# Counters and Sequence Generators

CMSC 313

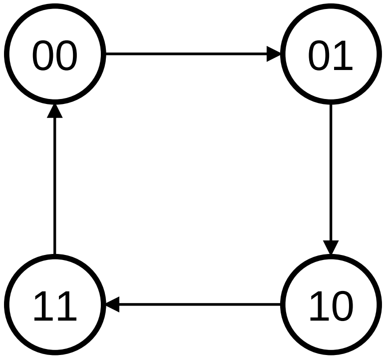
Raphael Elspas

# Counter

- Sometimes we want to count numbers and store their values in a series of flip flops.
- We’ll design a counter with several flip flops connected in a cascade.
- Example: if we want to count 0-7, we need 3 flip flops to store values 000-111, and then connect them in the right way so that after 111, the values go back to zero.
- The tricky part is coming up with the design that works for every state

# 0-3 Counter

- Let's design a counter that counts 0→1→2→3→0 (back to 0 in a loop) with JK flip flops that is triggered by positive edge clock cycles.
- We know we will need 2 flip flops.
- Let's draw a state table and state diagram for all state transitions




| $Q_A$ | $Q_B$ | $Q_A+$ | $Q_B+$ |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |

# 0-3 Counter (cont.)

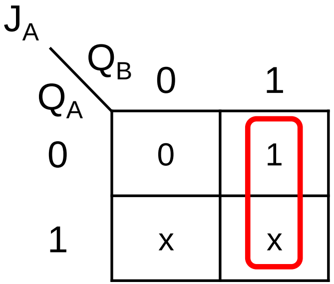
- We need to find how to connect the JK flip flops together.
- Let's find the JK inputs required to convert $Q_A$ to $Q_A$+ and $Q_B$ to $Q_B$+
- Extend state table so JK values are colinear with State transitions
- Values are labeled x when there are multiple ways to get a transition. E.g.to go from $Q_A = 0$ to $Q_A = 0$, **either** J=0, K=0 **or** J=0, K=1. Since K can be either value, we mark it as a don't care.

| $Q_A$ | $Q_B$ | $Q_A$+ | $Q_B$+ | $J_A$ | $K_A$ | $J_B$ | $K_B$ |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | X | 1 | X |
| 0 | 1 | 1 | 0 | 1 | X | X | 1 |
| 1 | 0 | 1 | 1 | X | 0 | 1 | X |
| 1 | 1 | 0 | 0 | X | 1 | X | 1 |

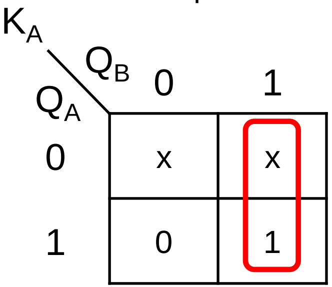
# 0-3 Counter (cont.)

- From all J K values design circuit with K Maps in terms of current states $Q_A$ and $Q_B$.

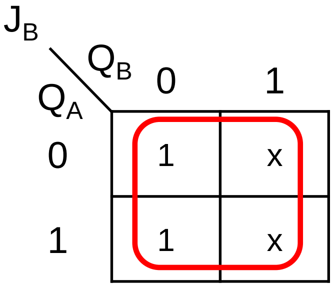
| $Q_A$ | $Q_B$ | $Q_A+$ | $Q_B+$ | $J_A$ | $K_A$ | $J_B$ | $K_B$ |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | X | 1 | X |
| 0 | 1 | 1 | 0 | 1 | X | X | 1 |
| 1 | 0 | 1 | 1 | X | 0 | 1 | X |
| 1 | 1 | 0 | 0 | X | 1 | X | 1 |



