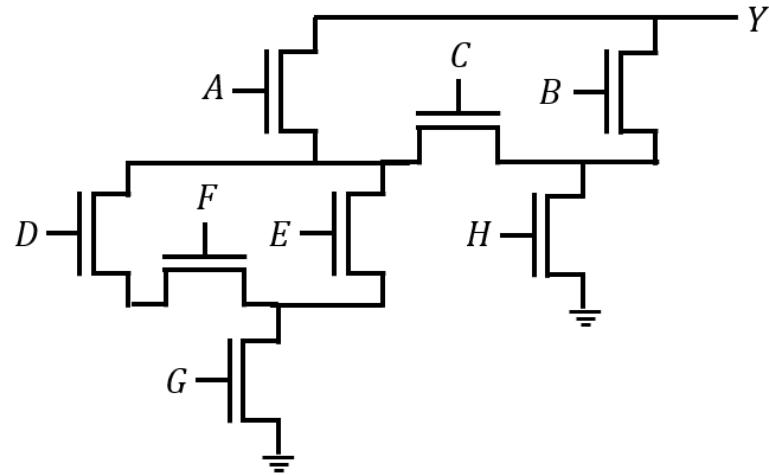
$$Y = \overline{A(D + CE) + B(E + CD)}$$



### Problem #4 (TR Sizing, 10 points)

$k = \frac{\mu_n}{\mu_p}$ .  $R_n$  is the resistance of a 1X NFET (whose width is  $w_{min}$ ). “ $h \times$ ” for a TR means that the width of the TR is  $h \cdot w_{min}$ .

The following figure shows the NFET network of a static CMOS logic gate.



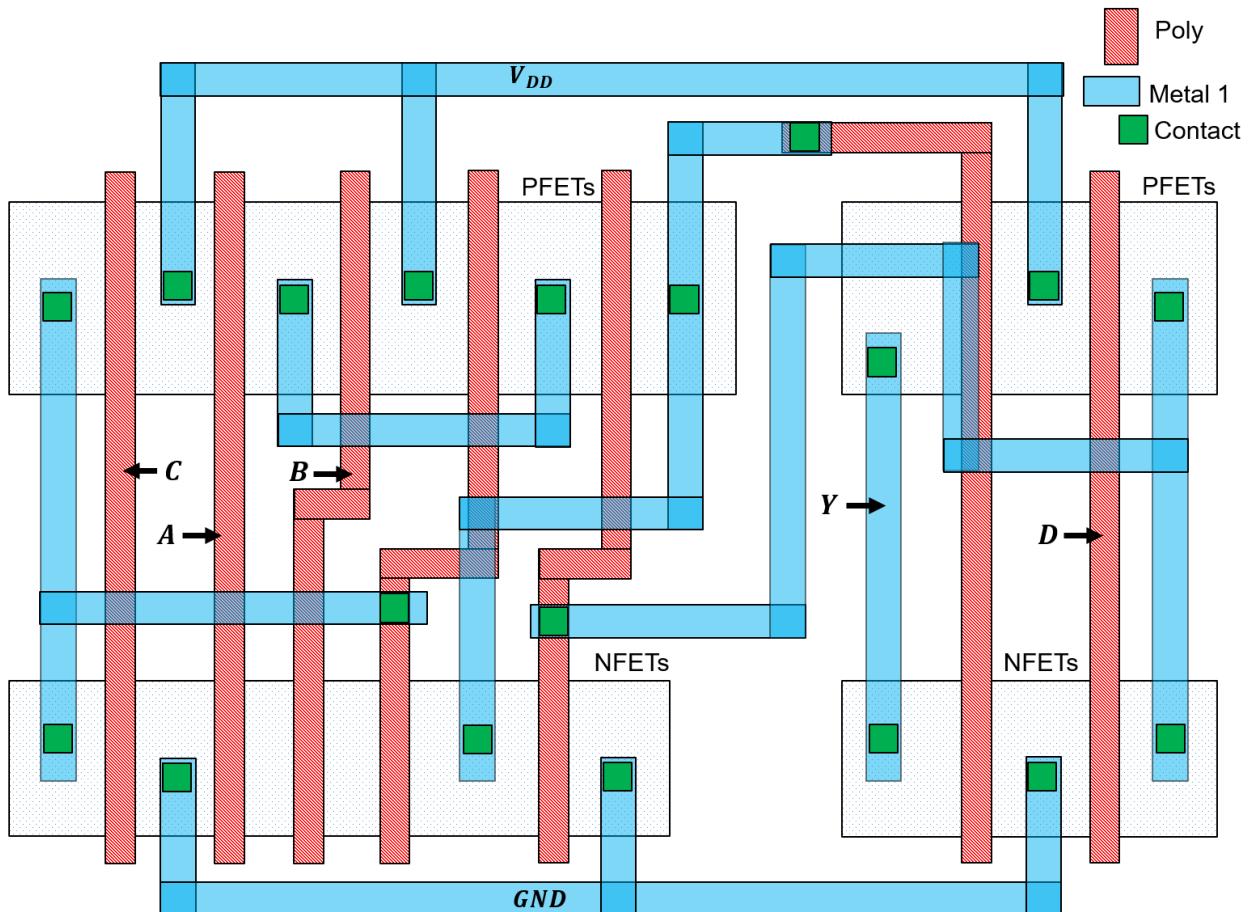
Size the transistors in the NFET network (show the size of each TR below). Timing constraint:  $\tau_f \leq R_n C_L$  ( $\tau_f$  is the worst-case fall delay). Try to minimize the total transistor width.

