# Binary Counters

ECE 2372 | Modern Digital System Design | Texas Tech University

# Lecture Overview

- Shift Registers
- Synchronous Binary Counters
- Counters for Other Sequences

ECE 2372 | Lecture 305

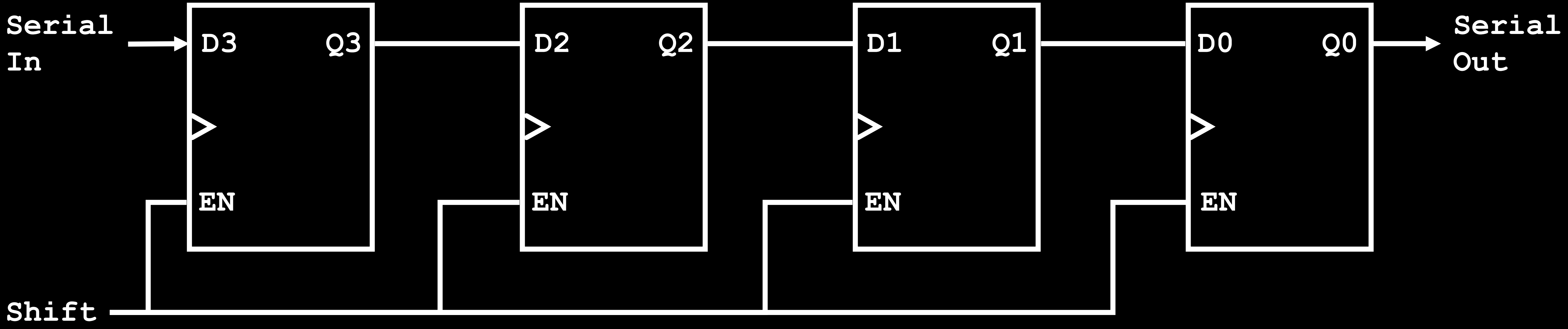
# Shift Registers

Binary Counters | Modern Digital System Design

# Shift Registers: Definition

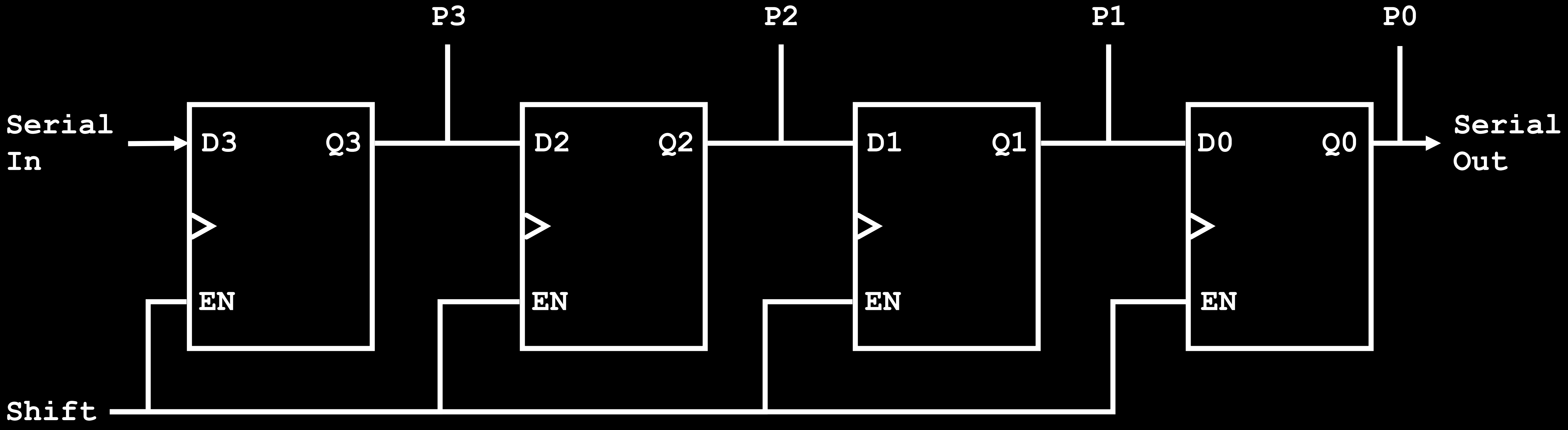
A **shift register** is a register in which binary data can be stored, and this data can be shifted to the left or right when a shift signal is applied.

# Shift Registers: Right-Shift Register



This type of Shift Register is also called a “Serial-in, Serial-out” shift register

# Shift Registers: Serial-In, Parallel-Out (SIPO)



# Shift Registers: SIPO Example

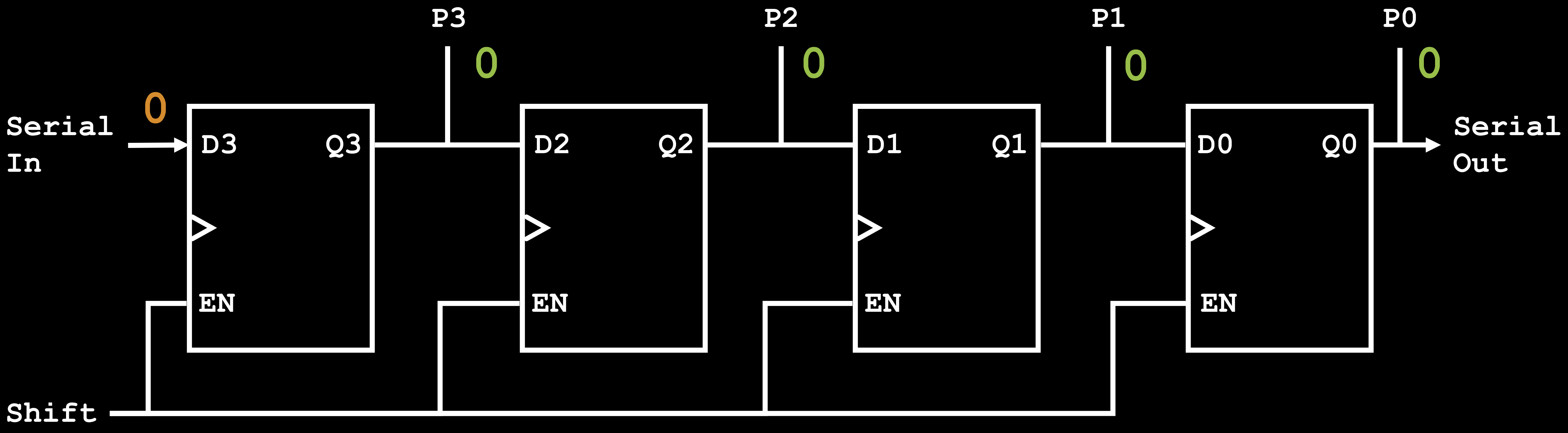
Suppose a 4-bit SIPO register like the one on the previous slide. If all the flip-flops are initially at rest, `Shift` is held at `1`, and the input bit-stream is,

$$SI = \{0, 1, 0, 1, 1, 0, 1\}$$

What will be the parallel output at the end of the 5th clock cycle?

