

EE466

VLSI System Design

Final Exam

Dec. 13, 2017. (1pm – 3pm)

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

Name:

WSU ID:

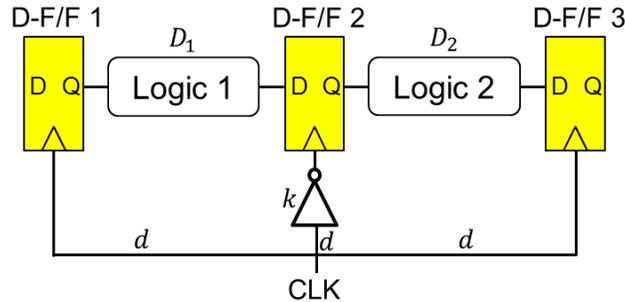
Problem	Points	
1	20	
2	30	
3	20	
4	20	
5	20	
6	20	
Total	130	

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

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

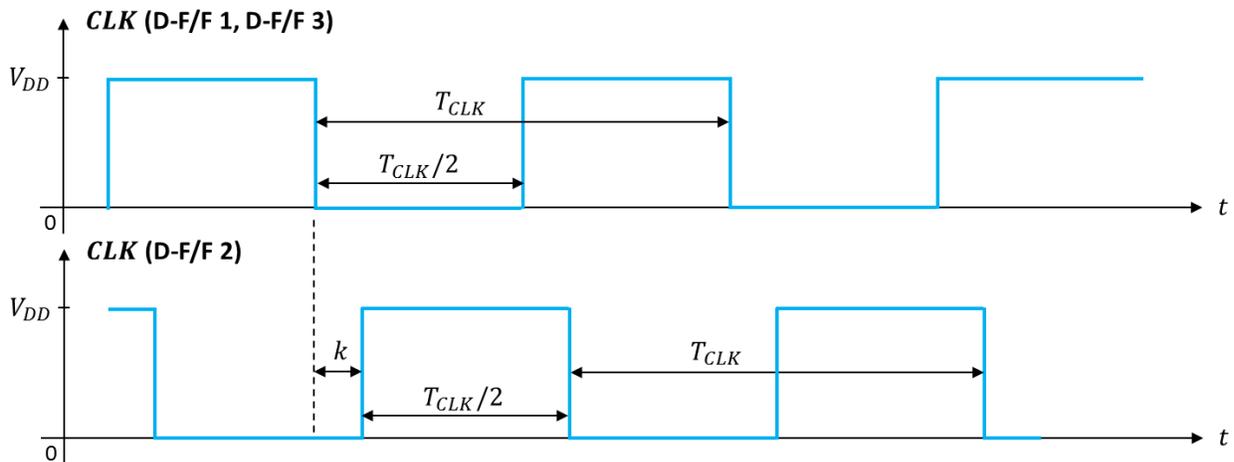
Problem #1 (Timing Analysis, 20 points)

The following shows a system consisting of two D-F/Fs and two logic stages (Logic 1 and Logic 2).

