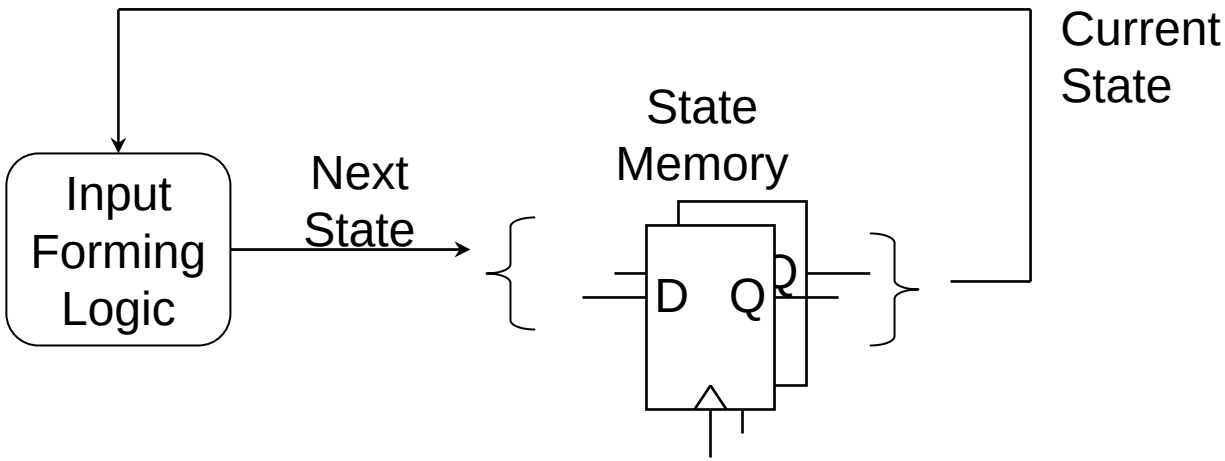
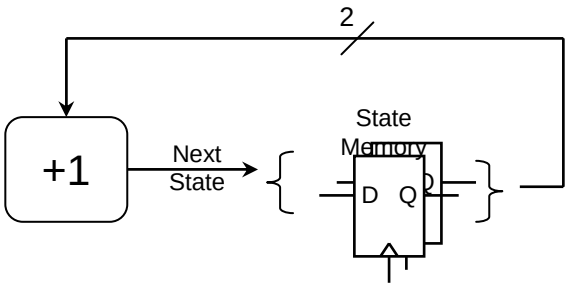
COUNTERS

Counters
Transition Tables
Moore Outputs
Counter Timing

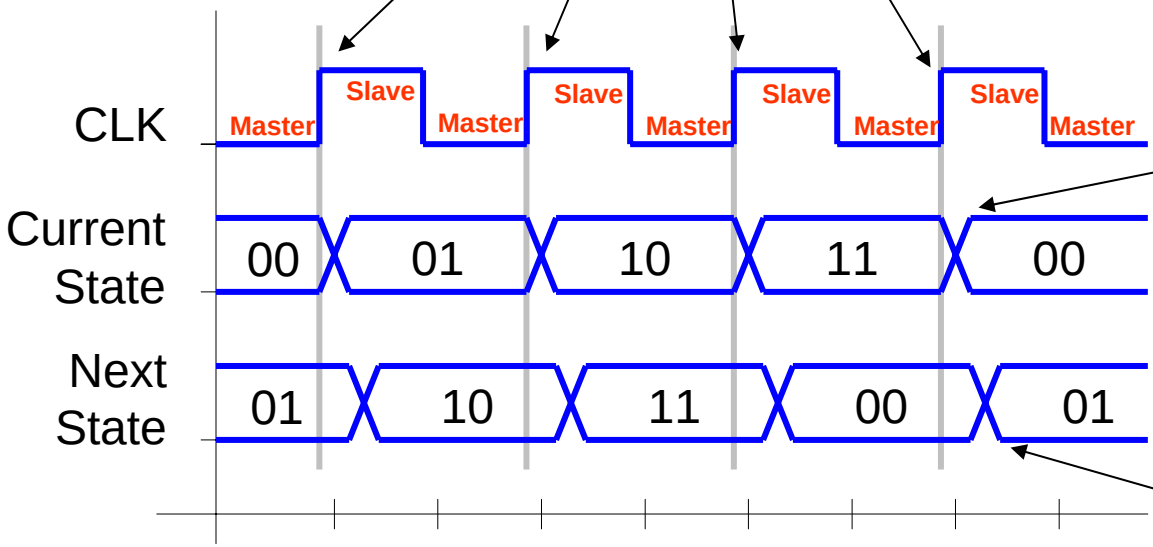
General Sequential Systems



A Sequential Counter



Clock edges…



The current state loads the next state values in response to the clock edge.

IFL reacts after some gate delays to produce a new next state.

Transition Table for 2-Bit Counter

Current State	Next State
00	01
01	10
10	11
11	00

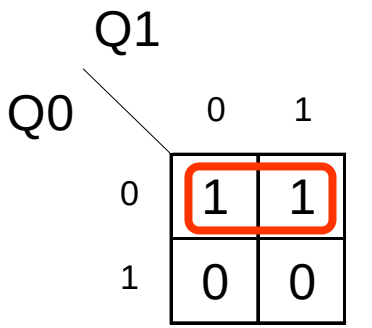
Current State		Next State	
Q1	Q0	N1	N0
0	0	0	1
0	1	1	0
1	0	1	1
1	1	0	0

It is the truth table for the input forming logic…

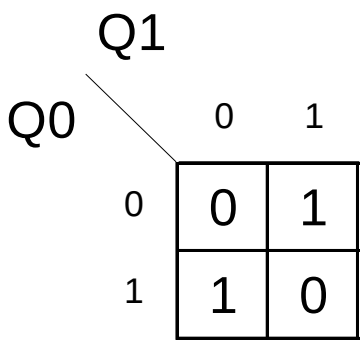
It describes what the *next state* values are as a function of the *current state* (clock is assumed)

Implementation of 2-Bit Counter

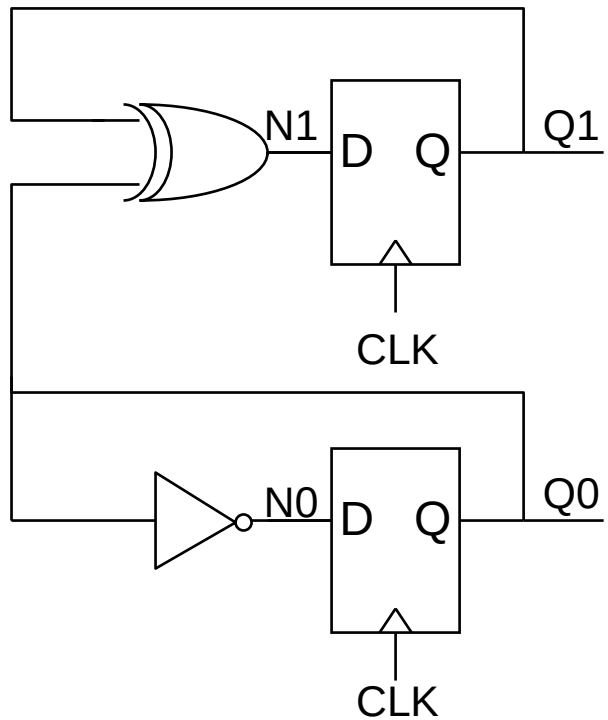
Current State		Next State	
Q1	Q0	N1	N0
0	0	0	1
0	1	1	0
1	0	1	1
1	1	0	0



N0 =Q0'