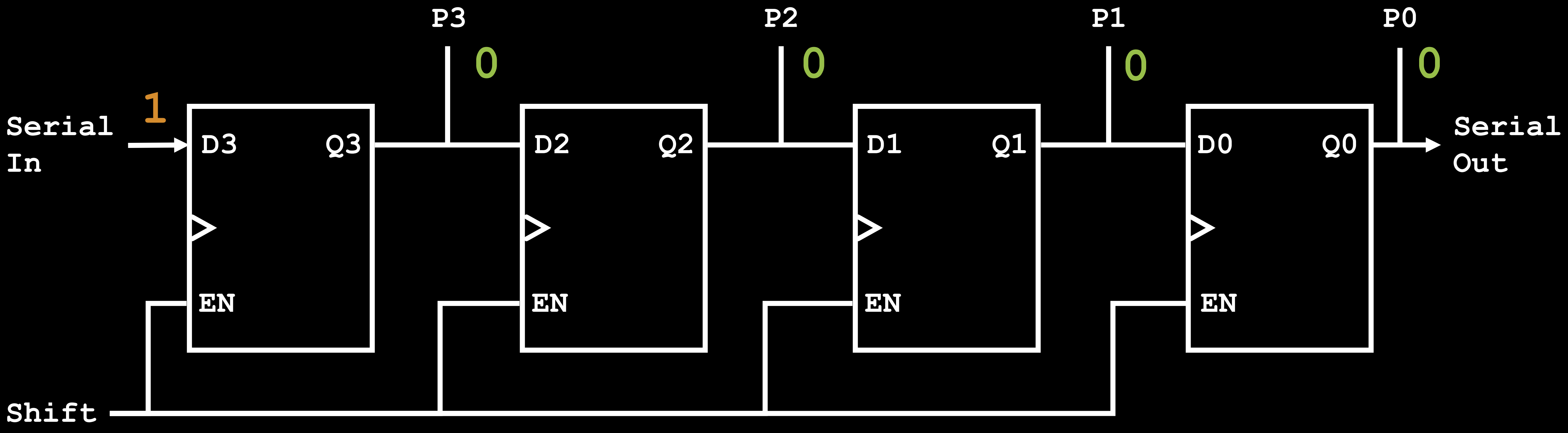
$$P = \{1, 1, 0, 1\}$$

# Shift Registers: SIPO Example



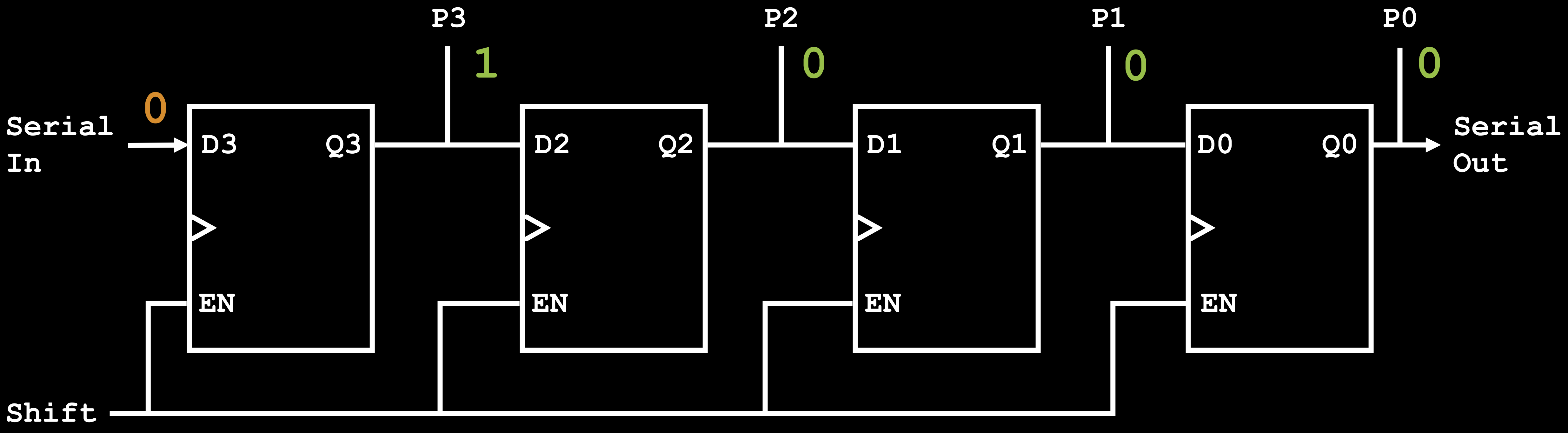
$SI = \{0, 1, 0, 1, 1, 0\}$ $P = \{0, 0, 0, 0\}$ $Cycle = 0$

# Shift Registers: SIPO Example



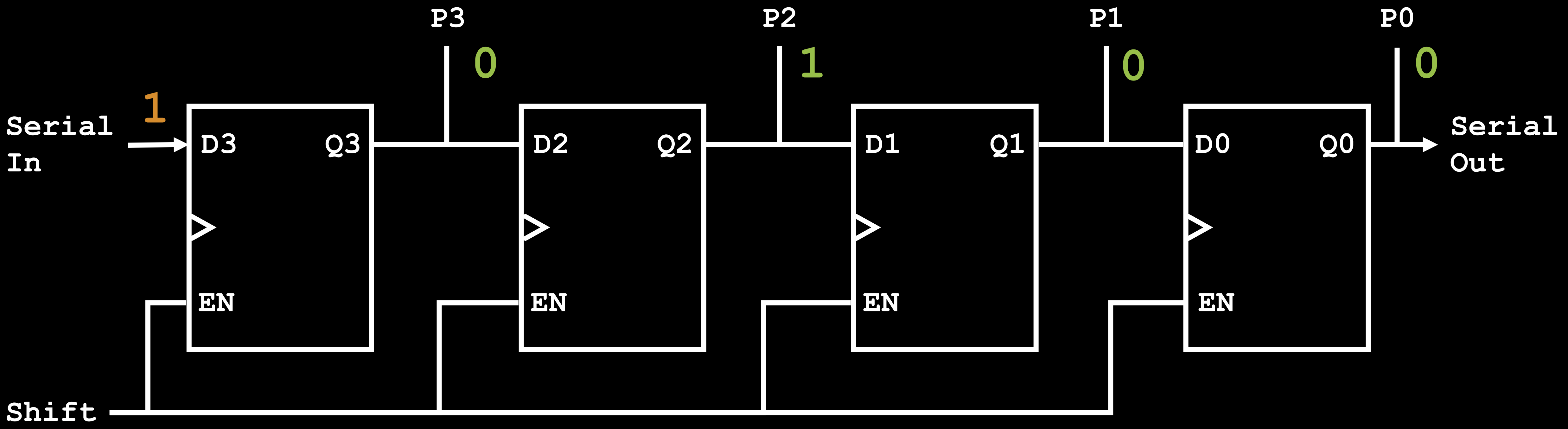
$SI = \{1, 0, 1, 1, 0\}$ $P = \{0, 0, 0, 0\}$ $Cycle = 1$