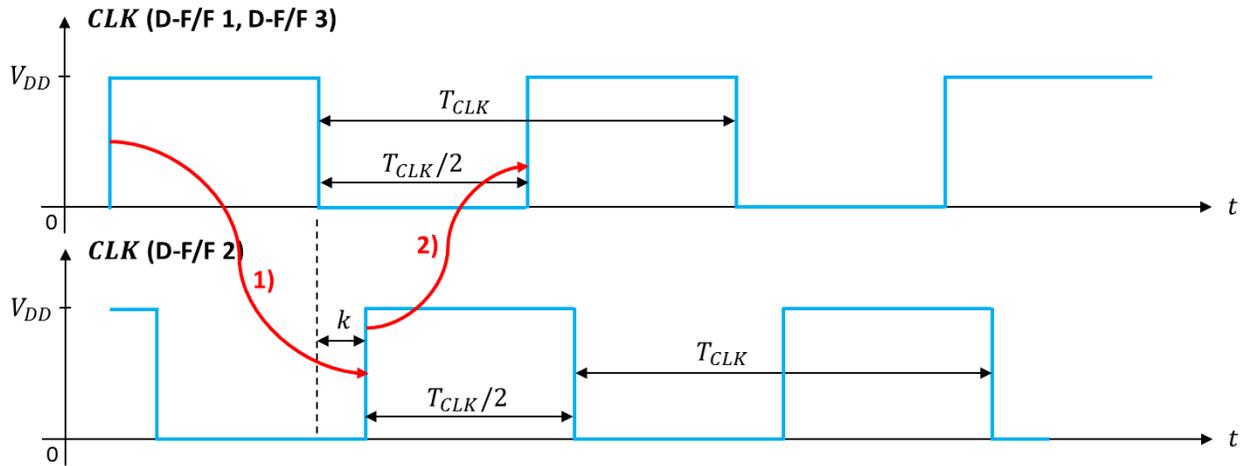


- D_n : Delay of Logic n
- d : Delay from the clock source to D-F/F 1 and the input of the inverter
- k : Delay of the inverter ($0 < k < T_{CLK}/4$)
- T_{CLK} : Clock period
- T_s : Setup time
- T_h : Hold time
- T_{CQ} : Clk-to-Q delay of a D-F/F
- Clock duty cycle: 50%

The following shows the waveforms of the clock at the clock input of D-F/F 1 and D-F/F 2 for better understanding.



- 1) Does the circuit have any setup time constraints? If yes, find all setup-time constraints (inequalities) for the system. If no, prove that the circuit does not have any setup time constraints (10 points).



1) For Logic 1: $d + T_{CQ} + D_1 \leq d + k + \frac{T_{CLK}}{2} - T_s$

$$T_{CQ} + D_1 \leq \frac{T_{CLK}}{2} + k - T_s$$

2) For Logic 2: $d + k + \frac{T_{CLK}}{2} + T_{CQ} + D_2 \leq d + T_{CLK} - T_s$

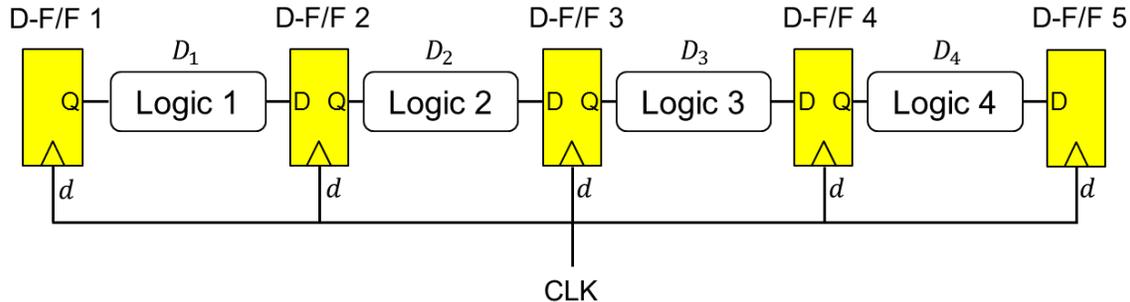
$$T_{CQ} + D_2 \leq \frac{T_{CLK}}{2} - k - T_s$$

- 2) Does the circuit have any hold time constraints? If yes, find all hold-time constraints (inequalities) for the system. If no, prove that the circuit does not have any hold time constraints (10 points).

Hold time constraints for a FF are used to guarantee that the FF captures a correct, stable input value. If the setup time constraints for Logic 1 are satisfied, the output of Logic 1 is correct and stable before the next clock rising edge arrives at D-F/F 2 as shown in the above figure. Thus, the above circuit does not need any hold time constraints.

Problem #2 (Timing Analysis, 30 points)

The following shows a pipelined logic. All the F/Fs are identical, i.e., they have the same setup time, hold time, and T_{CQ} .



The following shows the constants used in this problem.

- T_s : Setup time of a D-F/F
- T_h : Hold time of a D-F/F
- T_{CQ} : Clk-to-Q delay of a D-F/F
- D_n : Delay of Logic n
- T_{CLK} : Clock period
- d : Delay from the clock source to each D F/F.

The setup time constraint for Logic n is as follows:

$$d + T_{CQ} + D_n \leq d + T_{CLK} - T_s$$

If the delay of Logic 1 (D_1) is too large (i.e., $D_1 > T_{CLK} - T_s - T_{CQ}$), it violates the setup time constraint of Logic 1. In this case, we can intentionally apply clock skew to the system so that Logic 1 has a more loose setup time constraint while some of the other logic stages have tighter setup time constraints. This method is called “useful skew”.

