Cpr E 281 HW08
ELECTRICAL AND COMPUTER ENGINEERING
IOWA STATE UNIVERSITY

Flip Flops, Counter, and Registers
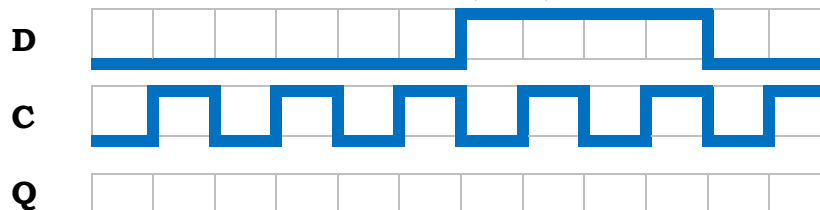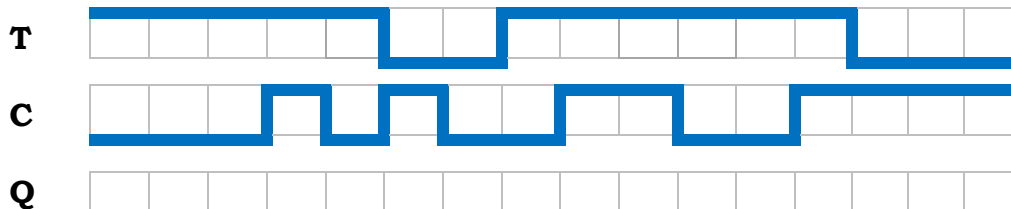## Assigned: Week 9
Due Date: Oct. 19, 2020

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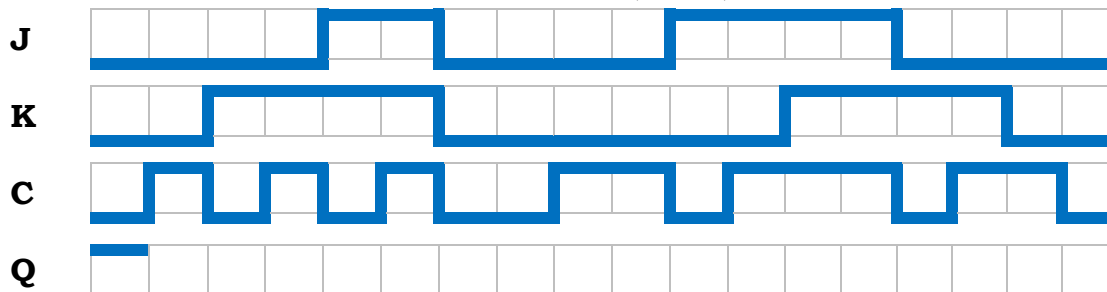
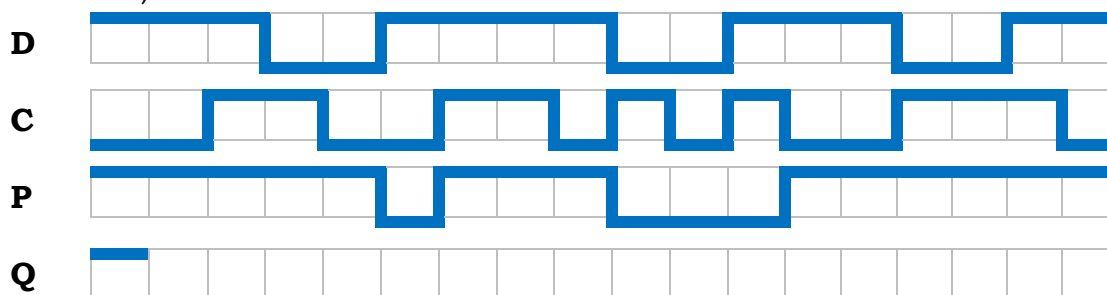A: A positive-edge-triggered D Flip-Flop (DFF).

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).

Cpr E 281 HW08

ELECTRICAL AND COMPUTER
ENGINEERING
IOWA STATE UNIVERSITY

Flip Flops, Counter, and Registers
Assigned: Week 9
Due Date: Oct. 19, 2020

P2 (10 points): Show how a D Flip-Flop (DFF) can be made using a T Flip-Flop (TFF). Your circuit must contain all of the functionality of a DFF (PRESET and CLEAR implementations are not necessary) but must use only one TFF and one 2-1 MUX.

P3. (10points)
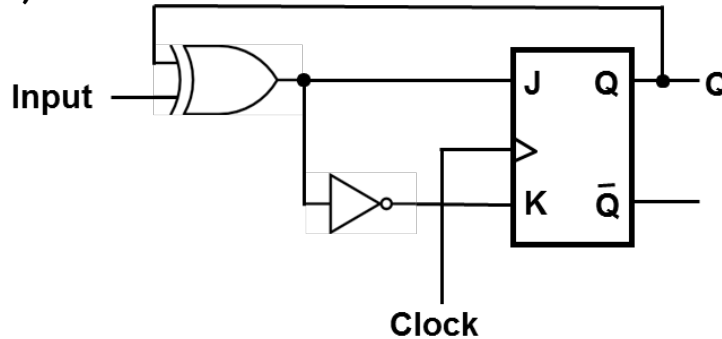
a) Complete the truth table for the circuit above.

