Design a 4-bit rotating parallel-to-serial converter circuit with inputs as given in the diagram to the right. The circuit accepts a clock signal, CLK, a 4-bit wide data word, X <X₃, X₂, X₁, X₀>, a load signal, LD, and a control signal, order. The LD signal will be asserted for periods of no more than one clock cycle. When X is loaded into the circuit, if order is 0, then the bits of X will appear at y rotating least significant bit first (X₀, X₁, X₂, X₃, X₀, …) If, when X is loaded, order is 1, then the bits of X will appear at y rotating most significant bit first (X₃, X₂, X₁, X₀, X₃, …). The value of order on any other cycle is ignored. The bits will continue rotating until LD is asserted again, at which point the circuit will load a new value. Using only flip-flops and multiplexors (mux), draw a circuit that has the specified functionality.

Label every input and output clearly, and show which boxes are FFs, with a small triangle on the CLK input. Label any mux data inputs with “0” and “1”.

There are two interesting solutions to this problem. The first, shown above, is to build a unidirectional shift register that is loaded either MSB or LSB first. The second interesting solution involves building a bi-directional shift register and remembering the value of order so as to agree with the specified behavior. To store the value of order, an additional flip-flop and mux must be used, achieving a setup wherein LD acts as an enable signal for the flip-flop storing the value of order.