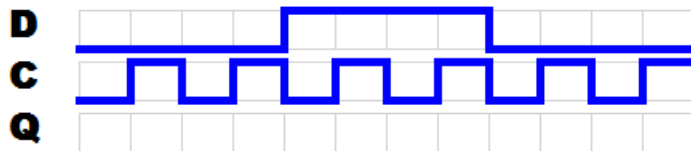
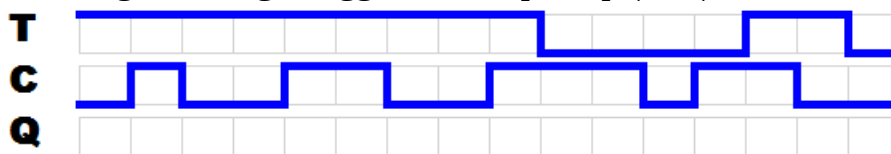


P1 (20 points): Complete the following timing diagrams for the specified components. The clock is C. You may assume that Q is initially at 0 unless specified otherwise.

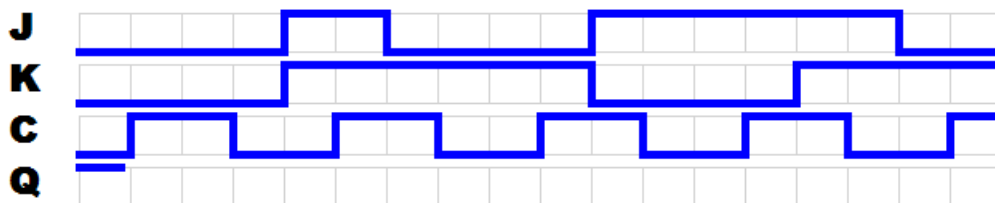
A: A positive-edge-triggered D Flip-Flop (DFF).



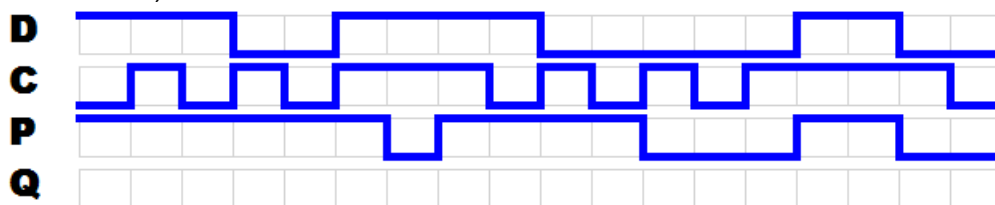
B: A negative-edge-triggered T Flip-Flop (TFF).



C: A positive-edge-triggered JK Flip-Flop (JKFF).



D: A negative-edge-triggered DFF with active-low Preset P (preset occurs when P=0).



P2 (12 points): A given register file can support storing values in its 32 registers. Each register is designed to hold numbers ranging from -25 to +25 (in 2's complement) with no additional bits beyond those necessary to hold numbers in this range. Answer the following questions:

A: What is the **width** of the LD_DATA bus? (Note that **width** is the number of bits)

B: What is the width of each register?

C: What is the width of the RA bus?

D: What is the width of the WA bus?

E: How many DFFs exist in this register file?

F: What type of decoder is used in this register file?

P3 (16 points): Fill in the diagram below for the register file containing four 4-bit registers given the following data points:

