CS 140 Lecture 15
Sequential Modules

Professor CK Cheng

CSE Dept.

UC San Diego

Standard Sequential Modules

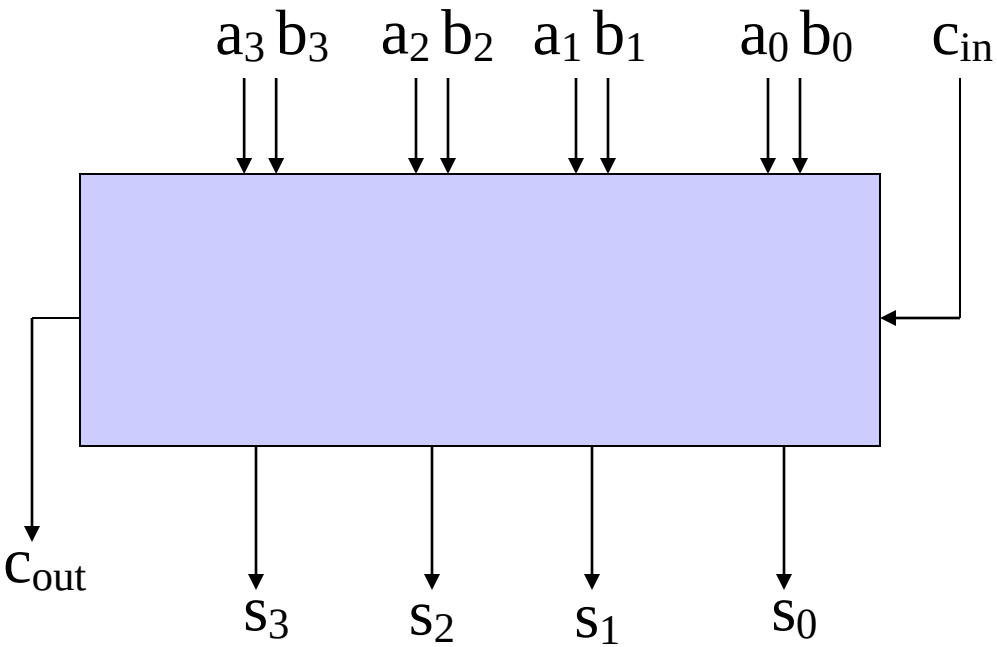
1. Serial Adders
2. Serial Multipliers
3. Register
4. Counter

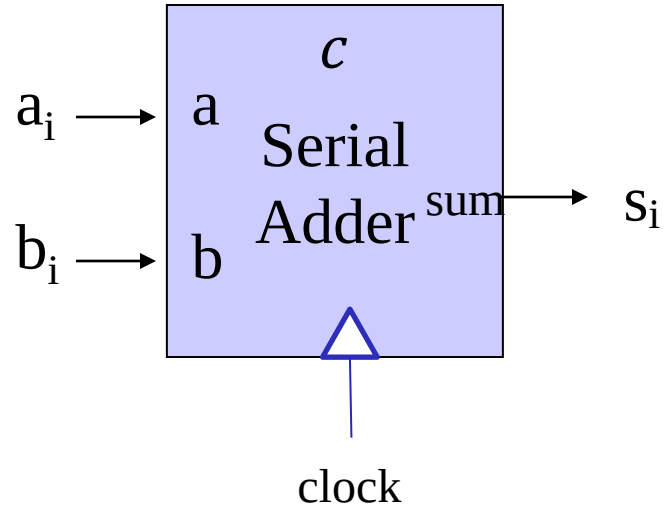
Motivation for Serial Adders and Multipliers

- Tradeoff of silicon area and system performance
  - Perform process in a series of time
- Utilization of FPGA architecture
  - Slice operation bitwise
- Metrics of Cost, Speed, and Power
- Ad: Cheaper hardware, Fit for FPGA architecture, Pipelining for excellent throughput
- Dis: Longer latency