Input	Q(t)	J	K	Q(t+1)
0	0			
0	1			
1	0			
1	1			

b) Based on the truth table, could you identify which flip-flop it is?

P4 (15 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 operations of the registers are defined below:

- If J=0, 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=1, then the output Q is shifted to the right.
$$Q_3^{new} = X_0, Q_2^{new} = Q_3^{old}, Q_1^{new} = Q_2^{old}, Q_0^{new} = Q_1^{old}$$
- 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 will take on the values in X.
$$Q_3^{new} = X_3, Q_2^{new} = X_2, Q_1^{new} = X_1, Q_0^{new} = X_0$$

2

Cpr E 281 HW08
ELECTRICAL AND COMPUTER ENGINEERING
IOWA STATE UNIVERSITY

Flip Flops, Counter, and Registers
Assigned: Week 9
Due Date: Oct. 19, 2020

P5 (15 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 5, Register 1 contains the value 8, Register 2 contains the value 6, and Register 3 contains 14.

Writing is enabled.

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

The register file's output is 6.
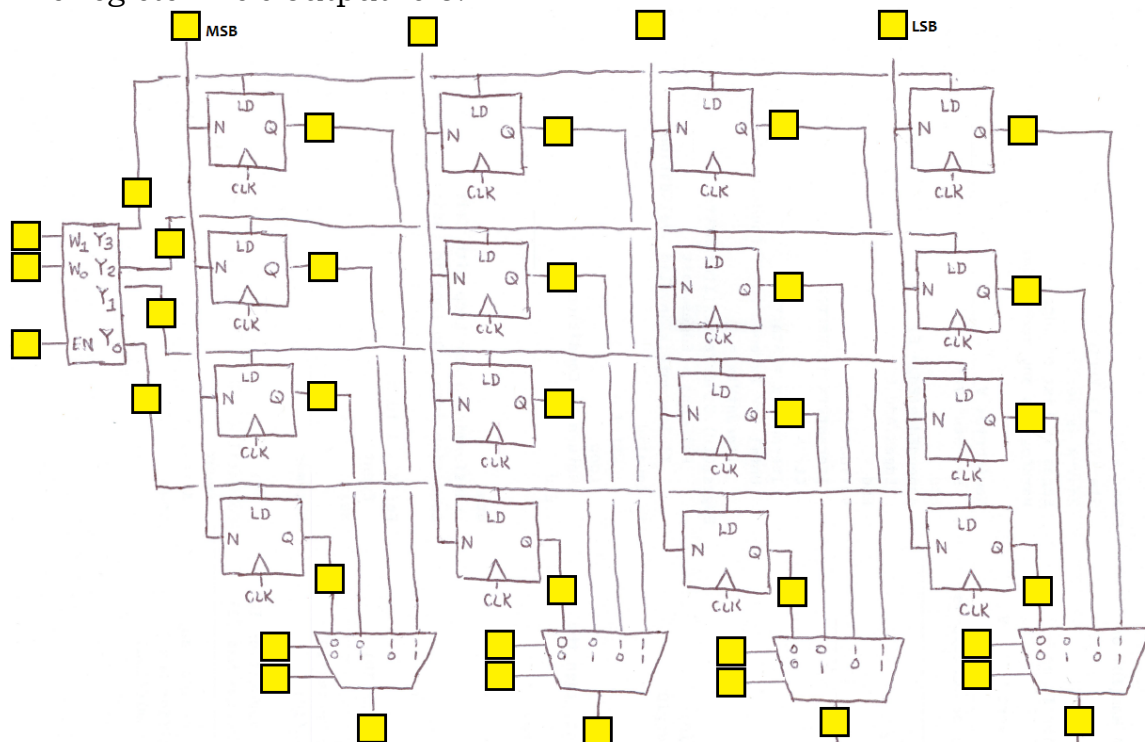
P6 (15 points): For the questions below, assume that the clock is 128 Hz (128 pulses per second at a fixed interval).

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?

E: How many flip-flops would be required to construct the counter in D

3

Cpr E 281 HW08
ELECTRICAL AND COMPUTER
ENGINEERING
IOWA STATE UNIVERSITY

Flip Flops, Counter, and Registers
Assigned: Week 9
Due Date: Oct. 19, 2020

P7 (10 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:
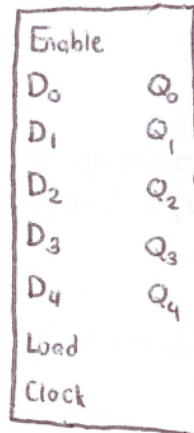
A: Design a 5-bit down counter.
B: Design a counter which produces the following sequence: 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 5, 6, 7....

P8 (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?