# Lecture 8:

## Sequential Digital Systems and Synchronous Counters

# In this lecture we will:

Introduce the main ideas of a Synchronous Digital System (SDS).

Examine how simple flip-flops may be used to define a generic and completely general SDS.

Examine binary counters in detail and how to design them.

Show how to design a controlled counter which includes "don't care" states.

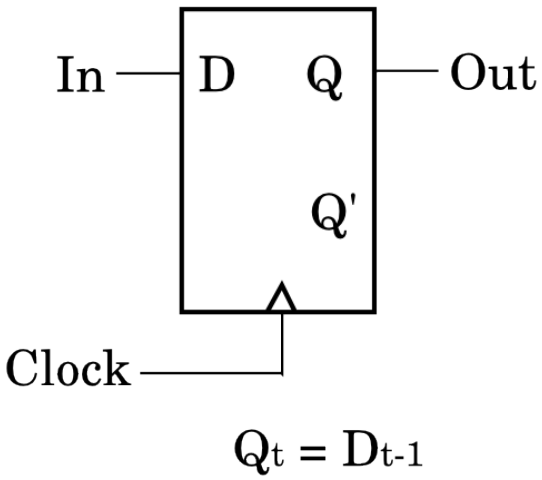
# The D type Flip-Flop

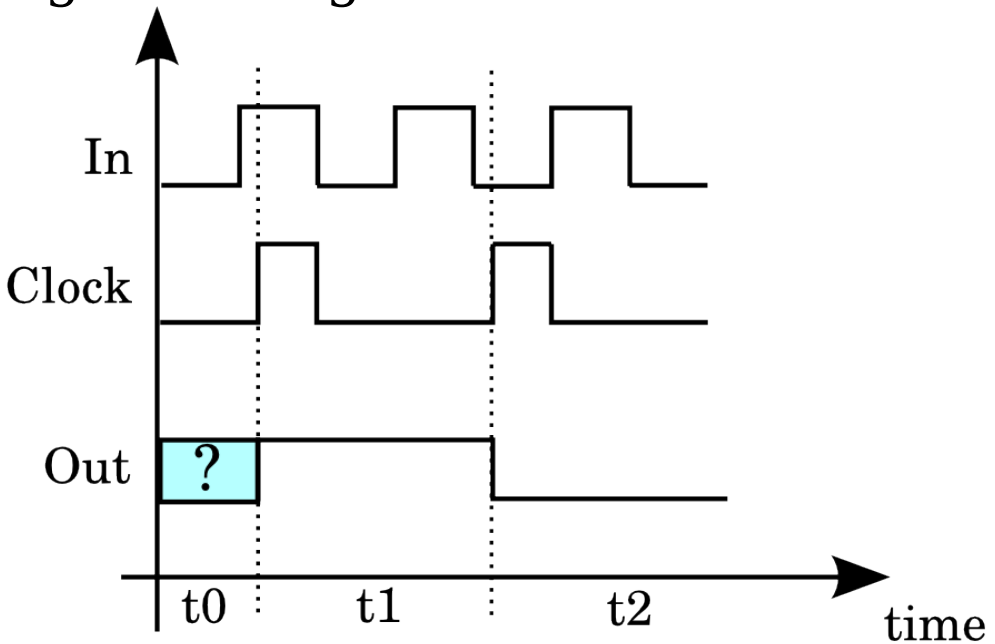
Last lecture we saw that a flip flop is a one-bit memory

It can be "read" at any time

It is "written" when the clock input changes value

The edge may be either rising or falling



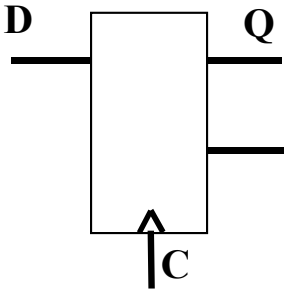



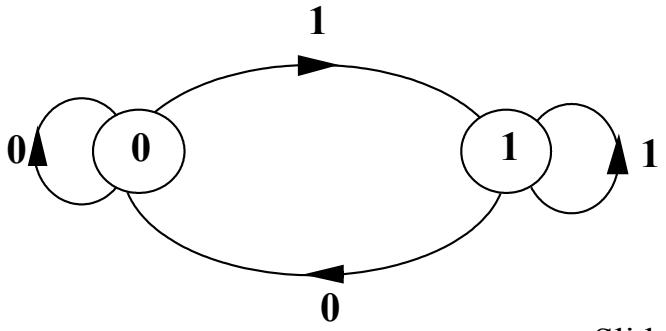
# The D-Q flip flop

The D-Q flip flop is a (very) simple example of a synchronous digital system.

It can be in one of two possible states, and its change of state is controlled by a single clock signal.