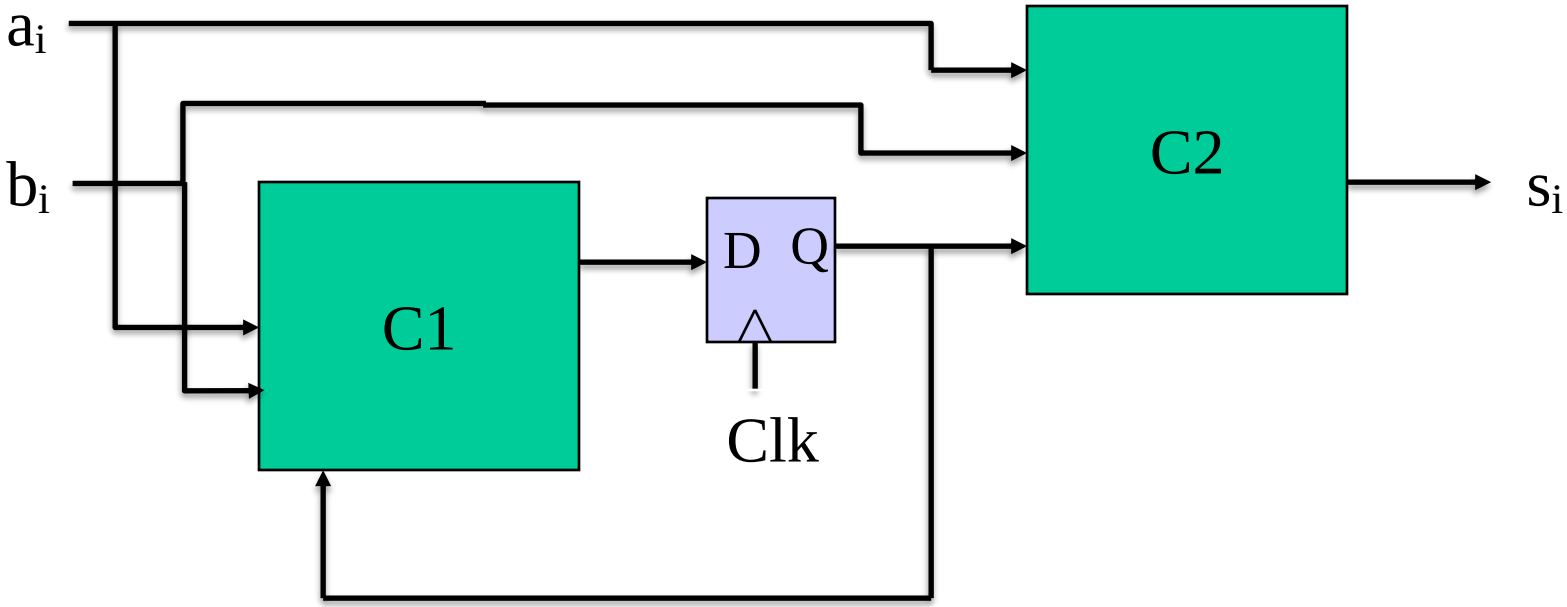
Serial Adder: Perform serial bit-addition

At time i, read a_i and b_i. Produce s_i and c_{i+1}
Internal state stores c_i. Carry bit c_0 is initialized as c_{in}





Serial Adder using D F-F



Feed a_i and b_i and generate s_i at time i.
Where are c_i and c_{i+1}?