A: 2.5X      B: 5X      C: 5X      D: 5X

E: 2.5X      F: 5X      G: 5X      H: 2.5X

## Problem #5 (Layout, 20 points)

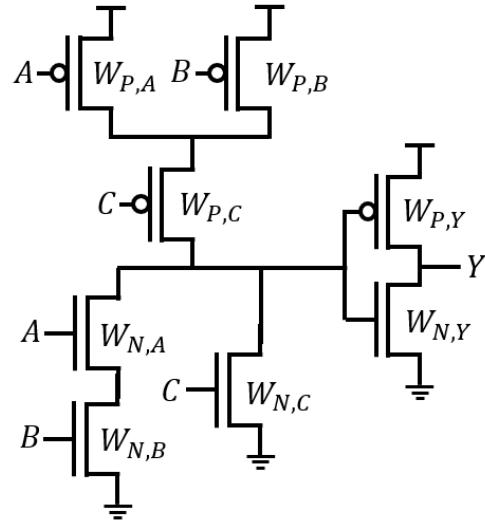
Express the output  $Y$  as a Boolean function of the input signals, A, B, C, and D.



$$Y = AB\bar{C} + \bar{D}$$

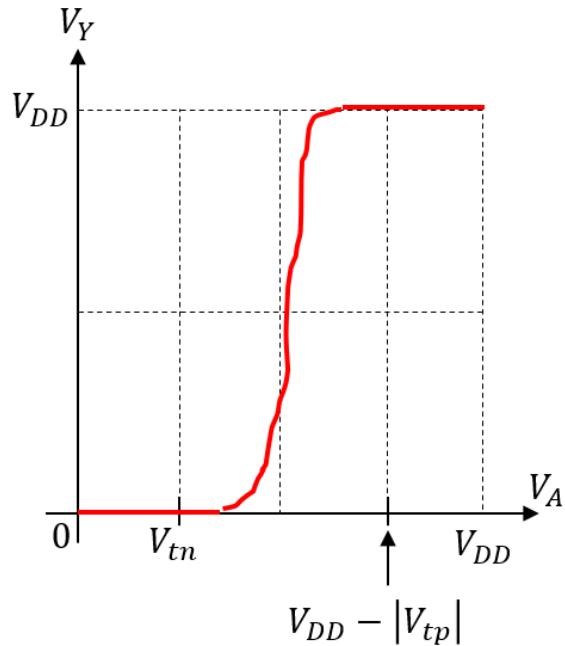
## Problem #6 (DC Analysis, 20 points)

The following shows a design of  $Y = A \cdot B + C$ .



