# 9 Sequential logic circuits – counters

## 9.1 Objectives

The function of the counters is defined, and their general properties are studied. The advantages and disadvantages of implementing asynchronous and synchronous counters with JK flip-flops running in *toggle* configuration are analyzed. Several integrated circuits implementing 4-bit count-up and up-down counters are presented, while focusing on their behavior and specific details.

## 9.2 Theoretical considerations

The counters are sequential logic circuits, which count the number of clock cycles registered on their clock input. They are implemented with flip-flop bistables. The number of flip-flops sets the number of bits used for counting. The number of states defines the *counter capacity*. In an *asynchronous* counter the flip-flops trigger at different moments of time. For *synchronous* counters, the flip-flops trigger simultaneously at moments defined by the clock signal state. Considering the state succession, counters can be *direct*, *inverse* or *reversible*. In general, direct counters count upwards, inverse counters count downwards, and reversible counters are capable of counting both ways. The counting can be *binary*, *decadic* or *modulo p*. The sequence counted by a *modulo p* counter contains *p* ($p<2^n$) *n*-bit values.

### 9.2.1 The asynchronous count-up binary counter

The asynchronous count-up counter has the simplest structure, being based solely on JK flip-flops running in *toggle* configuration (J=K=1). No additional circuitry is required. The counter is asynchronous because, excepting the rank 0 flip-flop, which is connected to the CLK signal, the clock input for any higher rank flip-flop is connected to the output of the inferior rank flip-flop. Figure 9. 1 highlights the implementation with JK flip-flops of a 3-bit asynchronous count-up binary counter. The counter size can be extended by adding more flip-flops. **Note**: The active-low $\overline{RST}$ signal resets the counter to 0, asynchronously (the effect takes place immediately).

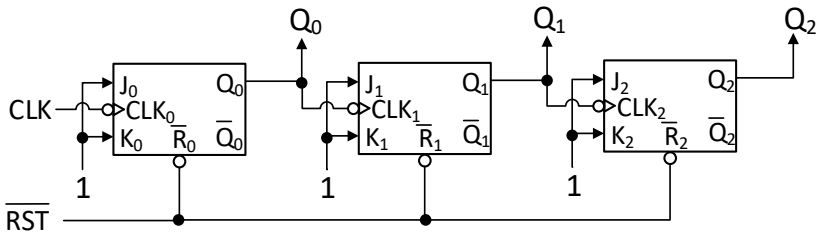



Figure 9. 1 3-bit asynchronous count-up binary counter implemented using JK flip-flops

The timing diagram in Figure 9. 2 shows the various states registered on the outputs of the asynchronous count-up binary counter, implemented with 3 JK flip-flops. The counting loops is 0÷7. Each falling edge of the CLK advances the counting by +1. Also,

a falling edge on the output state of a flip-flop triggers the toggling of the superior rank flip-flop. The transition from $111_2$ to $000_2$ is the slowest, because all flip-flops must toggle in sequential order. The drawback: considering that CLK cycle period must be higher than the slowest transition, increasing the size of the counter limits the working frequency.