We compare the following three methods for the useful skew.

1) We apply clock skew to D-F/F 2 only. In this case, the delay from the clock source to D-F/F 2 becomes $d + \alpha$ where α is positive. Assume $D_2 < T_{CLK} - T_s - T_{CQ}$. Find the range (minimum and maximum) of α , i.e., represent the minimum and the maximum values of α as functions of the constants listed above (5 points).

$$\text{Logic 1: } d + T_{CQ} + D_1 \leq d + \alpha + T_{CLK} - T_s$$

$$\text{Logic 2: } d + \alpha + T_{CQ} + D_2 \leq d + T_{CLK} - T_s$$

$$\therefore D_1 + T_{CQ} - T_{CLK} + T_s \leq \alpha \leq -D_2 - T_{CQ} + T_{CLK} - T_s$$

2) We apply the same clock skew to D-F/F 2, D-F/F 3, D-F/F 4, and D-F/F 5. In this case, the delay from the clock source to the four D-F/Fs (D-F/F 2, 3, 4, 5) becomes $d + \beta$ where β is positive. Assume $D_n < T_{CLK} - T_s - T_{CQ}$ for $n = 2,3,4,5$. Find the minimum and the maximum values of β , i.e., represent the minimum and the maximum values of β as functions of the constants listed above (10 points).

$$\text{Logic 1: } d + T_{CQ} + D_1 \leq d + \beta + T_{CLK} - T_s$$

$$\text{Logic } n \text{ (} n = 2,3,4\text{): } d + \beta + T_{CQ} + D_n \leq d + \beta + T_{CLK} - T_s$$

$$\therefore D_1 + T_{CQ} - T_{CLK} + T_s \leq \beta$$

3) We apply different clock skews to D-F/F 2, D-F/F 3, D-F/F 4, and D-F/F 5. In this case, the delay from the clock source to D-F/F m (where $m = 2,3,4,5$) becomes $d + \Delta_m$ where Δ_m is positive and Δ_5 is given ($\Delta_5 + T_{CLK} - T_s - T_{CQ} - D_4 > 0$). Find the minimum and the maximum values of Δ_2 , i.e., represent the minimum and the maximum values of Δ_2 as functions of the constants listed above ($T_s, T_h, T_{CQ}, D_1, D_2, D_3, D_4, T_{CLK}, \Delta_5$) (15 points).

$$\text{Logic 1: } d + T_{CQ} + D_1 \leq d + \Delta_2 + T_{CLK} - T_s$$

$$\text{Logic } n \text{ (} n = 2,3,4\text{): } d + \Delta_n + T_{CQ} + D_n \leq d + \Delta_{n+1} + T_{CLK} - T_s$$

The three inequalities for $n = 2,3,4$ are represented as follows:

$$\Delta_2 - \Delta_3 \leq T_{CLK} - T_s - T_{CQ} - D_2$$

$$\Delta_3 - \Delta_4 \leq T_{CLK} - T_s - T_{CQ} - D_3$$

$$\Delta_4 - \Delta_5 \leq T_{CLK} - T_s - T_{CQ} - D_4$$

To maximize Δ_2 , we should maximize Δ_3 , for which we should maximize Δ_4 .

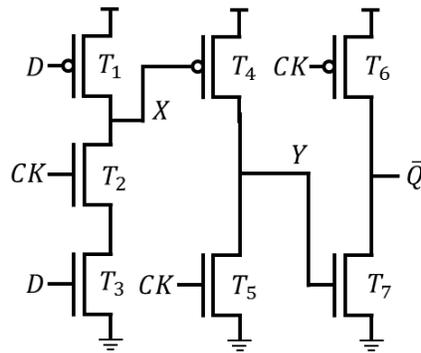
From the last inequality, the maximum value of Δ_4 is $\Delta_5 + T_{CLK} - T_s - T_{CQ} - D_4$.

Then, the maximum value of Δ_3 is $\Delta_5 + 2T_{CLK} - 2T_s - 2T_{CQ} - D_3 - D_4$.

