EE434

ASIC and Digital Systems

Final Exam

May 4, 2021. (1pm – 4pm)

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

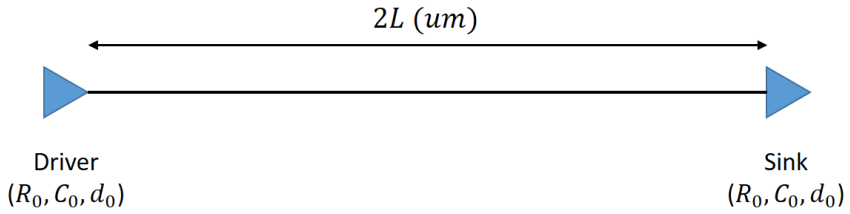
Name:

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)

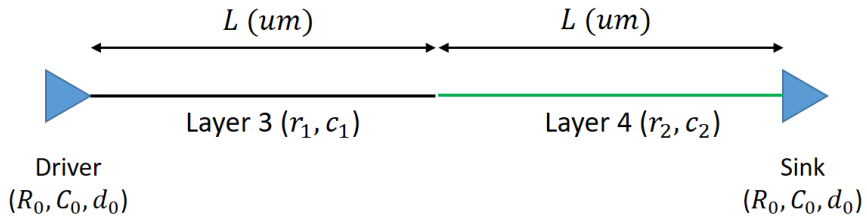
In the figure below, the driver is driving the sink through a long wire. The wire length is $2L$ (um).



- 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

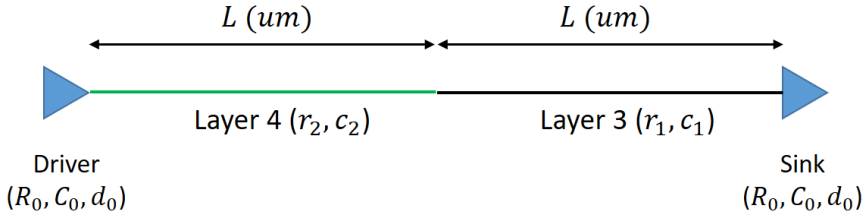
Answer the following questions.

(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 $\boldsymbol{d_A}$ 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 $\boldsymbol{d_B}$ from the output of the driver to the input of the sink as a function of the parameters above.



$$d_B = R_0(c_2L + c_1L + C_0) + r_2L(c_1L + C_0) + \frac{1}{2}r_2c_2L^2 + r_1LC_0 + \frac{1}{2}r_1c_1L^2$$

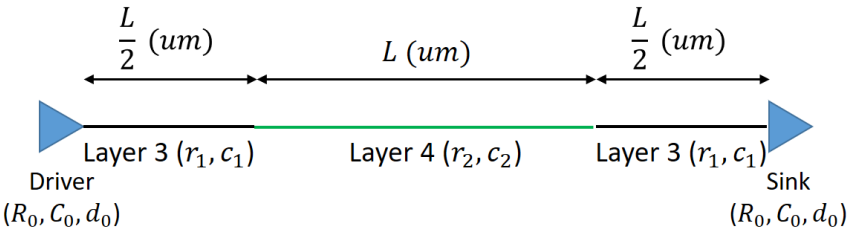
(c) (6 points) When are they ($\boldsymbol{d_A}$ and $\boldsymbol{d_B}$) equal? Show the condition for $\boldsymbol{d_A = d_B}$.

$$d_A - d_B = (r_1 c_2 - r_2 c_1)L^2, \text{ so } d_A = d_B \Leftrightarrow \frac{r_2}{r_1} = \frac{c_2}{c_1}.$$

(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

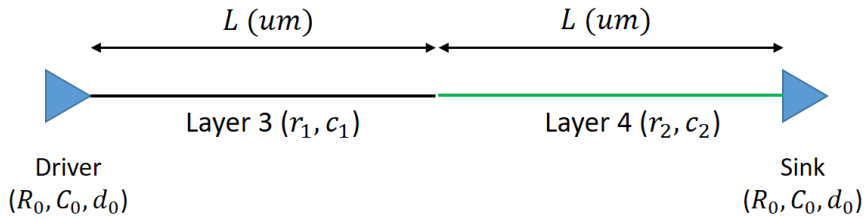
$$d = R_0(c_1 L + c_2 L + C_0) + r_1 L C_0 + r_2 L C_0 + \sum_{j=1}^{n} \left(r_1 w_{3,j}\left(C_{3,j}\right)\right) + \sum_{j=1}^{n} \left(r_2 w_{4,j}\left(C_{4,j}\right)\right)$$

$$+ \sum_{j=1}^{n} \frac{1}{2} r_1 c_1 {w_{3,j}}^2 + \sum_{j=1}^{n} \frac{1}{2} r_2 c_2 {w_{4,j}}^2$$

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.