# Shift Registers: SIPO Example



$SI = \{0, 1, 1, 0\}$ $P = \{1, 0, 0, 0\}$ $Cycle = 2$

# Shift Registers: SIPO Example

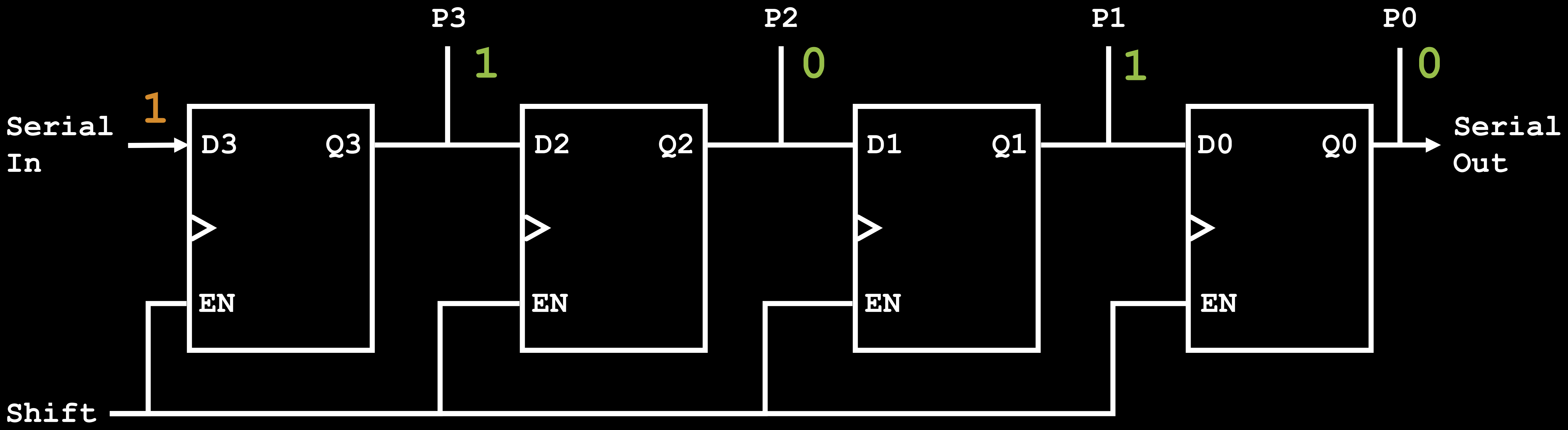


$SI = \{1, 1, 0\}$

$P = \{0, 1, 0, 0\}$

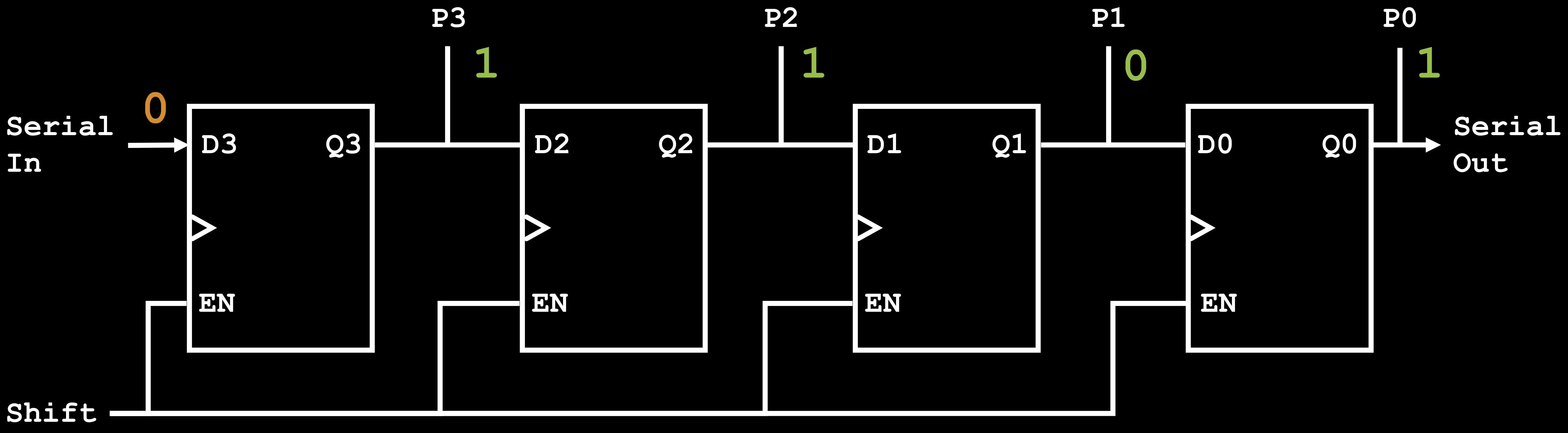
$Cycle = 3$

# Shift Registers: SIPO Example



$SI = \{1, 0\}$ $P = \{1, 0, 1, 0\}$ $Cycle = 4$

# Shift Registers: SIPO Example



$SI = \{0\}$ $P = \{1, 1, 0, 1\}$ $Cycle = 5$

# Shift Registers: SIPO Example

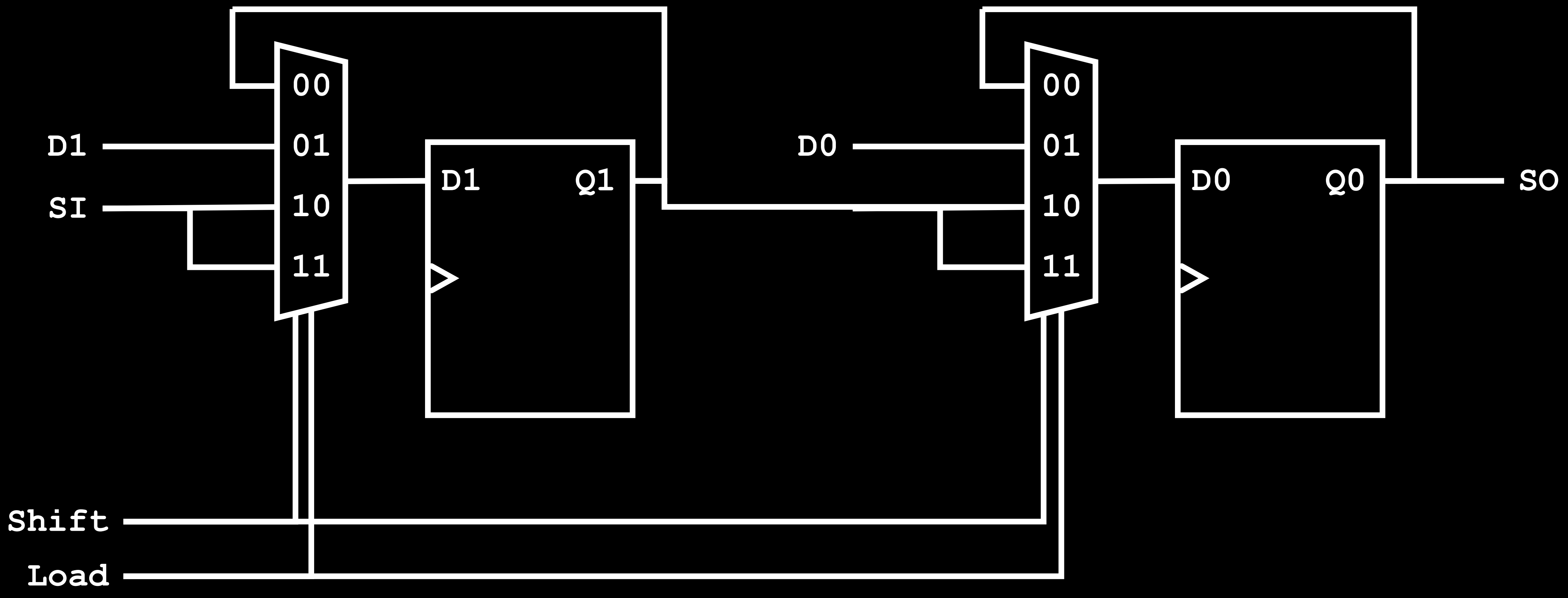
Suppose a 4-bit SIPO register like the one on the previous slide. If all the flip-flops are initially at rest, `Shift` is held at `1`, and the input bit-stream is,

$$SI = \{0, 1, 0, 1, 1, 0, 1\}$$

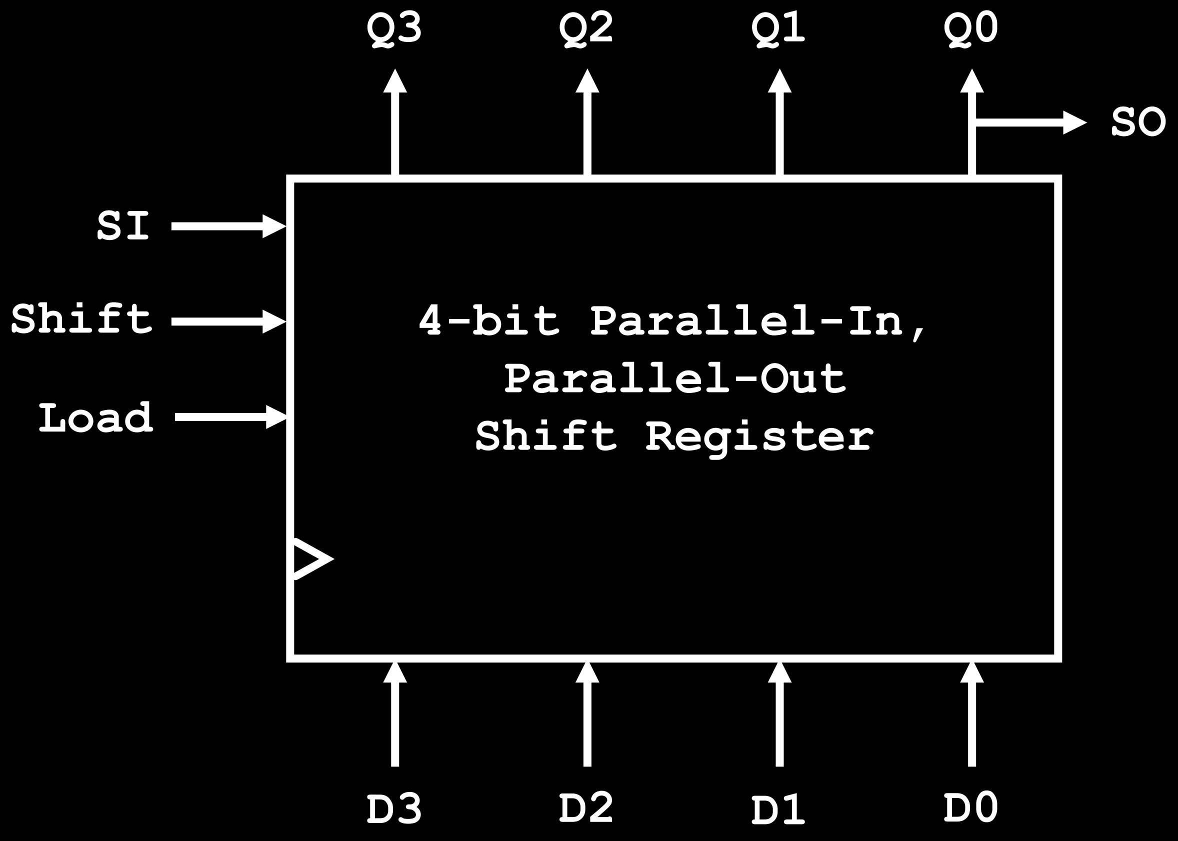
What will be the parallel output at the end of the 5th clock cycle?

$$P = \{1, 1, 0, 1\}$$

# Shift-Registers: Parallel-In, Parallel-Out (PIPO)



# Shift-Registers: Parallel-In, Parallel-Out (PIPO)



# Shift Registers: Parallel-In, Parallel-Out (PIPO)

| INPUTS | | NEXT STATE | | | | ACTION |
|---|---|---|---|---|---|---|
| Shift | Load | $Q_3^+$ | $Q_2^+$ | $Q_1^+$ | $Q_0^+$ | |
| 0 | 0 | $Q_3$ | $Q_2$ | $Q_1$ | $Q_0$ | NO CHANGE |
| 0 | 1 | $D_3$ | $D_2$ | $D_1$ | $D_0$ | LOAD |
| 1 | X | $SI$ | $Q_3$ | $Q_2$ | $Q_1$ | SHIFT RIGHT |

# What questions do you have?

# EXAM QUESTION: Shift Registers

A 4-bit PIPO right shift register is initially at rest. What will the output, Q, be at the end of the following sequence on inputs?

A. Q = {1, 1, 1, 1}

B. Q = {0, 0, 1, 1}

C. Q = {1, 0, 1, 1}

D. Q = {0, 1, 1, 1}

| SI | Shift | Load | D |
|---|---|---|---|
| 0 | 0 | 1 | {0, 1, 0, 1} |
| 0 | 0 | 0 | {0, 1, 1, 0} |
| 1 | 1 | 0 | {0, 1, 1, 0} |
| 1 | 1 | 0 | {1, 0, 0, 1} |
| 0 | 0 | 0 | {1, 0, 1, 1} |
| 1 | 1 | 1 | {1, 1, 0, 1} |
| 0 | 1 | 0 | {1, 1, 1, 1} |

# EXAM QUESTION: Shift Registers

A 4-bit PIPO right shift register is initially at rest. What will the output, Q, be at the end of the following sequence on inputs?

~~A. Q = {1, 1, 1, 1}~~

~~B. Q = {0, 0, 1, 1}~~

~~C. Q = {1, 0, 1, 1}~~

D. Q = {0, 1, 1, 1}

| SI | Shift | Load | D | |
|---|---|---|---|---|
| 0 | 0 | 1 | {0, 1, 0, 1} | Q = {0, 1, 0, 1} |
| 0 | 0 | 0 | {0, 1, 1, 0} | Q = {0, 1, 0, 1} |
| 1 | 1 | 0 | {0, 1, 1, 0} | Q = {1, 0, 1, 0} |
| 1 | 1 | 0 | {1, 0, 0, 1} | Q = {1, 1, 0, 1} |
| 0 | 0 | 0 | {1, 0, 1, 1} | Q = {1, 1, 0, 1} |
| 1 | 1 | 1 | {1, 1, 0, 1} | Q = {1, 1, 1, 0} |
| 0 | 1 | 0 | {1, 1, 1, 1} | Q = {0, 1, 1, 1} |

ECE 2372 | Lecture 305

# Synchronous Binary Counters

Binary Counters | Modern Digital System Design

# Synchronous Binary Counters

What if we wanted to apply this same concept of chaining flip-flops together to count through a predetermined sequence of binary values?

# Synchronous Binary Counters

| A | B | C |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |
| 1 | 1 | 1 |

What do you notice about the relationship between A, B, and C that might be useful to us?