Then, the maximum value of Δ_2 is $\Delta_5 + 3T_{CLK} - 3T_s - 3T_{CQ} - D_2 - D_3 - D_4$.

$$\therefore D_1 + T_{CQ} - T_{CLK} + T_s \leq \Delta_2 \leq \Delta_5 + 3(T_{CLK} - T_s - T_{CQ}) - D_2 - D_3 - D_4$$

Problem #3 (Sequential Logic & Transistor Sizing, 20 points)



<Chang, JSSC'96>

The above figure shows a negative edge-triggered D-F/F: However, some of the transistors should be properly sized, otherwise it won't work correctly. Find all inequalities and/or equations required to guarantee that the circuit shown above works as a negative edge-triggered D-F/F. Use the following variables/constants. (Do not care about the sampling frequency, i.e., the clock period is sufficiently long).

- D : Data input of the F/F
- CK : Clock input
- \bar{Q} : Inverted data output
- R_m : ON resistance of transistor T_m
- V_{DD} : Supply voltage
- V_{tn} : Threshold voltage of an NFET
- V_{tp} : Threshold voltage of a PFET
- If \bar{Q} is driven by a ratioed logic, \bar{Q} should be less than or equal to $0.1V_{DD}$ for logical output 0 and greater than or equal to $0.9V_{DD}$ for logical output 1.
- If a node is floating, assume that the node holds its previous value (no leakage).

1) $D = 0, \bar{Q} = 0, CK \downarrow$ (\bar{Q} should become 1)

$X = 1$. T_4 : OFF. When CK was 1, T_5 was ON, so Y was 0. When CK switches to 0, T_5 is turned OFF, so Y is still 0. T_7 is OFF, so T_6 charges the output and \bar{Q} becomes 1.

2) $D = 1, \bar{Q} = 1, CK \downarrow$ (\bar{Q} should become 0)

When D and CK are 1, X is 0. T_4 : ON. T_5 : ON. When CK switches to 0, X becomes a floating node (so X is still 0 and T_4 is still ON) and T_5 is turned OFF, so Y becomes 1 and T_7 is ON. Since CK is also 0, T_6 is turned ON, so the output is determined by the ratioed logic composed of T_6 and T_7 . Since \bar{Q} should be 0 in this case, the following inequality should be satisfied:

$$\frac{R_7}{R_6 + R_7} \leq 0.1$$

3) $D = 0, \bar{Q} = 0, CK \downarrow$, then D switches arbitrarily (\bar{Q} should stay at 1)

In this case, T_7 should always be OFF, otherwise \bar{Q} will be 1 due to the inequality in 2). Previously, T_4 was OFF and Y was 0. In this case, even if D switches to 1 or 0, X will be 1, so T_4 is always OFF, so Y is always 0.

4) $D = 0, \bar{Q} = 0, CK \downarrow$, then D switches arbitrarily and CK goes to 1 (\bar{Q} should stay at 1)

If CK switches to 1, T_5 is turned ON. If D is 0, X is 1 and T_4 is OFF, so Y will always be 0, so T_7 is OFF and \bar{Q} does not change. If D is 1, however, X becomes 0, so T_4 is turned ON and Y is determined by the ratioed logic composed of T_4 and T_5 . If Y is sufficiently high, T_7 will be turned on and \bar{Q} will become 0 due to the inequality in 2), which should not happen. Thus, even when T_4 and T_5 are turned on at the same time, T_7 should be turned OFF, so the following inequality should be satisfied:

$$\left(\frac{R_5}{R_4 + R_5} \right) V_{DD} \leq V_{tn}$$

5) $D = 1, \bar{Q} = 1, CK \downarrow$, then D switches arbitrarily (\bar{Q} should stay at 0)

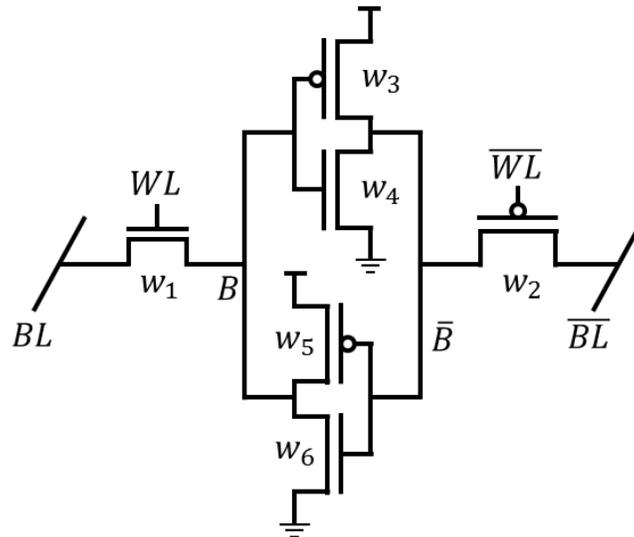
Since \bar{Q} should stay at 0, T_7 should always be ON. For that, Y should always be 1. Since CK becomes 0, T_5 is OFF. Since Y was 1 due to the inequality in 4), Y will always be 1 no matter how D changes.