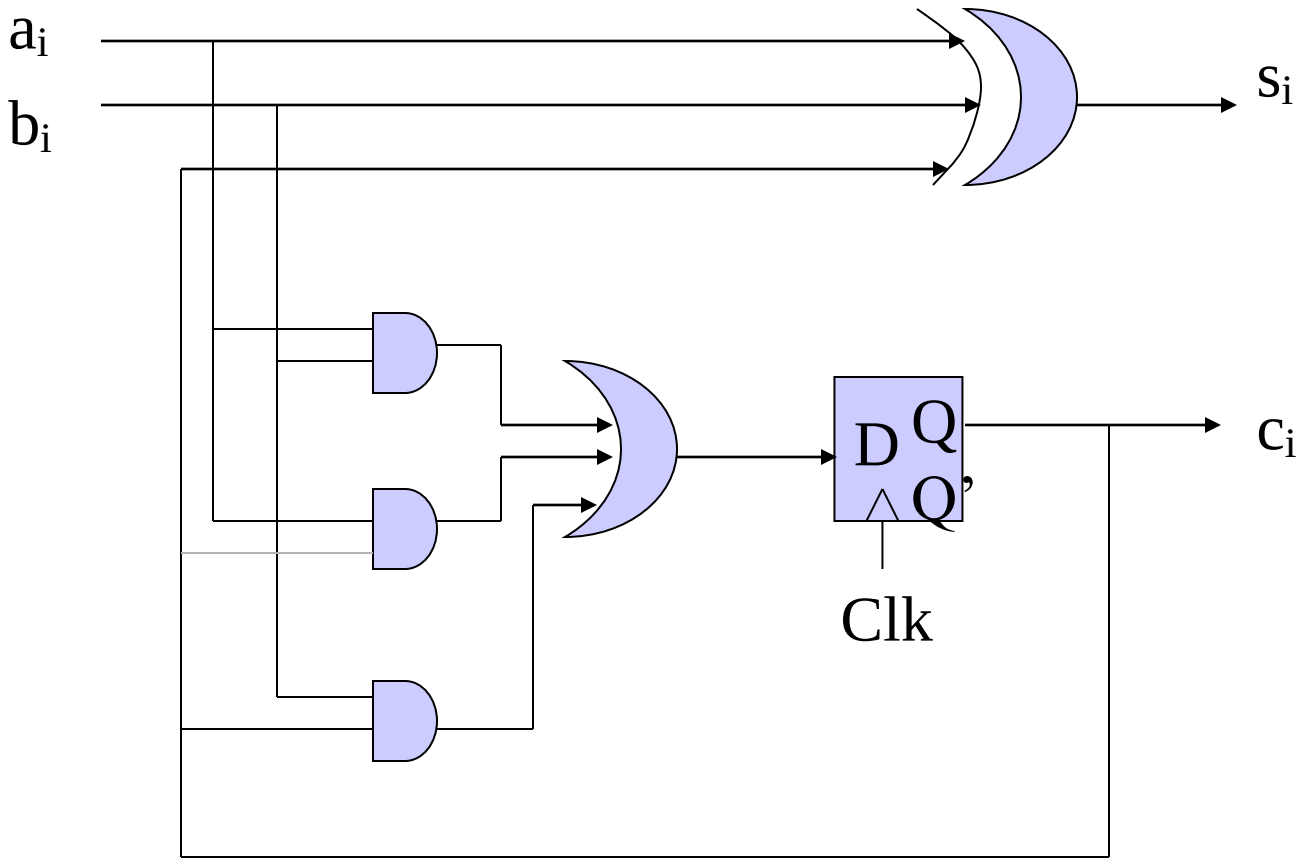
Serial Adder using a D Flip-Flop

id	a_i	b_i	c_i	c_{i+1}	s_i
0	0	0	0	0	0
1	0	0	1	0	1
2	0	1	0	0	1
3	0	1	1	1	0
4	1	0	0	0	1
5	1	0	1	1	0
6	1	1	0	1	0
7	1	1	1	1	1