$W_X$  is the width of the transistor X. Assume that the transistors have some proper widths.

- (1) Draw its DC characteristic curve for  $B = 1, C = 0$ , and  $A$  switches from 0 to 1. Just a rough sketch will be accepted. (10 points)

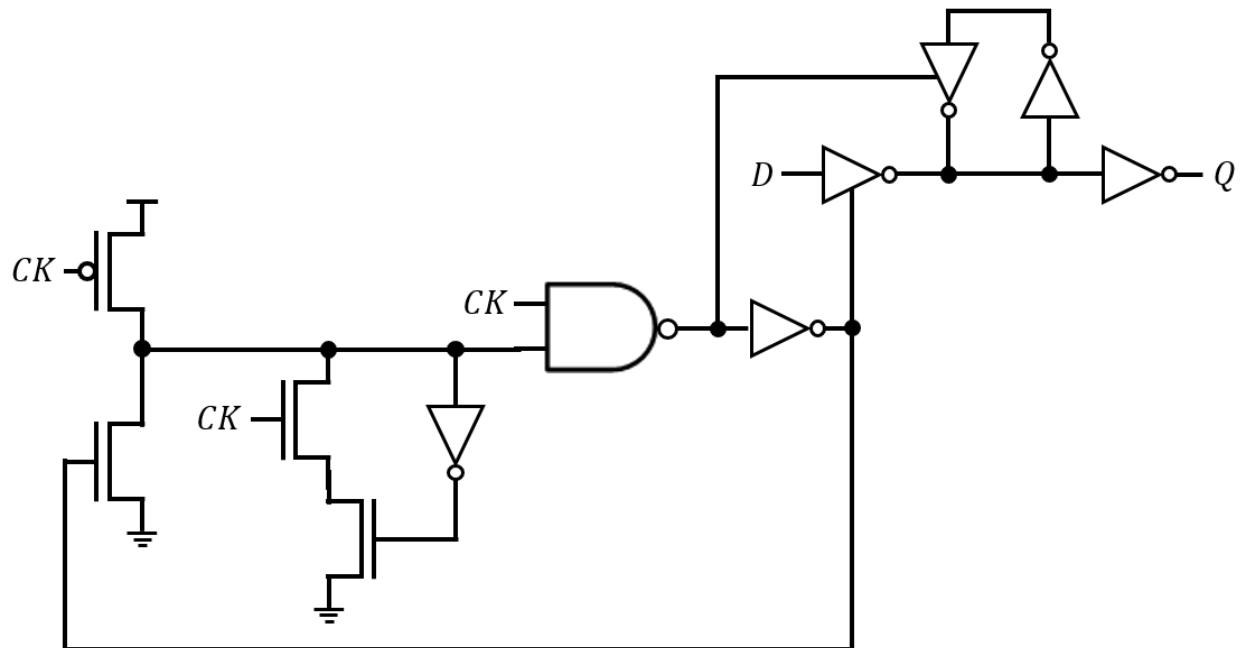


(2) Answer the following questions. Correct: +2 points, Wrong: -2 points, No answer: 0. Min: 0 points.

- If  $W_{N,A}$  goes up, then the DC curve is shifted upward always at  $V_A = \frac{V_{DD}}{2}$ . (**True** / **False**)
- If  $W_{N,C}$  goes up, then the DC curve is shifted upward always at  $V_A = \frac{V_{DD}}{2}$ . (**True** / **False**)
- If  $W_{P,A}$  goes up, then the DC curve is shifted downward always at  $V_A = \frac{V_{DD}}{2}$ . (**True** / **False**)
- If  $W_{P,C}$  goes up, then the DC curve is shifted downward always at  $V_A = \frac{V_{DD}}{2}$ . (**True** / **False**)
- If  $W_{N,Y}$  goes up, then the DC curve is shifted upward always at  $V_A = \frac{V_{DD}}{2}$ . (**True** / **False**)