# Synchronous Binary Counters

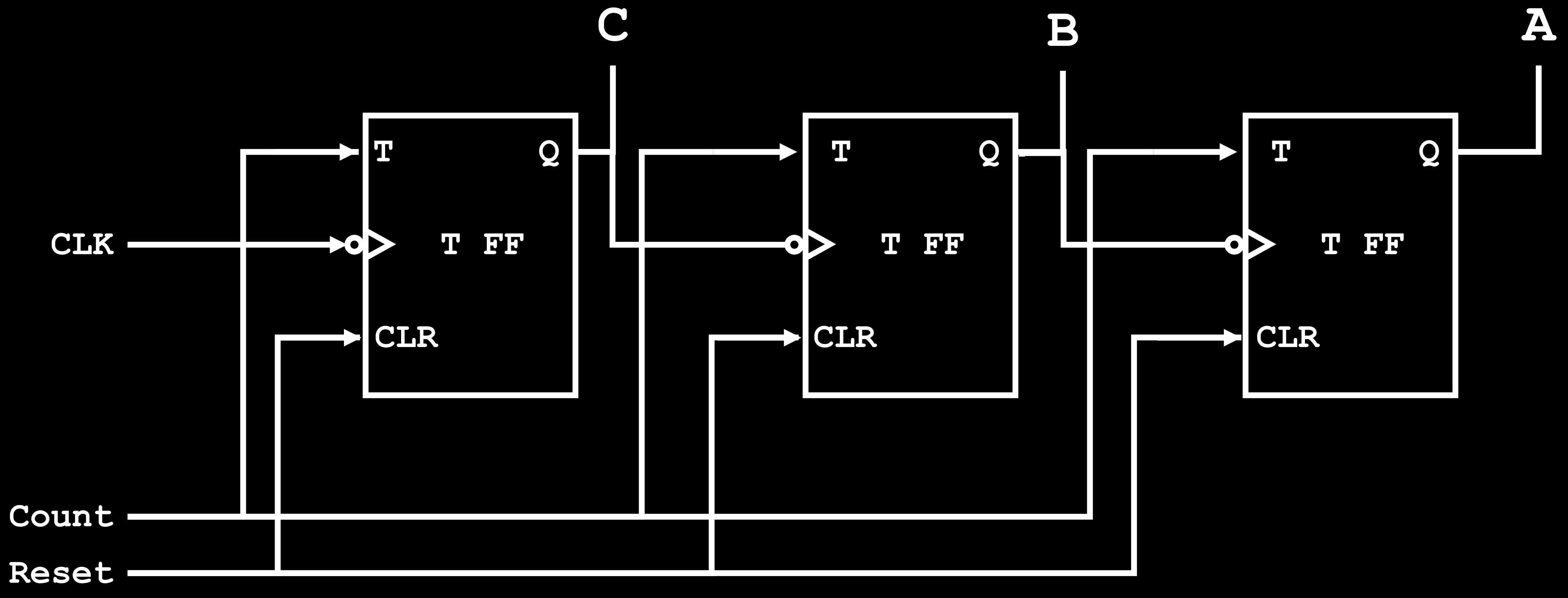
| A | B | C |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |
| 1 | 1 | 1 |

What do you notice about the relationship between A, B, and C that might be useful to us?

Notice that every bit toggles when there is a falling edge on the bit to its right.

What flip-flop likes to toggle?

# Synchronous Binary Counters



# What questions do you have?

# EXAM QUESTION: Synchronous Binary Counter

A 4-bit synchronous binary counter is initially at rest. What will the state of the counter be at the end of the following sequence of inputs?

A. {A, B, C} = 3’b111

B. {A, B, C} = 3’b100

C. {A, B, C} = 3’b101

D. {A, B, C} = 3’b110

| Count | Reset |
|---|---|
| 1 | 0 |
| 1 | 1 |
| 1 | 0 |
| 1 | 0 |
| 0 | 0 |
| 1 | 0 |
| 1 | 0 |

# EXAM QUESTION: Synchronous Binary Counter

A 4-bit synchronous binary counter is initially at rest. What will the state of the counter be at the end of the following sequence of inputs?

~~A. {A, B, C} = 3'b111~~

B. {A, B, C} = 3'b100

~~C. {A, B, C} = 3'b101~~

~~D. {A, B, C} = 3'b110~~

| Count | Reset | |
|---|---|---|
| 1 | 0 | {A,B,C} = 3'b001 |
| 1 | 1 | {A,B,C} = 3'b000 |
| 1 | 0 | {A,B,C} = 3'b001 |
| 1 | 0 | {A,B,C} = 3'b010 |
| 0 | 0 | {A,B,C} = 3'b010 |
| 1 | 0 | {A,B,C} = 3'b011 |
| 1 | 0 | {A,B,C} = 3'b100 |

# Counters for Other Sequences

Binary Counters | Modern Digital System Design

# Counters for Other Sequences

- Usually, our binary counters will count in the traditional way.
- Sometimes, we want counters that can move through an arbitrary sequence.
- To do this, we’ll need to apply some combinational logic design.

# Counters for Other Sequences

Let’s design a 3-bit gray code counter

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

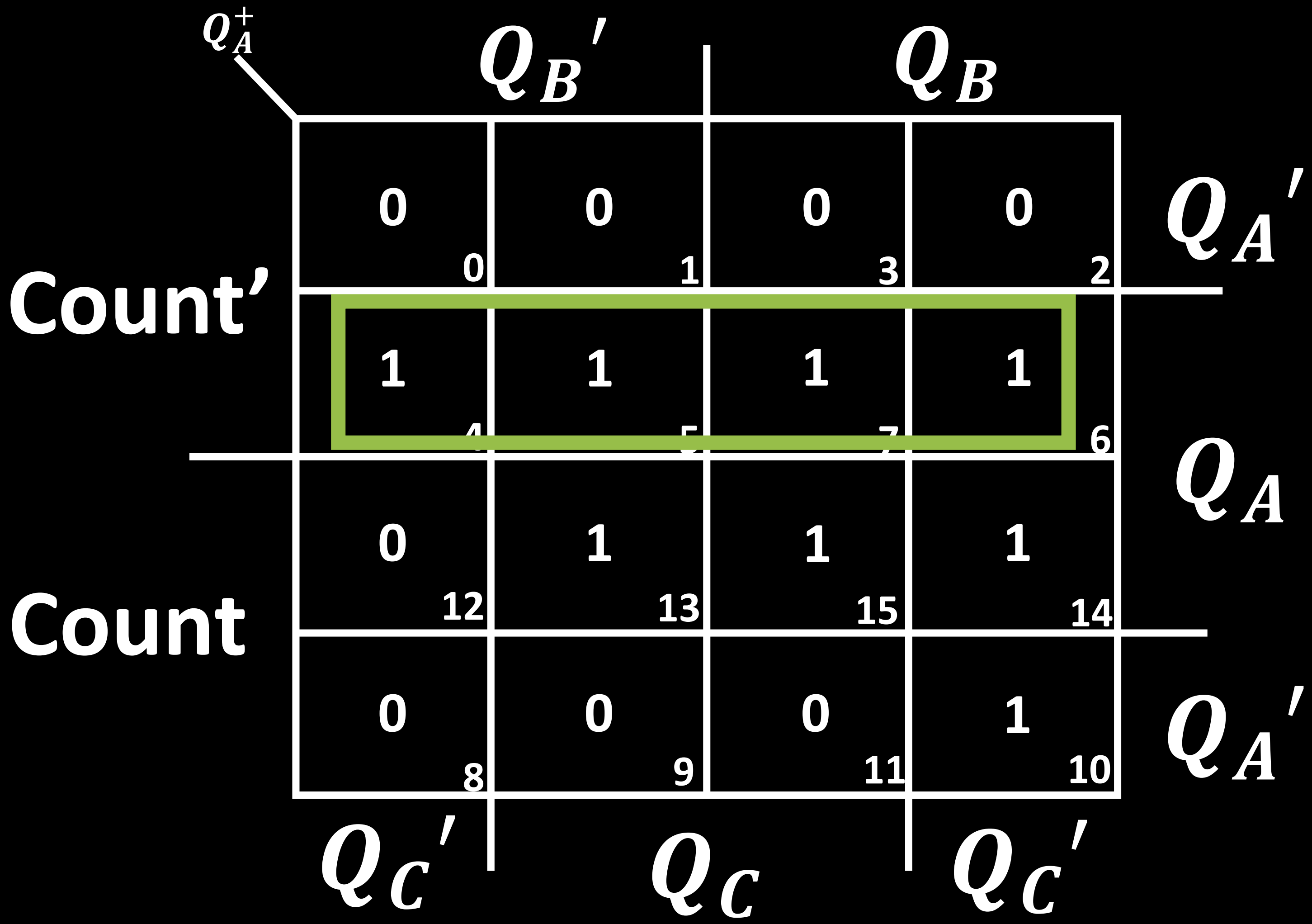
| $Count$ | $Q_A$ | $Q_B$ | $Q_C$ | $Q_A^+$ | $Q_B^+$ | $Q_C^+$ |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 | 1 |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