N1 =Q1⊕Q0

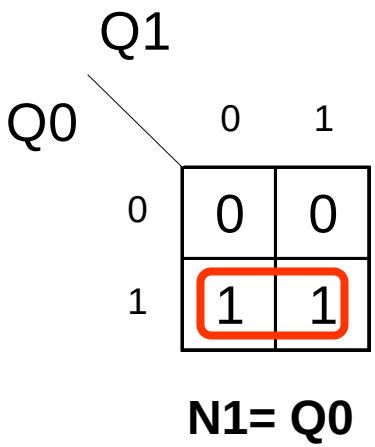


Example 2 – A Gray Code Counter

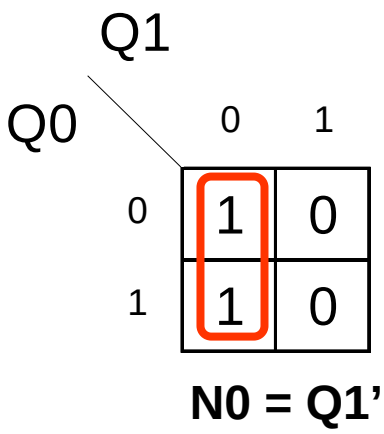
Q1	Q0	N1	N0
0	0	0	1
0	1	1	1
1	1	1	0
1	0	0	0

Ordering of TT rows chosen to make it easier to see the count pattern.

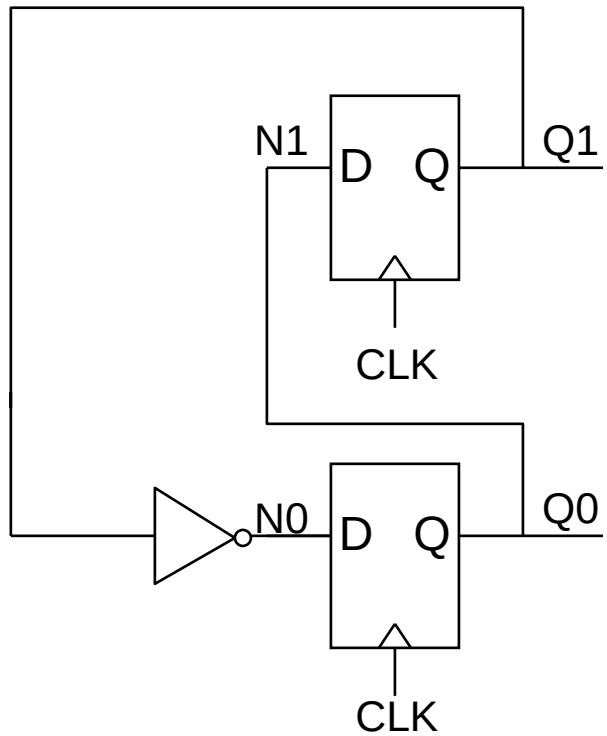
It doesn't change result.



N1= Q0



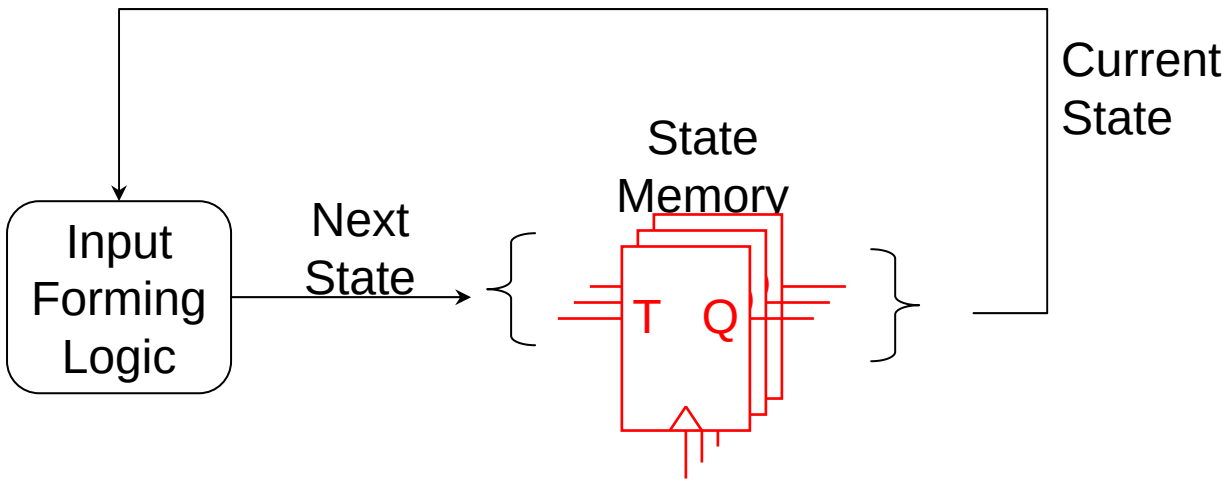
N0 = Q1'



General Counter Design Procedure

- Write transition table for counter
  - Use X's as appropriate
- Reduce each Nx variable to an equation
- Implement input forming logic (IFL) using gates
- Draw schematic using FF's + IFL

Counters With Alternative FF's

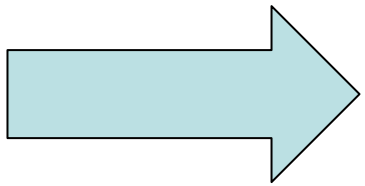


Excitation Table for T Flip Flop

Transition Table

T	Q	Q+
0	0	0
0	1	1
1	0	1
1	1	0

Rearranging rows and/or columns…



Excitation Table

Q	Q+	T
0	0	0
0	1	1
1	0	1
1	1	0

Tells how inputs affect output

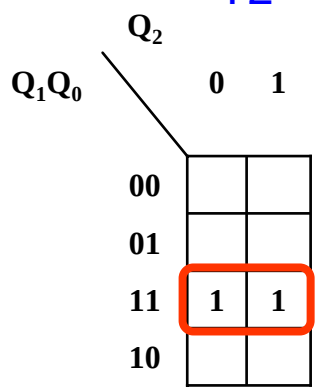
Tells what input to apply to achieve desired output transition

In the end, they are the same table, just rearranged…

TFF Counter Design Using Augmented Transition Table

Current State			Next State			TFF Inputs		
Q2	Q1	Q0	N2	N1	N0	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

T2

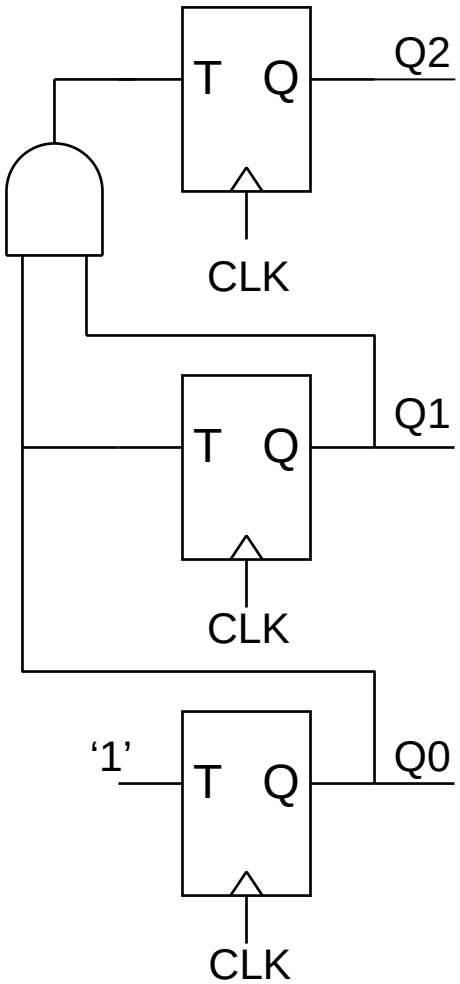


T2 = Q1•Q0
T1 = Q0
T0 = '1'

Next state values

Inputs to apply to achieve desired next state

TFF Counter Design

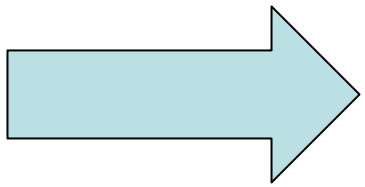


Excitation Table for JK Flip Flop

J	K	Q	Q+
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Tells how inputs affect output

Rearranging rows and/or columns…



Q	Q+	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Tells what inputs to apply to achieve desired output transition

Do you see why the X's?