### Problem #7 (Sequential Logic, 10 points)

What does the following circuit do? Describe its functionality in as much detail as possible. (D: data input. CK: clock. Q: data output)



Positive-edge-triggered D FF

**EE434**  
**ASIC and Digital Systems**

**Midterm Exam**  
**Mar. 6, 2024. (2:10pm – 3pm)**  
**Instructor: Dae Hyun Kim**

**Name:**

**WSU ID:**

Problem	Points	
1	10	
2	20	
3	10	
4	10	
5	10	
6	10	
7 (Bonus)	10	
Total	80	

## Problem #1 (15 points)

Answer the following questions for a digital system that has multiple pipeline stages.

Correct: +2 points. Wrong: -2 points. No answer: 0 points. Minimum: 0 points.

(Notice: Suppose a problem says, "If A happens, then B happens. (True / False)". In this case, if B can happen in some cases and cannot happen in some other cases, the answer is False. In other words, the answer is True only if B always happens when A happens.

For example, "If  $x > 0$  and  $y > 10$ , then  $y > x$ ." This is false because  $y < x$  can also happen (e.g.,  $x=20$ ,  $y=15$ ).

On the other hand, "If  $x^*y > 0$  and  $y > 0$ , then  $x > 0$ ." This is true because if  $y > 0$ , then dividing both sides of  $x^*y > 0$  by  $y$  leads to  $x > 0$ .

(1) Assume WNS < 0 and TNS < 0. If you reduce the delay of a net in the system, WNS goes up. (True / **False**)

(2) Assume WNS < 0 and TNS < 0. If you reduce the delay of a net in the system, TNS goes up. (True / **False**)

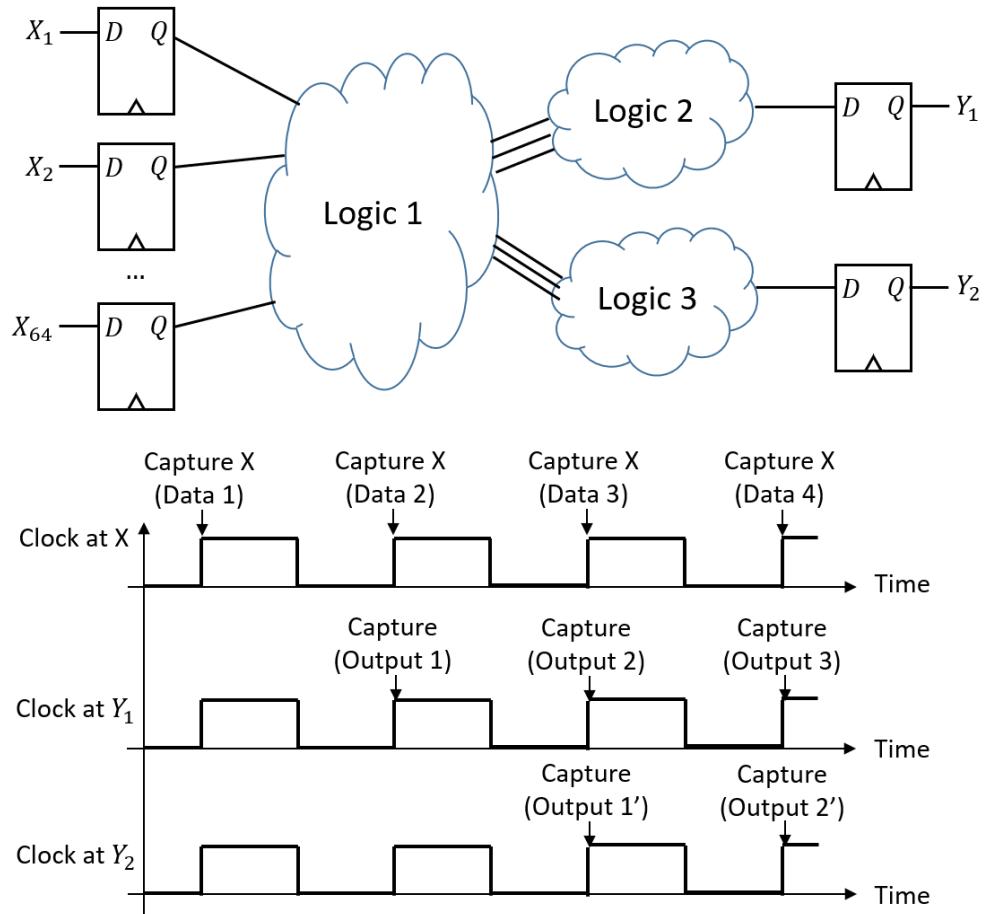
(3) Assume WNS = 0. If the delay of a gate goes up, then WNS goes down. (True / **False**)

