Chapter 11. Counters and Shift Registers

Read Chapter 11 for additional information.

11.1. Counters

A *counter* is a finite-state machine whose state diagram contains a single cycle.

An $n-bit\ binary\ counter$ has n flip-flops and 2^n states , which are visited in the order
$0, 1, \cdots, 2^n - 1, 0, 1, \cdots, 2^n - 1, \cdots$

We saw an implementation of a 3-bit binary counter using T flip-flops.

Characteristic equation:

$Q^* = TQ' + T'Q = T \oplus Q$

$T = Q \oplus Q^*$

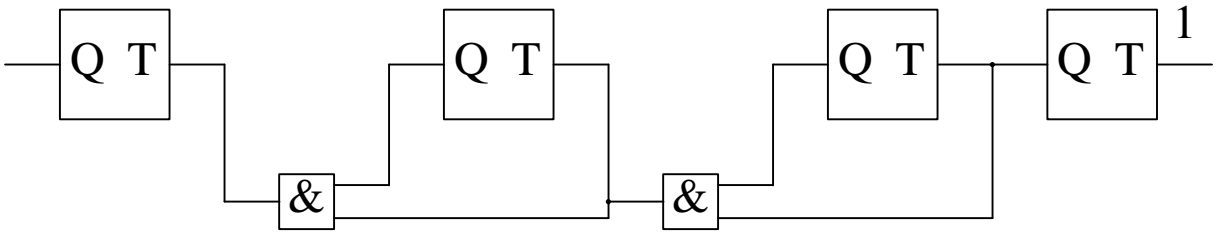
Q2	Q1	Q0	Q2*	Q1*	Q0*	T2	T1	T0
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	1
0	1	0	0	1	1	0	0	1
0	1	1	1	0	0	1	1	1
1	0	0	1	0	1	0	0	1
1	0	1	1	1	0	0	1	1
1	1	0	1	1	1	0	0	1
1	1	1	0	0	0	1	1	1

$T_0 = 1$
$T_1 = Q_0$
$T_2 = Q_1 \cdot Q_0$

For an n-bit counter, Q_i changes one clock cycle after $Q_{i-1} \cdots Q_1 Q_0 = 1 \cdots 11$. Therefore,
$T_i = Q_{i-1} \cdot \cdots \cdot Q_1 \cdot Q_0$

$T_0 = 1$
$T_1 = Q_0$
$T_2 = Q_1 \cdot Q_0$
$\cdots$
$T_i = Q_{i-1} \cdot \cdots \cdot Q_1 \cdot Q_0$
$\cdots$
$T_n = Q_{n-1} \cdot \cdots \cdot Q_1 \cdot Q_0$

Implementation:



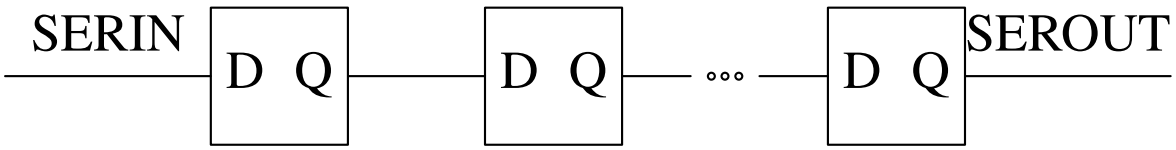
To count only when CNT=1:

$$T_0 = CNT$$
$$T_1 = CNT \cdot Q_0$$
$$T_2 = CNT \cdot Q_1 \cdot Q_0$$
$$\cdots$$
$$T_i = CNT \cdot Q_{i-1} \cdot \cdots \cdot Q_1 \cdot Q_0$$
$$\cdots$$
$$T_n = CNT \cdot Q_{n-1} \cdot \cdots \cdot Q_1 \cdot Q_0$$

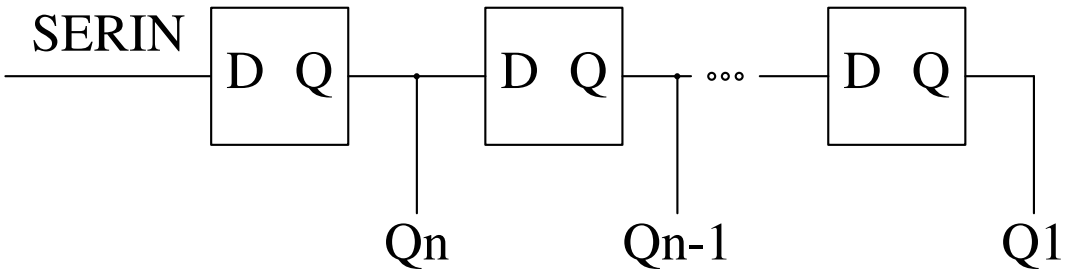
11.2. Shift Registers

A *shift register* is an *n*-bit register with a provision for shifting its data by one position at every clock cycle.