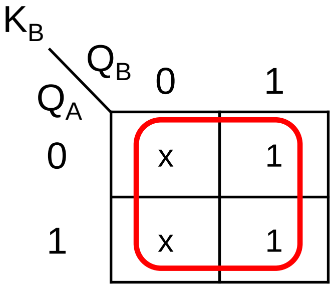

$J_A = Q_B$




$K_A = Q_B$

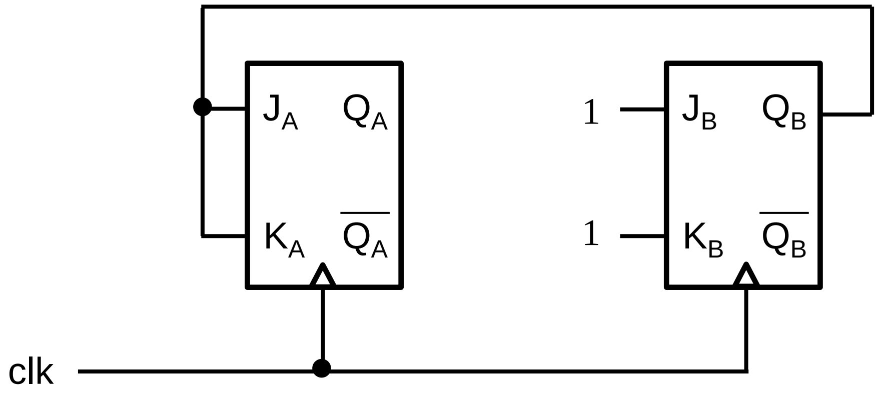



$J_B = 1$




$K_B = 1$

# 0-3 Counter (cont.)



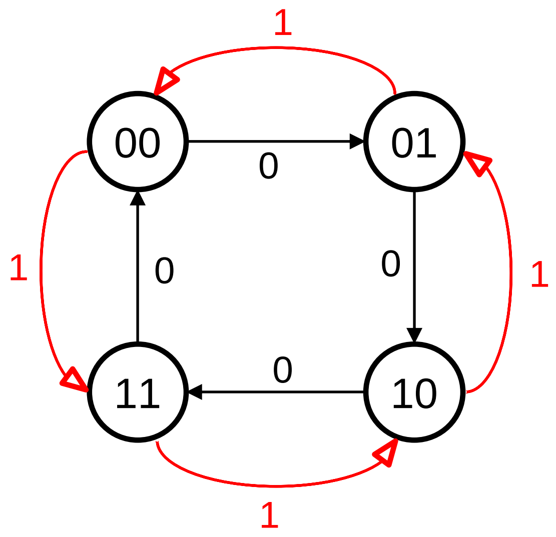

You can read the value of the counter by reading the values at $Q_A$ and $Q_B$.

# Bidirectional counter

- Let's make a 2 bit counter that counts both up and down
  0→1→2→3→(back to 0) and 3→2→1→0→(back to 3)
- We want to control it going up or down, so let's use an input for control
- Another constraint: The MSB shall use a T flip flop and the LSB shall use a D flip flop
- Let's design!

# State diagram and table

We now need another set of arrows on our diagram, as well as inputs describing what the next state is depending on input




| X | $Q_A$ | $Q_B$ | $Q_A+$ | $Q_B+$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 |

# State design

- Populate T and D with excitation table

Combined excitation table for T and D

| Q | Q+ | T | D |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

| X | $Q_A$ | $Q_B$ | $Q_A+$ | $Q_B+$ | T | D |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 |

# State design

| X | $Q_A$ | $Q_B$ | $Q_A+$ | $Q_B+$ | T | D |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 |

T

| x \ $Q_AQ_B$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 |

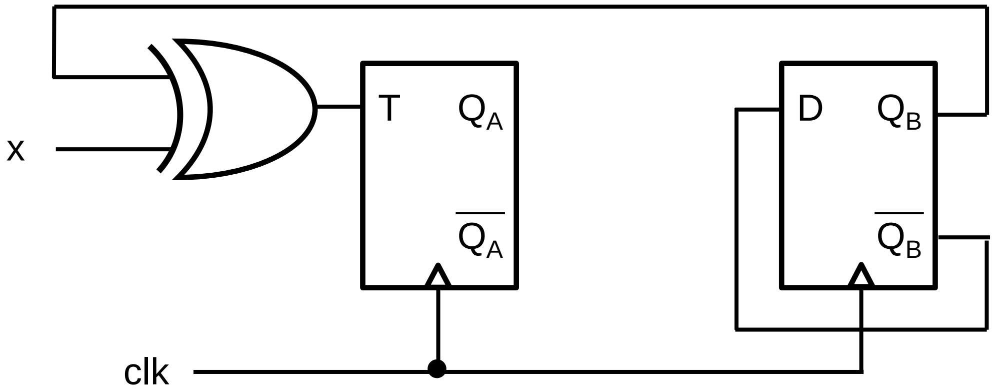
$$T = \bar{x}Q_B + x\overline{Q_B} = x \oplus Q_B$$

D

| x \ $Q_AQ_B$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |

$$D = \overline{Q_B}$$

# 0-3 up/down counter



# Sequence Generator

- What if we want to count ranges that are not a power of 2?
- We will have unused states
- Its possible to start in one of the states and have undefined behavior.
- We can either:
  - We can force our flip flops to start with a certain states
  - We can interpolate states
- What if we want to count out of order?
- A sequence generator is a counter where any sequence of numbers will repeat in a cycle.