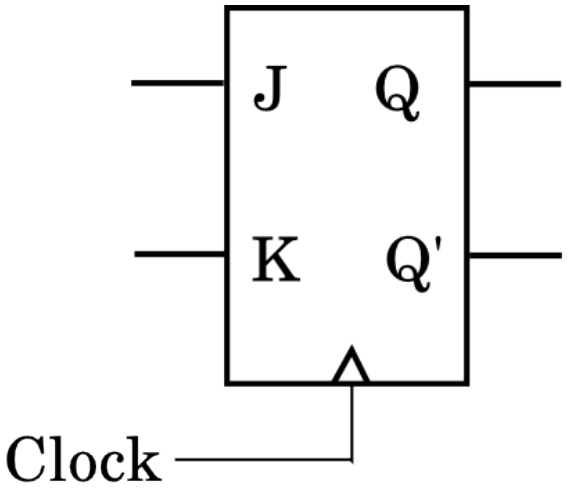
It can be represented by a finite state machine diagram.



# The J-K flip-flop

Digital designers also use the so called J-K flip-flop which has two data inputs (and of course, a clock input)

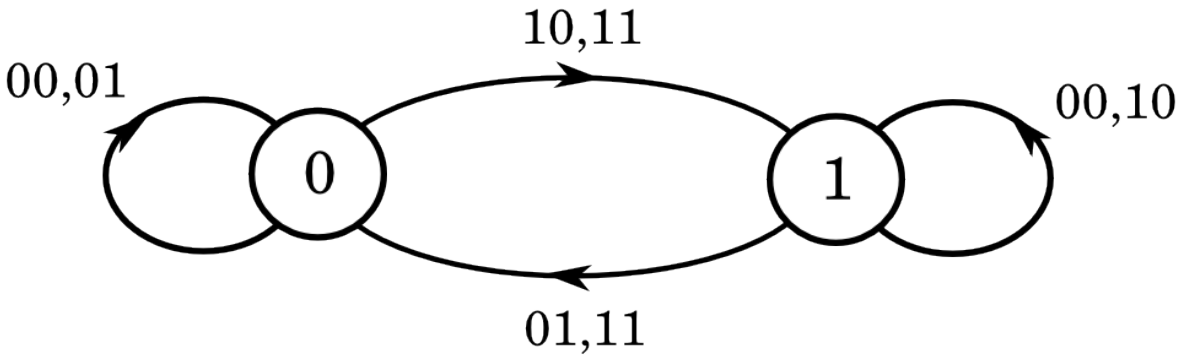



| J K | Function | Next Q |
|---|---|---|
| 0 0 | Unchanged | Q |
| 0 1 | Reset | Zero |
| 1 0 | Set | One |
| 1 1 | Toggle | Q’ |

This transition table defines the operation of the flip-flop.

# The Transition Diagram of the J-K Flip Flop

| J K | Function | Next Q |
|---|---|---|
| 0 0 | Unchanged | Q(now) |
| 0 1 | Reset | Zero |
| 1 0 | Set | One |
| 1 1 | Toggle | Q'(now) |




Even though the flip-flop has two inputs, it has only two states (not four).

It has two inputs and four possible transitions

# Sequential Digital Circuits

A flip-flop is an example of a sequential circuit. It goes through a sequence of states controlled by its inputs.

Another example is the counter in a digital watch. It will go through a sequence of states: display0 -> display1 -> display2 etc. States will change at exactly 1 second intervals