$$Q_A^+(Count, Q_A, Q_B, Q_C) = \sum_m (4, 5, 6, 7, 10, 13, 14, 15)$$

$$Q_B^+(Count, Q_A, Q_B, Q_C) = \sum_m (2, 3, 6, 7, 9, 10, 11, 14)$$

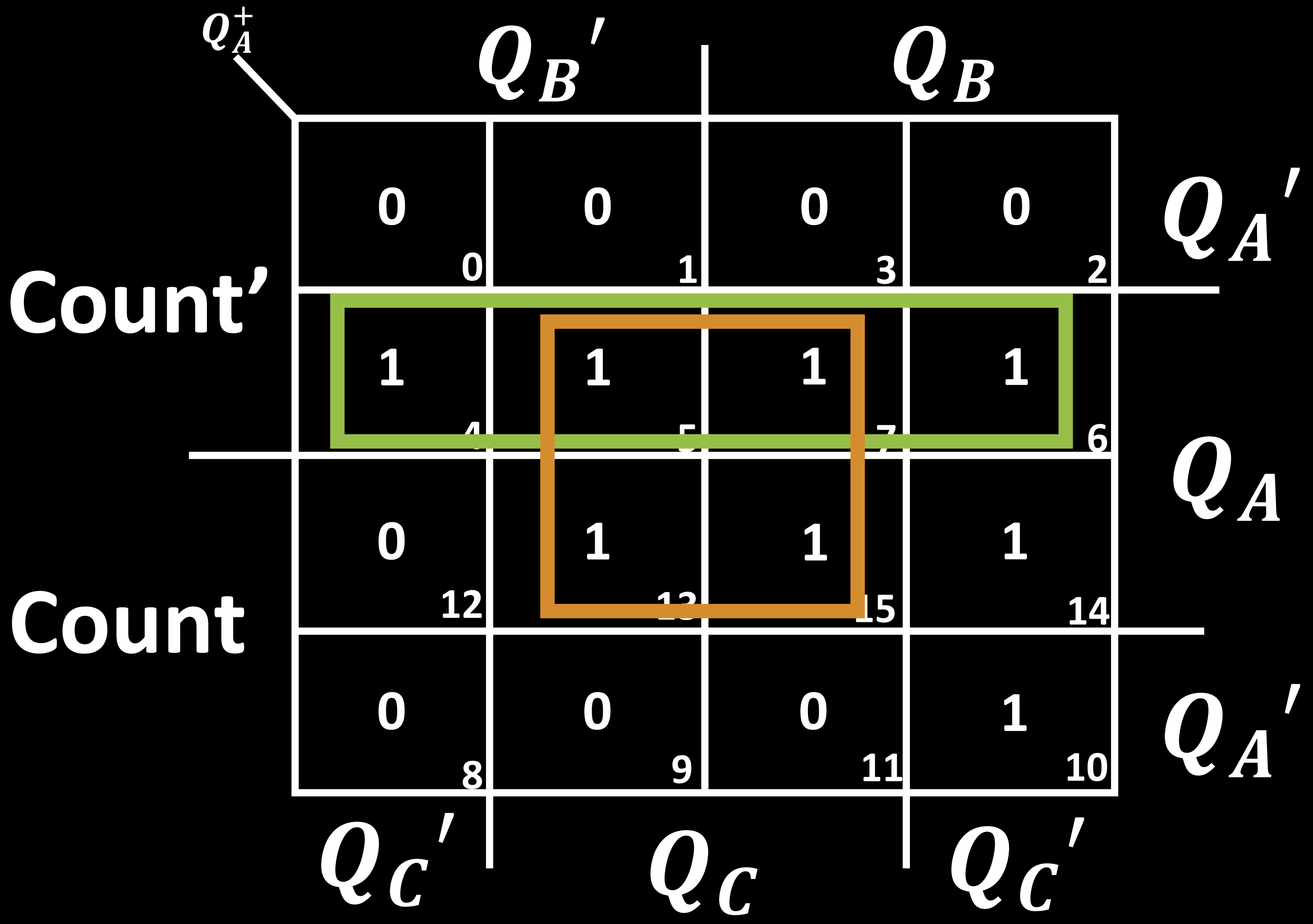
$$Q_C^+(Count, Q_A, Q_B, Q_C) = \sum_m (1, 3, 5, 7, 8, 9, 14, 15)$$

| $Q_A^+$ | $Q_B'$ | | $Q_B$ | | |
|---|---|---|---|---|---|
| Count' | 0<br>0 | 0<br>1 | 0<br>3 | 0<br>2 | $Q_A'$ |
| | 1<br>4 | 1<br>5 | 1<br>7 | 1<br>6 | $Q_A$ |
| Count | 0<br>12 | 1<br>13 | 1<br>15 | 1<br>14 | |
| | 0<br>8 | 0<br>9 | 0<br>11 | 1<br>10 | $Q_A'$ |
| | $Q_C'$ | $Q_C$ | | $Q_C'$ | |



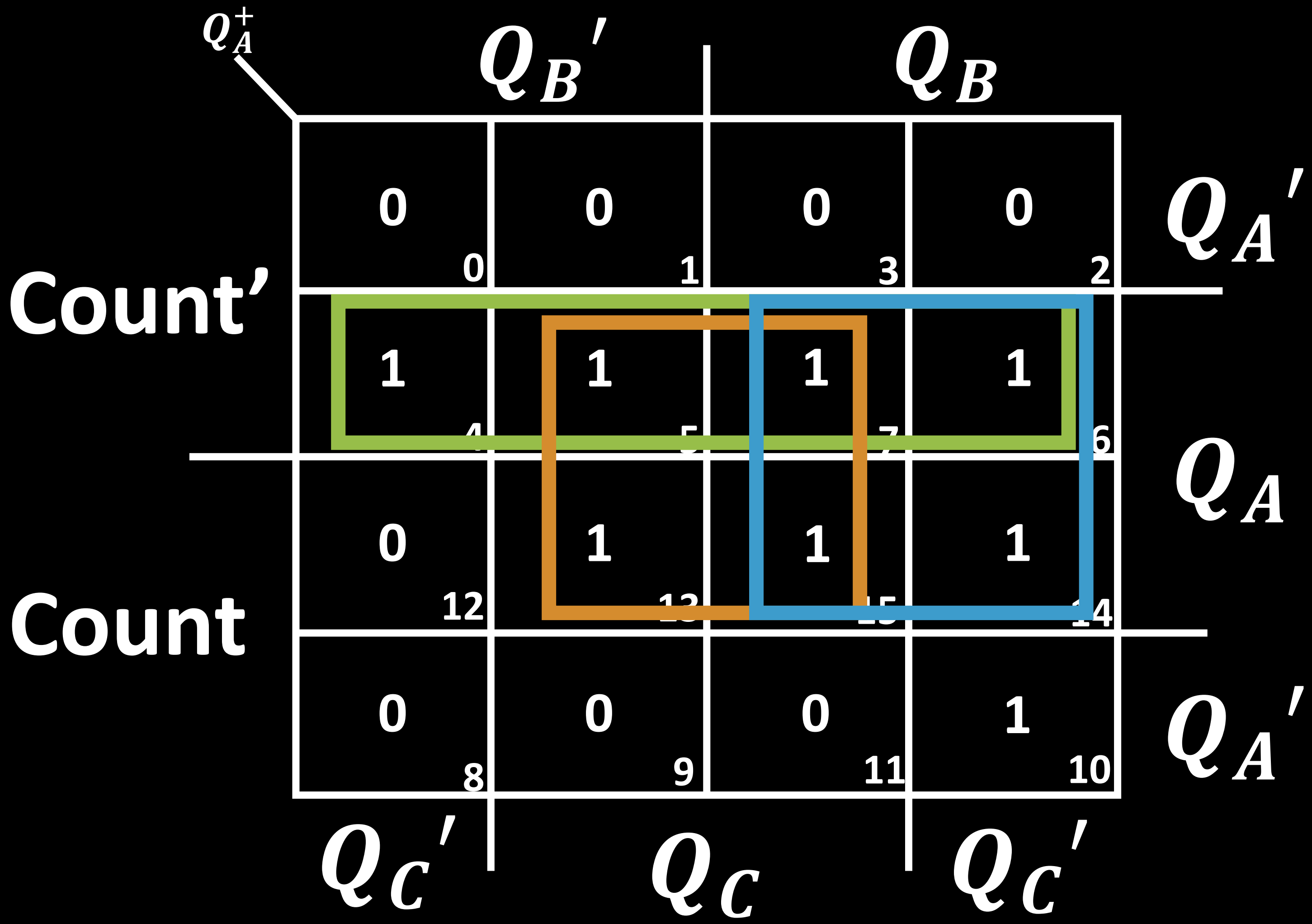
| $Q_A^+$ | | $Q_B'$ | | $Q_B$ | |
|---|---|---|---|---|---|
| | | $Q_C'$ | $Q_C$ | $Q_C$ | $Q_C'$ |
| Count’ | $Q_A'$ | 0<br>0 | 0<br>1 | 0<br>3 | 0<br>2 |
| | $Q_A$ | 1<br>4 | 1<br>5 | 1<br>7 | 1<br>6 |
| Count | $Q_A$ | 0<br>12 | 1<br>13 | 1<br>15 | 1<br>14 |
| | $Q_A'$ | 0<br>8 | 0<br>9 | 0<br>11 | 1<br>10 |

$$Q_A^+ = Count'Q_A +$$



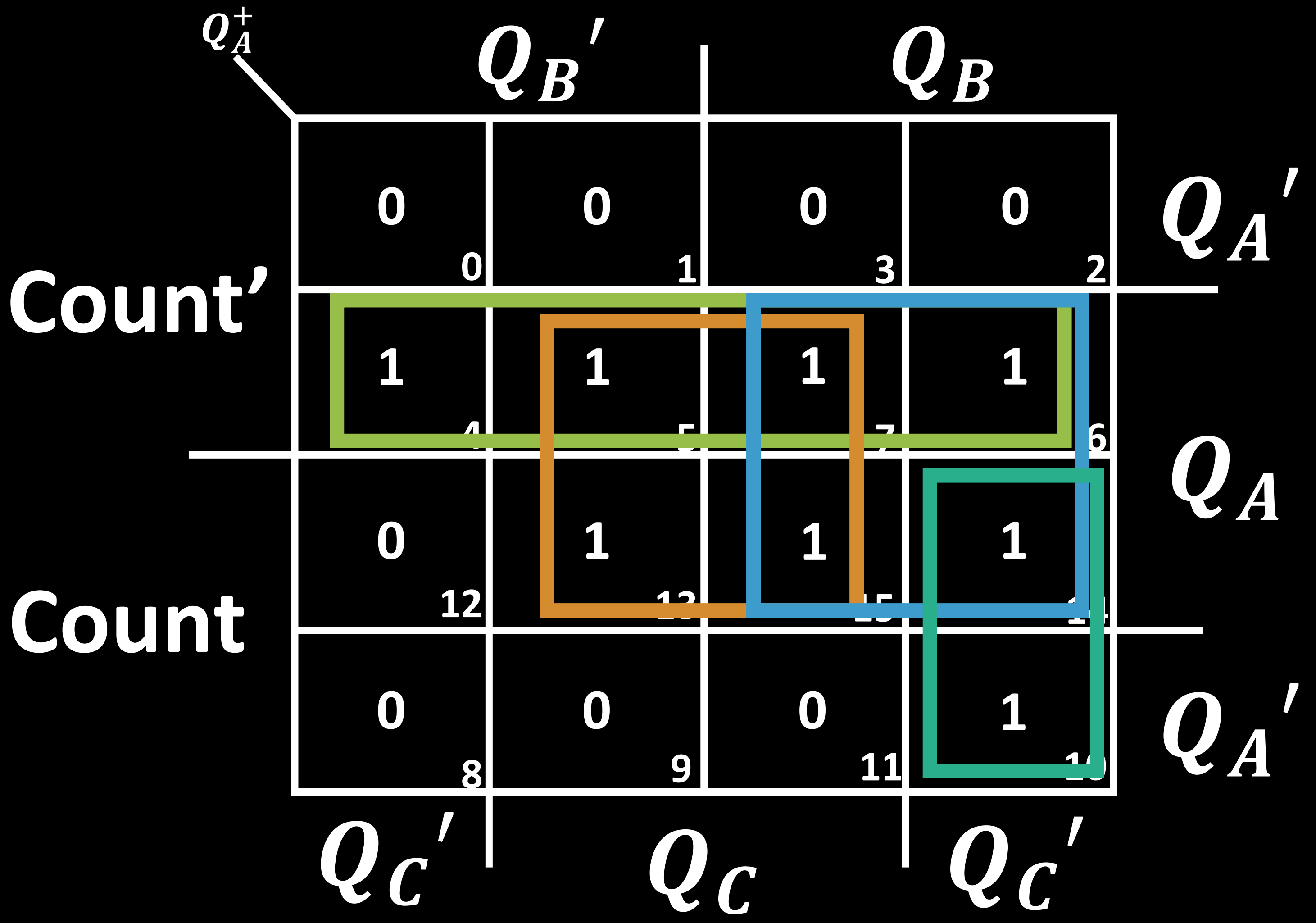
| $Q_A^+$ | $Q_B'$ | | $Q_B$ | | |
|---|---|---|---|---|---|
| | $Q_C'$ | $Q_C$ | | $Q_C'$ | |
| Count' | 0<br>0 | 0<br>1 | 0<br>3 | 0<br>2 | $Q_A'$ |
| | 1<br>4 | 1<br>5 | 1<br>7 | 1<br>6 | $Q_A$ |
| Count | 0<br>12 | 1<br>13 | 1<br>15 | 1<br>14 | |
| | 0<br>8 | 0<br>9 | 0<br>11 | 1<br>10 | $Q_A'$ |