In the end, they are the same table, just rearranged…

JKFF Gray Code Counter Design

Q2	Q1	Q0	N2	N1	N0	J2	K2	J1	K1	J0	K0
0	0	0	0	0	1	0	X	0	X	1	X
0	0	1	0	1	1	0	X	1	X	X	0
0	1	0	1	1	0	1	X	X	0	0	X
0	1	1	0	1	0	0	X	X	0	X	1
1	0	0	0	0	0	X	1	0	X	0	X
1	0	1	1	0	0	X	0	0	X	X	1
1	1	0	1	1	1	X	0	X	0	1	X
1	1	1	1	0	1	X	0	X	1	X	0

J2 = Q1Q0'
K2 = Q1'Q0'

J1 = Q2'Q0
K1 = Q2Q0

J0 = Q2Q1 + Q2'Q1' = K0'
K0 = Q2'Q1 + Q2Q1' = Q2 © Q1

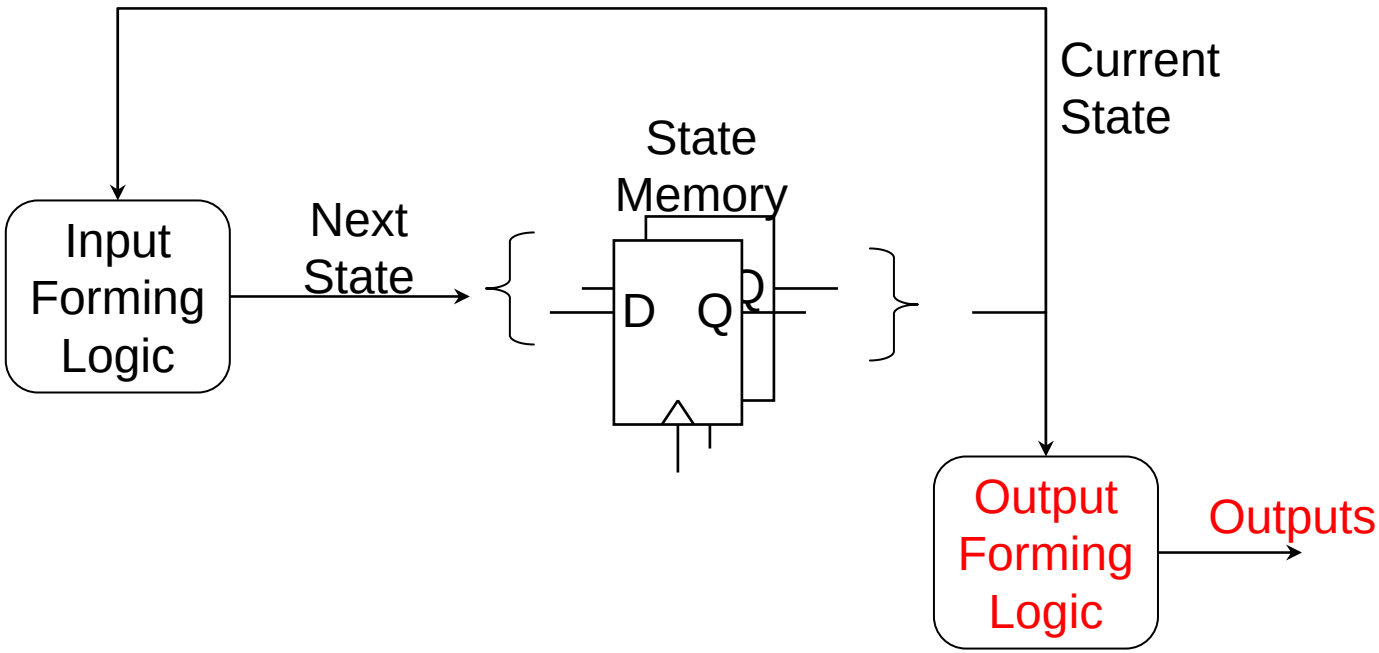
Next state values

Inputs to apply to achieve desired next state

General Counter Design Procedure

- Write transition table for counter
  - Use X's as appropriate
  - If not DFF's, augment table
- Reduce each FF input variable to equation
- Implement IFL with gates
- Draw schematic using FF's + gates

Counters With Outputs



Outputs = f(CurrentState)