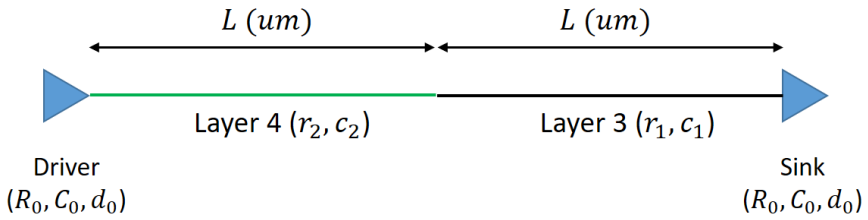
(f) (10 points) If $r_1 = r_2$ and $c_1 > c_2$, how would you route it to minimize the total delay?

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



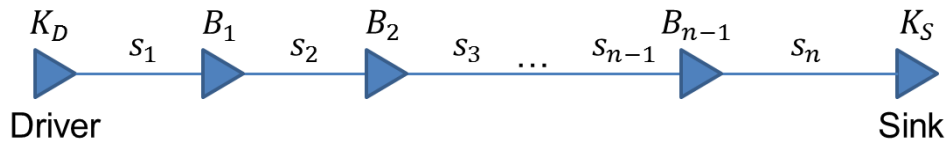
(g) (10 points) If $c_1 = c_2$ and $r_1 > r_2$, how would you route it to minimize the total delay?

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., $R_1, R_2, \ldots$)
- 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 (Ω/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

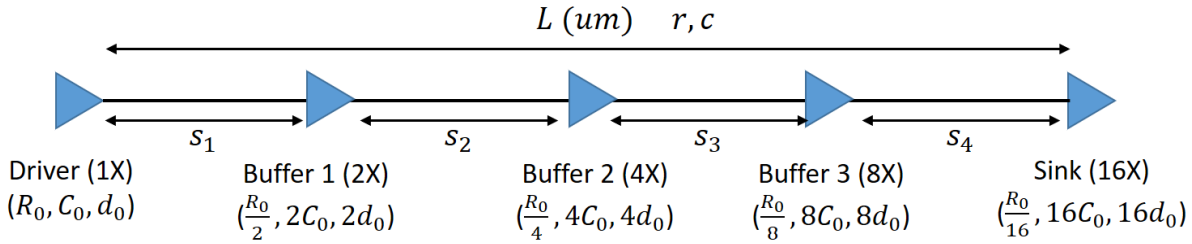
$$\tau_k = R_{k-1}(c_k \cdot s_k + C_k) + r_k \cdot s_k \cdot C_k + \frac{1}{2} r_k c_k {s_k}^2$$

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).

$d = R_0(cs_1 + 2C_0) + rs_1 2C_0 + \frac{1}{2}rcs_1{}^2 + \frac{R_0}{2}(cs_2 + 4C_0) + rs_2 4C_0 + \frac{1}{2}rcs_2{}^2 + \frac{R_0}{4}(cs_3 + 8C_0) + rs_3 8C_0 + \frac{1}{2}rcs_3{}^2 + \frac{R_0}{16}(cs_4 + 16C_0) + rs_4 16C_0 + \frac{1}{2}rcs_4{}^2 + D$ where D is the sum of the internal delays of the buffers.

$\frac{\partial d}{\partial s_1} = R_0 c + 2rC_0 + rcs_1 - \frac{R_0}{16}c - 16rC_0 - rcs_4 = 0$, so $s_1 = s_4 - \frac{15R_0}{16r} + 14\frac{C_0}{c}$

$\frac{\partial d}{\partial s_2} = \frac{R_0 c}{2} + 4rC_0 + rcs_2 - \frac{R_0}{16}c - 16rC_0 - rcs_4 = 0$, so $s_2 = s_4 - \frac{7R_0}{16r} + 12\frac{C_0}{c}$

$\frac{\partial d}{\partial s_3} = \frac{R_0 c}{4} + 8rC_0 + rcs_3 - \frac{R_0}{16}c - 16rC_0 - rcs_4 = 0$, so $s_3 = s_4 - \frac{3R_0}{16r} + 8\frac{C_0}{c}$