$$Q_A^+ = Count'Q_A + Q_AQ_C +$$



| $Q_A^+$ | $Q_B'$ | | $Q_B$ | | |
|---|---|---|---|---|---|
| Count' | 0<br>0 | 0<br>1 | 0<br>3 | 0<br>2 | $Q_A'$ |
| | 1<br>4 | 1<br>5 | 1<br>7 | 1<br>6 | $Q_A$ |
| Count | 0<br>12 | 1<br>13 | 1<br>15 | 1<br>14 | |
| | 0<br>8 | 0<br>9 | 0<br>11 | 1<br>10 | $Q_A'$ |
| | $Q_C'$ | $Q_C$ | | $Q_C'$ | |

$$Q_A^+ = Count'Q_A + Q_AQ_C + Q_AQ_B +$$



| $Q_A^+$ | $Q_B'$ $Q_C'$ | $Q_B'$ $Q_C$ | $Q_B$ $Q_C$ | $Q_B$ $Q_C'$ |
|---|---|---|---|---|
| Count' $Q_A'$ | 0<br>0 | 0<br>1 | 0<br>3 | 0<br>2 |
| Count' $Q_A$ | 1<br>4 | 1<br>5 | 1<br>7 | 1<br>6 |
| Count $Q_A$ | 0<br>12 | 1<br>13 | 1<br>15 | 1<br>14 |
| Count $Q_A'$ | 0<br>8 | 0<br>9 | 0<br>11 | 1<br>10 |

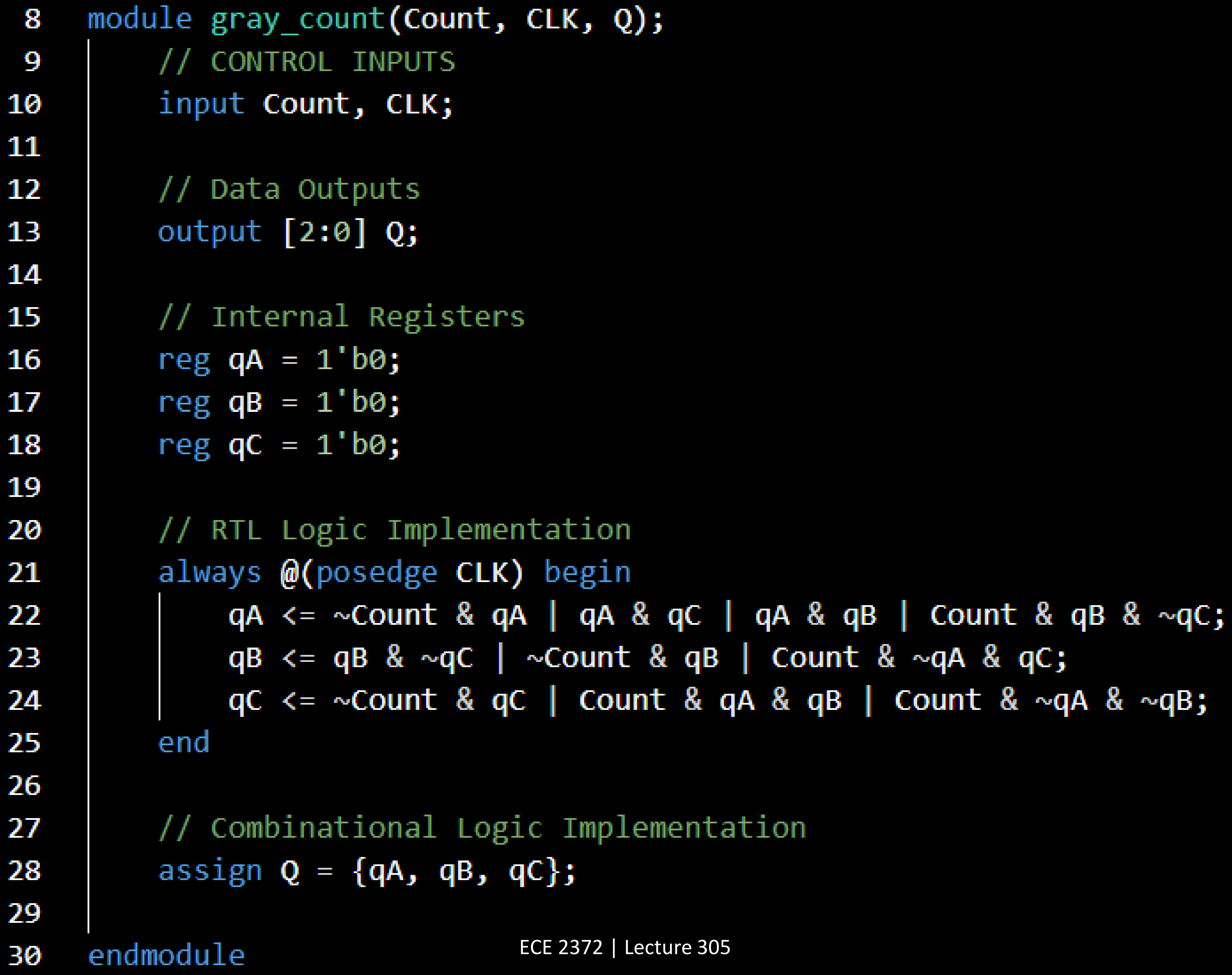
$$Q_A^+ = Count'Q_A + Q_AQ_C + Q_AQ_B + CountQ_BQ_C'$$

Solving the other K-maps yields,

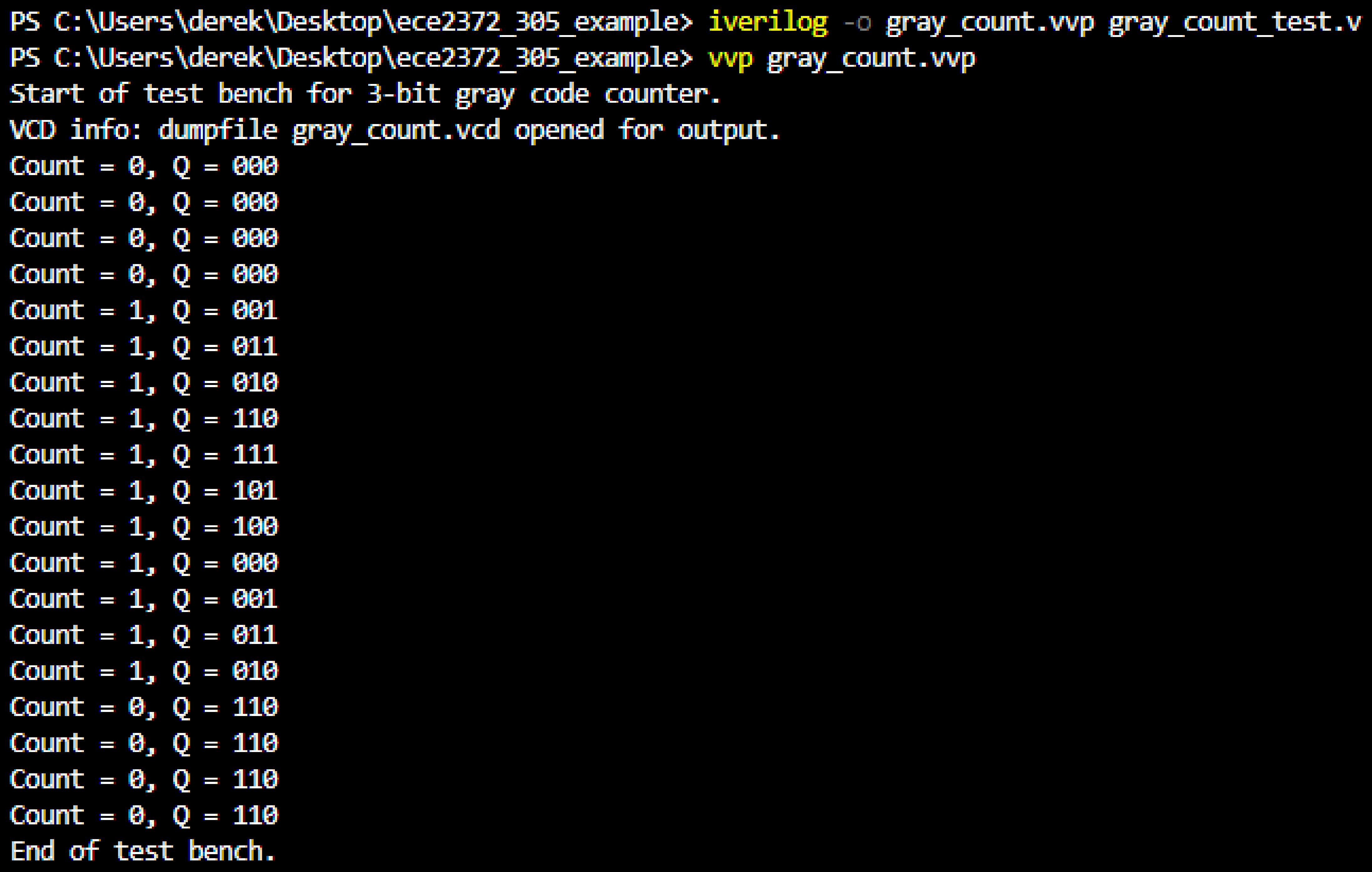
$$Q_A^+ = Count' Q_A + Q_A Q_C + Q_A Q_B + Count Q_B Q_C{}'$$

