EE434

ASIC and Digital Systems

Final Exam

May 5, 2015. (1pm – 3pm)

Instructor: Dae Hyun Kim (daehyun@eecs.wsu.edu)

Name:

WSU ID:

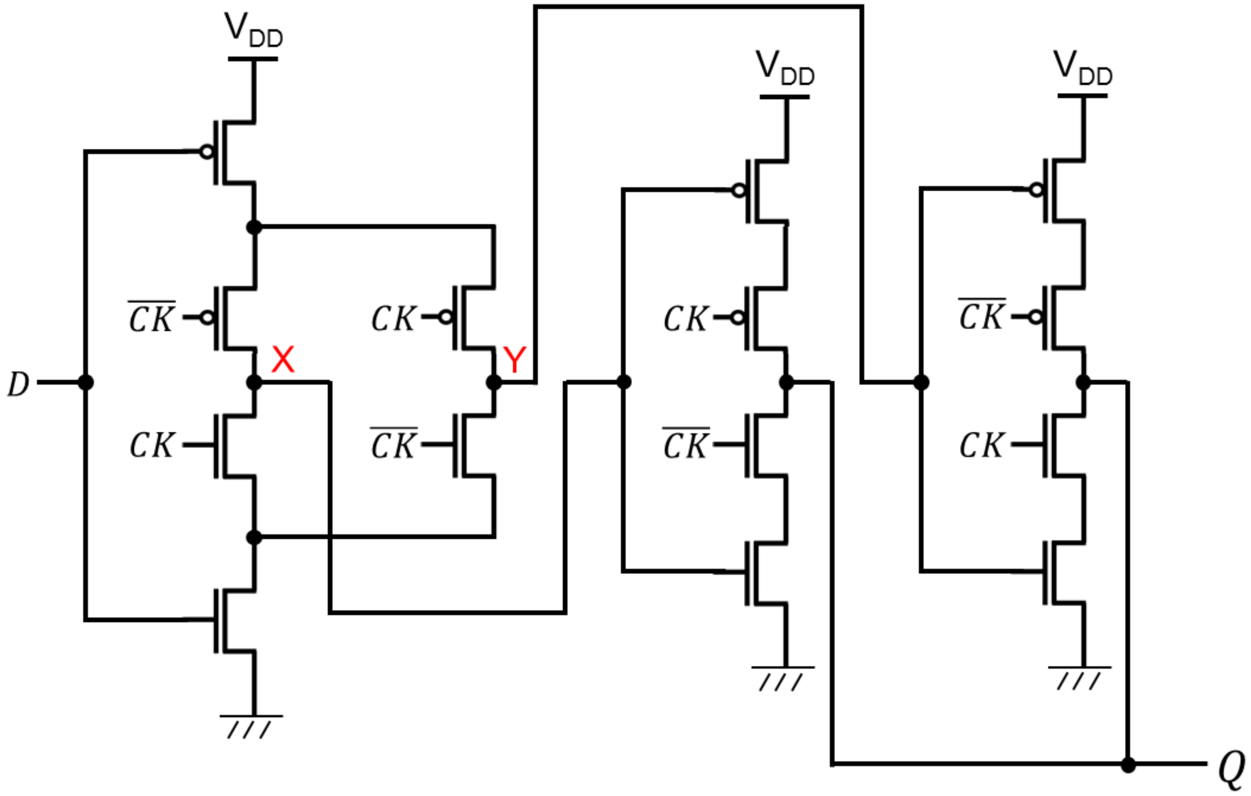
Problem	Points	
1	10	
2	10	
3	10	
4	10	
5	10	
6	10	
7	10	
8	10	
Total	80	

* 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).