Counters With Outputs

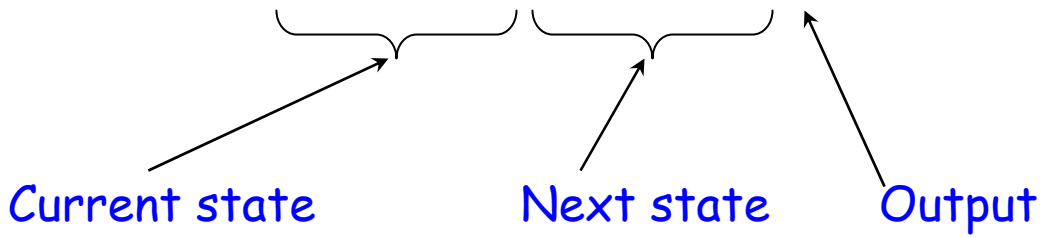
Z=1 when count={0,3,6}

Q2	Q1	Q0	Z
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

Z is called a Moore or static output.
It is a function only of the current state

Combined Transition Table

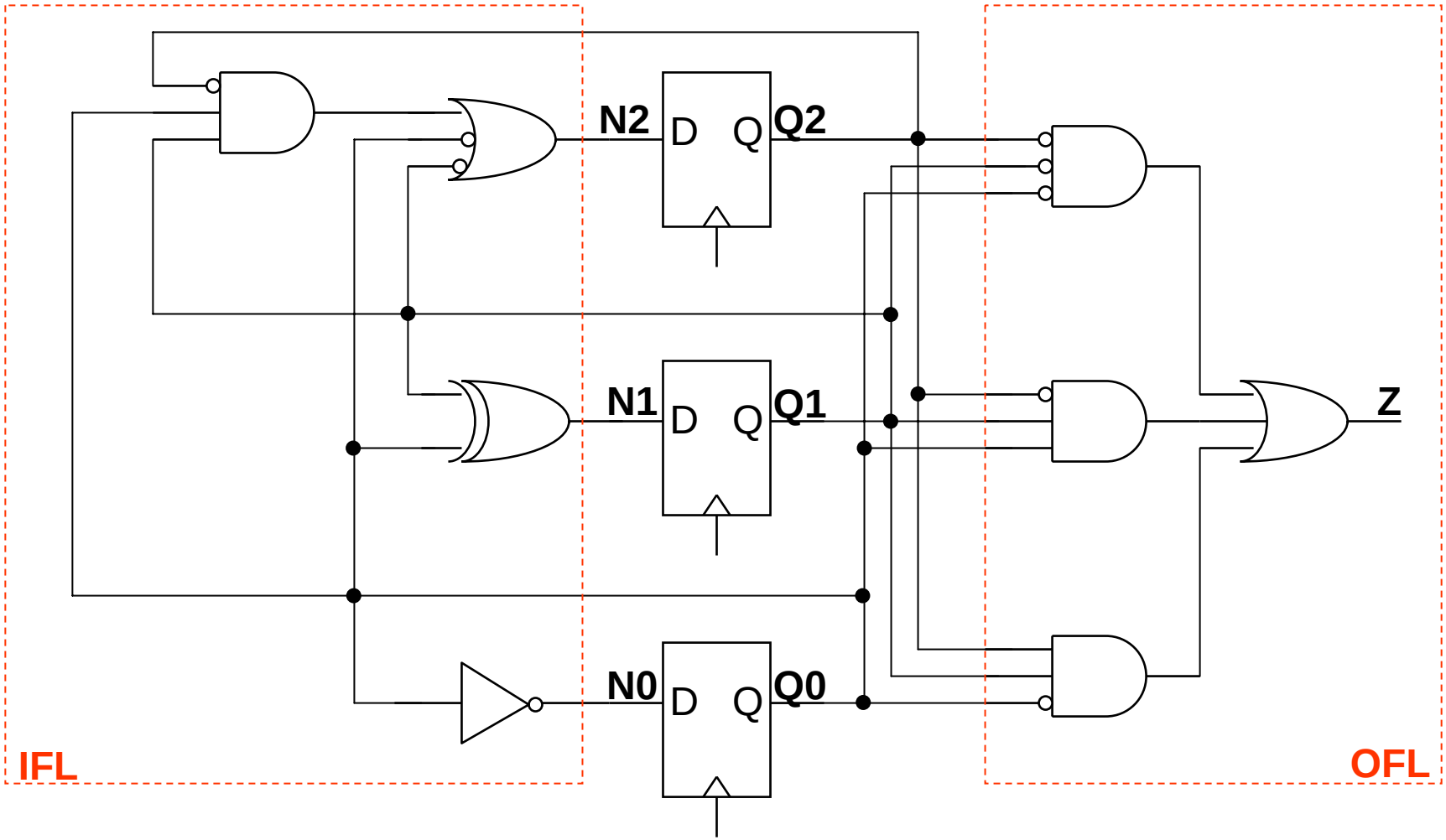
Q2	Q1	Q0	N2	N1	N0	Z
0	0	0	0	0	1	1
0	0	1	0	1	0	0
0	1	0	0	1	1	0
0	1	1	1	0	0	1
1	0	0	1	0	1	0
1	0	1	1	1	0	0
1	1	0	1	1	1	1
1	1	1	0	0	0	0



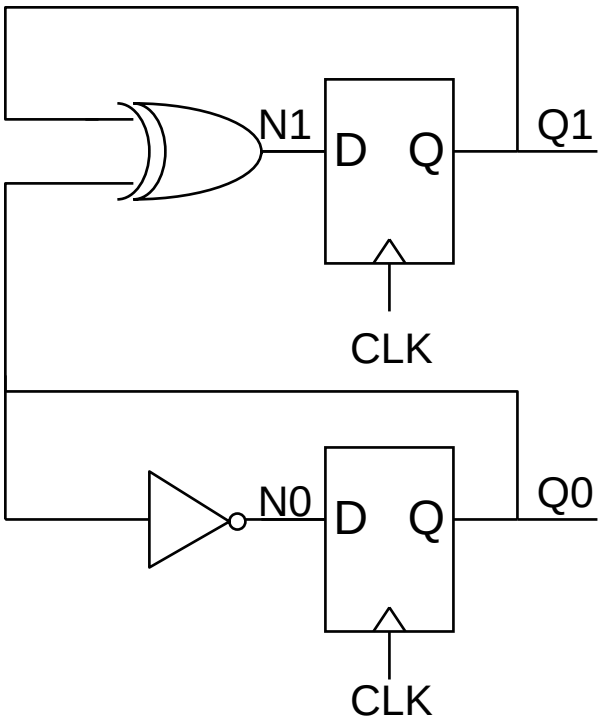
Z = Q2'Q1'Q0' + Q2'Q1Q0 + Q2Q1Q0'

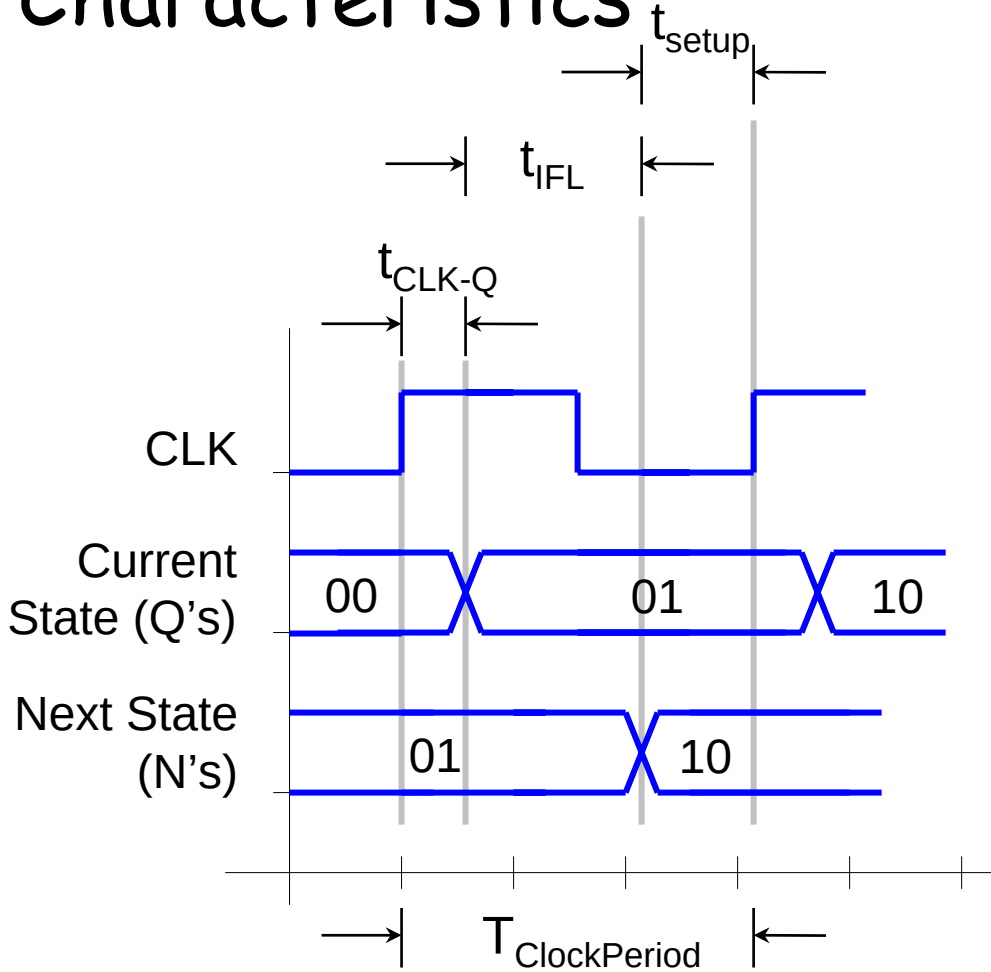
(implement OFL with gates)

