1. Consider a modified SR-FF as depicted in the figure below.

(a) Sketch a typical set of timing diagrams that can demonstrate the operational behavior of the flip-flop. Describe the difference between this modified SR-FF and the original one described in lecture.

(b) Draw the finite state machine for the modified SR-FF.

(c) Figure 1 shows another extension of the SR-FF, called the data latch. Sketch the timing diagram for the output $Q$ of the data latch given that $Q = 1$ initially.

2. Consider the design of a multiplier which accepts two 2-bit non-negative binary numbers – $A_1A_0$ and $B_1B_0$ – as inputs and yields the 4-bit output $Y_3Y_2Y_1Y_0$. Assume that the stock room has an ample supply of basic gates as well as full adders.

(a) Draw the circuit that performs the partial product of $B_0$ and $A_1A_0$. How about the product of $B_1$ and $A_1A_0$?
(b) Use your result in Part (a) to implement the multiplier. *Hint:* use a cascade of full adders and beware of the shift.

(c) Find the logic function for $Y_3$, the multiplication overflow bit.

3. The stock room has an ample supply of basic gates and toggle flip-flops.

   (a) Design a 2-bit modulo-4 counter that counts **backward**. Suppose that your counter’s initial display is 3 in decimal, it should show $3 \rightarrow 2 \rightarrow 1 \rightarrow 0 \rightarrow 3 \rightarrow 2 \rightarrow 1 \rightarrow 0 \rightarrow 3 \rightarrow 2 \rightarrow ...$

   (b) Sketch the finite state machine describing the operation of your reverse counter in Part (a). *Hint:* You may need more than two states here!

   (c) We would like to add another input signal labeled **RESET** to clear the display of the counter to 11 whenever it is activated. Draw the modified counter circuit with the additional **RESET** input.

4. Consider the **JK flip-flop** with inputs {$J, K, CLK$} and outputs {$Q, \overline{Q}$}, operating according to the following rules:

   - When both $J$ and $K$ are low, the flip-flop stays put (no change at output $Q$).
   - When $J = 0$ and $K = 1$, the flip-flop resets ($Q = 0$).
   - When $J = 1$ and $K = 0$, the flip-flop sets ($Q = 1$).
   - When both $J$ and $K$ are high, the flip-flop will toggle its output $Q$ at every falling edge of the clock input.

   (a) Show how to use the JK flip-flops to design a Modulo-8 counter.

   (b) Draw timing diagrams of your counter’s outputs {$Q_2, Q_1, Q_0$} along with a clock $CLK$ signal to demonstrate its operations.

   (c) Draw the finite state machine for your Modulo-8 counter.

   (d) Design a Modulo-6 counter, again using the JK flip-flops and basic gates.

   (e) Can we use the JK flip-flops as memory cells to store binary data? Justify why or why not.

Due date: **October 12** in class