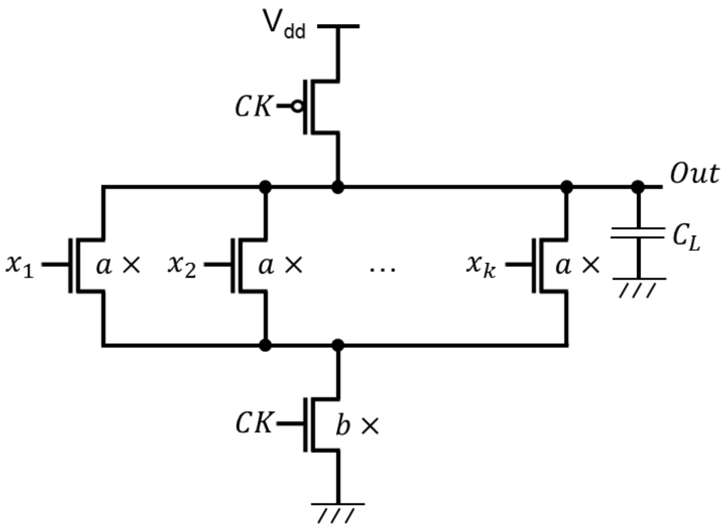
When CK goes to 0 (falling edge), X is floating, Y is $\overline{D}$, and Q is $\overline{X}(= D)$.

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

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 ($\mathrm{k} \geq 10$) 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: $\frac{1}{4} R_n C_L \ \leq \ \left(\frac{R_n}{a} + \frac{R_n}{b}\right) C_L \ \leq 4 R_n C_L \quad \Rightarrow \quad \frac{1}{4} \leq \ \left(\frac{1}{a} + \frac{1}{b}\right) \ \leq 4$

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 $\frac{1}{4} \leq c \leq 4$.

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

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

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

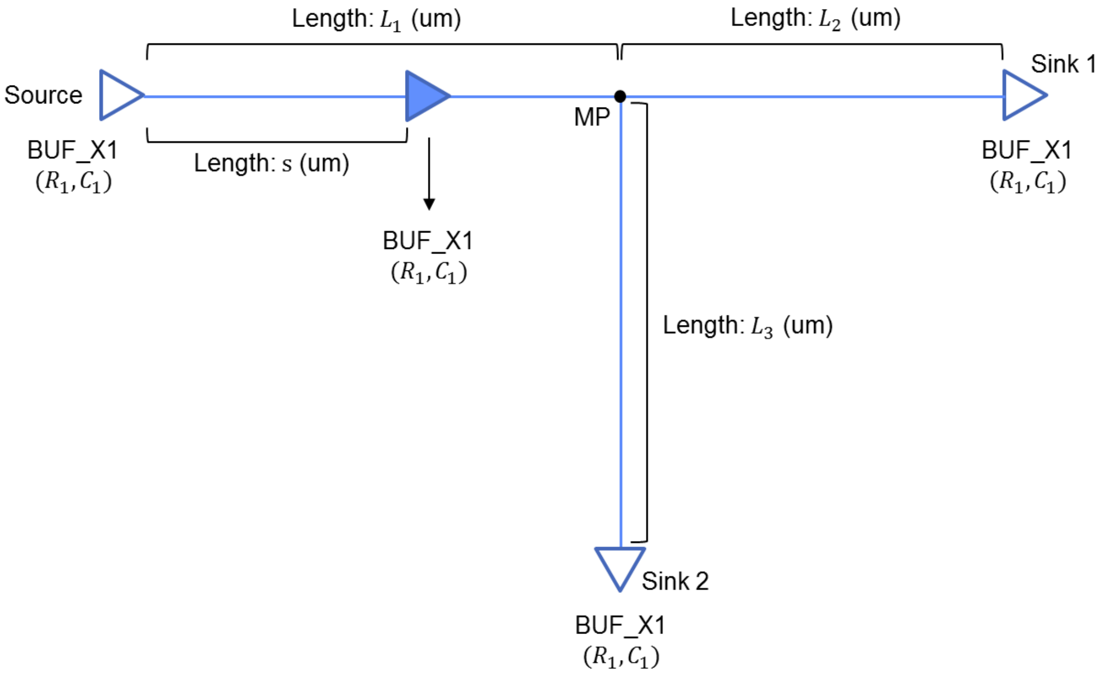
$$\Rightarrow \quad a = \frac{b}{bc-1} = \frac{1}{c}\left(1+\frac{1}{\sqrt{k}}\right)$$

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

We set c to 4 to minimize W.