$$Q_B^+ = Q_B Q_C' + Count' Q_B + Count Q_A' Q_C$$

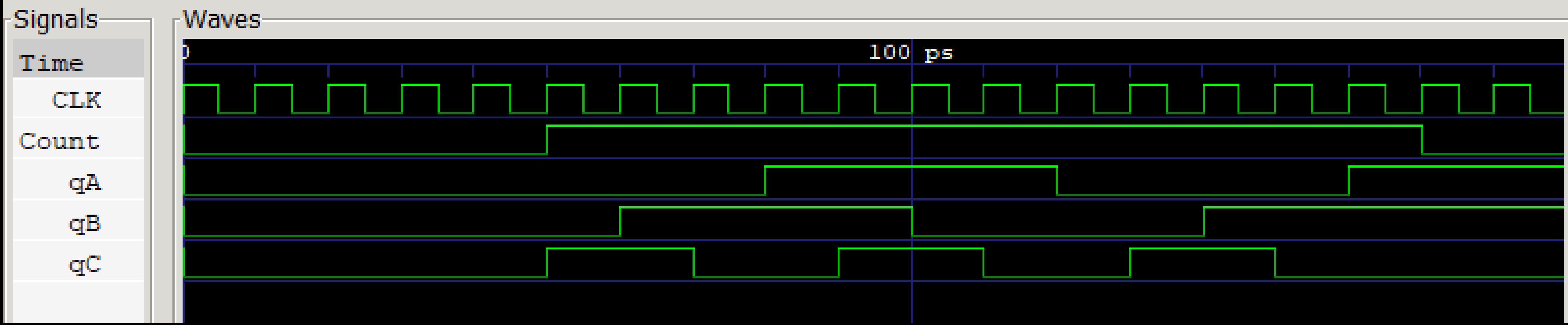
$$Q_C^+ = Count' Q_C + Count Q_A Q_B + Count Q_A' Q_B{}'$$



```verilog
module gray_count(Count, CLK, Q);
    // CONTROL INPUTS
    input Count, CLK;

    // Data Outputs
    output [2:0] Q;

    // Internal Registers
    reg qA = 1'b0;
    reg qB = 1'b0;
    reg qC = 1'b0;

    // RTL Logic Implementation
    always @(posedge CLK) begin
        qA <= ~Count & qA | qA & qC | qA & qB | Count & qB & ~qC;
        qB <= qB & ~qC | ~Count & qB | Count & ~qA & qC;
        qC <= ~Count & qC | Count & qA & qB | Count & ~qA & ~qB;
    end

    // Combinational Logic Implementation
    assign Q = {qA, qB, qC};

endmodule
```



```
PS C:\Users\derek\Desktop\ece2372_305_example> iverilog -o gray_count.vvp gray_count_test.v
PS C:\Users\derek\Desktop\ece2372_305_example> vvp gray_count.vvp
Start of test bench for 3-bit gray code counter.
VCD info: dumpfile gray_count.vcd opened for output.
Count = 0, Q = 000
Count = 0, Q = 000
Count = 0, Q = 000
Count = 0, Q = 000
Count = 1, Q = 001
Count = 1, Q = 011
Count = 1, Q = 010
Count = 1, Q = 110
Count = 1, Q = 111
Count = 1, Q = 101
Count = 1, Q = 100
Count = 1, Q = 000
Count = 1, Q = 001
Count = 1, Q = 011
Count = 1, Q = 010
Count = 0, Q = 110
Count = 0, Q = 110
Count = 0, Q = 110
Count = 0, Q = 110
End of test bench.
```


Signals
Waves
Time
CLK
Count
qA
qB
qC
0
100 ps

# What questions do you have?

# EXAM QUESTION: Arbitrary Binary Counters

What is the next-state equation for flip-flop A in a 2-bit arbitrary sequence binary counter with the given sequence?

*A.* $Q_A^+ = Count'Q_A + Q_A'Q_B' + Count'Q_AQ_B$

*B.* $Q_A^+ = Q_AQ_B + Count'Q_A'Q_B' + CountQ_AQ_B$

*C.* $Q_A^+ = \mathrm{Count}'\mathrm{Q_A} + \mathrm{Q_A}\mathrm{Q_B} + \mathrm{Count}\mathrm{Q}_\mathrm{A}'\mathrm{Q_B}'$

*D.* $Q_A^+ = Count'Q_A' + Q_A'Q_B + CountQ_AQ_B'$

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

# EXAM QUESTION: Arbitrary Binary Counters

What is the next-state equation for flip-flop A in a 2-bit arbitrary sequence binary counter with the given sequence?

~~A. $Q_A^+ = Count'Q_A + Q_A'Q_B' + Count'Q_AQ_B$~~

~~B. $Q_A^+ = Q_AQ_B + Count'Q_A'Q_B' + CountQ_AQ_B$~~

C. $Q_A^+ = \mathrm{Count'Q_A + Q_AQ_B + CountQ_A'Q_B'}$

~~D. $Q_A^+ = Count'Q_A' + Q_A'Q_B + CountQ_AQ_B'$~~

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

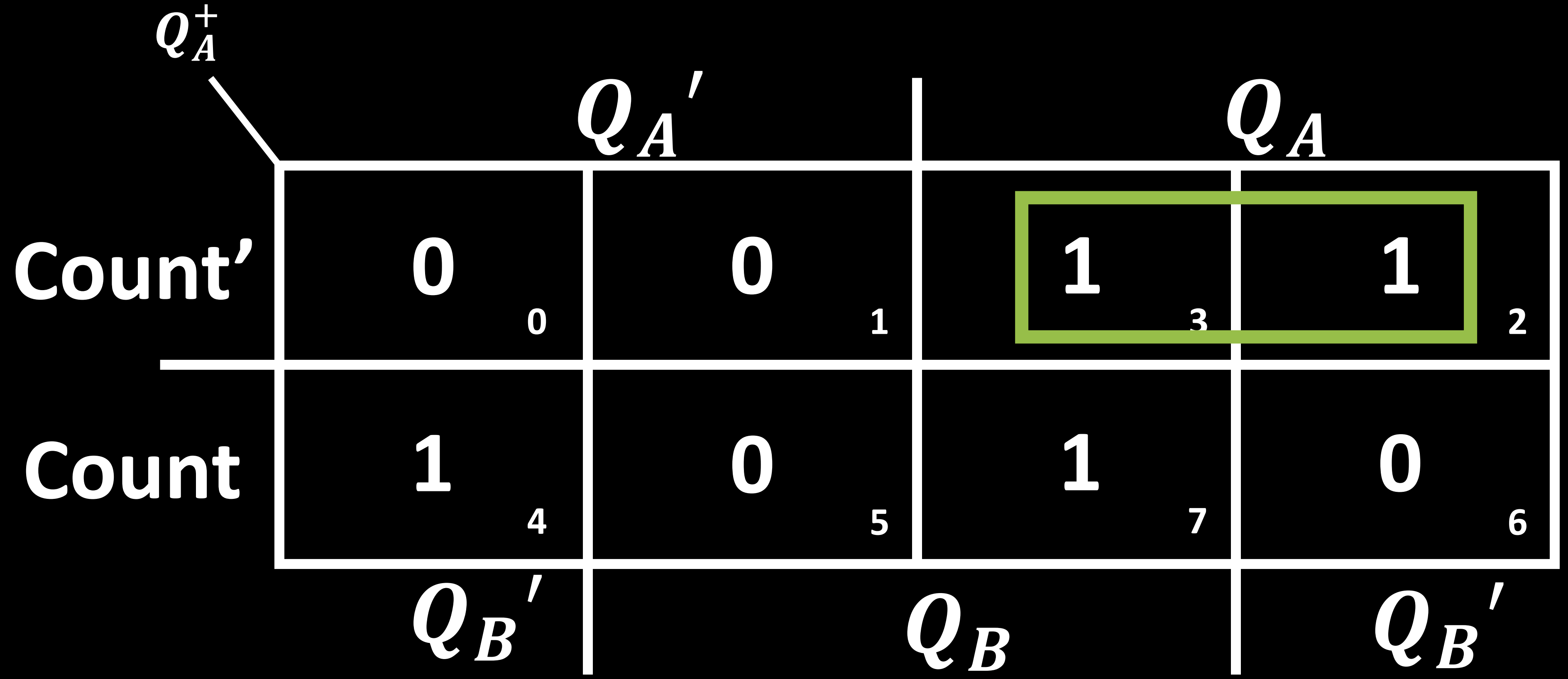
| $Q_A^+$ | $Q_B^+$ |
|---|---|
| 1 | 1 |
| 1 | 0 |
| 0 | 1 |
| 0 | 0 |