$$Q(t) = c_i$$
$$Q(t+1) = D = c_{i+1}$$

Serial Adder using a D Flip-Flop

Logic Diagram



Multiplication using Serial Addition

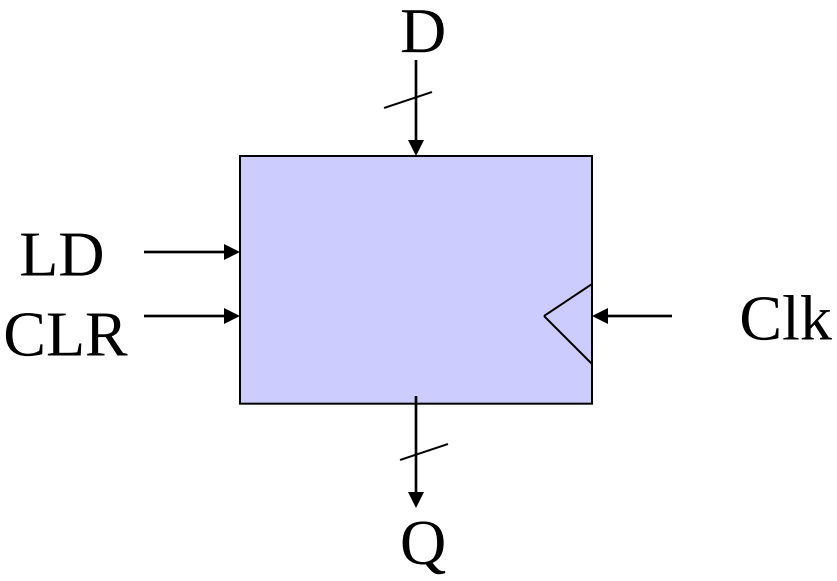
3 X 5 = 15

$$
\begin{array}{cccccc}
 & & & 0 & 1 & 1 \\
 & & \times & 1 & 0 & 1 \\
\hline
 & & & 0 & 1 & 1 \\
 & & 0 & 0 & 0 & \\
+ & 0 & 1 & 1 & & \\
\hline
 & 0 & 1 & 1 & 1 & 1
\end{array}
\qquad
\begin{array}{cccccc}
 & & & a_2 & a_1 & a_0 \\
 & & \times & b_2 & b_1 & b_0 \\
\hline
 & & & a_2b_0 & a_1b_0 & a_0b_0 \\
 & & a_2b_1 & a_1b_1 & a_0b_1 & \\
+ & a_2b_2 & a_1b_2 & a_0b_2 & & \\
\hline
m5 & m4 & m3 & m2 & m1 & m0
\end{array}
$$

For $m=AxB$, set $m^{(0)}=0$

At time i, perform $m^{(i+1)}=m^{(i)}+Ab_i2^i$

Register



Q (t+1) = (0, 0, .. , 0)	if CLR = 1
= D	if LD = 1 and CLR = 0
= Q (t)	if LD = 0 and CLR = 0

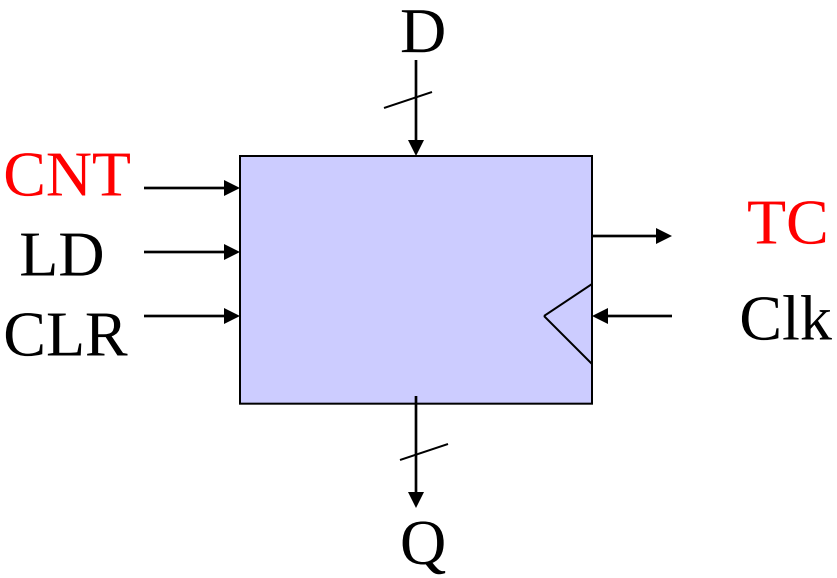
Counter

- Program Counter
- Address Keeper: FIFO, LIFO
- Clock Divider
- Sequential Machine

Counter

- Modulo-n Counter
- Modulo Counter (m<n)
- Counter (a-to-b)
- Counter of an Arbitrary Sequence
- Cascade Counter