6) $D = 1, \bar{Q} = 1, CK \downarrow$, then D switches arbitrarily and CK goes to 1 (\bar{Q} should stay at 0)

When CK switches back to 1, T_5 is turned ON, so no matter whether T_4 is turned ON or not, Y will be 0 (if T_4 is ON, Y is close to 0 due to the inequality in 4)). Thus, T_7 is OFF, but T_6 is turned OFF too, so \bar{Q} does not change.

Problem #4 (Memory, 20 points)

The following figure shows a 6T SRAM cell. However, it uses an NFET and a PFET for the access transistors as shown in the figure. The wordline (WL) is set to 1 to access the cell and 0 to forbid accessing the cell. Notice that \overline{WL} is connected to the gate of the PFET access transistor. The length of each TR is set to the minimum length l .



Assume that $B = 0$ ($\overline{B} = 1$) is stored in the SRAM cell. Derive all inequalities and/or equations for the widths of the six transistors to write 1 to the SRAM cell (in this case, BL and \overline{BL} are set to 1 and 0, respectively, and WL and \overline{WL} are set to 1 and 0, respectively, to access the cell and write 1 into the cell). Use the following variables/constants.

- w_m : The width of the transistor w_m
- V_{DD} : Supply voltage
- V_{tn} : Threshold voltage of an NFET
- V_{tp} : Threshold voltage of a PFET

Note: You can use the transistor current formula or the voltage-divider formula. You don't need to simplify the inequalities or equations, but you should mention exactly what conditions should be satisfied to be able to write 1 to the SRAM cell.

B: B is driven to $V_{DD} - V_{tn}$ by BL and to 0 by w_6 . V_B is determined by the following equation:

$$I_1 = \frac{1}{2} \mu_n c_{ox} \frac{w_1}{l} (V_{DD} - V_B - V_{tn})^2 = I_6 = \mu_n c_{ox} \frac{w_6}{l} \left\{ (V_{DD} - V_{tn}) V_B - \frac{1}{2} V_B^2 \right\}$$

because w_1 is in saturation mode and w_6 is in linear mode. V_B should satisfy the following inequality:

$$V_B > V_{tn}$$

to turn on w_4 .

If we use a simplified voltage-divider model, $V_B = \frac{R_6}{R_1+R_6} V_{DD} > V_{tn}$.

\bar{B} : \bar{B} is driven to $|V_{tp}|$ by $\bar{B}\bar{L}$ and to 1 by w_3 . $V_{\bar{B}}$ is determined by the following equation:

$$I_2 = \frac{1}{2} \mu_p c_{ox} \frac{w_2}{l} (V_{\bar{B}} - |V_{tp}|)^2 = I_3 = \mu_p c_{ox} \frac{w_3}{l} \left\{ (V_{DD} - |V_{tp}|)(V_{DD} - V_{\bar{B}}) - \frac{1}{2} (V_{DD} - V_{\bar{B}})^2 \right\}$$

because w_2 is in saturation mode and w_3 is in linear mode. $V_{\bar{B}}$ should satisfy the following inequality:

$$V_{\bar{B}} < V_{DD} - |V_{tp}|$$

to turn on w_5 .

If we use a simplified voltage-divider model, $V_{\bar{B}} = \frac{R_2}{R_3+R_2} V_{DD} < V_{DD} - |V_{tp}|$.

More accurately speaking, we use the following equations:

$$I_1 = \frac{1}{2} \mu_n c_{ox} \frac{w_1}{l} (V_{DD} - V_B - V_{tn})^2 = I_6 = \mu_n c_{ox} \frac{w_6}{l} \left\{ (V_{\bar{B}} - V_{tn})V_B - \frac{1}{2} V_B^2 \right\}$$

$$I_2 = \frac{1}{2} \mu_p c_{ox} \frac{w_2}{l} (V_{\bar{B}} - |V_{tp}|)^2 = I_3 = \mu_p c_{ox} \frac{w_3}{l} \left\{ (V_{DD} - V_B - |V_{tp}|)(V_{DD} - V_{\bar{B}}) - \frac{1}{2} (V_{DD} - V_{\bar{B}})^2 \right\}$$

Once w_4 and w_5 are turned on, there will be a positive feedback between the two inverters.