UMBC

# Sequence Generator

- Let's create a sequence with pattern 1→3→2→7→1 (repeat)

Excitation table for JK

| Q Output | | Inputs | |
|---|---|---|---|
| Present State | Next State | $J_n$ | $K_n$ |
| 0 | 0 | 0 | x |
| 0 | 1 | 1 | x |
| 1 | 0 | x | 1 |
| 1 | 1 | x | 0 |

| $Q_A$ | $Q_B$ | $Q_C$ | $Q_A+$ | $Q_B+$ | $Q_C+$ | $J_A$ | $K_A$ | $J_B$ | $K_B$ | $J_C$ | $K_C$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | | | | | | | | | |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | x | 1 | x | x | 0 |
| 0 | 1 | 0 | 1 | 1 | 1 | 1 | x | x | 0 | 1 | x |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 | x | x | 0 | x | 1 |
| 1 | 0 | 0 | | | | | | | | | |
| 1 | 0 | 1 | | | | | | | | | |
| 1 | 1 | 0 | | | | | | | | | |
| 1 | 1 | 1 | 0 | 0 | 1 | x | 1 | x | 1 | x | 0 |

# Seq Gen (cont.)

| $Q_A$ | $Q_B$ | $Q_C$ | $Q_A+$ | $Q_B+$ | $Q_C+$ | $J_A$ | $K_A$ | $J_B$ | $K_B$ | $J_C$ | $K_C$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | | | | | | | | | |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | x | 1 | x | x | 0 |
| 0 | 1 | 0 | 1 | 1 | 1 | 1 | x | x | 0 | 1 | x |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 | x | x | 0 | x | 1 |
| 1 | 0 | 0 | | | | | | | | | |
| 1 | 0 | 1 | | | | | | | | | |
| 1 | 1 | 0 | | | | | | | | | |
| 1 | 1 | 1 | 0 | 0 | 1 | x | 1 | x | 1 | x | 0 |

## Kmaps

$J_A$

| $Q_A$ \ $Q_BQ_C$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | x | 0 | 0 | 1 |
| 1 | x | x | x | x |

$J_A = \overline{Q_C}$

$K_A$

| $Q_A$ \ $Q_BQ_C$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | x | x | x | x |
| 1 | x | x | 1 | x |

$K_A = 1$

$J_B$

| $Q_A$ \ $Q_BQ_C$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | x | 1 | x | x |
| 1 | x | x | x | x |

$J_B = 1$

$K_B$

| $Q_A$ \ $Q_BQ_C$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | x | x | 0 | 0 |
| 1 | x | x | 1 | x |

$K_B = Q_A$

$J_C$

| $Q_A$ \ $Q_BQ_C$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | x | x | x | 1 |
| 1 | x | x | x | x |

$J_C = 1$

$K_C$

| $Q_A$ \ $Q_BQ_C$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | x | 0 | 1 | x |
| 1 | x | x | 0 | x |