(4) Suppose all the paths in the pipeline stages have setup time violations. If the delay of a gate goes down, then WNS goes up. (True / **False**)

(5) Suppose all the paths in the pipeline stages have setup time violations. If the delay of a gate goes down, then TNS goes up. (**True** / False)

## Problem #2 (20 points)

The following figure shows a pipeline stage. X is the input and Y is the output. Y has two output signals,  $Y_1$  and  $Y_2$ . The system spec is as follows:



- Data is fed into the system every cycle (captured by the FFs on the left).
- Each data set “Data #” generates two outputs, “Output #” and “Output #’”.
  - Output # is generated by Logic 1 and Logic 2. This is captured by  $Y_1$ .
  - Output #’ is generated by Logic 1 and Logic 3. This is captured by  $Y_2$ .
- $X - Y_1$  is a single-cycle path as shown in the waveform above.
- $X - Y_2$  is a multi-cycle path (two cycles) as shown in the waveform above.

Parameters:

- $L_{i-k}$ : the logic delay from input  $X_i$  to output  $Y_k$  (e.g.,  $L_{64-2}$  means the logic delay from input  $X_{64}$  to output  $Y_2$ )
- $s$ : setup time of a flip-flop
- $h$ : hold time of a flip-flop
- $x$ : delay from the clock source to the clock pin of a flip-flop

- $c$ : clk-to-Q delay of a flip-flop
- $T_{CLK}$ : clock period
- $\text{MIN}, \text{MAX}$ : MIN, MAX operators

(1) Show all the setup time inequalities for the design (5 points). (Use the MIN, MAX operators.)

$$x + c + \text{MAX}(L_{1-1}, \dots, L_{64-1}) \leq x + T_{CLK} - s$$

$$x + c + \text{MAX}(L_{1-2}, \dots, L_{64-2}) \leq x + 2 \cdot T_{CLK} - s$$

(2) Show all the hold time inequalities for the design (5 points). (Use the MIN, MAX operators.)

$$x + c + \text{MIN}(L_{1-1}, \dots, L_{64-1}) \geq x + h$$

$$x + c + \text{MIN}(L_{1-2}, \dots, L_{64-2}) \geq x + 2 \cdot T_{CLK} + h$$

Answer the following questions for the figure above. Correct: +2 points, Wrong: -2 points, No answer: 0. Min: 0 points.

(3) Assume WNS < 0. In this case, if  $s$  goes up, WNS goes down. (**True** / False)

(4) Assume WNS < 0. In this case, if the delay of Logic 1 goes up, WNS goes down. (**True** / False)

(5) Assume WNS < 0. In this case, if the delay of Logic 3 goes up, WNS goes down. (**True** / **False**)

Answer the following questions for the figure above. Correct: +2 points, Wrong: -2 points, No answer: 0. Min: 0 points.

(6) Assume TNS < 0. In this case, if  $s$  goes up, TNS goes down. (**True** / False)

(7) Assume TNS < 0. In this case, if the delay of Logic 1 goes up, TNS goes down. (**True** / False)

(8) Assume TNS < 0. In this case, if the delay of Logic 2 goes up, TNS goes down. (**True** / **False**)

### Problem #3 (Static CMOS, 10 points)

Design the following function (draw a transistor-level schematic) using the static CMOS logic design methodology. Available input:  $A, B, C, D$ . Try to minimize the # TRs.