Modulo-n Counter

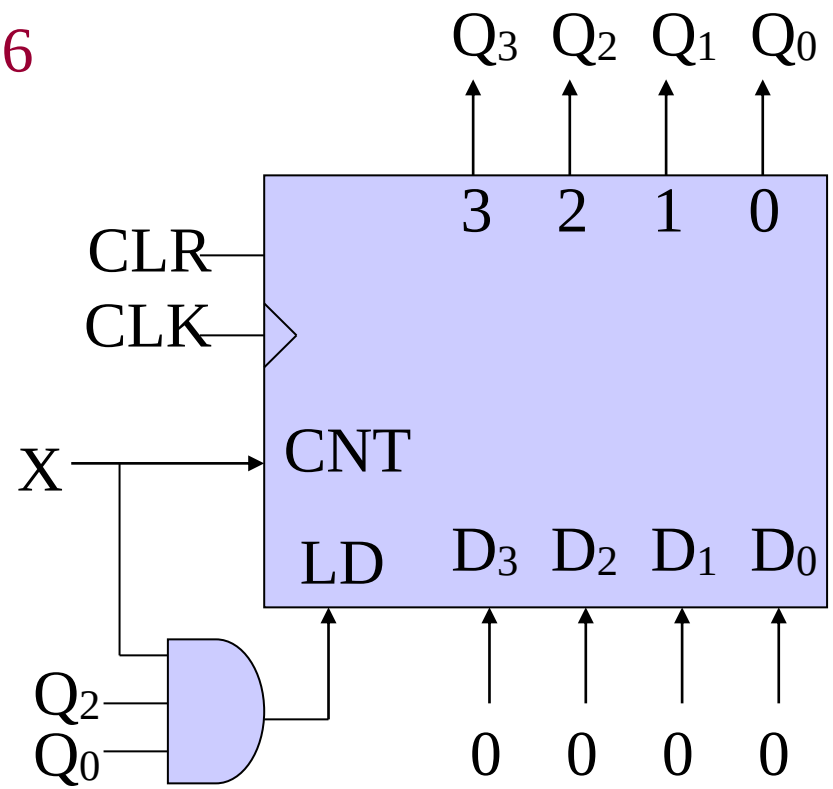


Q (t+1)	= (0, 0, .. , 0)	if CLR = 1
	= D	if LD = 1 and CLR = 0
	= (Q(t)+1)mod n	if LD = 0, CNT = 1 and CLR = 0
	= Q (t)	if LD = 0, CNT = 0 and CLR = 0
TC	= 1	if Q (t) = n-1 and CNT = 1
	= 0	otherwise

Modulo-m Counter (m< n)

Given a mod 16 counter, construct
a mod-m counter (0 < m < 16) with AND, OR, NOT gates

m = 6

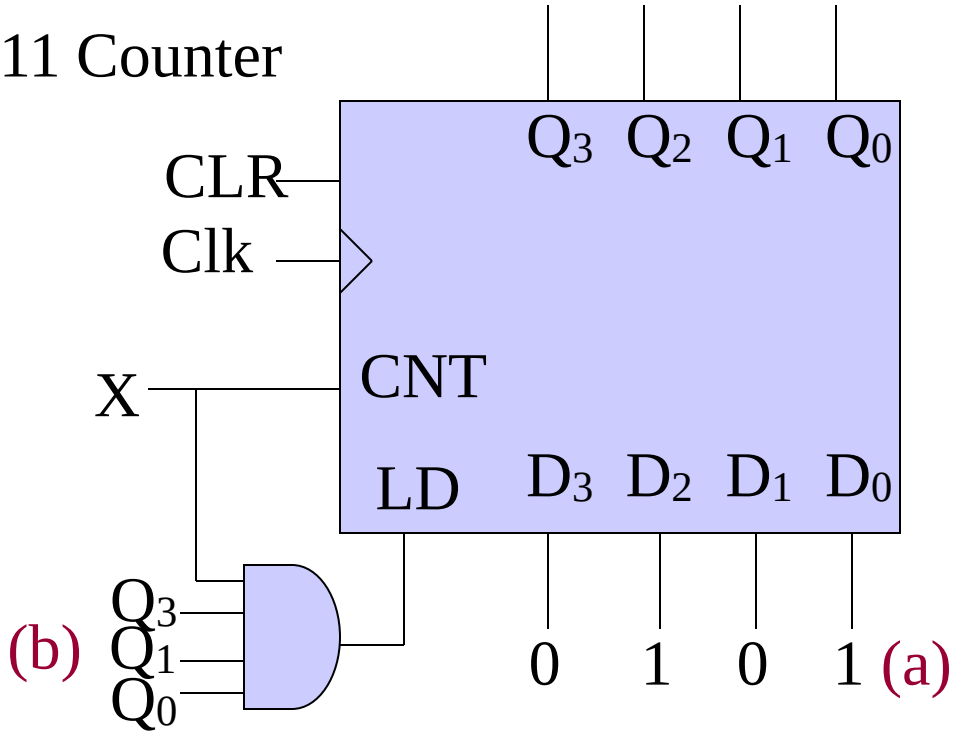


Set LD = 1 when X = 1 and ($Q_3Q_2Q_1Q_0$) = (0101), ie m-1

Counter (a-to-b)

Given a mod 16 counter, construct an a-to-b counter ($0 < a < b < 15$)