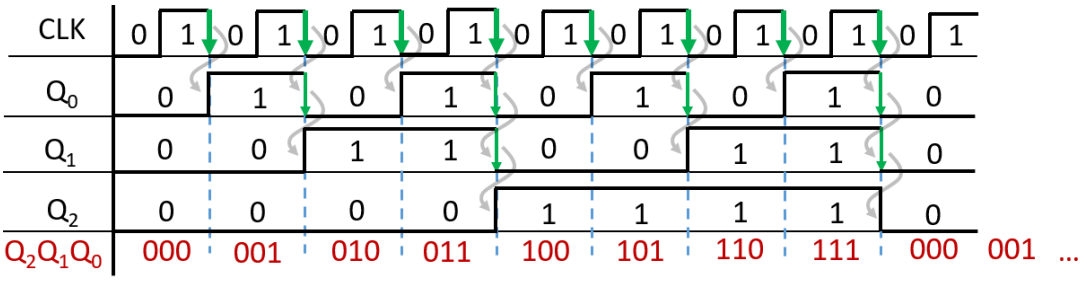



Figure 9. 2 The timing diagram for a 3-bit asynchronous count-up binary counter

Analyzing the waveforms for outputs $Q_i$ in Figure 9. 2, it can be noticed that each flip-flop generates a clock period, double the amount of clock period generated at the inferior rank. Consequently, the JK flip-flops running in *toggle* configuration are *frequency dividers* with a factor of 2. Compared to CLK signal, $Q_0$, $Q_1$ and $Q_2$ apply a division factor of 2, 4, and 8, respectively.

### 9.2.2 The synchronous count-up serial binary counter

The frequency limitation problem encountered for asynchronous counters is partially solved by synchronous serial counters. In this case, the JK flip-flops trigger synchronously, being connected to the same CLK signal, as in Figure 9. 3. The J and K inputs are connected, however the boolean function controlling their values can be extracted by analyzing the behavior of each bit during the counting loop. Table 9. 1 highlights the values registered on the outputs for the loop $0 \div 15$.

Table 9. 1 Outputs of a 4-bit counter

| Nr. | $Q_3$ | $Q_2$ | $Q_1$ | $Q_0$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 2 | 0 | 0 | 1 | 0 |
| 3 | 0 | 0 | 1 | 1 |
| 4 | 0 | 1 | 0 | 0 |
| 5 | 0 | 1 | 0 | 1 |
| 6 | 0 | 1 | 1 | 0 |
| 7 | 0 | 1 | 1 | 1 |
| 8 | 1 | 0 | 0 | 0 |
| 9 | 1 | 0 | 0 | 1 |
| 10 | 1 | 0 | 1 | 0 |
| 11 | 1 | 0 | 1 | 1 |
| 12 | 1 | 1 | 0 | 0 |
| 13 | 1 | 1 | 0 | 1 |
| 14 | 1 | 1 | 1 | 0 |
| 15 | 1 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 | 0 |

Analyzing each column $Q_i$, the following associations can be noticed: $Q_0$ toggles each clock cycle; $Q_1$ toggles when $Q_0$=1; $Q_2$ toggles when $Q_1$=1 and $Q_0$=1; $Q_3$ toggles when $Q_2$=1 and $Q_1$=1 and $Q_0$=1. By analogy, a flip-flop should toggle when all flip-flops of inferior rank have a value of 1. Consequently, the following boolean expression are true for inputs $J_i$ and $K_i$:

$$\begin{aligned} J_0 = K_0 = 1 \\ J_1 = K_1 = Q_0 \\ J_2 = K_2 = Q_1 \cdot Q_0 \\ J_3 = K_3 = Q_2 \cdot Q_1 \cdot Q_0 \end{aligned} \quad (9.\ 1)$$

In a serial configuration, 2-input AND gates are connected serially, to implement the operations required, because they have a simple structure. The drawback with the serial connection is the increase of propagation delay with the increase of flip-flop number, when extending the counter size. The consequence is a partial limitation of the clock frequency.

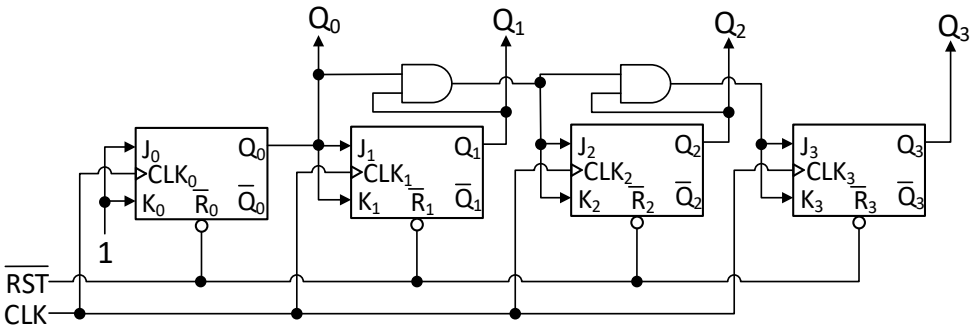



Figure 9. 3 4-bit synchronous count-up serial counter implemented using JK flip-flops