$K_C = \overline{Q_A}Q_B$

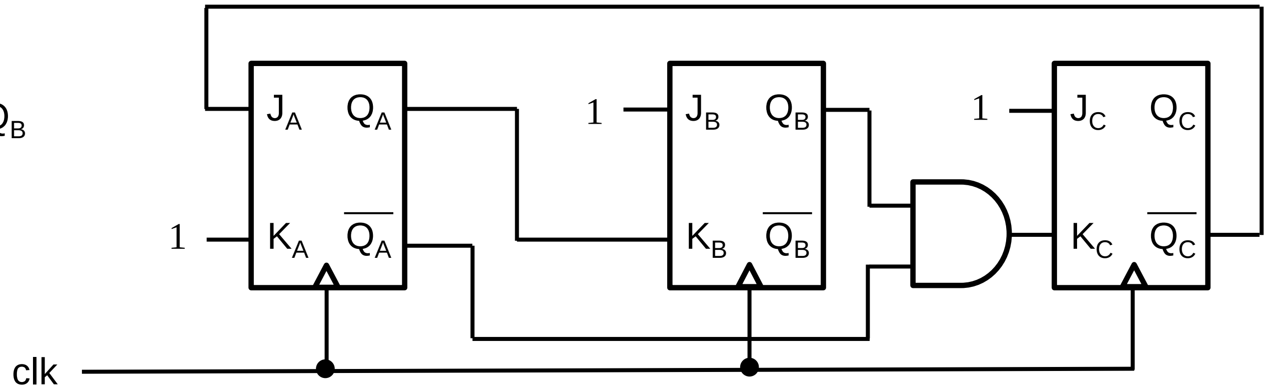
# Seq Gen

- We can draw a circuit from our equations

$J_A = \overline{Q_C}$  $K_A = 1$

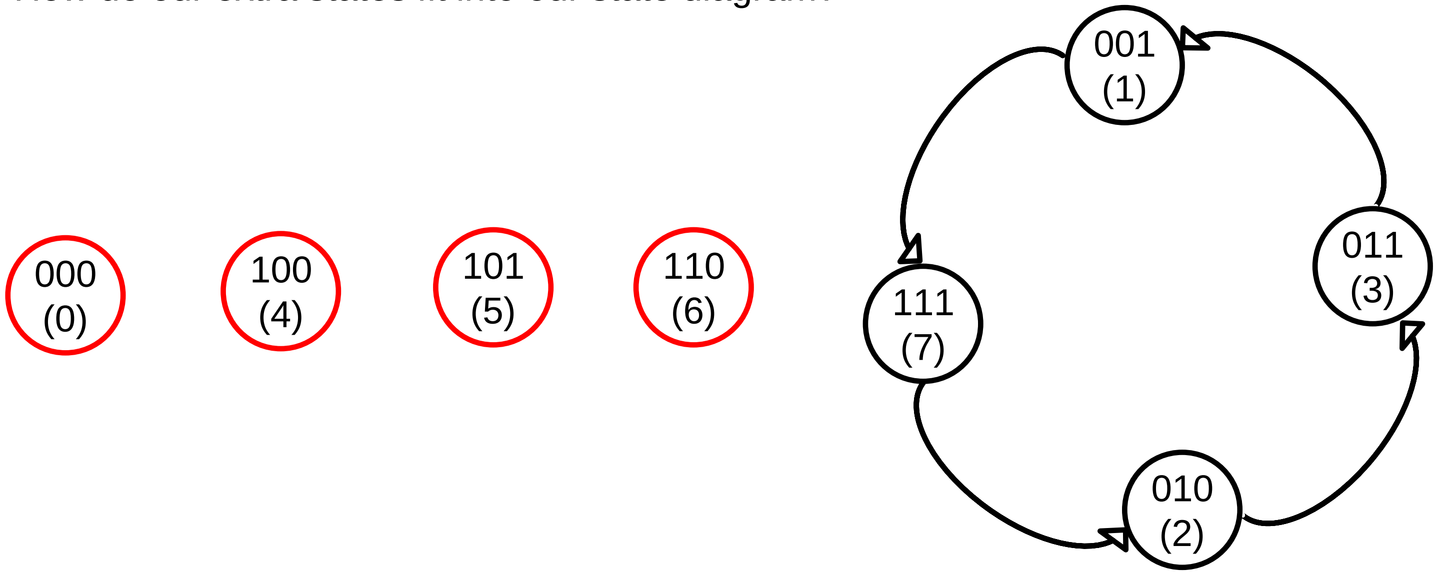
$J_B = 1$  $K_B = Q_A$

$J_C = 1$  $K_C = \overline{Q_A}Q_B$



## Seq Gen State diagram

How do our extra states fit into our state diagram?



UMBC

# Seq Gen (cont.)

| $Q_A$ | $Q_B$ | $Q_C$ | $Q_A+$ | $Q_B+$ | $Q_C+$ | $J_A$ | $K_A$ | $J_B$ | $K_B$ | $J_C$ | $K_C$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | | | | 1 | 1 | 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | X | 1 | X | X | 0 |
| 0 | 1 | 0 | 1 | 1 | 1 | 1 | X | X | 0 | 1 | X |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 | X | X | 0 | X | 1 |
| 1 | 0 | 0 | | | | 1 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 1 | | | | 0 | 1 | 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | | | | 1 | 1 | 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 1 | x | 1 | x | 1 | x | 0 |

- Let's find out what happens to the other states and their transitions.
- From J and K equations, find what values the don't cares are being used as (zeros or ones)

$J_A = \overline{Q_C}$ $K_A = 1$

$J_B = 1$ $K_B = Q_A$

$J_C = 1$ $K_C = \overline{Q_A}Q_B$

# Seq Gen (cont.)

| $Q_A$ | $Q_B$ | $Q_C$ | $Q_A+$ | $Q_B+$ | $Q_C+$ | $J_A$ | $K_A$ | $J_B$ | $K_B$ | $J_C$ | $K_C$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | X | 1 | X | X | 0 |
| 0 | 1 | 0 | 1 | 1 | 1 | 1 | X | X | 0 | 1 | X |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 | X | X | 0 | X | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 1 | x | 1 | x | 1 | x | 0 |

From new J and K values, determine what the next states will be.

