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)

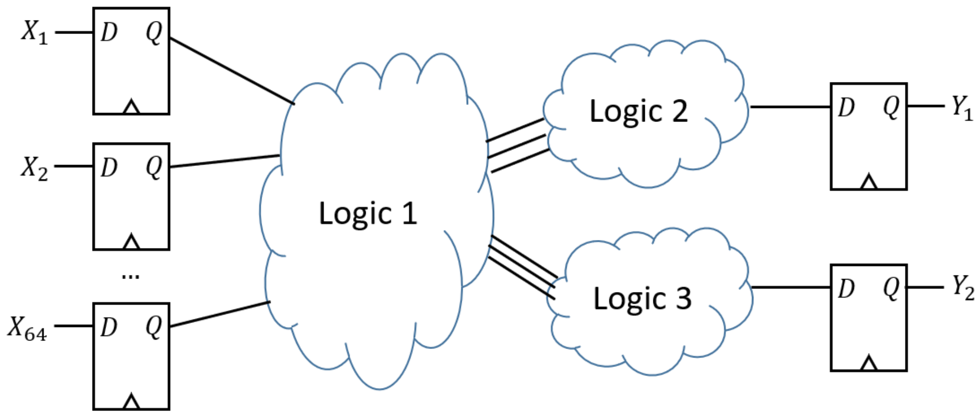
(3) Assume WNS = 0. If the delay of a gate goes up, then WNS goes down. (True / False)

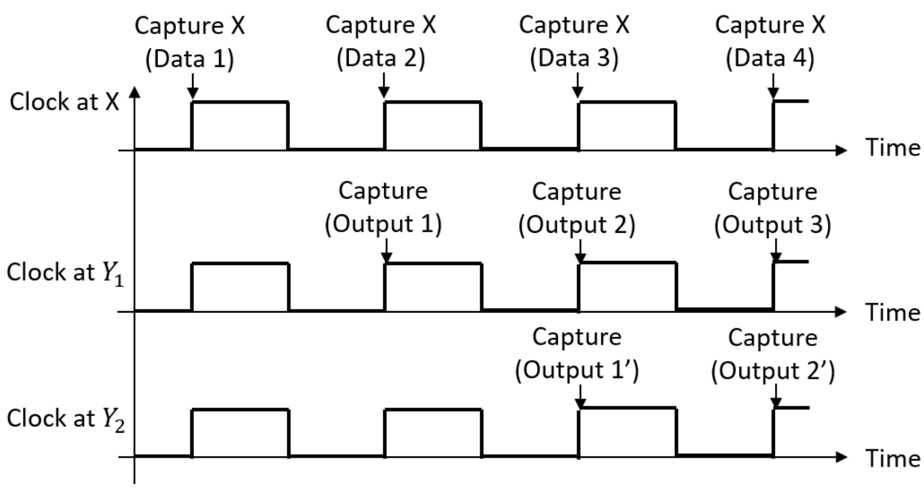
(4) Suppose all the paths in the pipeline stages have setup time violations. If the delay of a gate goes down, then WNS goes up. (True / False)

(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
- MIN, MAX: MIN, MAX operators

(1) Show all the setup time inequalities for the design (5 points). (Use the MIN, MAX operators.)

$$x + c + MAX(L_{1-1}, \dots, L_{64-1}) \leq x + T_{CLK} - s$$

$$x + c + MAX(L_{1-2}, \dots, L_{64-2}) \leq x + 2 \cdot T_{CLK} - s$$

(2) Show all the hold time inequalities for the design (5 points). (Use the MIN, MAX operators.)

$$x + c + MIN(L_{1-1}, \dots, L_{64-1}) \geq x + h$$

$$x + c + MIN(L_{1-2}, \dots, L_{64-2}) \geq x + 2 \cdot T_{CLK} + h$$

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 < # TRs $\leq$ 22. 5 points if # 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.

$\overline{\bar{A} + \{B \cdot \overline{C + D} + \bar{B} \cdot (C + D)\}} = \overline{\bar{A} + B \cdot X + \bar{B} \cdot \bar{X}}$ (where $X = \overline{C + D}$)

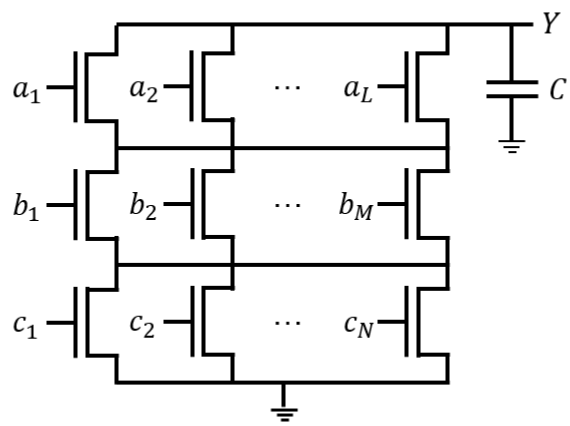
In this case, we design X and use an inverter to generate $\bar{X}$ (3 NFETs + 3 PFETs).

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: $\tau_f \le R_n C$ (τ_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.)

$$a_1 = \cdots = a_L = aX. b_1 = \cdots = b_M = bX. c_1 = \cdots = c_L = cX.$$

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, $x = \sqrt{\frac{L}{N}} \cdot z$. $y = \sqrt{\frac{M}{N}} \cdot z$.