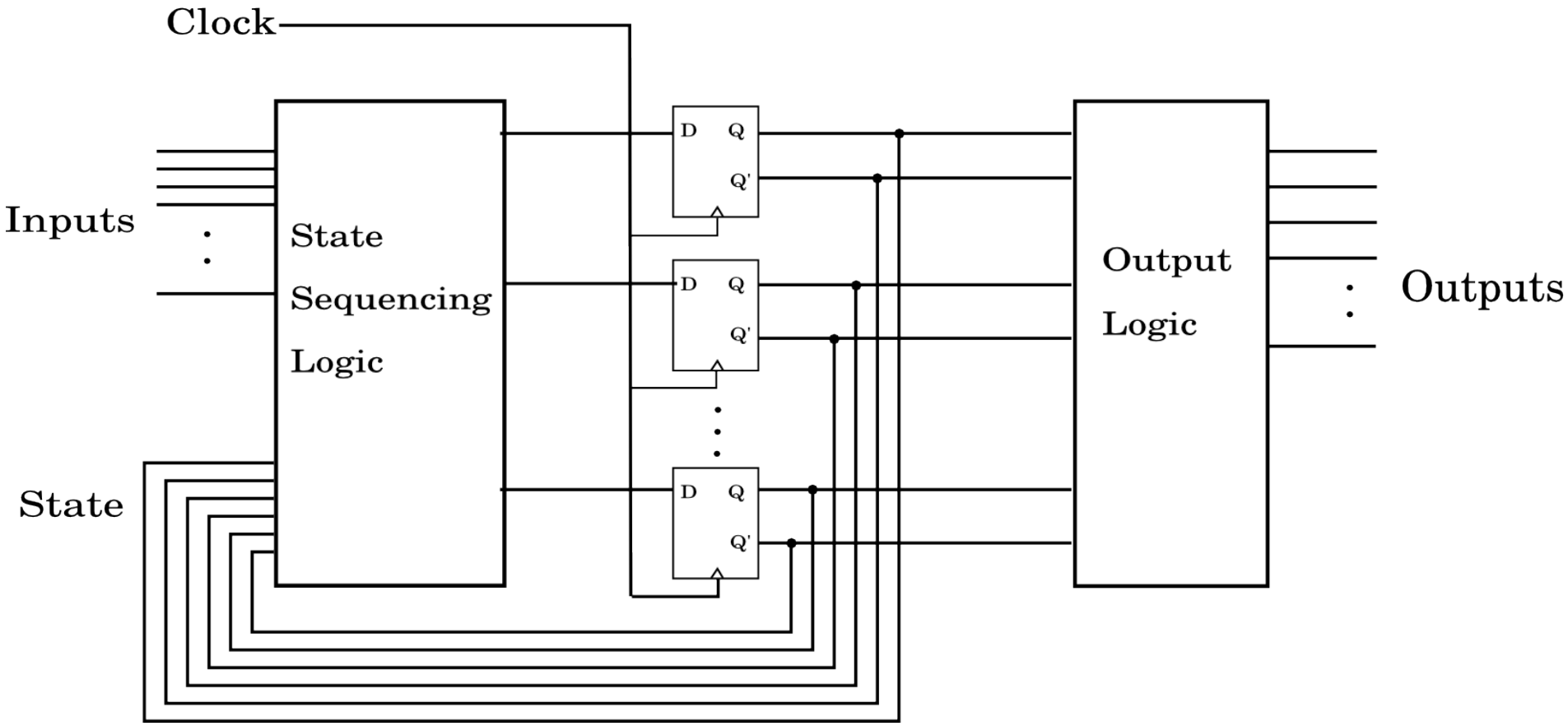
# Synchronous Circuits

Synchronous circuits are those where state changes occur at exact times controlled by a clock.

Synchronous circuits are by definition sequential. Not all sequential circuits are synchronous.

With the simple D type flip-flop and our knowledge of combinational digital circuits, we can construct a general model with which any synchronous circuit can be implemented!

# Synchronous Digital Circuits




Both the state sequencing (input) and output control logic are combinational circuits.

# Synchronous Digital Circuits

The number of outputs of the circuit depends on the problem.

The outputs depend only on the State bits Qi.

Q[k](t+1) = D[k](t) D-type Flip-Flop law

D[k] = F( I[1], I[2], .., Q[1], Q[2], ...) Input logic

Out[k] = G(Q[1], Q[2], ...) Output logic

# Synchronous Binary Counters

We will look at binary counters as an example of synchronous sequential circuit design.

Simple binary counters have just a clock input.

Their output is a repeating sequence of binary numbers. The state flip-flop outputs can be used directly as outputs (there is no output logic).

# Synchronous Binary Counters

For a two-bit counter, there are four states, 0 (00), 1 (01), 2 (10), and 3 (11)

There are 6 complete counting sequences:

0->1->2->3->0 etc

0->1->3->2->0 etc

0->2->3->1->0 etc

0->2->1->3->0 etc

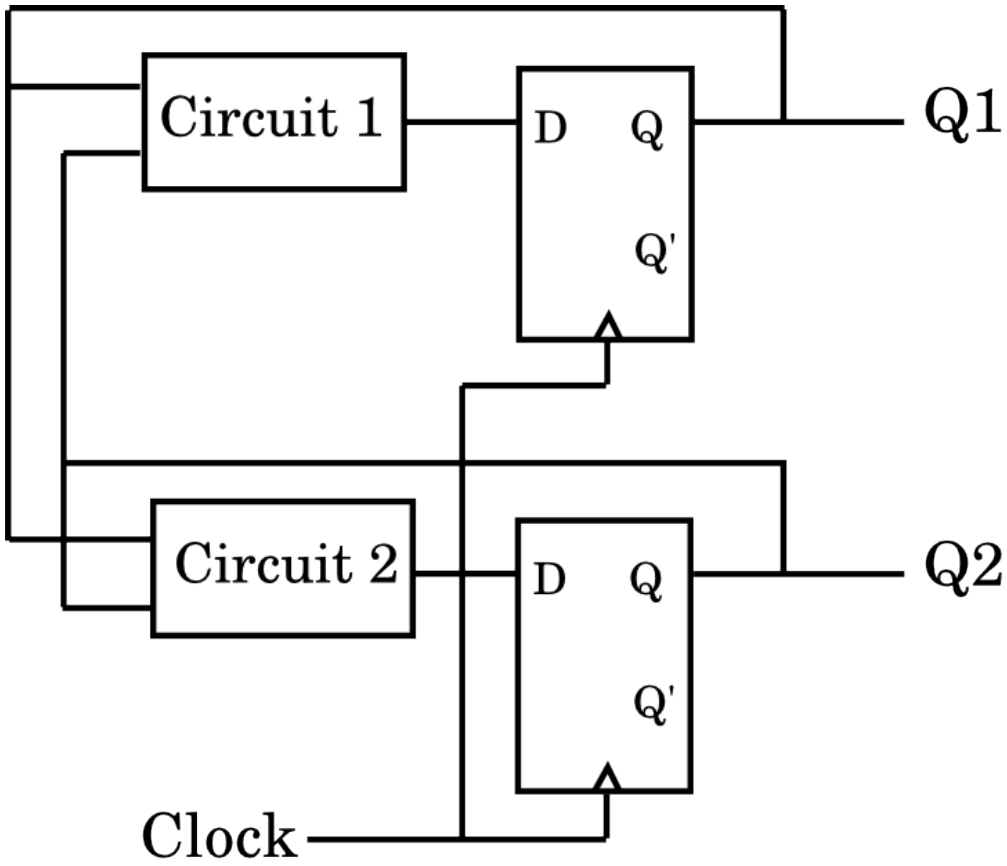
0->3->1->2->0 etc

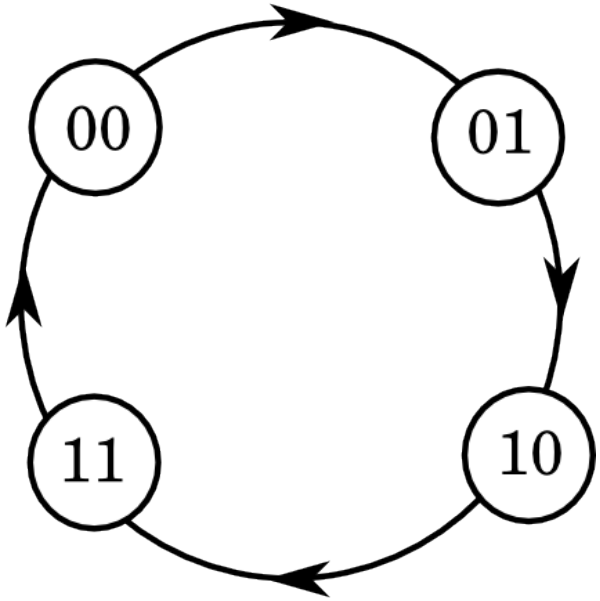
0->3->2->1->0 etc

And more if not all the states are used

# A Two-Bit Binary Up Counter

The sequence is: 0 -> 1 -> 2 -> 3 -> 0



# Two-Bit Binary Up Counter - Design

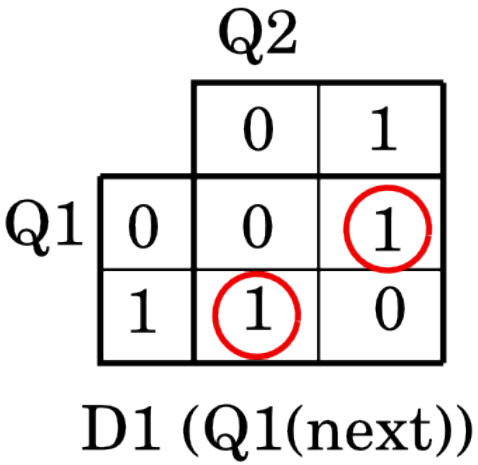
The first step is to construct the truth table of a synchronous circuit - its Transition Table.

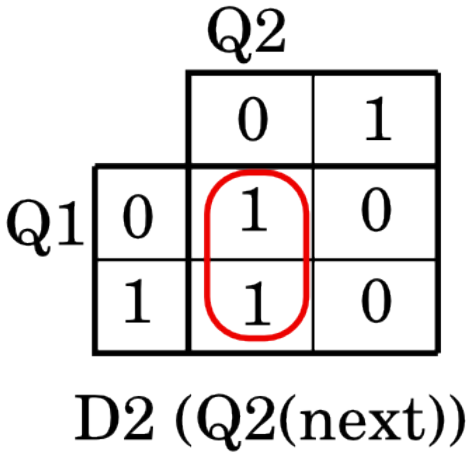
The transition table shows the state output values after the clock pulse (next) as a function of the input and state output values before the clock pulse (now).

Since for a D type flip-flop the output (Q) after the clock pulse is equal to the input (D) before the clock pulse, the transition table becomes a simple input/output truth table.