Problem #5 (Carry Look-Ahead Adder, 20 points)

The max. fanout is 4. The delay of an AND (OR) gate is Δ and the delay of a two-level (sum-of-product) logic is 2Δ . We are designing a 1024-bit carry look-ahead adder.

- 1) Represent c_{62} hierarchically using group-generated and group-propagated carries ($g_{i:k}, p_{i:k}$) and c_0 (primary carry-in), then compute the delay to compute c_{62} assuming all the primary input signals are available at time 0 (10 points).

$$c_{62} = g_{61} + p_{61} \cdot g_{60} + p_{61} \cdot p_{60} \cdot c_{60}$$

$$c_{60} = g_{59:56} + p_{59:56} \cdot g_{55:52} + p_{59:56} \cdot p_{55:52} \cdot g_{51:48} + p_{59:56} \cdot p_{55:52} \cdot p_{51:48} \cdot c_{48}$$

$$c_{48} = g_{47:32} + p_{47:32} \cdot g_{31:16} + p_{47:32} \cdot p_{31:16} \cdot g_{15:0} + p_{47:32} \cdot p_{31:16} \cdot p_{15:0} \cdot c_0$$

Delay computation

$$g_i, p_i: \Delta$$

$$g, p_{i:i-3}: \Delta + 2\Delta = 3\Delta$$

$$g, p_{i:i-15}: 3\Delta + 2\Delta = 5\Delta$$

$$c_{48}: 5\Delta + 2\Delta = 7\Delta$$

$$c_{60}: 7\Delta + 2\Delta = 9\Delta$$

$$c_{62}: 9\Delta + 2\Delta = 11\Delta$$

Thus, the delay is 11Δ .

- 2) Represent c_{827} hierarchically using group-generated and group-propagated carries ($g_{i:k}, p_{i:k}$) and c_0 (primary carry-in), then compute the delay to compute c_{827} assuming all the primary input signals are available at time 0 (10 points).

$$c_{827} = g_{826} + p_{826} \cdot g_{825} + p_{826} \cdot p_{825} \cdot g_{824} + p_{826} \cdot p_{825} \cdot p_{824} \cdot c_{824}$$

$$c_{824} = g_{823:820} + p_{823:820} \cdot g_{819:816} + p_{823:820} \cdot p_{819:816} \cdot c_{816}$$

$$c_{816} = g_{815:800} + p_{815:800} \cdot g_{799:784} + p_{815:800} \cdot p_{799:784} \cdot g_{783:768} + p_{815:800} \cdot p_{799:784} \cdot p_{783:768} \cdot c_{768}$$

$$c_{768} = g_{767:512} + p_{767:512} \cdot g_{511:256} + p_{767:512} \cdot p_{511:256} \cdot g_{255:0} + p_{767:512} \cdot p_{511:256} \cdot p_{255:0} \cdot c_0$$

$$p_{255:0} = p_{255:192} \cdot p_{191:128} \cdot p_{127:64} \cdot p_{63:0}$$

$$p_{63:0} = p_{63:48} \cdot p_{47:32} \cdot p_{31:16} \cdot p_{15:0}$$

$$p_{15:0} = p_{15:12} \cdot p_{11:8} \cdot p_{7:4} \cdot p_{3:0}$$

$$p_{3:0} = p_3 \cdot p_2 \cdot p_1 \cdot p_0$$

Delay computation

$$g_i, p_i: \Delta$$

$$g, p_{i:i-3}: \Delta + 2\Delta = 3\Delta$$

$$g, p_{i:i-15}: 3\Delta + 2\Delta = 5\Delta$$

$$g, p_{i:i-63}: 5\Delta + 2\Delta = 7\Delta$$

$$g, p_{i:i-255}: 7\Delta + 2\Delta = 9\Delta$$

$$c_{768}: 9\Delta + 2\Delta = 11\Delta$$

$$c_{816}: 11\Delta + 2\Delta = 13\Delta$$

$$c_{824}: 13\Delta + 2\Delta = 15\Delta$$

$$c_{827}: 15\Delta + 2\Delta = 17\Delta$$

Thus, the delay is 17Δ .