A 5-to-11 Counter

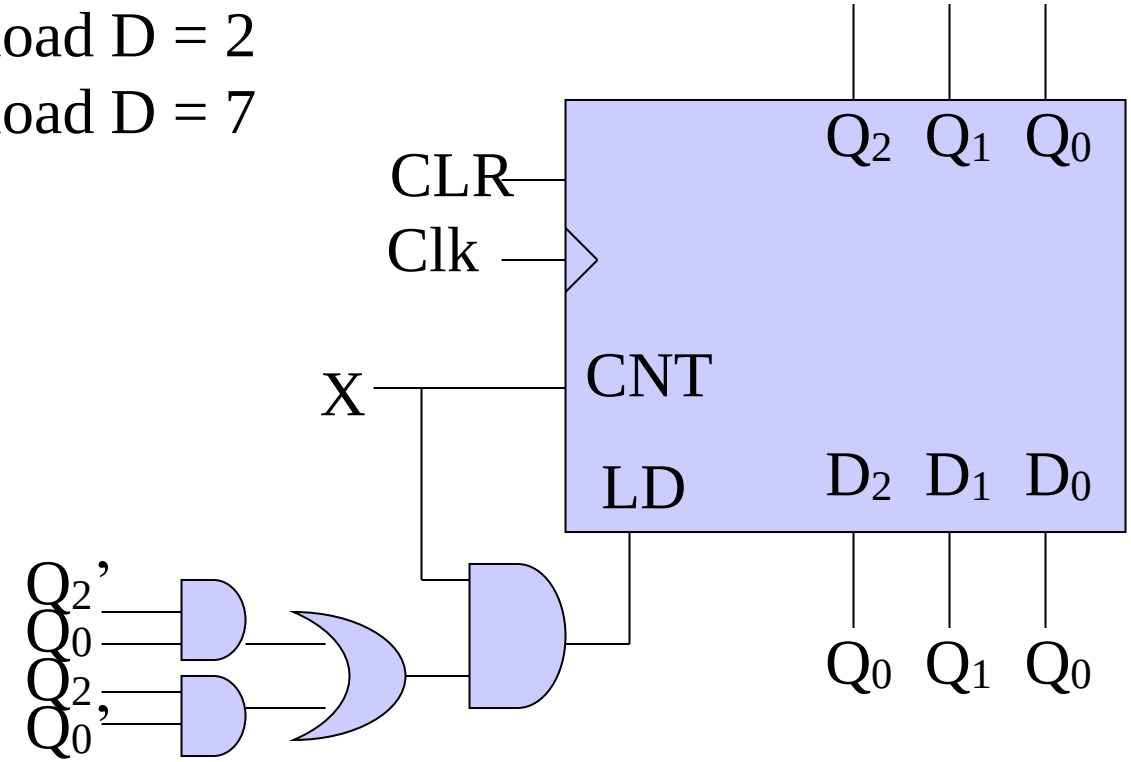


Set LD = 1 when X = 1 and $(Q_3Q_2Q_1Q_0)$ = b (in this case, 1011)

Counter of an Arbitrary Sequence

Given a mod 8 counter, construct
a counter with sequence 0 1 5 6 2 3 7

When Q = 1, load D = 5
When Q = 6, load D = 2
When Q = 3, load D = 7



Counter of an Arbitrary Sequence

Given a mod 8 counter, construct a counter with sequence 0 1 5 6 2 3 7

Id	$Q_2Q_1Q_0$	LD	D_2	D_1	D_0
0	000	0	-	-	-
1	001	1	1	0	1
2	010	0	-	-	-
3	011	1	1	1	1
4	100	-	-	-	-
5	101	0	-	-	-
6	110	1	0	1	0
7	111	0	-	-	-

K Mapping LD and D, we get

$$LD = Q'_2 Q_0 + Q_2 Q'_0$$

$$D_2 = Q_0$$
$$D_1 = Q_1$$
$$D_0 = Q_0$$

Counter of an Arbitrary Sequence

Example: Count in sequence 0 2 3 4 5 7 6

LD = 1 D = 2 When Q(t) = 0
LD = 1 D = 7 When Q(t) = 5
LD = 1 D = 6 When Q(t) = 7
LD = 1 D = 0 When Q(t) = 6

