Your Name: ___________________________________ netid:________________ Group #:

Name: _______________________________________ netid:________________

Name: _______________________________________ netid:________________

Name: _______________________________________ netid:________________

ECE 120 Worksheet 8: Shift registers and Counters

Before you come to discussion, please read Lecture Notes Sections 2.6, 2.7, and 3.1.

In this discussion, you will implement a synchronous counter. You will also design several serial shift registers.

1. Synchronous Counters

1. Using D flip-flops, implement a synchronous counter that generates the output sequence

000 → 010 → 011 → 101 → 110 → 000....

Your answer should consist of a **next-state table**, **K-maps**, **Boolean expressions** (in minimal SOP form) for all D flip-flop inputs, and a **circuit drawing**.

S_2	S_1	S_0	S_2^+	S_1^+	S_0^+
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

S_1^+

S_0 \ S_2S_1	00	01	11	10
0				
1				

S_0^+

S_0 \ S_2S_1	00	01	11	10
0				
1				

S_2^+

S_0 \ S_2S_1	00	01	11	10
0				
1				

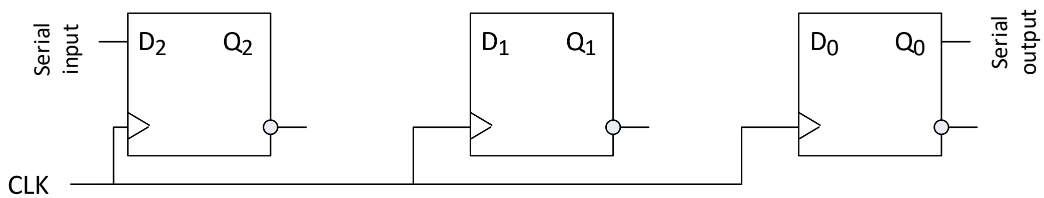
S_2^+ = ________ **S_1^+ =** __________________________ **S_0^+ =** ___________________________

2. A counter is *self-starting* if, regardless of its initial state, the counter eventually enters the desired sequence. Is your counter self-starting? Explain your answer.

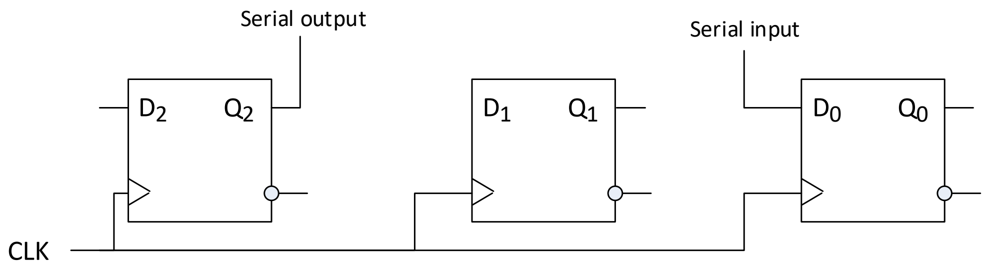
2. Serial shift registers

Draw missing connections to implement various shift registers.

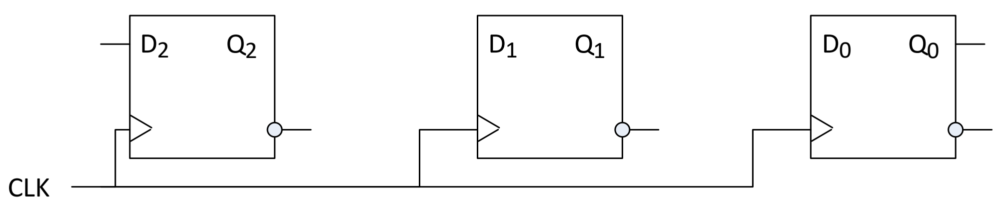
1. *Shift right*: All bits of the register move right by one position, and a new bit value from a serial input is stored in the most significant bit (leftmost flip-flop below).



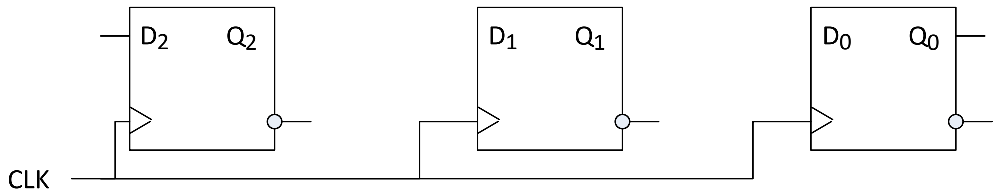
2. *Shift left*: All bits of the register move left by one position, and a new bit value from a serial input is stored in the least significant bit (rightmost flip-flow below).



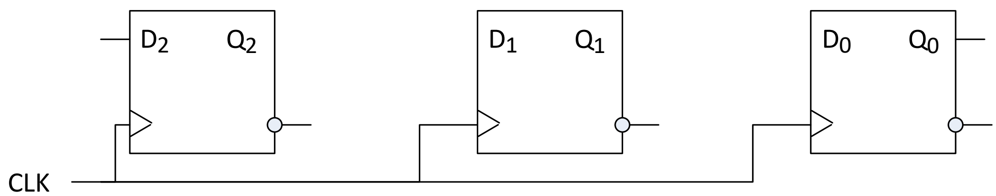
3. *Circular shift left* (also known as rotate left): the output of the leftmost flip-flop feeds back into the rightmost flip-flop. Hence each bit moves left, and the leftmost bit becomes the rightmost bit.



4. *Circular shift right* (also known as rotate right): the output of the rightmost flip-flop feeds back into the leftmost flip-flop. Hence each bit moves right, and the rightmost bit becomes the leftmost bit.

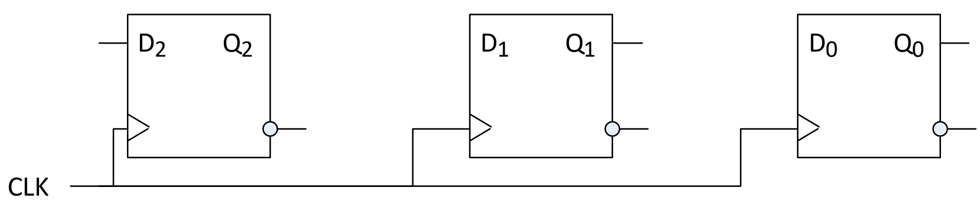


5. *Logical shift right*: The bits in the register are treated as an unsigned number (or simply a set of bits). All bits of the register move right (towards less significant digits) by one position, and a 0 is stored in the most significant bit (leftmost flip-flop below). Logical shift right corresponds to division by 2 for unsigned numbers.



---

6. *Arithmetic shift left*: The register contents are treated as a 2's complement number, so a 0 is shifted into the least significant bit (the rightmost flip-flop). Arithmetic shift left corresponds to multiplication by 2 for 2's complement numbers. Notice that *arithmetic* and *logical* shift left are identical.



---

7. *Arithmetic shift right*: The register contents are treated as a 2's complement number, so the sign bit (leftmost flip-flop below) is copied back into itself. Arithmetic shift right corresponds to division by 2 for 2's complement numbers.