Homework 2: Flip-Flops and Registers

Solve the **five** following problems. Show all your work/process. Please neatly write or type your answers, and please scan or upload your answers in a single PDF file to the assignment submission on Carmen. Answers may be graded for correctness, thoroughness, completion, or some combination.

2.1 Use the adapter design pattern approach to implement a D flip-flop using an SR flip-flop.

a. Show your work and draw the circuit for the D flip-flop.

b. For the D flip-flop you drew in part a, will the propagation delay from the clock to the output for the D flip-flop be the same, greater, or smaller than the propagation delay from the clock to the output for the SR flip-flop inside? Explain your answer.

c. For the D flip-flop you drew in part a, will the setup time of the D flip-flop be the same, greater, or smaller than the setup time of the SR flip-flop inside? Explain your answer.

2.2 Redesign the right-shift register circuit shown in Figure 1 using four D flip-flops with clock enable (CE) input, four 2-to-1 MUXes, and one single OR gate. The final design should have the same parallel-in parallel-out and serial-in serial-out functionality, and the same combinations for *Sh* and *L* should perform the same operations (keep state, load, and right shift) according to Figure 2. Write at least 1 complete sentence describing what you did and why it works.

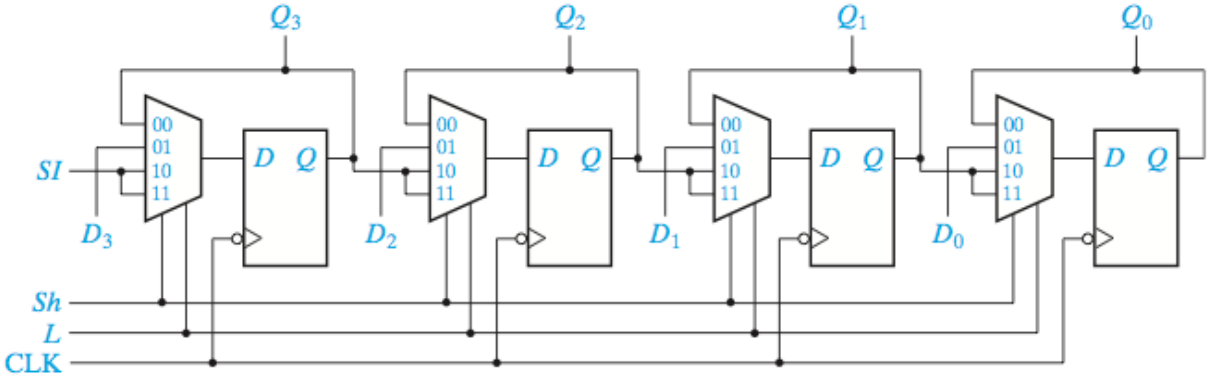


Figure 1: Parallel-in parallel-put right-shift register (Figure 12-10 from textbook).

Inputs		Next State				
Sh (Shift)	*L* (Load)	Q_3^+	Q_2^+	Q_1^+	Q_0^+	Action
0	0	Q_3	Q_2	Q_1	Q_0	No change
0	1	D_3	D_2	D_1	D_0	Load
1	X	*SI*	Q_3	Q_2	Q_1	Right shift

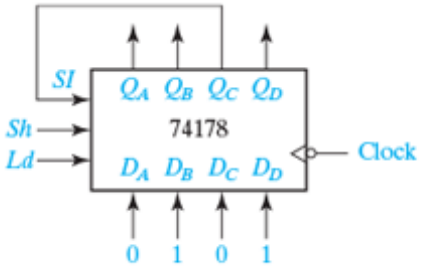
Figure 2: Table showing *Sh* and *L* input combinations with corresponding circuit actions.

2.3 Design a left-shift register similar to that of Figure 1. Your left-shift register should shift left if $Sh = 1$, load if $Sh = 0$ and $L = 1$, and keep its state if $Sh = L = 0$.

a. Draw the circuit using four D flip-flops and four 4-to-1 MUXes. Write at least 1 complete sentence describing what you did and why it works.

b. Derive the next-state equations for the four flip-flops. The equations will be in terms of inputs Sh and L, serial-in input SI, parallel-in data D_3, D_2, D_1, D_0, and current states Q_3, Q_2, Q_1, Q_0.

2.4 A 74178 shift register is described by the given block diagram and truth table in Figure 3. All state changes occur on the $1 - 0$ transition of the clock. The shift register is connected as shown in the block diagram. Complete the timing diagram.