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

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 BUF_X1: R_1
- Input capacitance of BUF_X1: C_1
- Unit wire resistance: $r\ (\Omega/\mu m)$
- Unit wire capacitance: $c\ (F/\mu m)$
- $\frac{C_1}{c} + L_2 + L_3 < L_1$

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

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

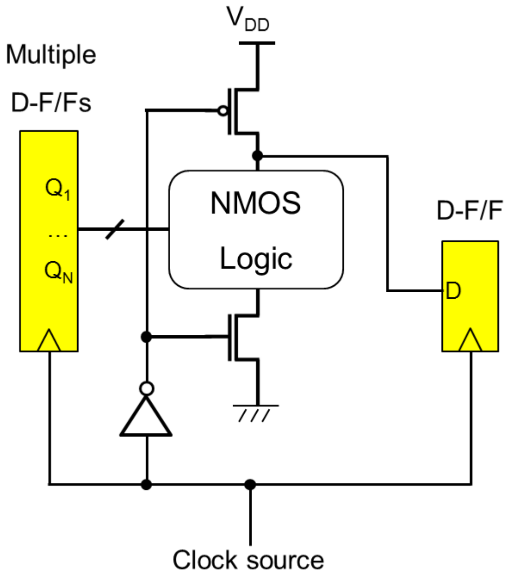
$$\frac{d\tau}{ds} = R_1 \cdot c + r \cdot C_1 + r \cdot c \cdot s - R_1 \cdot c - r(c \cdot L_2 + c \cdot L_3 + 2C_1) - r \cdot c \cdot (L_1 - s) = 0$$

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

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

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

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



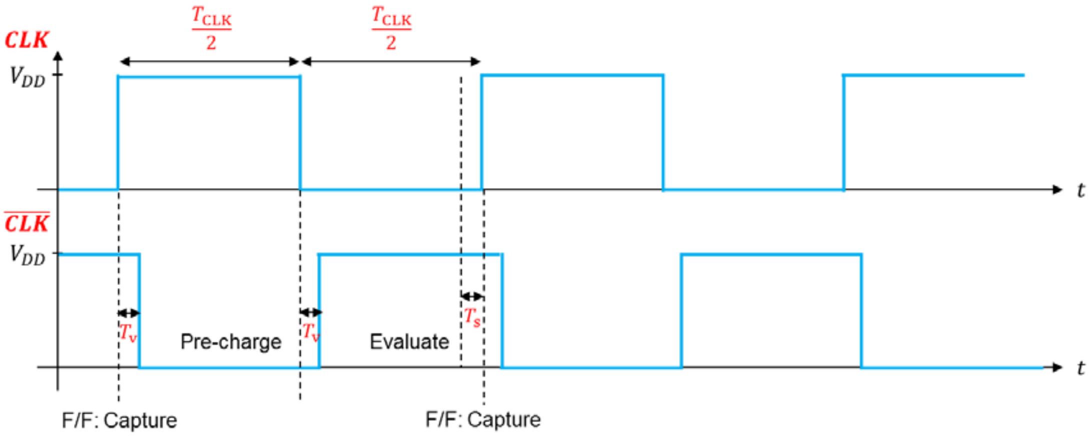
- Setup time: T_{s}
- Clock period: T_{CLK}
- D-F/F internal delay: T_{CQ}
- Clock skew: 0
- NMOS logic delay: T_{logic}
- Inverter delay: T_{v}
- Clock duty cycle: 50%

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:

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

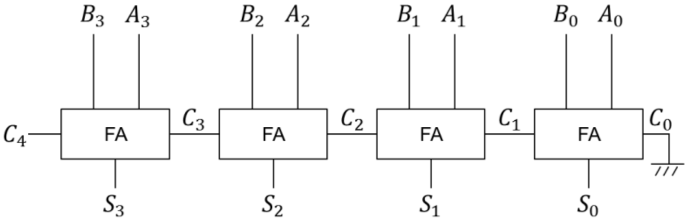
As the following figure shows, when the clock goes from low to high, the D-F/Fs capture their input signals. After T_{v}, $\overline{CLK}$ goes from high to low and the dynamic CMOS circuit starts pre-charging the output capacitor. When the clock goes from high to low, $\overline{CLK}$ goes from low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



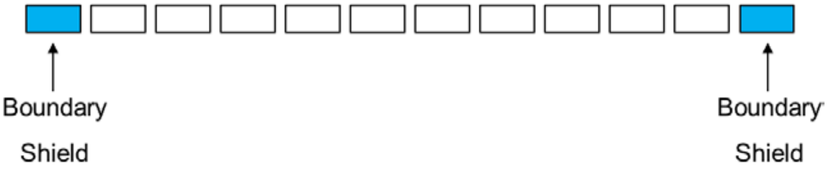
$$\boxed{\therefore\ T_{\text{logic}} \leq \frac{T_{\text{CLK}}}{2} - T_{\text{s}} - T_{\text{v}}}$$

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

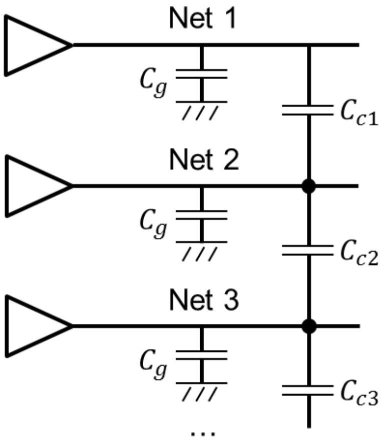
Let's design a four-bit ripple carry adder (RCA) with C_0 tied to ground as shown below.



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

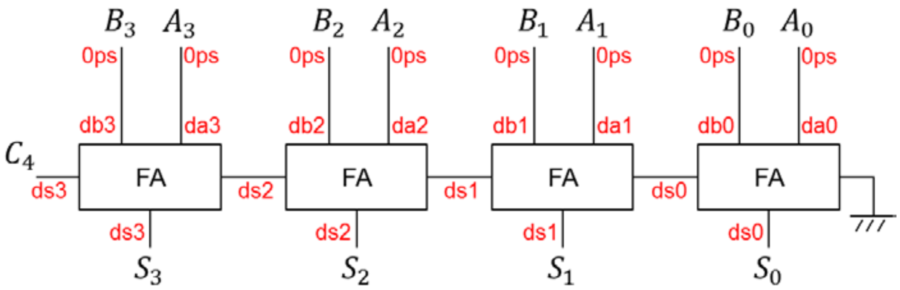


You are supposed to use ***eight*** routing tracks ***for signal*** and the other ***two*** routing tracks ***for shielding*** tied to V$_{DD}$ or V$_{SS}$. Wire resistance is negligible and each wire has a ground capacitor (C_g) and a coupling capacitor as follows:



- Output resistance of the buffers driving the wires: 500Ω
- 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).