### 9.2.3 The synchronous count-up parallel binary counter

To remove the limitations imposed by the serial connection, a solution is to use multiple-input AND gates, which implement the expected boolean expressions simultaneously, as in Figure 9. 4. The propagation delay is dictated by the slowest AND gate. In matters of complexity, the slowest gate has the highest number of inputs, however its time delay is significantly lower than 2-input AND gates connected serially. Consequently, the working frequency is less affected when increasing the counter size.

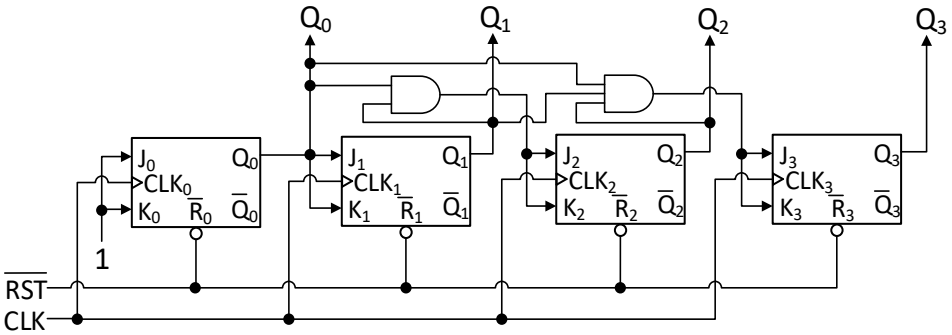
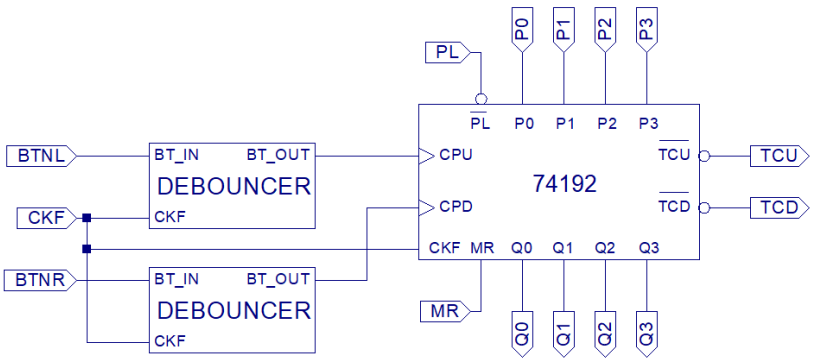



Figure 9. 4 4-bit synchronous count-up parallel counter implemented using JK flip-flops

### 9.2.4 The synchronous up-down decade counter 74192

Circuit TTL 74192 in Figure 9. 5 is a 4-bit up-down counter, running the counting loop $0 \div 9$, associated with the decimal digits. The counter has an asynchronous reset command MR (Master Reset). The reset sets the counter to 0. A second asynchronous command $\overline{PL}$ (Parallel Load) sets the counter to the 4-bit value registered on inputs $P_{0:3}$. The reset has priority over the load. The load command $\overline{PL}$ is active low. The current value of the counter can be read on outputs $Q_{3:0}$. The effect of the signals is presented in the function table from Figure 9. 5.

To enable the count-up, the clock signal must be connected to $CP_U$ (Count Up), and $CP_D$ (Count Down) must be connected to 1. To enable the count-down, the clock signal must be connected to $CP_D$, and $CP_U$ must be connected to 1.




| MR | $\overline{PL}$ | $CP_U$ | $CP_D$ | $Q_{3:0}^{t+1}$ |
|---|---|---|---|---|
| 1 | X | X | X | $(0000)_2$ |
| 0 | 0 | X | X | $P_{3:0}$ |
| 0 | 1 | ↑ | 1 | $Q_{3:0}^{t}+1$ |
| 0 | 1 | 1 | ↑ | $Q_{3:0}^{t}-1$ |
| 0 | 1 | 1 | 1 | $Q_{3:0}^{t}$ |