# Two-Bit Counter Truth Table and Karnaugh Map

| (now) Q1 | Q2 | (next) Q1 | Q2 |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |

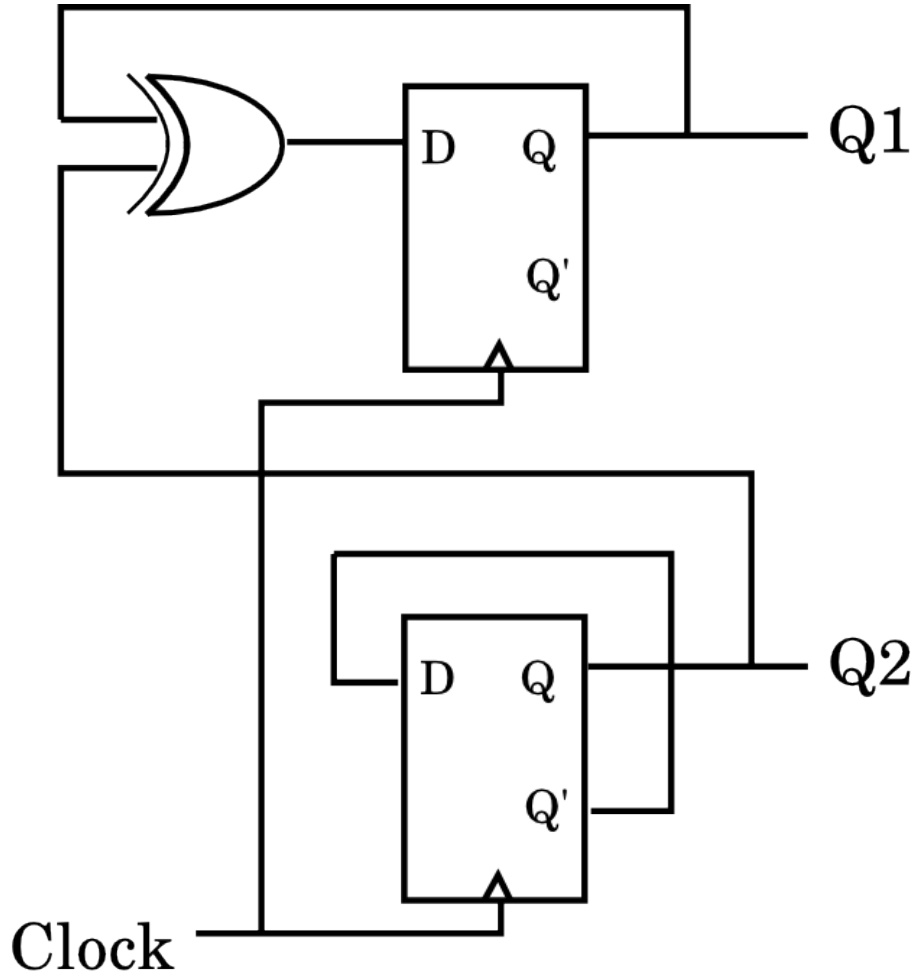


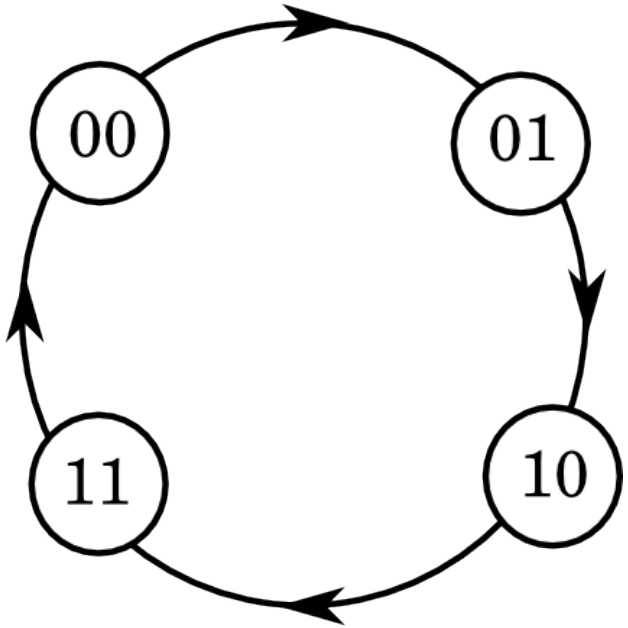




$$D1 = Q1' \cdot Q2 + Q1 \cdot Q2' = Q1 \oplus Q2$$

$$D2 = Q2'$$

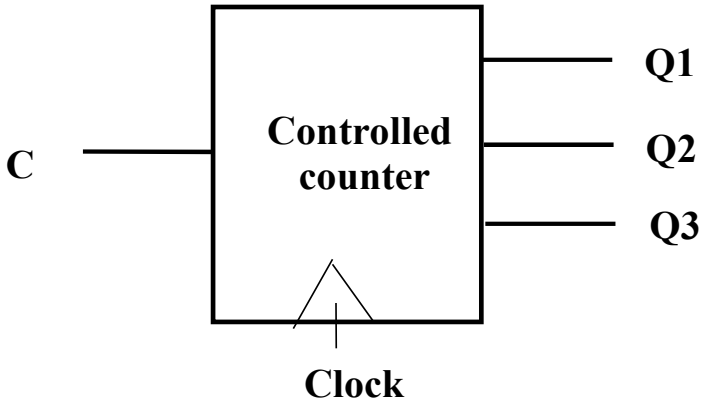
# Two-Bit Counter- The Final Circuit



# Design of a controlled 3-bit counter with don't care states

We are given the following description of a synchronous sequential 3-bit binary counter:




When input C=0 the counter counts up even numbers: 000 -> 010 -> 100 -> 110 -> 000 -> etc

When input C=1 the counter counts down odd numbers: 000 -> 111 -> 101-> 011-> 001-> 000

# Problem Time: What does the state transition diagram look like?

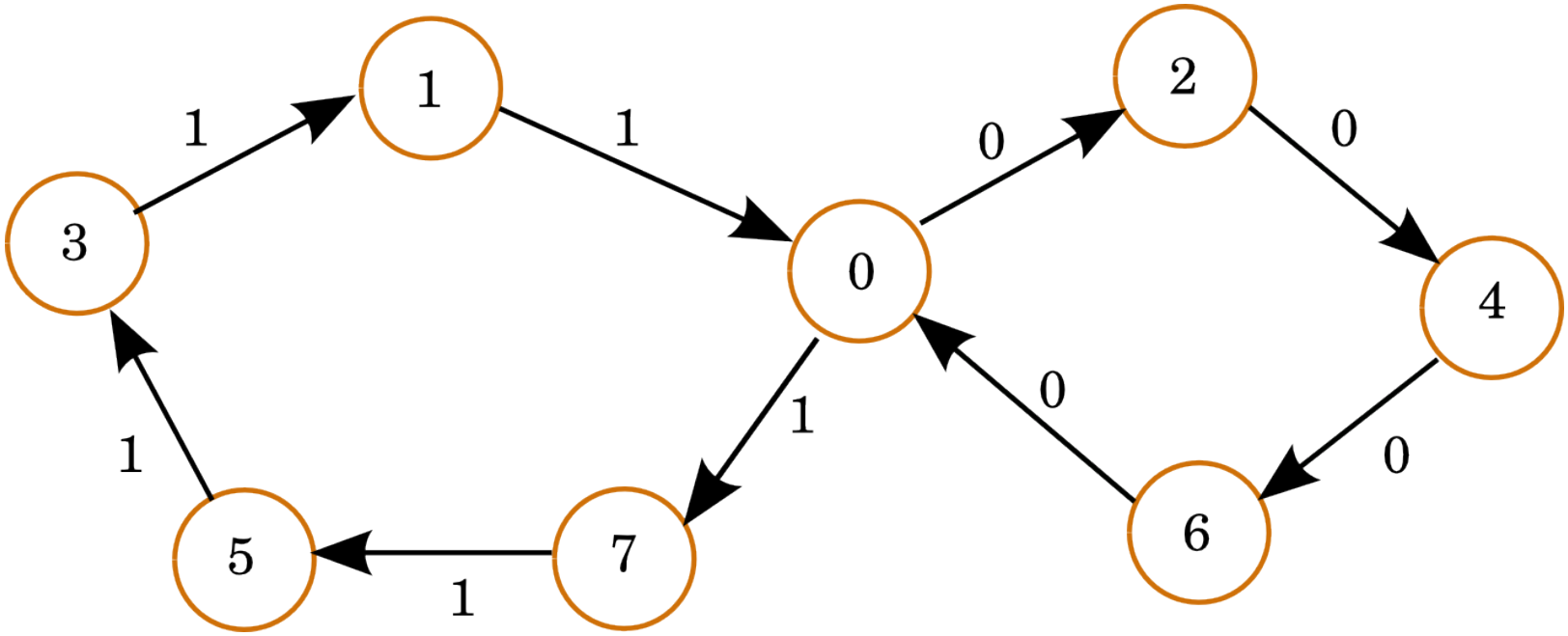
When input C=0 the counter counts up even numbers:
000 -> 010 -> 100 -> 110 -> 000 ->

When input C=1 the counter counts down odd numbers:
000 -> 111 -> 101-> 011-> 001-> 000

# Problem Time: What does the state transition diagram look like?

When input C=0 the counter counts up even numbers:
000 -> 010 -> 100 -> 110 -> 000 ->

When input C=1 the counter counts down odd numbers:
000 -> 111 -> 101-> 011-> 001-> 000



# Controlled 3-bit counter

The specification shows that not all states are included in the counting sequences.

However, these "don't care" states must included in the design.

We must check to see that the circuit is safe when we switch it on.