From $s_1 + s_2 + s_3 + s_4 = L$, we get $4s_4 - \frac{25R_0}{16r} + 34\frac{C_0}{c} = L$.

$$s_1 = \frac{L}{4} - \frac{35R_0}{64r} + \frac{11C_0}{2c}$$

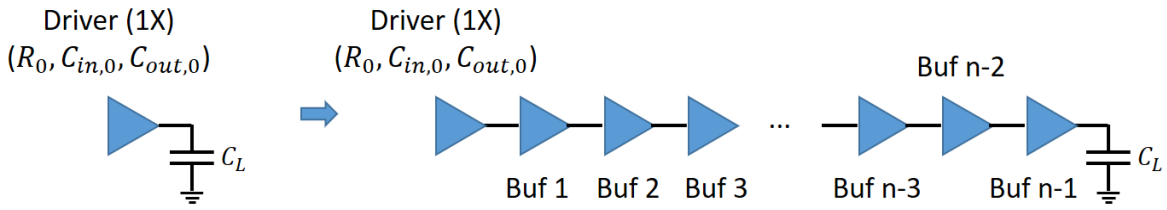
$$s_2 = \frac{L}{4} - \frac{3R_0}{64r} + \frac{7C_0}{2c}$$

$$s_3 = \frac{L}{4} + \frac{13R_0}{64r} - \frac{C_0}{2c}$$

$$s_4 = \frac{L}{4} + \frac{25R_0}{64r} - \frac{17C_0}{2c}$$

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 ×, 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 ×, the size of Buf 2 is k^2 ×, the size of Buf 3 is k^3 ×, 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 × to B × $(B > k^5)$, I should upsize Buf 3 too for optimal buffer insertion. (**True** / False)

- If I upsize Buf 5 from k^5 × to B × $(B > k^5)$, 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**)

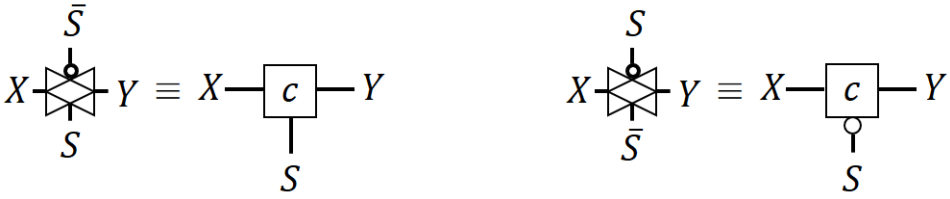
- If C_L goes down and I don't want to size the buffers, I should remove some of the buffers (i.e., reduce n) for optimal buffer insertion. (**True** / False)

Problem #5 (Transmission Gates, 10 points)

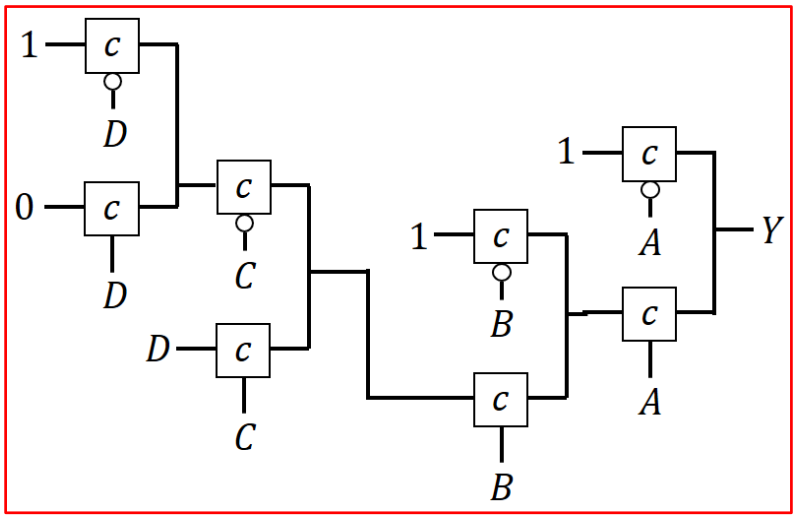
Design the following function (draw a transmission-gate-level schematic) using transmission gates only. Available input: $A, B, C, D, 0\ (GND), 1\ (VDD)$.

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

You can use the following symbols for transmission gates.