Figure 9. 5 Testing circuit for counter 74192 (left) and the function table (right)

At count-up, the active low output flag $\overline{TC}_U$ (Terminal Count Up) signals the upper bound (by having $\overline{TC}_U = 0$), when reaching value $Q_{3:0}$=1001. Similarly, at count-down, the active low output flag $\overline{TC}_D$ (Terminal Count Down) signals the lower bound (by having $\overline{TC}_D = 0$), when reaching value $Q_{3:0}$=0000. **Note**: The signaling takes place in the second part of the clock cycle, as shown in Figure 9. 6. The expressions for $\overline{TC}_U$ and $\overline{TC}_D$ are:

$$\overline{TC}_U = \overline{Q_3 \cdot Q_0 \cdot \overline{CP}_U}$$
$$\overline{TC}_D = \overline{\overline{Q_3} \cdot \overline{Q_2} \cdot \overline{Q_1} \cdot \overline{Q_0} \cdot \overline{CP}_D} \quad (9.\ 2)$$

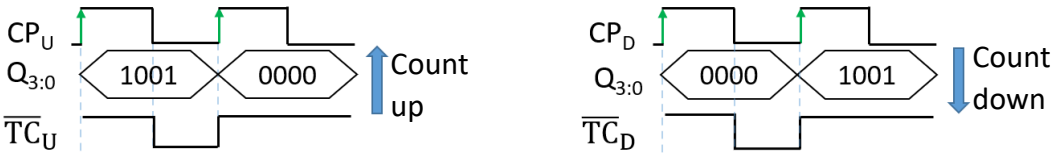
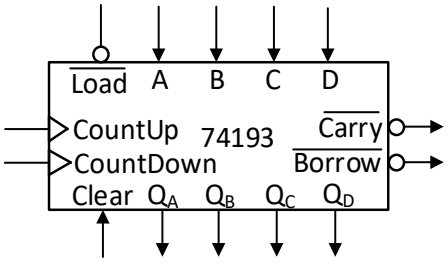



Figure 9. 6 Signaling the upper and lower bounds for 74192

The 74192 counter has *synchronous self-correction* mechanism: in case the current value is outside $0 \div 9$, the state will return to the counting loop after several clock cycles.

### 9.2.5 The synchronous up-down binary counter 74193

Circuit TTL 74193 in Figure 9. 7 is the binary version of counter 74192. It counts the complete 4-bit counting loop in range $0 \div 15$. The inputs and outputs have different names, but their functionality is the same as for 74192.




| Clear | $\overline{Load}$ | CountUp | CountDown | $Q_{D:A}^{t+1}$ |
|---|---|---|---|---|
| 1 | X | X | X | $(0000)_2$ |
| 0 | 0 | X | X | $(DCBA)_2$ |
| 0 | 1 | ↑ | 1 | $Q_{D:A}^{t}+1$ |
| 0 | 1 | 1 | ↑ | $Q_{D:A}^{t}-1$ |
| 0 | 1 | 1 | 1 | $Q_{D:A}^{t}$ |

Figure 9. 7 The symbol for counter 74193 (left) and the function table (right)

The signals Clear and $\overline{Load}$ represent the asynchronous reset and load with the binary value $DCBA_2$, respectively. The inputs CountUp and CountDown are the clock signal inputs. When connecting the clock to one of them, the other input must be connected to

1. The current value of the counter can be read on outputs $Q_{D:A}$. According to the timing diagrams in Figure 9. 8, the output $\overline{\text{Carry}}$ signals the maximum value 15 when counting up, and the output $\overline{\text{Borrow}}$ signals the minimum value 0 when counting down. The signaling takes place only when the clock signal is 0. Accordingly, the boolean expressions for $\overline{\text{Carry}}$ and $\overline{\text{Borrow}}$ are:

$$\begin{aligned}\overline{\text{Carry}} &= \overline{Q_3 \cdot Q_2 \cdot Q_1 \cdot Q_0 \cdot \overline{\text{CountUp}}} \\ \overline{\text{Borrow}} &= \overline{\overline{Q_3} \cdot \overline{Q_2} \cdot \overline{Q_1} \cdot \overline{Q_0} \cdot \overline{\text{CountDown}}}\end{aligned} \tag{9. 3}$$