Sh	Ld	Q_A^+	Q_B^+	Q_C^+	Q_D^+
0	0	Q_A	Q_B	Q_C	Q_D
0	1	D_A	D_B	D_C	D_D
1	X	SI	Q_A	Q_B	Q_C

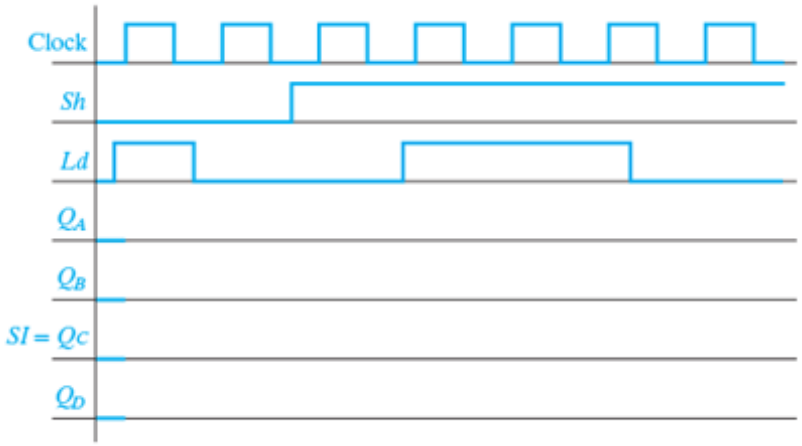


Figure 3: 74178 shift register block diagram, truth table, and timing diagram.

2.5 A digital system can perform any four-variable bitwise logic function, but it may take several clock cycles. (A bitwise logic function performs the same logic function on each bit. For example: 0101 NAND 1100 = 1011.) Recall that the NAND operation is functionally complete, which means we can do any logic function by a series of NAND operations.

Figure 4 shows four 8-bit registers: A, B, C, and D. These four registers are connected via 8-bit buses. Registers A and B can output onto the X bus. Registers C and D can output onto the Y bus. To implement the bitwise logic functions, there are 8 NAND gates whose inputs are the X and Y buses. The 8 NAND gates perform bitwise NAND operation on the bits of X and Y. The output result R of the NAND gates can be loaded into any of the registers A, B, C, or D.

En is a tri-state buffer enable which allows the register to output onto the X or Y bus. CE is a clock enable which allows the register to load the result R. Additionally, there is another tri-state buffer on the right of the circuit which can output onto the Y bus. The enable signal and input signal for this tri-state buffer are not indicated on the figure because you will figure out what they should be.

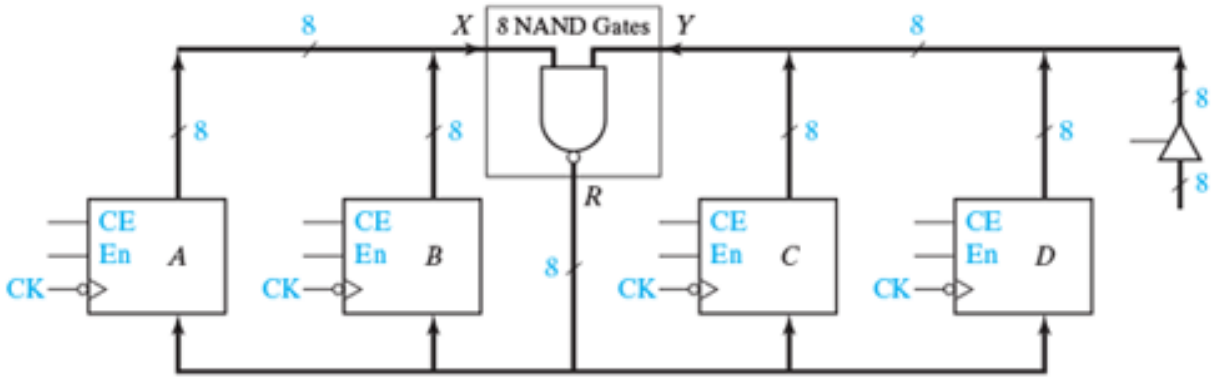


Figure 4: Circuit for performing bitwise logic functions using four 8-bit registers and NAND gates.

a. Show how to connect a 2-to-4 decoder (with inverting outputs) such that the next rising edge of the clock will load the result R into register A, B, C, or D based on control inputs $G_0G_1 = 00, 01, 10,$ or 11, respectively. Add the decoder, the input signals G_0 and G_1, and any other logic you need to the diagram shown in Figure 4. Write at least 1 complete sentence describing what you added and why it works.

b. Show how to connect three control signals, E_0, E_1, and E_2, to the registers such that $E_0 = 0$ places the A register contents on the X bus, $E_0 = 1$ places B onto the X bus, $E_1E_2 = 00$ places C onto the Y bus, $E_1E_2 = 01$ places D onto the Y bus, $E_1E_2 = 10$ places 00000000 onto the Y bus, and $E_1E_2 = 11$ places 11111111 onto the Y bus. You will use a few additional logic gates. (Hint: Connect E_2 to the input signal of the tri-state buffer on the right side of the circuit.) Add the input signals E_0, E_1, and E_2, and any other logic you need to the diagram shown in Figure 4 along with what you added for part a. Write at least 1 complete sentence describing what you added and why it works.