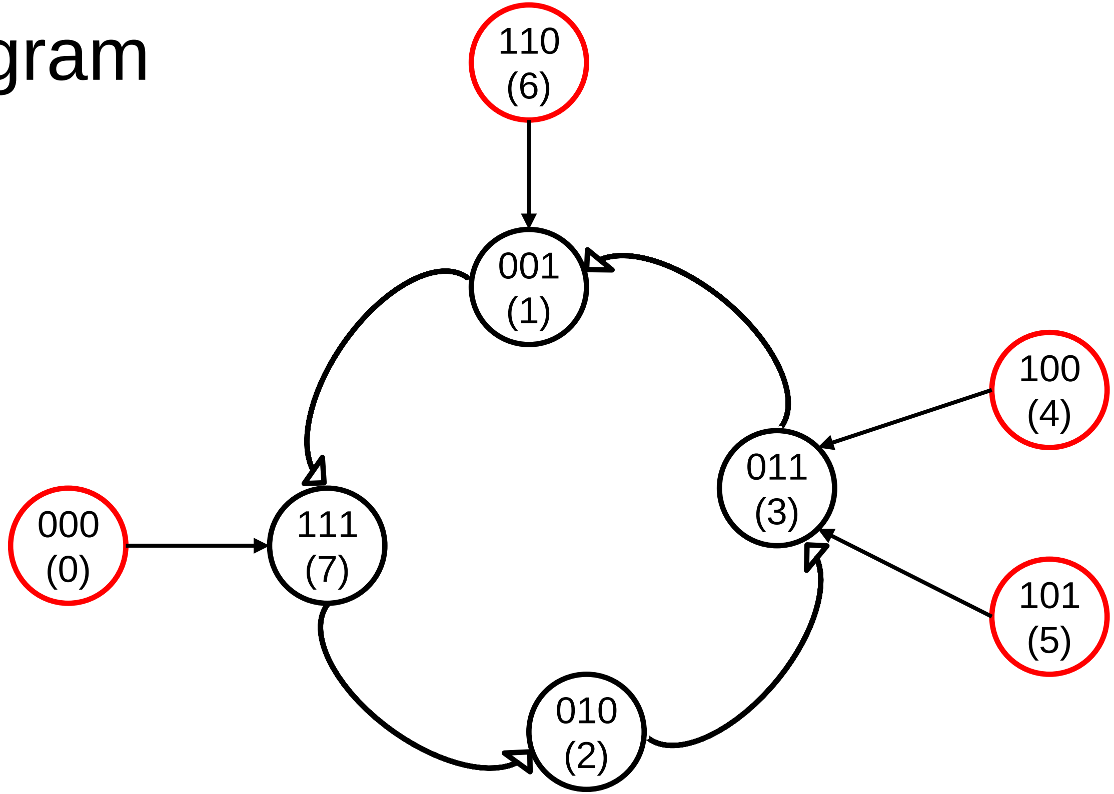
$J_A = \overline{Q_C}$ $K_A = 1$

$J_B = 1$ $K_B = Q_A$

$J_C = 1$ $K_C = \overline{Q_A}Q_B$

# Seq Gen State diagram

| $Q_A$ | $Q_B$ | $Q_C$ | $Q_A+$ | $Q_B+$ | $Q_C+$ |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 | 0 | 1 |



# Any state diagram

- A random state diagram may have a counting mechanism, but it also may be random.
- There may be states that are “islands”.
- You can use the techniques for counters to convert any state diagram into a circuit.
- If you have leftover states that you don’t care about, you can also interpolate where unassigned states will go based on the circuitry by completing the table retroactively as we showed

# Summary

- A sequence generator increments through a series of numbers eventually returning to the first number
- If not all values are used within a sequence, not all values will appear within the state diagram. In order to have a complete state diagram, it needs to be derived which states will succeed other states to ensure that a random starting state will eventually insert into the sequence.

# References