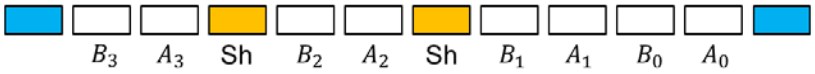
ds0 = MAX(da0, db0)+ε where ε is the full adder delay.

ds1 = MAX(da1, db1, ds0)+ε

ds2 = MAX(da2, db2, ds1)+ε

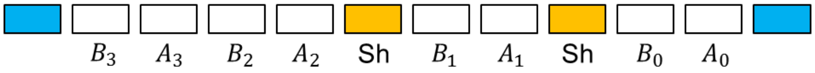
ds3 = MAX(da3, db3, ds2)+ε

Coupling capacitance



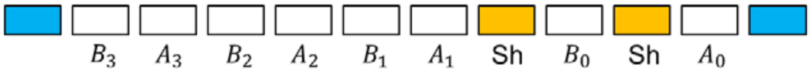
$$(B_3\ A_3\ B_2\ A_2\ B_1\ A_1\ B_0\ A_0) = (3C_c, 3, 3, 3, 3, 4, 4, 3)$$

$$\Rightarrow delay = (125, 125, 125, 125, 125, 150, 150, 125)$$



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

$$\Rightarrow delay = (125, 150, 150, 125, 125, 125, 125, 125)$$



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

$$\Rightarrow delay = (125, 150, 150, 150, 150, 125, 100, 100)$$

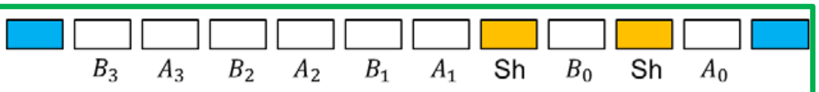
Case 1: ds0 = 125+40 = 165ps   ds1 = 165+40 = 205ps   ds2 = 205+40 = 245ps

ds3 = 245+40 = 285ps

Case 2: ds0 = 100+40 = 140ps   ds1 = 150+40 = 190ps   ds2 = 190+40 = 230ps

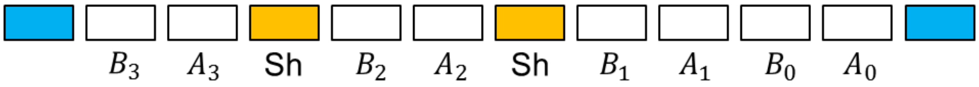
ds3 = 230+40 = 270ps

Thus, Case 2 is the best.



Delay = 270ps

Example) Suppose the following is my assignment result.

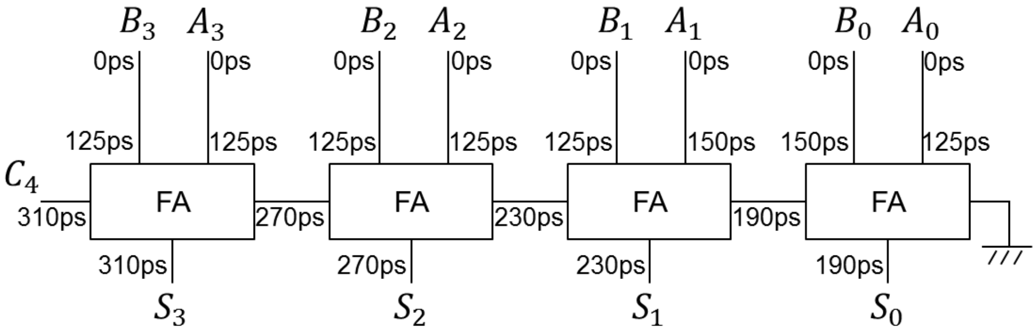


The following shows the total capacitance of each wire:

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

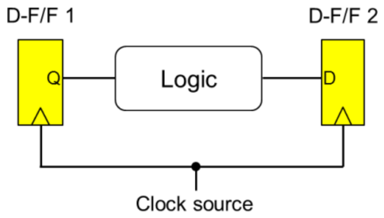
The following shows the arrival time at each node:

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



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: $T_\mathrm{skew} = cd_2 - cd_1$
- Logic delay: T_logic
- Clock period: T_CLK

Ideally, the following inequalities should be satisfied:

1. Setup time: $T_\mathrm{s} \leq T_\mathrm{CLK} + T_\mathrm{skew} - T_\mathrm{logic} - T_\mathrm{CQ}$
2. Hold time: $T_\mathrm{h} \leq T_\mathrm{CQ} + T_\mathrm{logic} - T_\mathrm{skew}$