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

10 points if # TRs  $\leq 20$ . 7 points if  $20 < \# \text{TRs} \leq 22$ . 5 points if  $\# \text{TRs} > 22$ .

$$Y = \overline{\bar{A} + \{B \oplus (C + D)\}} = \overline{\bar{A} + \{B \cdot \overline{C + D} + \bar{B} \cdot (C + D)\}} = \overline{\bar{A} + B \cdot \bar{C} \cdot \bar{D} + \bar{B} \cdot (C + D)}$$

This uses 7 NFETs and 7 PFETs (and 4 NFETs and 4 PFETs for the four inverters), which is a total of 11 NFETs and 11 PFETs.

$$\overline{\bar{A} + \{B \cdot \overline{C + D} + \bar{B} \cdot (C + D)\}} = \overline{\bar{A} + B \cdot X + \bar{B} \cdot \bar{X}} \text{ (where } X = \overline{C + D})$$

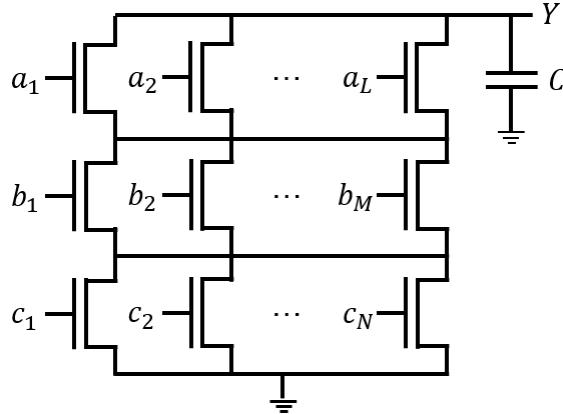
In this case, we design  $X$  and use an inverter to generate  $\bar{X}$  (3 NFETs + 3 PFETs).

Then, design the above expression, which requires 5 NFETs and 5 PFETs (and 2 NFETs and 2 PFETs for the two inverters to generate  $\bar{A}$  and  $\bar{B}$ ), which is a total of 10 NFETs and 10 PFETs.

## Problem #4 (TR Sizing, 10 points)

$R_n$  is the resistance of a 1X NFET (whose width is  $w_{min}$ ). “ $h \times$ ” for a TR means that the width of the TR is  $h \cdot w_{min}$ .

The following figure shows the NFET network of a static CMOS logic gate.



Notice that  $L, M, N$  are positive integer constants.

Timing constraint:  $\tau_f \leq R_n C$  ( $\tau_f$  is the worst-case fall delay).

Find the optimal sizes of the TRs that minimize the total transistor width. (Just saying “3X for all the TRs” will get 0 points. You should optimize the total width.)

$$a_1 = \dots = a_L = aX. b_1 = \dots = b_M = bX. c_1 = \dots = c_N = cX.$$

Constraints:  $\frac{1}{a} + \frac{1}{b} + \frac{1}{c} = 1$ . Minimize  $f = L \cdot a + M \cdot b + N \cdot c$ .

Let  $\frac{1}{a} = x, \frac{1}{b} = y, \frac{1}{c} = z$ . Then,  $x + y + z = 1$ . Minimize  $g = \frac{L}{x} + \frac{M}{y} + \frac{N}{z}$ .

$\frac{\partial g}{\partial x} = -\frac{L}{x^2} + \frac{N}{z^2} = 0, \frac{\partial g}{\partial y} = -\frac{M}{y^2} + \frac{N}{z^2} = 0$ . Thus,  $x = \sqrt{\frac{L}{N}} \cdot z, y = \sqrt{\frac{M}{N}} \cdot z$ .