# Controlled 3-bit counter

Possible ways of dealing with unused states:

1. If the counter finds itself in one of the unused states, it should return to a known state after one clock pulse.

2. The counter can return to any state.

# Controlled 3-bit counter - Design

Step 1

The transition table is produced.

The don't care outputs X indicate a state which is not part of the counting sequence:

| C | Q1 | Q2 | Q3 | D1 | D2 | D3 |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 | X | X | X |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | X | X | X |
| 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | X | X | X |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | X | X | X |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | X | X | X |
| 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 | X | X | X |
| 1 | 1 | 0 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | X | X | X |
| 1 | 1 | 1 | 1 | 1 | 0 | 1 |

# Controlled 3-bit counter - Design

Step 2: The Karnaugh map minimisation.

D1

| C, Q1 \ Q2, Q3 | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 | X | X | 1 |
| 01 | 1 | X | X | 0 |
| 11 | X | 0 | 1 | X |
| 10 | 1 | 0 | 0 | X |

D2

| C, Q1 \ Q2, Q3 | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1 | X | X | 0 |
| 01 | 1 | X | X | 0 |
| 11 | X | 1 | 0 | X |
| 10 | 1 | 0 | 0 | X |

$D1 = C'Q_1'Q_2+C'Q_1Q_2'+CQ_3'+CQ_1Q_2$

$D2 = Q_2'Q_3'+Q_1Q_2'$

D3

| C, Q1 \ Q2, Q3 | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 | X | X | 0 |
| 01 | 0 | X | X | 0 |
| 11 | X | 1 | 1 | X |
| 10 | 1 | 0 | 1 | X |

$D3 = CQ_2+CQ_3'+CQ_1$

# Design Strategy

If the counter can return to any state from an unknown state, then:

Retain the don’t care states

Check to see that the circuit is safe after design.

# Controlled 3-bit counter - Design

## Step 3: The Realised Transition Table (1s and 0s)

D1

| C, Q1 \ Q2, Q3 | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 | 0 | 1 | 1 |
| 01 | 1 | 1 | 0 | 0 |
| 11 | 1 | 0 | 1 | 1 |
| 10 | 1 | 0 | 0 | 1 |

D2

| C, Q1 \ Q2, Q3 | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1 | 0 | 0 | 0 |
| 01 | 1 | 1 | 0 | 0 |
| 11 | 1 | 1 | 0 | 0 |
| 10 | 1 | 0 | 0 | 0 |

$D1 = C'Q_1'Q_2+C'Q_1Q_2'+CQ_3'+CQ_1Q_2$

$D2 = Q_2'Q_3'+Q_1Q_2'$

D3

| C, Q1 \ Q2, Q3 | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 | 0 | 0 | 0 |
| 01 | 0 | 0 | 0 | 0 |
| 11 | 1 | 1 | 1 | 1 |
| 10 | 1 | 0 | 1 | 1 |

$D3 = CQ_2+CQ_3'+CQ_1$

# Controlled 3-bit counter - Design

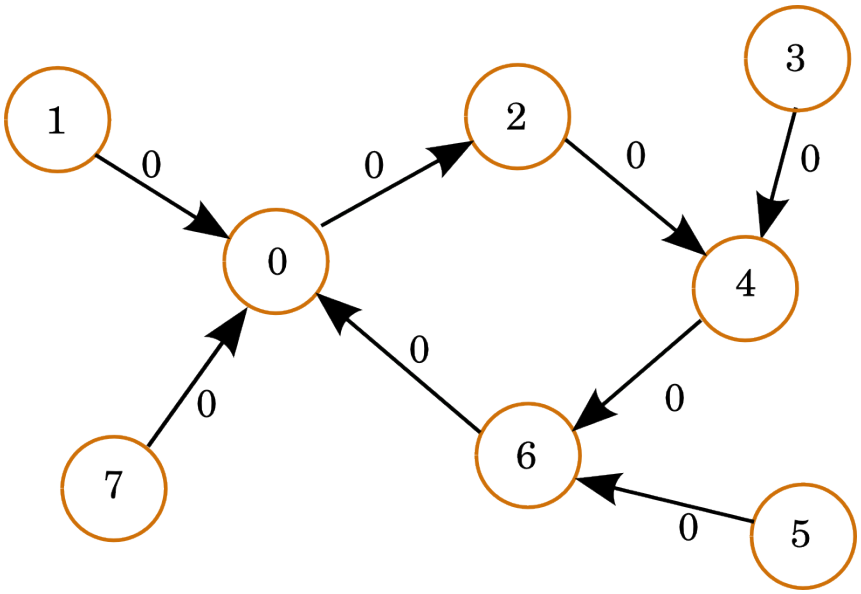
Step 4: Produce the correct transition table (without don't cares) & the transition diagram.

Indicate the state transitions by State Numbers (0 to 7).

Specifications satisfied because for either control input case the counter will eventually reach State 0.