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., μ_p and μ_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 ($\Delta_1,\ \Delta_2,\ \Delta_3,\ \Delta_4 > 0$):

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

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:

- $T_\mathrm{s},\ T_\mathrm{h},\ T_\mathrm{CQ},\ T_\mathrm{CLK},\ T_\mathrm{skew},\ T_\mathrm{logic},\ \Delta_1,\ \Delta_2,\ \Delta_3,\ \Delta_4$

The new range of T_skew: $[(cd_2 - \Delta_2) - (cd_1 + \Delta_1),\ (cd_2 + \Delta_2) - (cd_1 - \Delta_1)]$

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

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

Setup time)

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

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

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

$$\left[T_{\text{logic}} - \Delta_4, T_{\text{logic}} + \Delta_4\right] \leq T_r$$

$$\therefore\ T_{\text{logic}} \leq T_{\text{CLK}} + \{(cd_2 - \Delta_2) - (cd_1 + \Delta_1)\} - T_{\text{s}} - \left\{T_{\text{CQ}} + \Delta_3\right\} - \Delta_4$$

$$\boxed{T_{\text{logic}} \leq (T_{\text{CLK}} + T_{\text{skew}} - T_{\text{s}} - T_{\text{CQ}}) - (\Delta_1 + \Delta_2 + \Delta_3 + \Delta_4)}$$

Hold time)

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

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

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

$$\left[T_{\text{logic}} - \Delta_4, T_{\text{logic}} + \Delta_4\right] \geq T_r$$

$$\therefore\ T_{\text{logic}} \geq T_{\text{h}} - \left\{T_{\text{CQ}} - \Delta_3\right\} + \{(cd_2 + \Delta_2) - (cd_1 - \Delta_1)\} + \Delta_4$$

$$\boxed{T_{\text{logic}} \geq (T_{\text{h}} - T_{\text{CQ}} + T_{\text{skew}}) + (\Delta_1 + \Delta_2 + \Delta_3 + \Delta_4)}$$

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

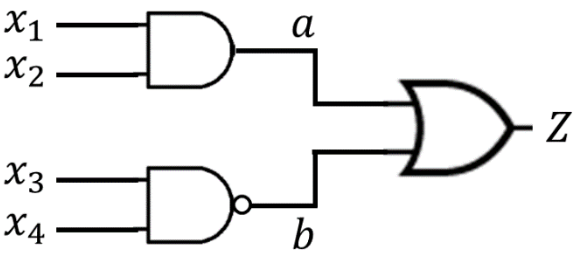
$CI_0 = 1$

	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	
	${S_i}^1$:	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 2	${S_i}^0$:	1	1	1	0	0	0	1	1	0	0	0	1	1	0		
	${CO_i}^0$:	0		1		1		0		1		1		1			
	${S_i}^1$:	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	
Step 3	${S_i}^0$:	0	0	1	0	0	0	1	1	0	1	0	1				
	${CO_i}^0$:	1				1				1							
	${S_i}^1$:	0	0	1	1	0	1	0	0	0	1	1	0	1	1	0	0
	${CO_i}^1$:	1				1				1				1			
Step 4	${S_i}^0$:	0	0	1	1	0	0	1	1								
	${CO_i}^0$:	1															
	${S_i}^1$:	0	0	1	1	0	1	0	0	0	1	1	0	1	1	0	0
	${CO_i}^1$:	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.