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

$$Q_A^+(Count, Q_A, Q_B) = \sum\nolimits_m (2, 3, 4, 7)$$

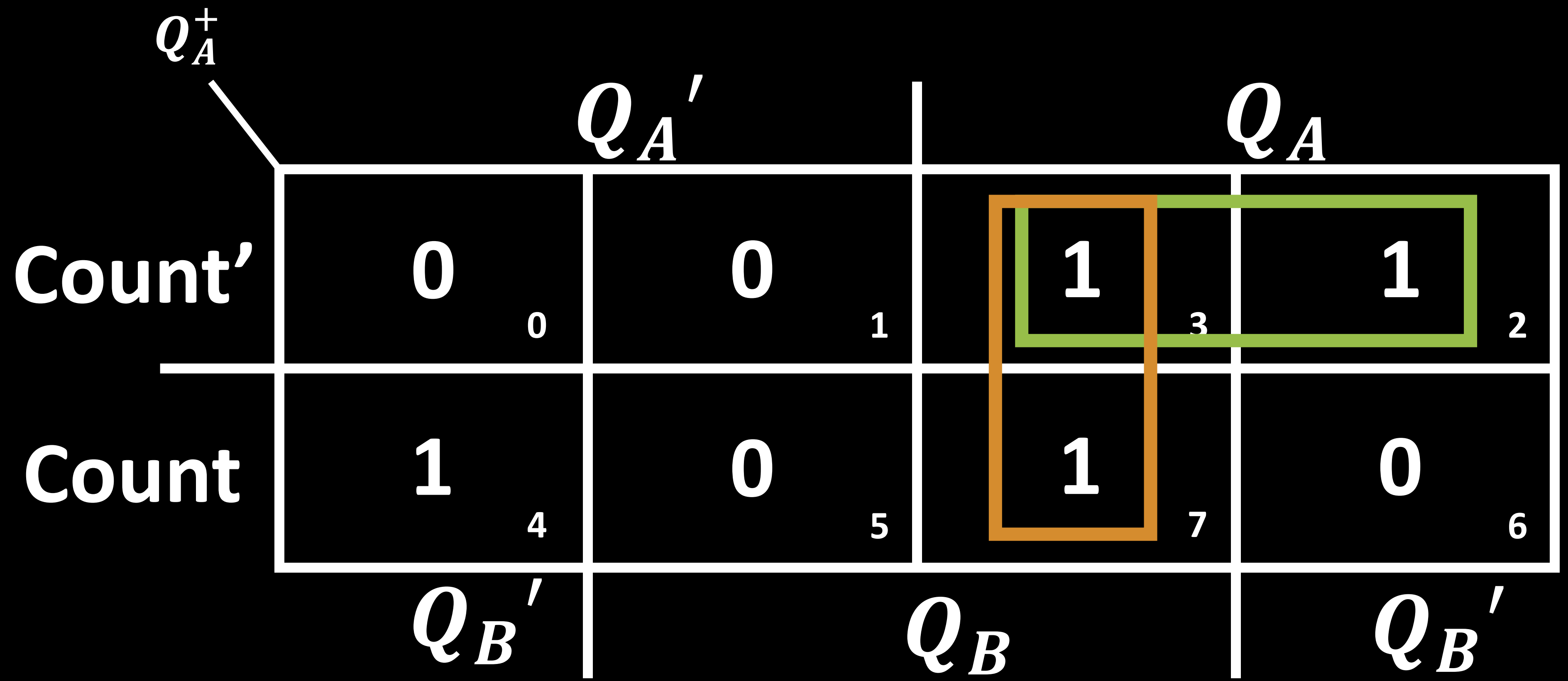
$$Q_A^+(Count, Q_A, Q_B) = \sum\nolimits_m (2, 3, 4, 7)$$

| $Q_A^+$ | $Q_A'$ | | $Q_A$ | |
|---|---|---|---|---|
| Count’ | 0 $_{0}$ | 0 $_{1}$ | 1 $_{3}$ | 1 $_{2}$ |
| Count | 1 $_{4}$ | 0 $_{5}$ | 1 $_{7}$ | 0 $_{6}$ |
| | $Q_B'$ | $Q_B$ | | $Q_B'$ |



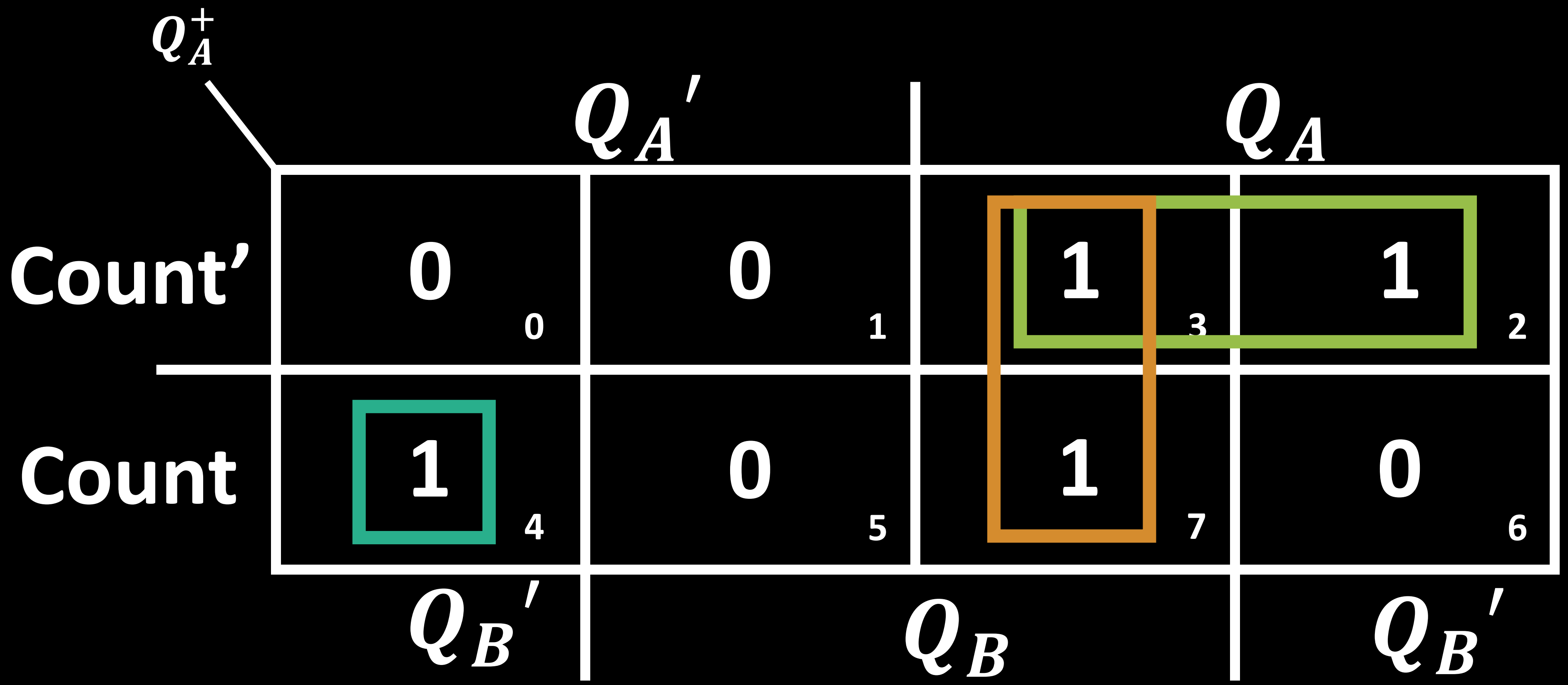
| $Q_A^+$ | $Q_A'$ | | $Q_A$ | |
|---|---|---|---|---|
| Count’ | 0<br>0 | 0<br>1 | 1<br>3 | 1<br>2 |
| Count | 1<br>4 | 0<br>5 | 1<br>7 | 0<br>6 |
| | $Q_B'$ | $Q_B$ | | $Q_B'$ |

$$Q_A^+(Count\,, Q_A, Q_B) = Count'Q_A +$$



| $Q_A^+$ | $Q_A'$ | | $Q_A$ | |
|---|---|---|---|---|
| | $Q_B'$ | $Q_B$ | $Q_B$ | $Q_B'$ |
| Count’ | 0<br>0 | 0<br>1 | 1<br>3 | 1<br>2 |
| Count | 1<br>4 | 0<br>5 | 1<br>7 | 0<br>6 |

$$Q_A^+(Count\,, Q_A, Q_B) = Count'Q_A + Q_AQ_B +$$



| $Q_A^+$ | $Q_A'$ | | $Q_A$ | |
|---|---|---|---|---|
| | $Q_B'$ | $Q_B$ | $Q_B$ | $Q_B'$ |
| Count’ | 0 <br> 0 | 0 <br> 1 | 1 <br> 3 | 1 <br> 2 |
| Count | 1 <br> 4 | 0 <br> 5 | 1 <br> 7 | 0 <br> 6 |

$$Q_A^+(Count\,, Q_A, Q_B) = Count'Q_A + Q_AQ_B + CountQ_A'Q_B{}'$$

# EXAM QUESTION: Arbitrary Binary Counters

What is the next-state equation for flip-flop A in a 2-bit arbitrary sequence binary counter with the given sequence?

~~A. $Q_A^+ = Count' Q_A + Q_A' Q_B' + Count' Q_A Q_B$~~

~~B. $Q_A^+ = Q_A Q_B + Count' Q_A' Q_B' + Count Q_A Q_B$~~

C. $Q_A^+ = \mathrm{Count}' \mathrm{Q_A} + \mathrm{Q_A Q_B} + \mathrm{Count} \mathrm{Q}_A' \mathrm{Q_B}'$

~~D. $Q_A^+ = Count' Q_A' + Q_A' Q_B + Count Q_A Q_B'$~~

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

$$Q_A^+(Count\,, Q_A, Q_B) = Count' Q_A + Q_A Q_B + Count Q_A' Q_B'$$

# Lecture Recap

- Shift Registers
- Synchronous Binary Counters
- Counters for Other Sequences

# What questions do you have?

# Binary Counters

ECE 2372 | Modern Digital System Design | Texas Tech University