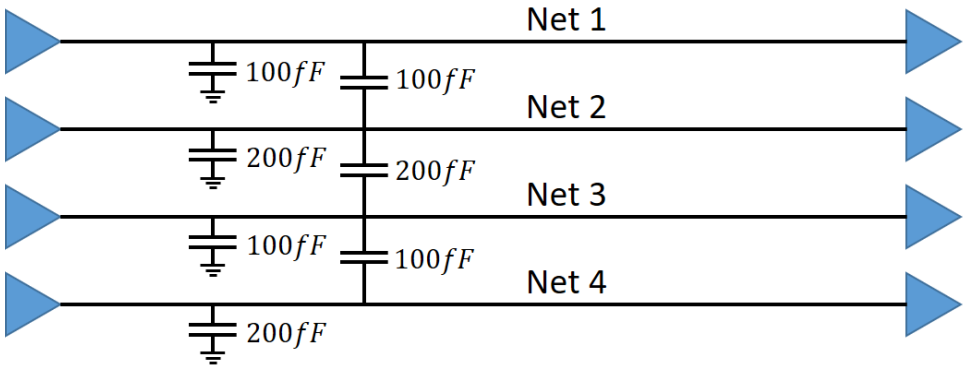


TGs ≤14: 10 points, ≤16: 7 points, ≤18: 4 points



Problem #6 (Capacitive Coupling & STA, 20 points)

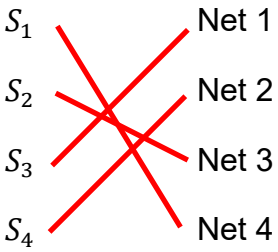
The following figure shows four nets and their ground and coupling capacitances. You can estimate the delay of a net by $R \cdot C_{eff}$ where R is the output resistance of the driver driving the net and C_{eff} is the effective capacitance of the net.



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.

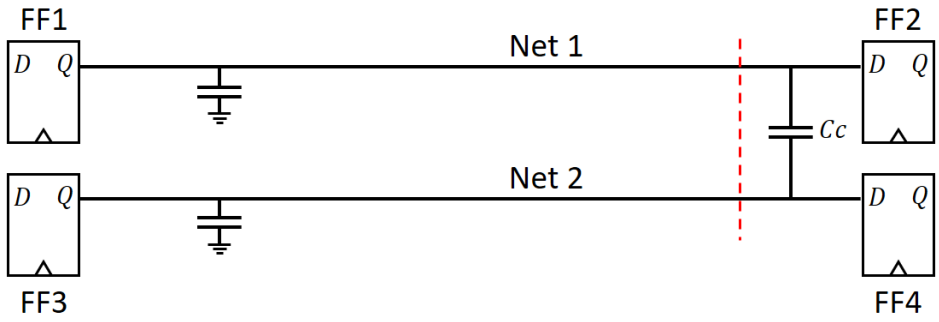
- R: $1k\Omega$ for all the drivers.
- All the drivers have the same arrival time, AT=0ps.
- S_1 and S_2 always switch in the same direction.
- S_3 and S_4 always switch in the same direction.
- S_1 (or S_2) and S_3 (or S_4) always switch in opposite directions.

RT (at the sink)
S_1: 250ps
S_2: 550ps
S_3: 150ps
S_4: 700ps

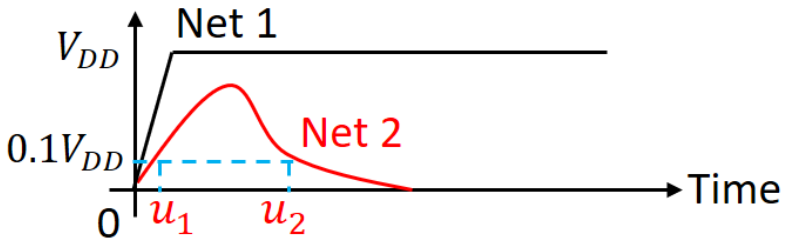


Problem #7 (Crosstalk & STA, 20 points)

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 ($k = 1\sim2$)
- s_k: setup time of FF_k ($k = 1\sim4$)
- h_k: hold time of FF_k ($k = 1\sim4$)
- x_k: delay from the clock source to the clock pin of FF_k ($k = 1\sim4$)
- c_k: clk-to-Q delay of FF_k ($k = 1\sim4$)
- 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 $\ll$ 0” could happen if many paths have very small delays. (**True** / False)