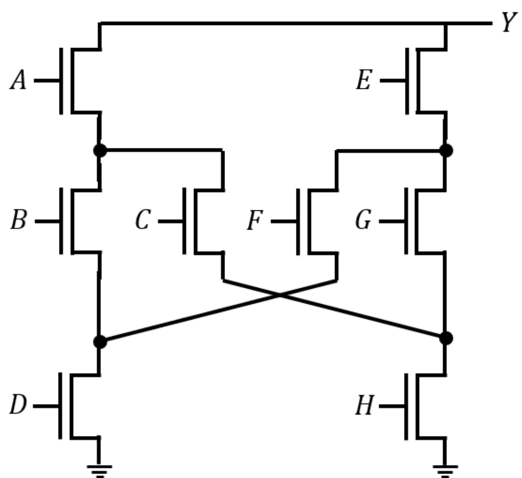
From the constraint, $z\left(\sqrt{\frac{L}{N}} + \sqrt{\frac{M}{N}} + 1\right) = 1$, so

$$z = \frac{\sqrt{N}}{\sqrt{L}+\sqrt{M}+\sqrt{N}}.\ x = \frac{\sqrt{L}}{\sqrt{L}+\sqrt{M}+\sqrt{N}}.\ y = \frac{\sqrt{M}}{\sqrt{L}+\sqrt{M}+\sqrt{N}}.$$

$a_1 = \cdots = a_L = \frac{\sqrt{L}+\sqrt{M}+\sqrt{N}}{\sqrt{L}}\times$. $b_1 = \cdots = b_M = \frac{\sqrt{L}+\sqrt{M}+\sqrt{N}}{\sqrt{M}}\times$. $c_1 = \cdots = c_L = \frac{\sqrt{L}+\sqrt{M}+\sqrt{N}}{\sqrt{N}}\times$.

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).



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

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:

slack (for hold time) = arrival time – 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)

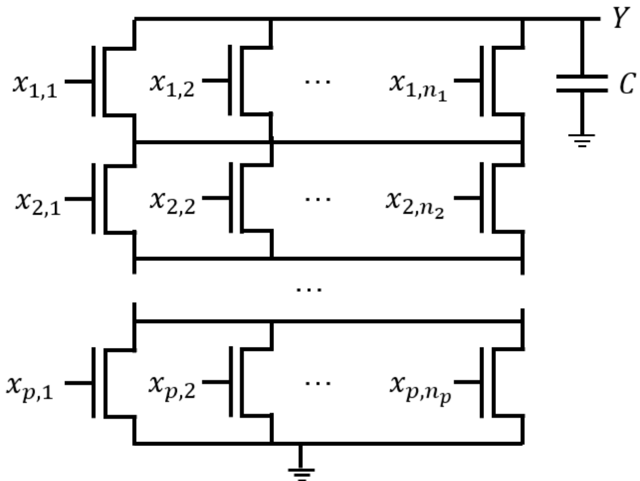
(4) Assume TNS < 0. In this case, can "TNS < WNS" happen? (Yes / No)

(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 ×" 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 $Y = \overline{(x_{1,1} + \cdots + x_{1,n_1}) \cdot (x_{2,1} + \cdots + x_{2,n_2}) \cdot \ldots \cdot (x_{p,1} + \cdots + x_{p,n_p})}$.



Notice that $p, n_1, n_2, \ldots, n_p$ 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: $\tau_f \le R_n C$ (τ_f is the worst-case fall delay).

Find the optimal sizes of the TRs that minimize the total transistor width. (Just saying "n_p × for all the TRs" will get 0 points. You should optimize the total width.)

$$x_{1,1} = \cdots = x_{1,n_1} = a_1 X.$$

$$x_{2,1} = \cdots = x_{2,n_2} = a_2 X.$$

$$\ldots$$

$$x_{p,1} = \cdots = x_{p,n_p} = a_p X.$$

Constraints: $\frac{1}{a_1} + \frac{1}{a_2} + \cdots + \frac{1}{a_p} = 1$. Minimize $f = n_1 \cdot a_1 + n_2 \cdot a_2 + \cdots + n_p \cdot a_p$.

Let $\frac{1}{a_1} = c_1, \frac{1}{a_2} = c_2, \ldots, \frac{1}{a_p} = c_p$. Then, $c_1 + c_2 + \cdots + c_p = 1$. Minimize $g = \frac{n_1}{c_1} + \frac{n_2}{c_2} + \cdots + \frac{n_p}{c_p}$.

$\frac{\partial g}{\partial c_k} = -\frac{n_k}{{c_k}^2} + \frac{n_p}{{c_p}^2} = 0$ (for $k = 1,2,\ldots,p-1$).

Thus, $c_k = \sqrt{\frac{n_k}{n_p}} \cdot c_p$.

From the constraint, $\left(\sqrt{\frac{n_1}{n_p}} + \sqrt{\frac{n_2}{n_p}} + \cdots + \sqrt{\frac{n_p}{n_p}}\right) \cdot c_p = 1$.

Thus, $c_p = \frac{\sqrt{n_p}}{\sqrt{n_1}+\sqrt{n_2}+\cdots+\sqrt{n_p}}$. Similarly, $c_k = \frac{\sqrt{n_k}}{\sqrt{n_1}+\sqrt{n_2}+\cdots+\sqrt{n_p}}$.

Answer:

$$x_{1,1} = \cdots = x_{1,n_1} = \frac{\sqrt{n_1} + \sqrt{n_2} + \cdots + \sqrt{n_p}}{\sqrt{n_1}} \times$$

$$x_{2,1} = \cdots = x_{2,n_2} = \frac{\sqrt{n_1} + \sqrt{n_2} + \cdots + \sqrt{n_p}}{\sqrt{n_2}} \times$$

$$\cdots$$

$$x_{p,1} = \cdots = x_{p,n_p} = \frac{\sqrt{n_1} + \sqrt{n_2} + \cdots + \sqrt{n_p}}{\sqrt{n_p}} \times$$