Counter With A Moore Output



Counter Delay Characteristics





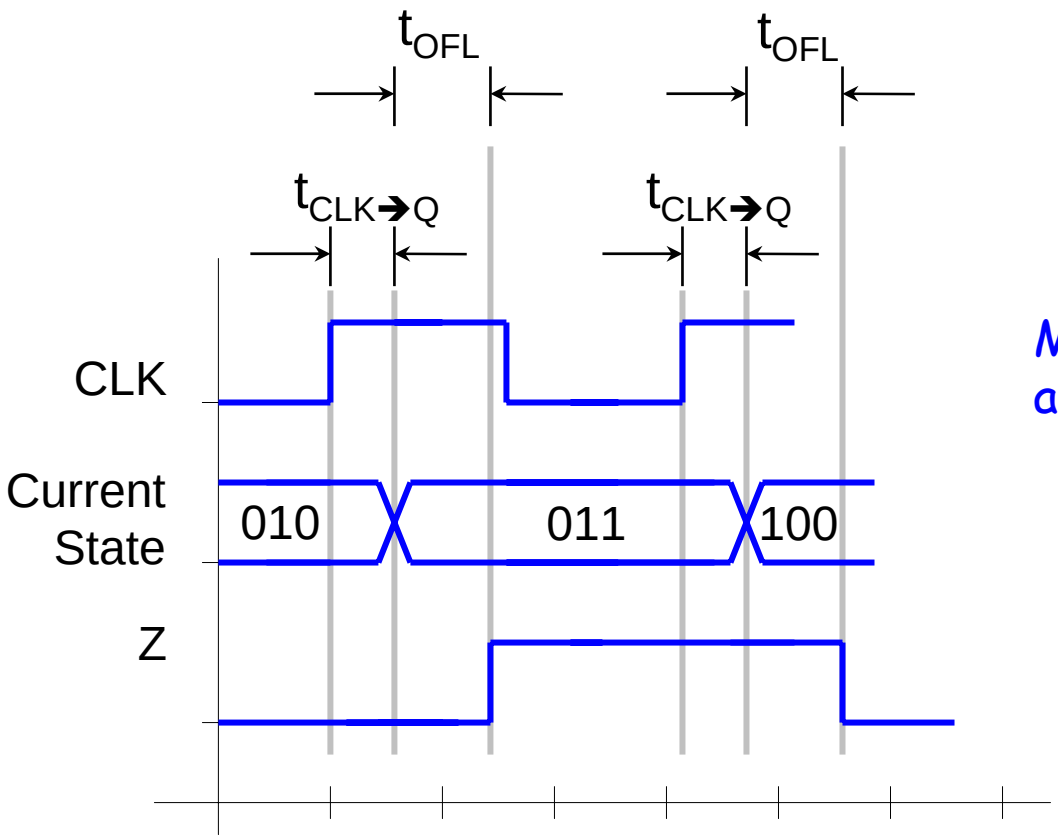
$$T_{ClockPeriod} >= t_{CLK-Q} + t_{IFL} + t_{setup}$$

Counter Delay Example

- $t_{CLK \rightarrow Q}$ = 1ns
- t_{IFL} = 7ns
- t_{setup} = 2ns

- $T_{ClockPeriod}$ >= 10ns (10×10^{-9} sec)
- ClockRate <= 100MHz (100×10^{6} Hz)
  - Rate = 1/Period

Next State and Output Timings



More outputs appear $t_{CLK \rightarrow Q} + t_{OFL}$ after the clock edge.

Counters and Timing

- The count sequence affects the size of IFL
- The size of the IFL affects t_{IFL}
- t_{IFL} affects $T_{ClockPeriod}$
- Choosing binary counts is not always fastest or smallest

Example Problem

- Design a 3-bit binary counter
- The Z output is TRUE iff count value is even
- The Y output is TRUE iff count value is multiple of 3

COUNTERS WITH INPUTS

Inputs

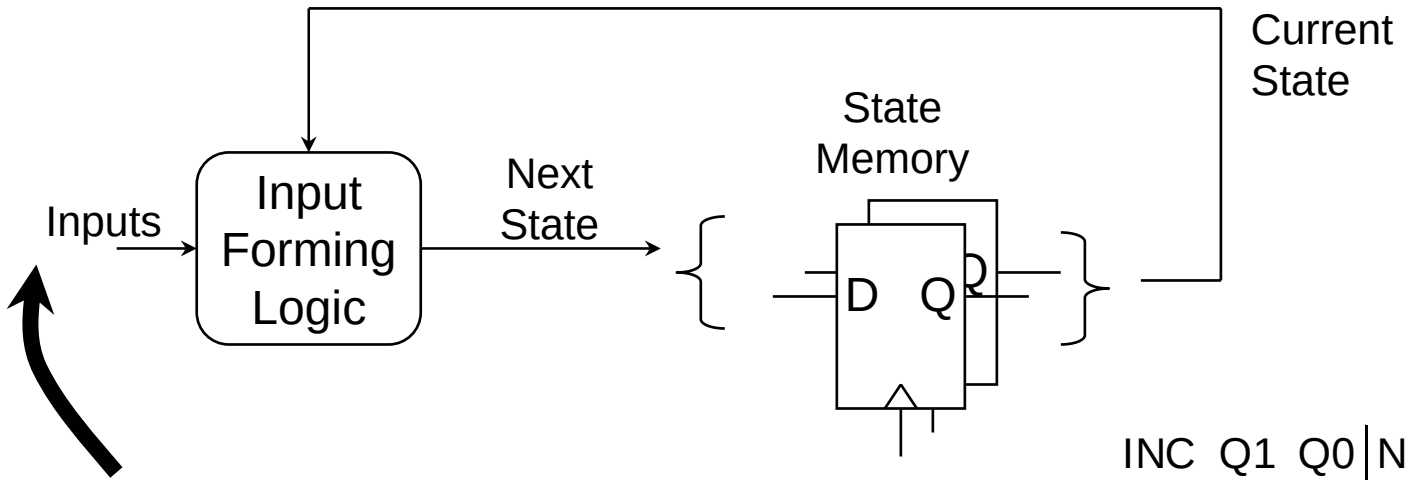
Mealy Outputs

Mealy vs. Moore

Counters With Inputs

- What if you don't want counter to increment every single cycle?

An Incrementable Counter



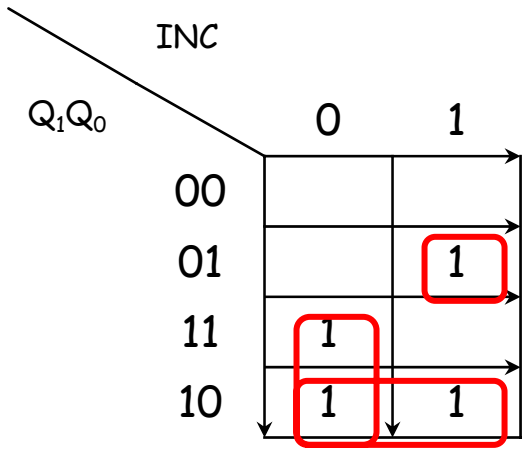
Note that the next state is a function of the input 'INC' as well as the current state.

INC	Q1	Q0	N1	N0
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	1	1
1	0	0	0	1
1	0	1	1	0
1	1	0	1	1
1	1	1	0	0

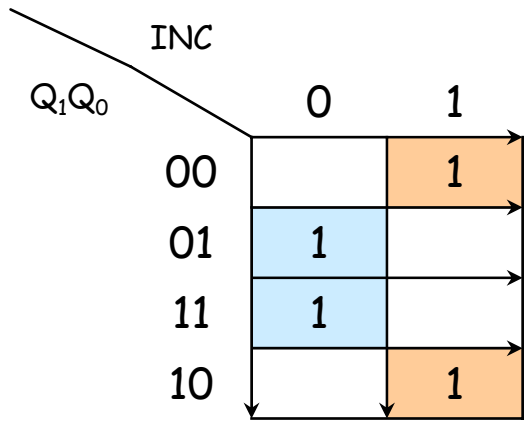
Incrementable Counter Derivation

INC	Q1	Q0	N1	N0
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	1	1
1	0	0	0	1
1	0	1	1	0
1	1	0	1	1
1	1	1	0	0

Doing the KMaps for this results in:



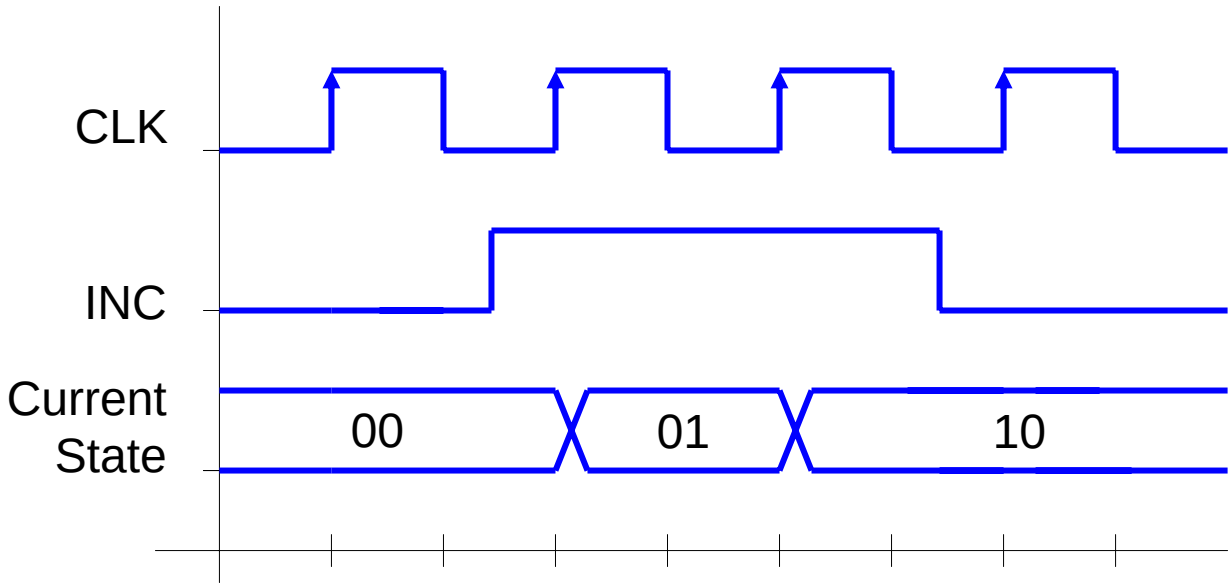
N1



N0

N1 = (INC•Q1'•Q0) + (INC'•Q1) + (Q1•Q0')
N0 = (INC'•Q0) + (INC•Q0') = INC⊕Q0

Incrementable Counter Behavior



The counter increments on the clock edge only when INC is asserted

Counters With More Inputs

	CLR	INC	Q1	Q0	N1	N0
'CLR' = 'INC' = 0 No state transition	0	0	0	0	0	0
	0	0	0	1	0	1
	0	0	1	0	1	0
	0	0	1	1	1	1
'CLR' = 0 'INC' = 1 Counter Increments	0	1	0	0	0	1
	0	1	0	1	1	0
	0	1	1	0	1	1
	0	1	1	1	0	0
'CLR' = 1 'INC' = 0 Counter resets to '00'	1	0	0	0	0	0
	1	0	0	1	0	0
	1	0	1	0	0	0
	1	0	1	1	0	0
'CLR' = 1 'INC' = 1 What should it do?	1	1	0	0	?	?
	1	1	0	1	?	?
	1	1	1	0	?	?
	1	1	1	1	?	?

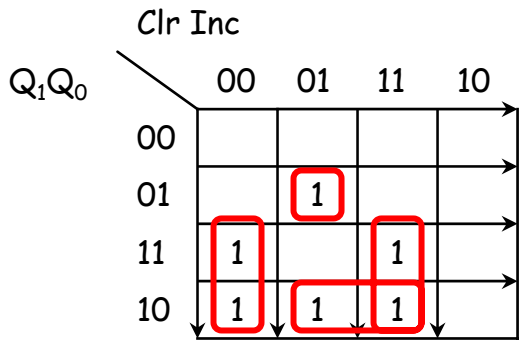
Precedence of INC vs. CLR?

1. Could do nothing
2. Could give INC precedence
3. Could give CLR precedence
4. Assume INC=CLR=1 will never occur

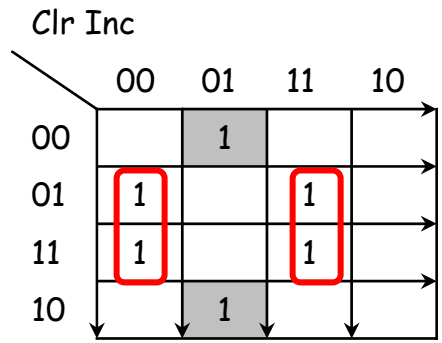
- You decide when you draw the transition table

Case 1 – Do Nothing

CLR	INC	Q1	Q0	N1	N0
0	0	0	0	0	0
0	0	0	1	0	1
0	0	1	0	1	0
0	0	1	1	1	1
0	1	0	0	0	1
0	1	0	1	1	0
0	1	1	0	1	1
0	1	1	1	0	0
1	0	0	0	0	0
1	0	0	1	0	0
1	0	1	0	0	0
1	0	1	1	0	0
1	1	0	0	0	0
1	1	0	1	0	1
1	1	1	0	1	0
1	1	1	1	1	1



N1



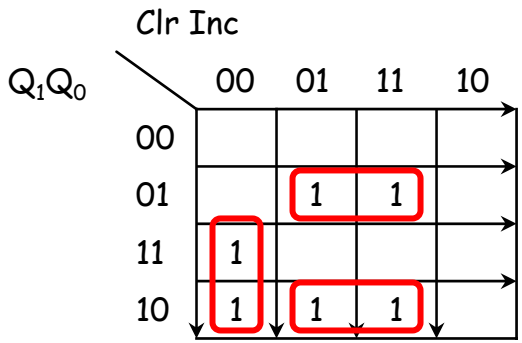
N0

N1 = CLR'•INC•Q1'•Q0 +
CLR'•INC'•Q1 +
CLR•INC•Q1 +
INC•Q1•Q0'

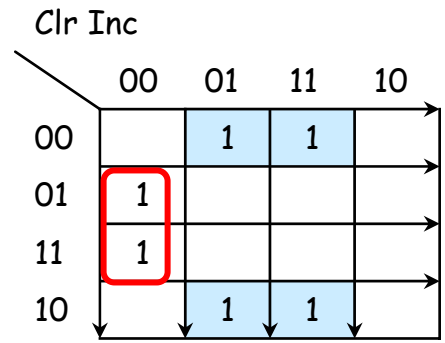
N0 = CLR'•INC'•Q0 +
CLR•INC•Q0 +
CLR'•INC•Q0'

Case 2 – Give INC Precedence

CLR	INC	Q1	Q0	N1	N0
0	0	0	0	0	0
0	0	0	1	0	1
0	0	1	0	1	0
0	0	1	1	1	1
0	1	0	0	0	1
0	1	0	1	1	0
0	1	1	0	1	1
0	1	1	1	0	0
1	0	0	0	0	0
1	0	0	1	0	0
1	0	1	0	0	0
1	0	1	1	0	0
1	1	0	0	0	1
1	1	0	1	1	0
1	1	1	0	1	1
1	1	1	1	0	0



N1



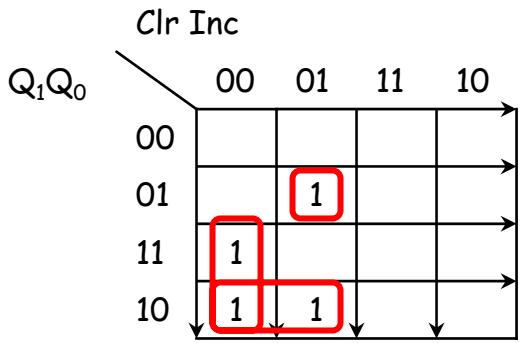
N0

N1 = INC•Q1'•Q0 + CLR'•INC'•Q1 + INC•Q1•Q0'