Problem #6 (Prefix Adder, 20 points)

The delay of an AND (OR) gate is Δ and the delay of a two-level (sum-of-product) logic is 2Δ . We are designing a 1024-bit Kogge-Stone adder.

- 1) Represent c_{52} hierarchically using group-generated and group-propagated carries ($g_{i:k}, p_{i:k}$) and c_0 (primary carry-in), then compute the delay to compute c_{52} assuming all the primary input signals are available at time 0 (10 points).

$$c_{52} = g_{51:in} = g_{51:20} + p_{51:20} \cdot g_{19:in}$$

$$g_{19:in} = g_{19:4} + p_{19:4} \cdot g_{3:in}$$

$$g_{3:in} = g_{3:0} + p_{3:0} \cdot c_0$$

Delay computation

$$g_i: \Delta, p_i: 2\Delta$$

$$g, p_{i:i-1} = 2\Delta + 2\Delta = 4\Delta$$

$$g, p_{i:i-3} = 4\Delta + 2\Delta = 6\Delta$$

$$g_{3:in}: 6\Delta + 2\Delta = 8\Delta$$

$$g, p_{i:i-7} = 6\Delta + 2\Delta = 8\Delta$$

$$g, p_{i:i-15} = 8\Delta + 2\Delta = 10\Delta$$

$$g, p_{i:i-31} = 10\Delta + 2\Delta = 12\Delta$$

$$g_{19:in} = \text{MAX}(10\Delta, 8\Delta) + 2\Delta = 12\Delta$$

$$g_{51:in} = \text{MAX}(12\Delta, 12\Delta) + 2\Delta = 14\Delta$$

Thus, the delay is 14Δ .

- 2) Represent c_{987} hierarchically using group-generated and group-propagated carries ($g_{i:k}, p_{i:k}$) and c_0 (primary carry-in), then compute the delay to compute c_{52} assuming all the primary input signals are available at time 0 (10 points).

$$c_{987} = g_{986:in} = g_{986:475} + p_{986:475} \cdot g_{474:in}$$

$$g_{474:in} = g_{474:219} + p_{474:219} \cdot g_{218:in}$$

$$g_{218:in} = g_{218:91} + p_{218:91} \cdot g_{90:in}$$

$$g_{90:in} = g_{90:27} + p_{90:27} \cdot g_{26:in}$$

$$g_{26:in} = g_{26:11} + p_{26:11} \cdot g_{10:in}$$

$$g_{10:in} = g_{10:3} + p_{10:3} \cdot g_{2:in}$$

$$g_{2:in} = g_{2:1} + p_{2:1} \cdot g_{0:in}$$

$$g_{0:in} = g_0 + p_0 \cdot c_0$$

$$g_i: \Delta, p_i: 2\Delta$$

$$g_{0:in}: 4\Delta$$

$$g_{2:in} = \text{MAX}(4\Delta, 4\Delta) + 2\Delta = 6\Delta$$

$$g_{10:in} = \text{MAX}(8\Delta, 6\Delta) + 2\Delta = 10\Delta$$

$$g_{26:in} = \text{MAX}(10\Delta, 10\Delta) + 2\Delta = 12\Delta$$

$$g, p_{i:i-63} = 12\Delta + 2\Delta = 14\Delta$$

$$g, p_{i:i-127} = 14\Delta + 2\Delta = 16\Delta$$

$$g, p_{i:i-255} = 16\Delta + 2\Delta = 18\Delta$$

$$g, p_{i:i-511} = 18\Delta + 2\Delta = 20\Delta$$

$$g_{90:in} = \text{MAX}(14\Delta, 12\Delta) + 2\Delta = 16\Delta$$

$$g_{218:in} = \text{MAX}(16\Delta, 16\Delta) + 2\Delta = 18\Delta$$

$$g_{474:in} = \text{MAX}(18\Delta, 18\Delta) + 2\Delta = 20\Delta$$

$$g_{986:in} = \text{MAX}(20\Delta, 20\Delta) + 2\Delta = 22\Delta$$

Thus, the delay is 22Δ .