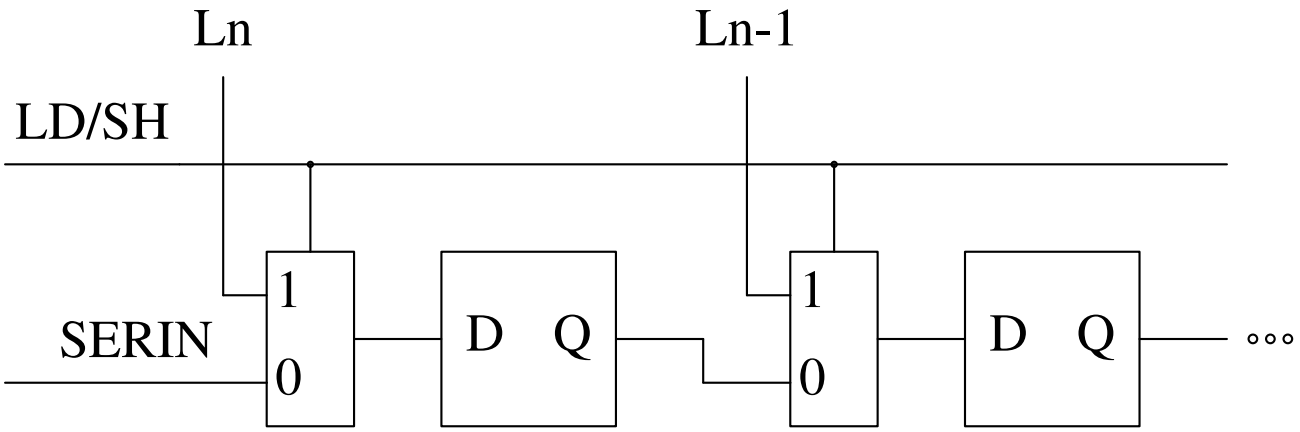
A *serial – in serial – out* shift register:



A *serial – in parallel – out* shift register:



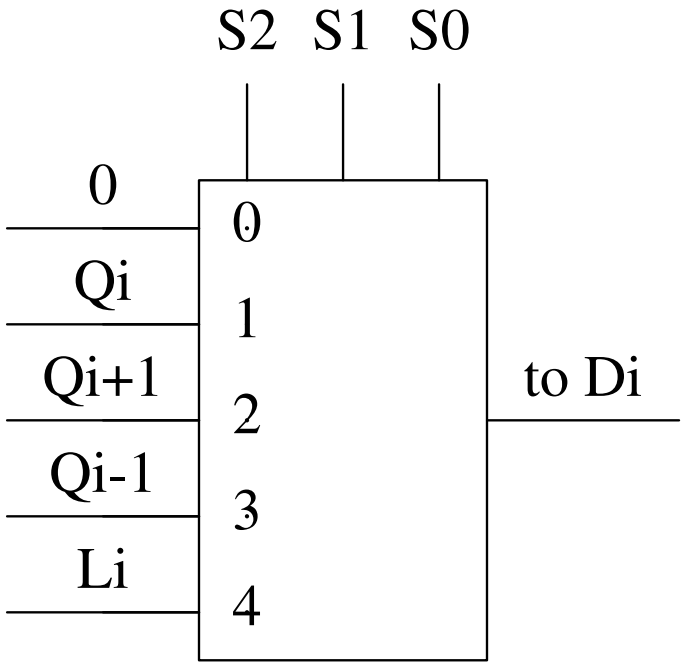
A *parallel – in serial – out* shift register:



With larger MUXes and more control inputs (also called select inputs) it is possible to design a shift-register with more functions.

Example:

S2	S1	S0	Operation
0	0	0	Clear
0	0	1	Hold
0	1	0	Shift right
0	1	1	Shift left
1	0	0	Load



An 8-bit *ring counter* goes through the sequence of states

00000001
00000010
00000100
. . .
01000000
10000000
00000001
00000010
. . .

This sequence can be implemented by an n-bit shift-register as follows.

Connect Qn to LEFTIN (on the righthand side, driving Q1).

Use parallel load to initialize the counter to state 00· · ·0001

Then use left shift for counting.

An 8-bit *twisted – ring* counter goes through the sequence of states

00000000
00000001
00000011
00000111
. . .
11111111
11111110
11111100
. . .
11000000
10000000
00000000
00000001
. . .

This sequence can be implemented by an *n*-bit shift-register as follows.
Connect Qn' to LEFTIN (on the righthand side, driving Q1).
Use clear to initialize the shift-register.
Then use left shift for counting.