# **Experiment 8**

# **Advanced Sequential CircuitDesign**

**Basic Concepts:**

Counters are sequential circuits which "count" through a specific state sequence. They can count up, count down, or count through other fixed sequences. Two distinct types are in common usage:

1. **Ripple Counters**

These circuits are called ripple counters because each edge sensitive transition causes a change in the next flip-flop’s state. The changes “ripple” upward through the chain of flip-flops.

In the ripple counter, clock is connected to the flip-flop clock input on the LSB bit flip-flop. For all other bits, a flip-flop output is connected to the clock input, thus circuit is not truly synchronous. Output change is delayed more for each bit toward the MSB. The ripple counter is shown in Figure 1.

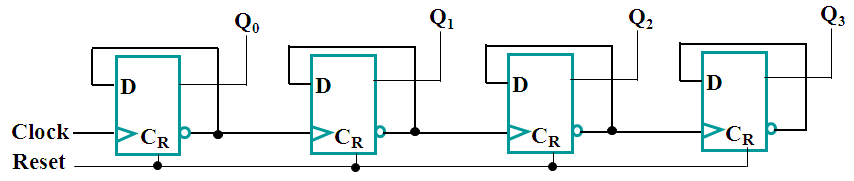


Figure 1: Ripple Counter

1. **Synchronous Counters**

Clock is directly connected to the flip-flop clock inputs. Therefore, all flip-flops change state at the same time.The synchronous counter is shown in Figure 2.

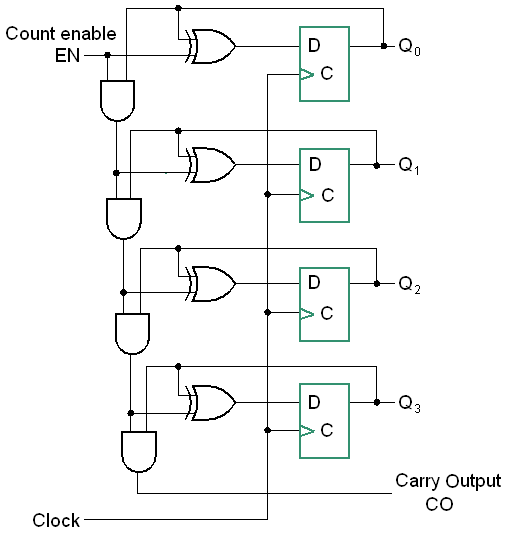


Figure 2: Synchronous Counter

**Procedure**

**Part 1 (Workbench Only):**

Using a 4 bit binary counter with parallel load and an AND gate, construct a BCD counter that repeats for a count from equivalent to decimal 0 to equivalent to decimal 9.

Minimize the number of inputs to the AND gate as much as you can.

CLK

Load

Count

D0

D1

D2

D3

Q0

Q1

Q2

Q3

CO

**Counter**

1

?

**Part 2 (Workbench and Implementation):**

Using two binary counters and logic gates, design and implement a binary counter that repeats for a count from decimal 11 (000010112) to decimal60 (001111002). In addition, you need to add certain input and logic to the counter to initialize it synchronously to “11” when the signal INIT is “1”.

Load

Count

D0

D1

D2

D3

Q0

Q1

Q2

Q3

CO

**Counter**

CLK

CLK

Load

Count

D0

D1

D2

D3

Q4

Q5

Q6

Q7

CO

**Counter**

1

INIT

?

?

?

?

**74163 (Sync 4-bit Binary Counter)**

**4-Bit Counter truth table:**

**\_\_\_ \_\_\_\_**

**CLR | LOAD | ENP | ENT | CLK | A B D C | QA QB QC QD RCO**

**0 | X | X | X | X | X X X X | 0 0 0 0 0**

**1 | 0 | 0 | 0 | POS | X X X X | A B C D \*1**

**1 | 1 | 1 | 1 | POS | X X X X | Count \*1**

**1 | 1 | 1 | X | X | X X X X | QA0 QB0 QC0 QD0 \*1**

**1 | 1 | X | 1 | X | X X X X | QA0 QB0 QC0 QD0 \*1**

**Part 3 (Workbench and implementation):**

Use D flip-flops in IC 7474 and gates to design and implement a binary counter with the following repeated binary sequence: 0, 2, 4, 5, 6, 7.

D

C

*A*

CLK

D

C

D

C

*B*

*C*

*C'*

*B'*

*A'*

**Part 4 (Workbench and implementation):**

Use J-K flip-flops in IC 7476 and gates to design and implement a binary counter with the same repeated binary sequence in part 3.