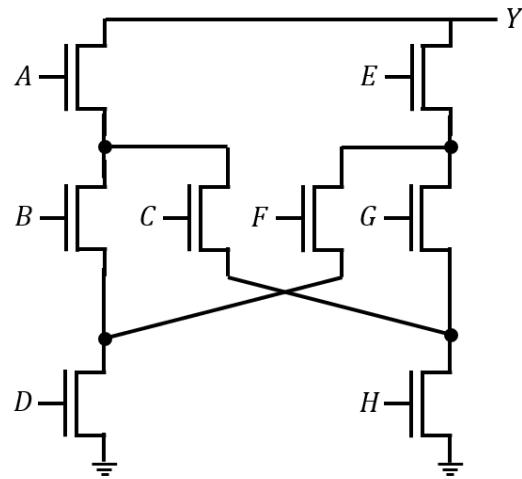
From the constraint,  $z \left( \sqrt{\frac{L}{N}} + \sqrt{\frac{M}{N}} + 1 \right) = 1$ , so

$$z = \frac{\sqrt{N}}{\sqrt{L} + \sqrt{M} + \sqrt{N}}, x = \frac{\sqrt{L}}{\sqrt{L} + \sqrt{M} + \sqrt{N}}, y = \frac{\sqrt{M}}{\sqrt{L} + \sqrt{M} + \sqrt{N}}.$$

$$a_1 = \dots = a_L = \frac{\sqrt{L} + \sqrt{M} + \sqrt{N}}{\sqrt{L}} \times, b_1 = \dots = b_M = \frac{\sqrt{L} + \sqrt{M} + \sqrt{N}}{\sqrt{M}} \times, c_1 = \dots = c_N = \frac{\sqrt{L} + \sqrt{M} + \sqrt{N}}{\sqrt{N}} \times.$$

### Problem #5 (Analysis, 10 points)

Express the output  $Y$  as a Boolean function of the input signals ( $A, B, C, D, E, F, G, H$ ).



$$\begin{aligned}\bar{Y} = & A \cdot (B \cdot D + B \cdot F \cdot G \cdot H + C \cdot H + C \cdot G \cdot F \cdot D) + E \\ & \cdot (G \cdot H + G \cdot C \cdot B \cdot D + F \cdot D + F \cdot B \cdot C \cdot H)\end{aligned}$$

## Problem #6 (STA, 10 points)

For the setup-time analysis, we use the slack = required time – arrival time. We can define a similar metric for the hold-time analysis as follows:

$$\text{slack (for hold time)} = \text{arrival time} - \text{required time}.$$

Notice that a hold-time violation occurs when the delay (arrival time) of a signal is too small. Thus, if we use the definition above, a positive slack means no hold-time violation and a negative slack means a hold-time violation. We can define WNS and TNS in the same way.

Answer the following questions for the hold-time analysis of a pipeline stage (between two flip-flop stages). Correct: +2 points. Wrong: -2 points. No answer: 0 points.

Minimum: 0 points.

(1) Assume WNS < 0 and TNS < 0. If you increase the delay of a net in the critical path, WNS goes up. (**True** / False)

(2) Assume WNS < 0 and TNS < 0. If you increase the delay of a gate in the critical path, TNS goes up. (**True** / False)

(3) Assume WNS < 0. In this case, can “WNS = TNS” happen? (i.e., can WNS be equal to TNS?) (**Yes** / No)

(4) Assume TNS < 0. In this case, can “TNS < WNS” happen? (**Yes** / No)

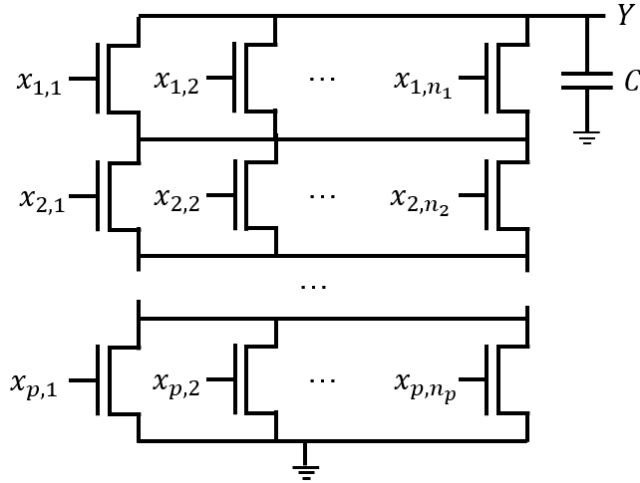
(5) Suppose  $T_{CLK} > 2 \cdot (T_S + T_H)$  where  $T_{CLK}$  is the clock period,  $T_S$  is the setup time of a FF, and  $T_H$  is the hold time of a FF. In this case, can a path have both setup-time and hold-time violations? (**Yes** / **No**)