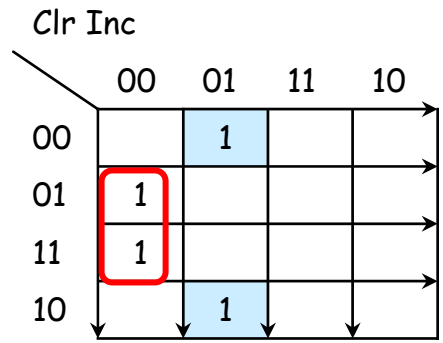
N0 = CLR'•INC'•Q0 + INC•Q0'

Case 3 – Give CLR Precedence

CLR	INC	Q1	Q0	N1	N0
0	0	0	0	0	0
0	0	0	1	0	1
0	0	1	0	1	0
0	0	1	1	1	1
0	1	0	0	0	1
0	1	0	1	1	0
0	1	1	0	1	1
0	1	1	1	0	0
1	0	0	0	0	0
1	0	0	1	0	0
1	0	1	0	0	0
1	0	1	1	0	0
1	1	0	0	0	0
1	1	0	1	0	0
1	1	1	0	0	0
1	1	1	1	0	0



N1



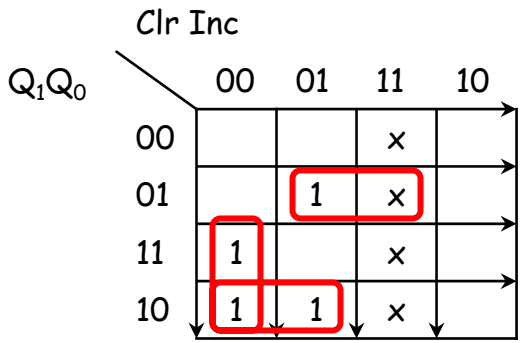
N0

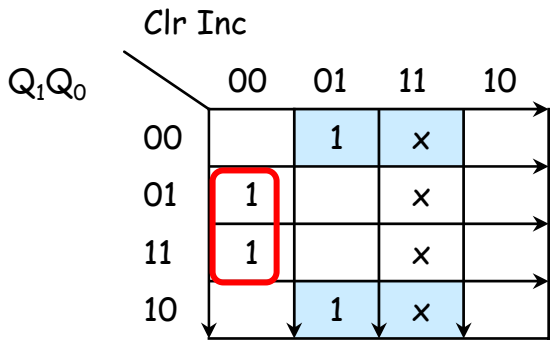
N1 = CLR'•INC•Q1'•Q0 +
CLR'•INC'•Q1 +
CLR'•Q1•Q0'

N0 = CLR'•INC'•Q0 +
CLR'•INC•Q0'

Case 4 – Assume INC=CLR=1 Will Never Occur

CLR	INC	Q1	Q0	N1	N0
0	0	0	0	0	0
0	0	0	1	0	1
0	0	1	0	1	0
0	0	1	1	1	1
0	1	0	0	0	1
0	1	0	1	1	0
0	1	1	0	1	1
0	1	1	1	0	0
1	0	0	0	0	0
1	0	0	1	0	0
1	0	1	0	0	0
1	0	1	1	0	0
1	1	0	0	X	X
1	1	0	1	X	X
1	1	1	0	X	X
1	1	1	1	X	X





N1 **N0**

N1 = CLR'•INC'•Q1 + INC•Q1'•Q0 + CLR'•Q1•Q0'

N0 = CLR'•INC'•Q0 + INC•Q0'

What happens in the real circuit when INC=CLR=1?

It depends on the final equations…

Case 4 – When INC=CLR=1 What Happens?

CLR	INC	Q1	Q0	N1	N0
0	0	0	0	0	0
0	0	0	1	0	1
0	0	1	0	1	0
0	0	1	1	1	1
0	1	0	0	0	1
0	1	0	1	1	0
0	1	1	0	1	1
0	1	1	1	0	0
1	0	0	0	0	0
1	0	0	1	0	0
1	0	1	0	0	0
1	0	1	1	0	0
1	1	0	0	0	1
1	1	0	1	1	0
1	1	1	0	0	1
1	1	1	1	0	0

N1 = CLR'•INC'•Q1 + INC•Q1'•Q0 + CLR'•Q1•Q0'

N0 = CLR'•INC'•Q0 + INC•Q0'

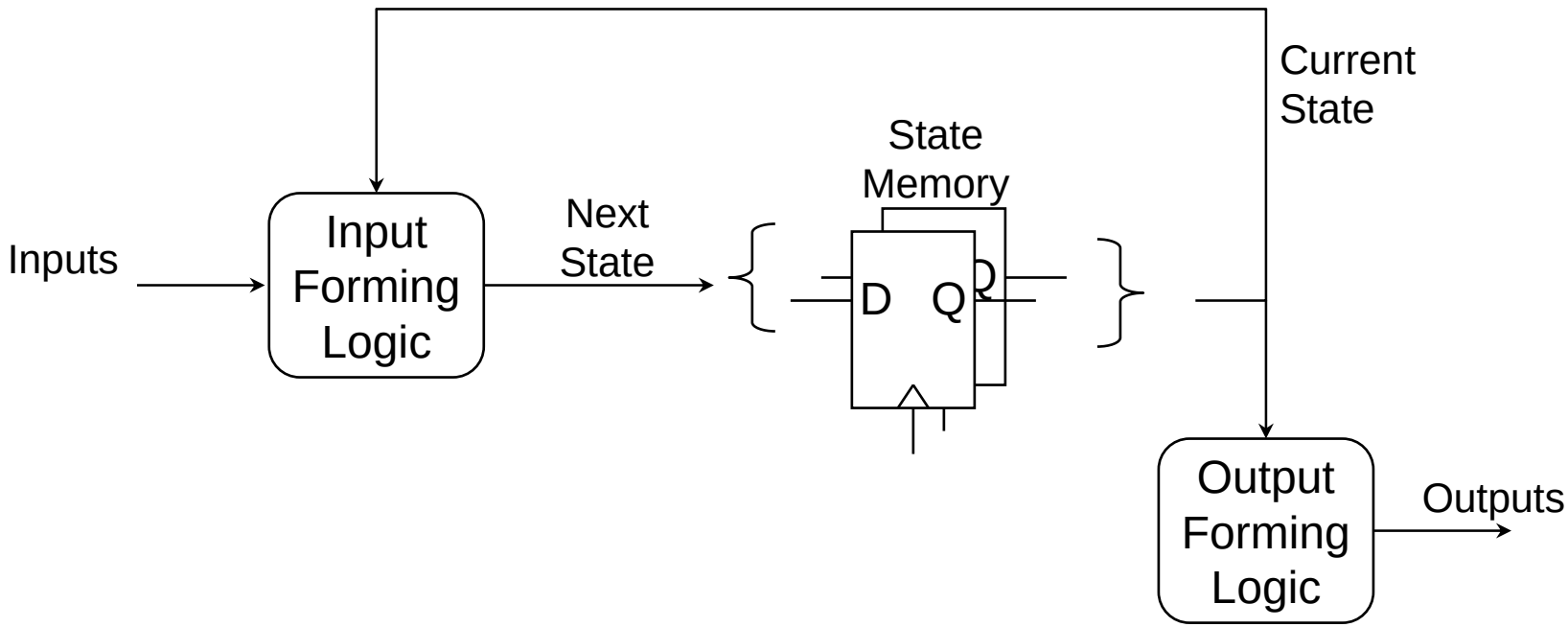
In the end, X's in TT's result in *some* predictable behavior, you just don't care what that is…