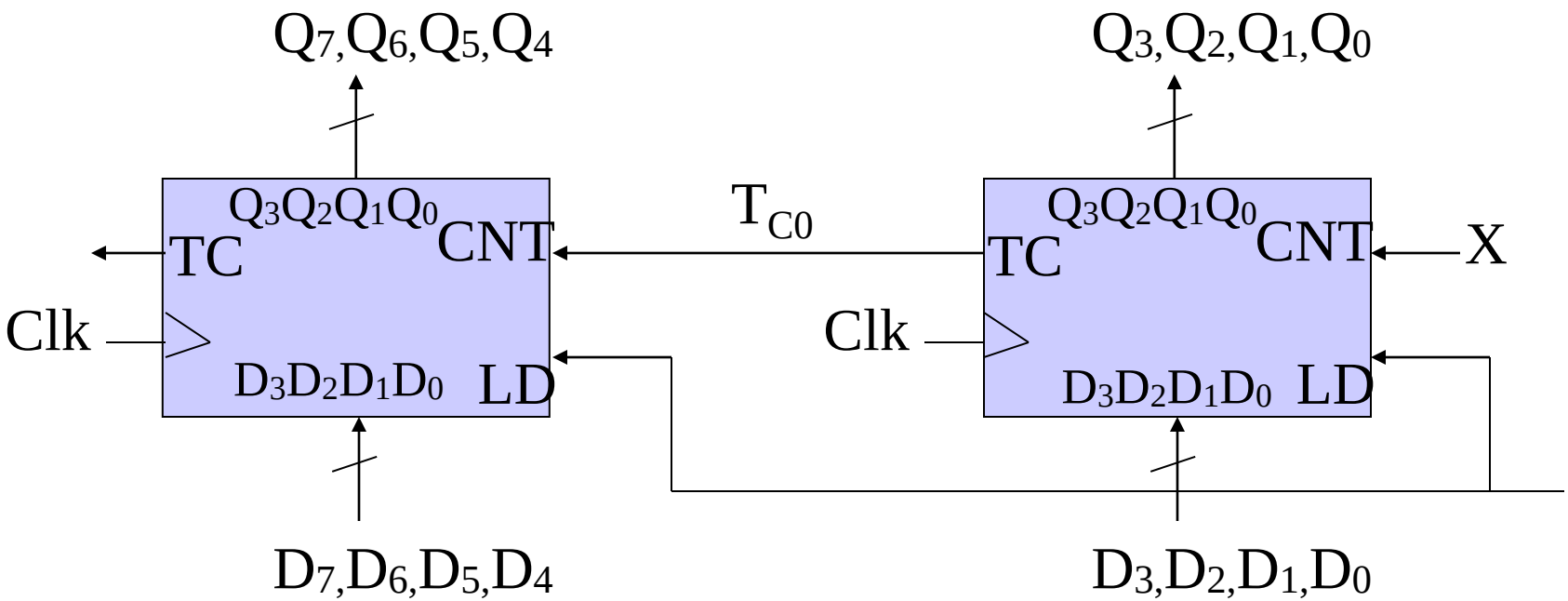
Through K-map, we derive

$LD = Q'_2 Q'_1 + Q_2 Q_0 + Q_2 Q_1$
$D_2 = Q_0$
$D_1 = Q'_1 + Q_0$
$D_0 = Q'_1 Q_0$

Id	$Q_2Q_1Q_0$	LD	D_2	D_1	D_0
0	000	1	0	1	0
1	001	-	-	-	-
2	010	0	-	-	-
3	011	0	-	-	-
4	100	0	-	-	-
5	101	1	1	1	1
6	110	1	0	0	0
7	111	1	1	1	0

Cascade Counter

A Cascade Modulo 256 Counter



Cascade Counter

TC = 1 when $(Q_3, Q_2, Q_1, Q_0) = (1,1,1,1)$ and X=1

$(Q_{7\,(t+1)}\ Q_{6\,(t+1)}\ Q_{5\,(t+1)}\ Q_{4\,(t+1)}) = (Q_{7\,(t)}\ Q_{6\,(t)}\ Q_{5\,(t)}\ Q_{4\,(t)}) + 1 \bmod 16$

when $T_{C0} = 1$

The circuit functions as a modulo 256 counter.

Time		0	1	2	3	…	13	14	15	16	17	18	19
$Q_{7\text{-}4}$		0	0	0	0	…	0	0	0	1	1	1	1
T_{C0}		0	0	0	0	…	0	0	1	0	0	0	0
$Q_{3\text{-}0}$		0	1	2	3	…	13	14	15	0	1	2	3

Conclusion

- Tradeoff of silicon area and system performance
  - Spatial vs. Temporal Dimension Expansion
  - Combinational vs. Sequential Implementation
  - Wires vs. Embedded States for Communication
- Metrics of Cost, Speed, and Power
- Ad: Cheaper hardware, Fit for FPGA architecture, Pipelining for excellent throughput
- Dis: Longer latency