## Problem #7 (TR Sizing, 10 points, Bonus)

$R_n$  is the resistance of a 1X NFET (whose width is  $w_{min}$ ). “ $h \times$ ” for a TR means that the width of the TR is  $h \cdot w_{min}$ .

The following figure shows the NFET network of a static CMOS logic gate  $Y =$

$$(x_{1,1} + \dots + x_{1,n_1}) \cdot (x_{2,1} + \dots + x_{2,n_2}) \cdot \dots \cdot (x_{p,1} + \dots + x_{p,n_p}).$$



Notice that  $p, n_1, n_2, \dots, n_p$  are all positive integer constants. For example,  $p = 3, n_1 = L, n_2 = M, n_3 = N$  for Problem #4. This problem is a generalization of Problem #4.

Timing constraint:  $\tau_f \leq R_n C$  ( $\tau_f$  is the worst-case fall delay).

Find the optimal sizes of the TRs that minimize the total transistor width. (Just saying “ $n_p \times$  for all the TRs” will get 0 points. You should optimize the total width.)

$$x_{1,1} = \dots = x_{1,n_1} = a_1 X.$$

$$x_{2,1} = \dots = x_{2,n_2} = a_2 X.$$

...

$$x_{p,1} = \dots = x_{p,n_p} = a_p X.$$

Constraints:  $\frac{1}{a_1} + \frac{1}{a_2} + \dots + \frac{1}{a_p} = 1$ . Minimize  $f = n_1 \cdot a_1 + n_2 \cdot a_2 + \dots + n_p \cdot a_p$ .

Let  $\frac{1}{a_1} = c_1, \frac{1}{a_2} = c_2, \dots, \frac{1}{a_p} = c_p$ . Then,  $c_1 + c_2 + \dots + c_p = 1$ . Minimize  $g = \frac{n_1}{c_1} + \frac{n_2}{c_2} + \dots + \frac{n_p}{c_p}$ .

$\frac{\partial g}{\partial c_k} = -\frac{n_k}{c_k^2} + \frac{n_p}{c_p^2} = 0$  (for  $k = 1, 2, \dots, p-1$ ).

Thus,  $c_k = \sqrt{\frac{n_k}{n_p}} \cdot c_p$ .

From the constraint,  $\left( \sqrt{\frac{n_1}{n_p}} + \sqrt{\frac{n_2}{n_p}} + \dots + \sqrt{\frac{n_p}{n_p}} \right) \cdot c_p = 1$ .

Thus,  $c_p = \frac{\sqrt{n_p}}{\sqrt{n_1} + \sqrt{n_2} + \dots + \sqrt{n_p}}$ . Similarly,  $c_k = \frac{\sqrt{n_k}}{\sqrt{n_1} + \sqrt{n_2} + \dots + \sqrt{n_p}}$ .

Answer:

$$x_{1,1} = \dots = x_{1,n_1} = \frac{\sqrt{n_1} + \sqrt{n_2} + \dots + \sqrt{n_p}}{\sqrt{n_1}} \times$$

$$x_{2,1} = \dots = x_{2,n_2} = \frac{\sqrt{n_1} + \sqrt{n_2} + \dots + \sqrt{n_p}}{\sqrt{n_2}} \times$$

...

$$x_{p,1} = \dots = x_{p,n_p} = \frac{\sqrt{n_1} + \sqrt{n_2} + \dots + \sqrt{n_p}}{\sqrt{n_p}} \times$$