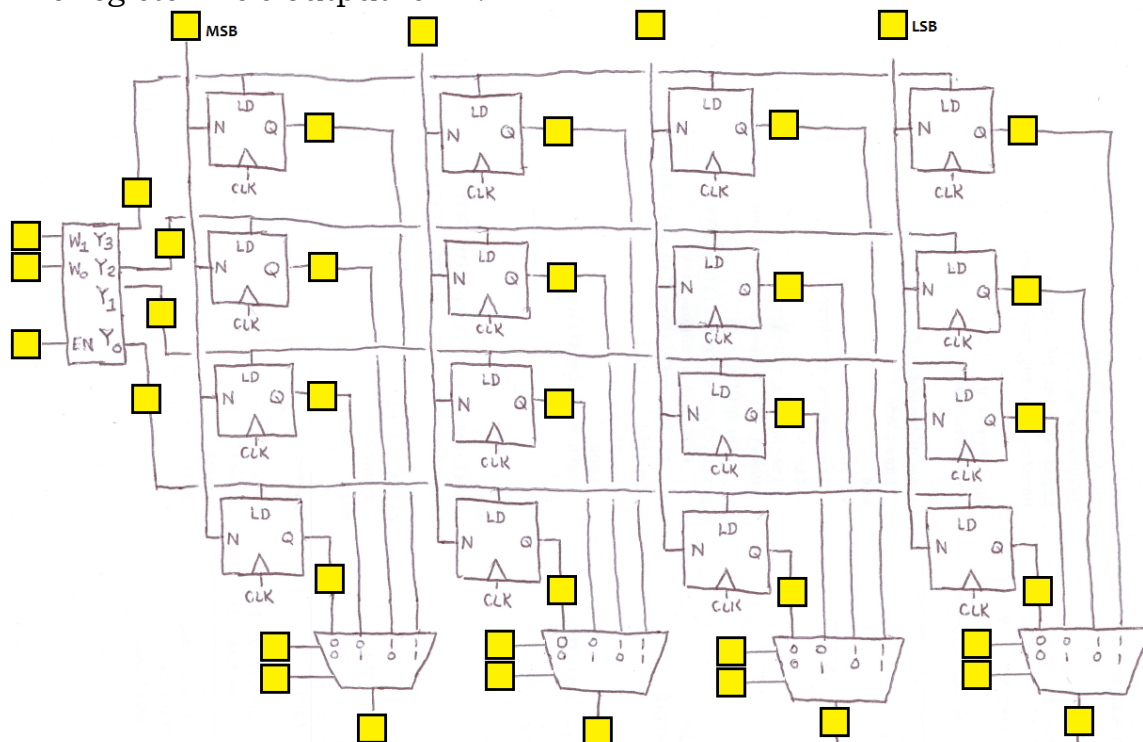
Register 0 contains the value 3, Register 1 contains the value 10,

Register 2 contains the value 14, and Register 3 contains 5.

Writing is enabled.

The user will write the value 7 to register 1 on the next clock cycle.

The register file's output is 14.



P4 (16 points): Design a four-bit register with both shift and parallel load features. The inputs of the register include a 2-bit input bus J as J_1J_0 , a 4-bit input bus X as $X_3X_2X_1X_0$, and a clock signal. The register will have a 4-bit output bus Q that represents the value stored in the register. You are allowed to use any number and size of the following: DFFs, MUXes, decoders, encoders, AND gates, OR gates, and NOT gates (Notice that you do not need all of them). The operation of the registers are defined below:

If $J=0$, then the output Q will take on the values in X. $Q_3^{new} = X_3, Q_2^{new} = X_2, Q_1^{new} = X_1, Q_0^{new} = X_0$

If $J=1$, then the output Q remains unchanged. $Q_3^{new} = Q_3^{old}, Q_2^{new} = Q_2^{old}, Q_1^{new} = Q_1^{old}, Q_0^{new} = Q_0^{old}$

If $J=2$, then the output Q is shifted to the left $Q_3^{new} = Q_2^{old}, Q_2^{new} =$

$$Q_1^{old}, Q_1^{new} = Q_0^{old}, Q_0^{new} = X_0$$

If $J=3$, then the output Q is shifted to the right. $Q_3^{new} = X_0, Q_2^{new} = Q_3^{old}, Q_1^{new} = Q_2^{old}, Q_0^{new} = Q_1^{old}$

P5 (16 points): For the questions below, assume that the clock is 640 Hz (640 pulses per second at a fixed interval).

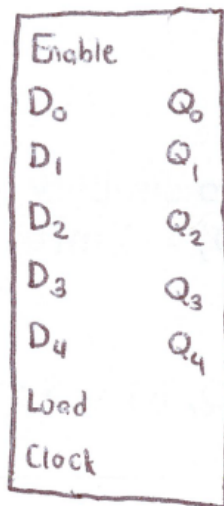
A: If the clock is connected to a modulo-10 counter, how long does it take the counter to count through each number and return to zero?

B: If the clock is connected to a modulo-32 counter, how long does it take the counter to count through each number and return to zero?

C: How many flip-flops would be required to construct a modulo-32 counter?

D: What type of counter would be necessary such that the counter will return to zero after exactly 64 seconds?

P6 (15 points): The component below is a 5-bit synchronous up-counter with synchronous parallel-load. Design the following components using 5-bit synchronous up-counters and other gates as necessary:



A: Design a 5-bit down counter.

B: Design a counter which produces the following sequence: 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 3, 4, 5, 6, 7....

P7 (5 points): What advantages are there to using a synchronous up-counter instead of an asynchronous up-counter? What advantages are there to using an asynchronous up-counter instead of a synchronous up-counter?