Moore vs. Mealy

Sequential Machines

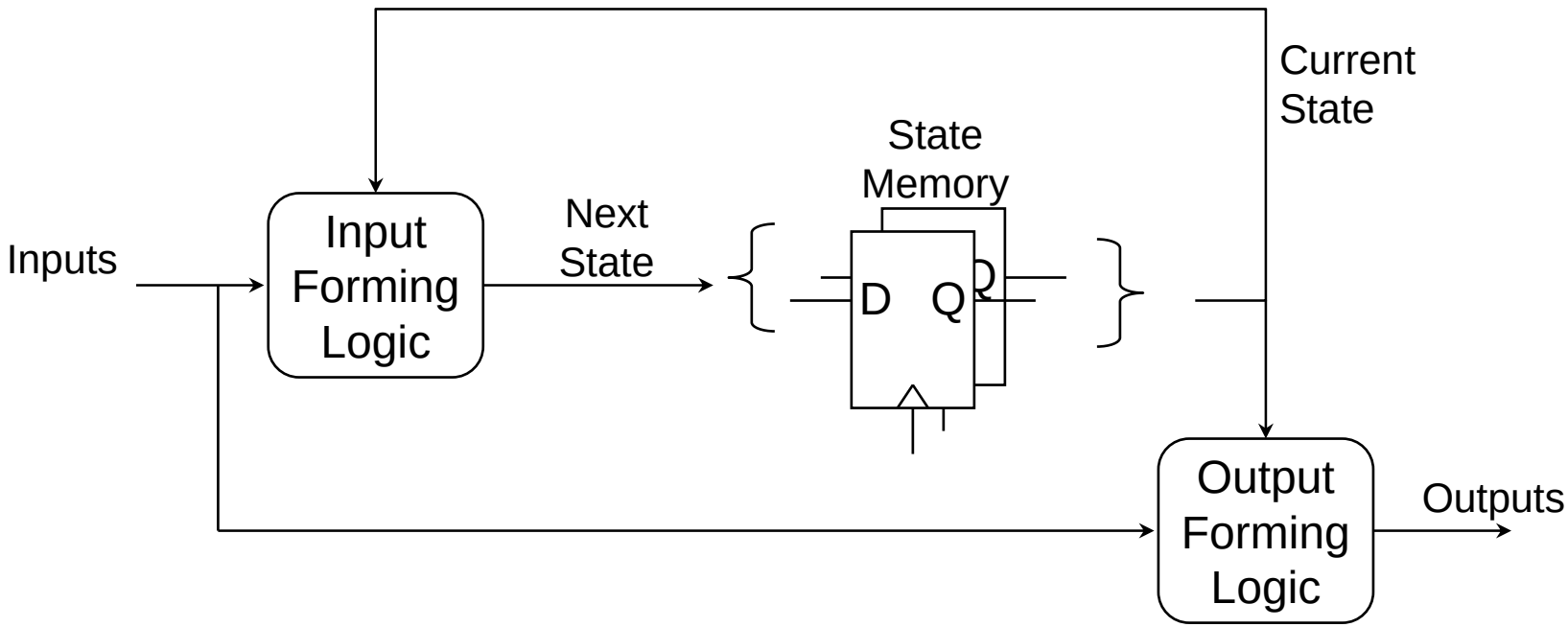
Moore Sequential Machine (static)

- Outputs are a function of current state only



Mealy Sequential Machine (dynamic)

- Outputs are a function of current state ***and*** inputs



Example: 2 Bit Counter

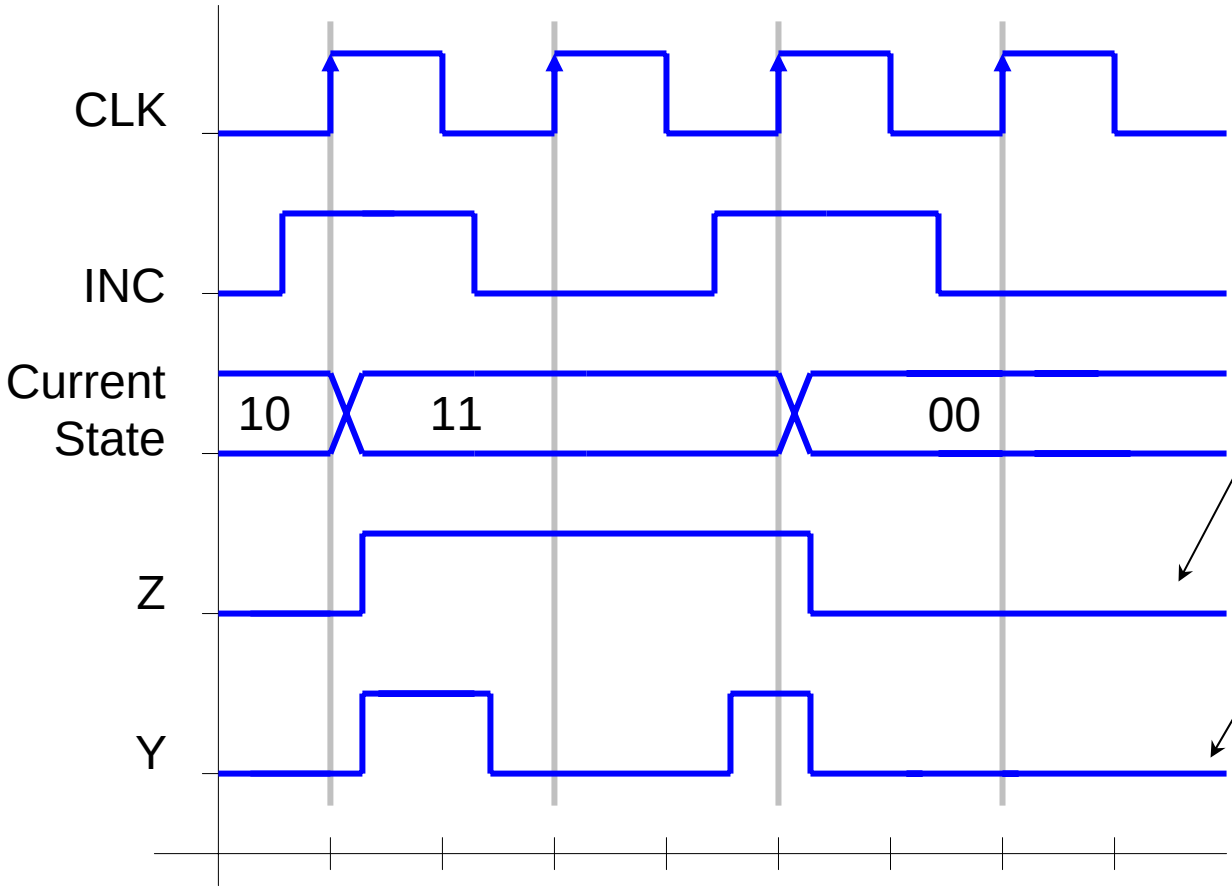
- Moore output (Z) on count=3
- Mealy output (Y) when count=3 and INC=1
  - Incrementing from 3 back to 0

INC	Q1	Q0	N1	N0	Z	Y
0	0	0	0	0	0	0
0	0	1	0	1	0	0
0	1	0	1	0	0	0
0	1	1	1	1	1	0
1	0	0	0	1	0	0
1	0	1	1	0	0	0
1	1	0	1	1	0	0
1	1	1	0	0	1	1

Moore outputs usually require multiple entries in TT.

Mealy outputs often require only one.

Mealy vs. Moore Timing



Z output is TRUE whenever count=3 (it is static function of current state)

Y output is TRUE whenever count=3 AND INC=1 (it is dynamic function of current state and input)