$$\mathrm{Z} = x_1x_2 + \overline{x_3x_4}$$

x_1 s-a-0: $Z_f = \overline{x_3x_4} \Rightarrow \mathrm{Z} \oplus Z_f = (x_1x_2 + \overline{x_3x_4}) \oplus \overline{x_3x_4} = 1 \Rightarrow (x_1\ x_2\ x_3\ x_4) = (1\ 1\ 1\ 1)$

x_1 s-a-1: $Z_f = x_2 + \overline{x_3x_4} \Rightarrow \mathrm{Z} \oplus Z_f = (x_1x_2 + \overline{x_3x_4}) \oplus (x_2 + \overline{x_3x_4}) = 1 \Rightarrow (x_1\ x_2\ x_3\ x_4) = (0\ 1\ 1\ 1)$

x_2 s-a-0: $Z_f = \overline{x_3x_4} \Rightarrow (x_1\ x_2\ x_3\ x_4) = (1\ 1\ 1\ 1)$

x_2 s-a-1: $Z_f = x_1 + \overline{x_3x_4} \Rightarrow (x_1\ x_2\ x_3\ x_4) = (1\ 0\ 1\ 1)$

x_3 s-a-0:
$Z_f = 1 \Rightarrow \mathrm{Z} \oplus Z_f = (x_1x_2 + \overline{x_3x_4}) \oplus 1 = 1 \Rightarrow (x_1\ x_2\ x_3\ x_4) = (0\ 0\ 1\ 1)\ or\ (0\ 1\ 1\ 1)\ or\ (1\ 0\ 1\ 1)$

x_3 s-a-1: $Z_f = x_1x_2 + \overline{x_4} \Rightarrow \mathrm{Z} \oplus Z_f = (x_1x_2 + \overline{x_3x_4}) \oplus (x_1x_2 + \overline{x_4}) = 1 \Rightarrow (x_1\ x_2\ x_3\ x_4) = (0\ 0\ 0\ 1)\ or\ (0\ 1\ 0\ 1)\ or\ (1\ 0\ 0\ 1)$

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

x_4 s-a-1: $Z_f = x_1x_2 + \overline{x_3} \Rightarrow (x_1\ x_2\ x_3\ x_4) = (0\ 0\ 1\ 0)\ or\ (0\ 1\ 1\ 0)\ or\ (1\ 0\ 1\ 0)$

a s-a-0: $Z_f = \overline{x_3x_4} \Rightarrow (x_1\ x_2\ x_3\ x_4) = (1\ 1\ 1\ 1)$

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

b s-a-0: $Z_f = x_1x_2 \Rightarrow \mathrm{Z} \oplus Z_f = (x_1x_2 + \overline{x_3x_4}) \oplus (x_1x_2) = 1 \Rightarrow (x_1\ x_2\ x_3\ x_4) = (0\ 0\ 0\ 0)\ or\ (0\ 1\ 0\ 0)\ or\ (1\ 0\ 0\ 0)\ or\ (0\ 0\ 0\ 1)\ or\ (0\ 1\ 0\ 1)\ or\ (1\ 0\ 0\ 1)\ or\ (0\ 0\ 1\ 0)$

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

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

$x_1\, x_2\, x_3\, x_4$	x_1 s-a-0 x_2 s-a-0 a s-a-0	x_1 s-a-1	x_2 s-a-1	x_3 s-a-0 x_4 s-a-0 a s-a-1 b s-a-1	x_3 s-a-1	x_4 s-a-1	b s-a-0
0 0 0 0							●
0 0 0 1					●		●
0 0 1 0						●	●
0 0 1 1				●			
0 1 0 0							●
0 1 0 1					●		●
0 1 1 0						●	●
0 1 1 1		●		●			
1 0 0 0							●
1 0 0 1					●		●
1 0 1 0						●	●
1 0 1 1			●	●			
1 1 0 0							
1 1 0 1							
1 1 1 0							
1 1 1 1	●						

1) Needs (1 1 1 1) to cover the 1st column.

2) Needs (0 1 1 1) to cover the 2nd column. This also covers the 4th column.

3) Needs (1 0 1 1) to cover the 3rd column.

4) Needs two more vectors to cover the 5th, 6th, and 7th columns.

=> (0 0 1 0) and (0 0 0 1) (there exist total 18 different minimial sets).

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