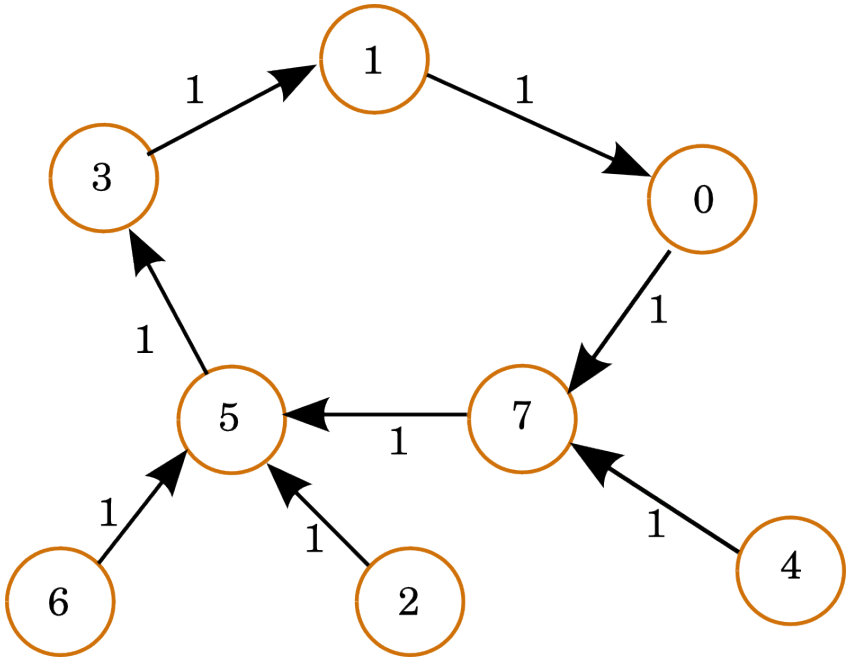
| C | Q1 | Q2 | Q3 | D1 | D2 | D3 | $S_{(tn)}$ | $S_{(tn+1)}$ |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 2 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 2 | 4 |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 3 | 4 |
| 0 | 1 | 0 | 0 | 1 | 1 | 0 | 4 | 6 |
| 0 | 1 | 0 | 1 | 1 | 1 | 0 | 5 | 6 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 6 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | 7 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 7 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 | 1 | 2 | 5 |
| 1 | 0 | 1 | 1 | 0 | 0 | 1 | 3 | 1 |
| 1 | 1 | 0 | 0 | 1 | 1 | 1 | 4 | 7 |
| 1 | 1 | 0 | 1 | 0 | 1 | 1 | 5 | 3 |
| 1 | 1 | 1 | 0 | 1 | 0 | 1 | 6 | 5 |
| 1 | 1 | 1 | 1 | 1 | 0 | 1 | 7 | 5 |

# 3-bit counter Transition Table for C=0




| C | Q1 | Q2 | Q3 | D1 | D2 | D3 | $S_{(tn)}$ | $S_{(tn+1)}$ |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 2 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 2 | 4 |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 3 | 4 |
| 0 | 1 | 0 | 0 | 1 | 1 | 0 | 4 | 6 |
| 0 | 1 | 0 | 1 | 1 | 1 | 0 | 5 | 6 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 6 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | 7 | 0 |

# 3-bit counter Transition Table for C=1




| C | Q1 | Q2 | Q3 | D1 | D2 | D3 | $S_{(tn)}$ | $S_{(tn+1)}$ |
|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 7 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 | 1 | 2 | 5 |
| 1 | 0 | 1 | 1 | 0 | 0 | 1 | 3 | 1 |
| 1 | 1 | 0 | 0 | 1 | 1 | 1 | 4 | 7 |
| 1 | 1 | 0 | 1 | 0 | 1 | 1 | 5 | 3 |
| 1 | 1 | 1 | 0 | 1 | 0 | 1 | 6 | 5 |
| 1 | 1 | 1 | 1 | 1 | 0 | 1 | 7 | 5 |

# 3-bit counter Transition Table

Step 5. Build the circuit. Here we will assume that any basic gate (AND, OR, NAND, NOR, XOR, XNOR, Inverter) can be used. From the K-maps we have:

$$D1 = C'\bullet Q1'\bullet Q2 + C'\bullet Q1\bullet Q2' + C\bullet Q1\bullet Q2 + C\bullet Q3'$$
$$= C'\bullet(Q1 \oplus Q2) + C\bullet(Q1\bullet Q2 + Q3')$$

$$D2 = Q2'\bullet Q3' + Q1\bullet Q2'$$
$$= Q2'\bullet(Q1 + Q3')$$

$$D3 = C\bullet Q1 + C\bullet Q3' + C\bullet Q2$$
$$= C\bullet(Q1 + Q2 + Q3')$$
$$= C\bullet( (Q1 + Q3') + Q2)$$

**Common terms are bracketed as they can be shared between expressions**

# Controlled 3-bit counter - circuit