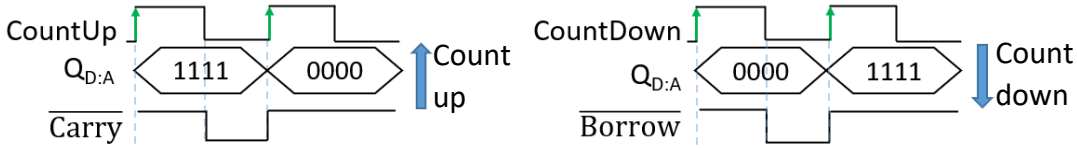
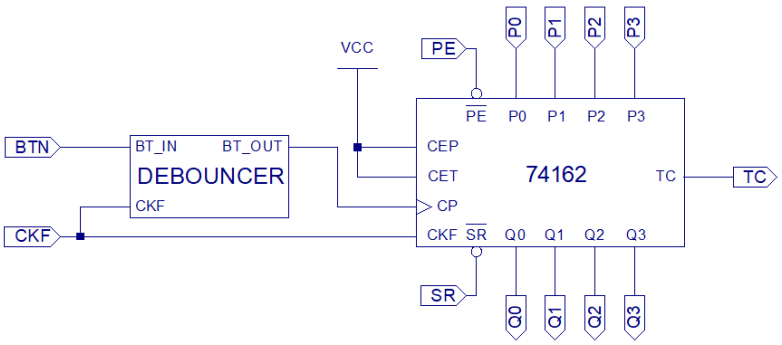



Figure 9. 8 Signaling the upper and lower bounds for 74193

### 9.2.6 The synchronous count-up decade counter 74162

Counter 74162 (Figure 9. 9) counts the 4-bit loop 0÷9. The current value of the counter can be read on outputs $Q_{3:0}$. The clock signal is applied on input CP (Clock Pulse). The active low command $\overline{\text{SR}}$ (Synchronous Reset) resets the counter synchronously (on the rising edge of the clock), setting its value to 0. The active low command $\overline{\text{PE}}$ (Parallel Enable) loads the counter synchronously with the 4-bit value on inputs $P_{3:0}$. According to function table in Figure 9. 9, the command $\overline{\text{SR}}$ has priority over $\overline{\text{PE}}$. Both have priority over counting.




| $\overline{\text{SR}}$ | $\overline{\text{PE}}$ | CET | CEP | CP | $Q_{3:0}^{t+1}$ |
|---|---|---|---|---|---|
| 0 | X | X | X | ↑ | $(0000)_2$ |
| 1 | 0 | X | X | ↑ | $P_{3:0}$ |
| 1 | 1 | 1 | 1 | ↑ | $Q_{3:0}^{t}+1$ |
| 1 | 1 | 0 | X | X | $Q_{3:0}^{t}$ |
| 1 | 1 | X | 0 | X | $Q_{3:0}^{t}$ |

Figure 9. 9 Testing circuit for counter 74162 (left) and the function table (right)

To activate the counting, signals CEP and CET must be set (CEP=CET=1), and synchronous commands $\overline{\text{SR}}$, $\overline{\text{PE}}$ must be disabled ($\overline{\text{SR}} = \overline{\text{PE}} = 1$). The output TC (Terminal Count) signals when reaching the upper bound $1001_2$, by enabling TC=1 (Figure 9. 10). The output TC is functional only when CET=1, according to the expression:

$$\text{TC} = Q_3 \cdot \overline{Q_2} \cdot \overline{Q_1} \cdot Q_0 \cdot \text{CET} \tag{9. 4}$$

Considering that values outside the counting loop can be loaded, the counter has *synchronous self-correction* mechanism, which brings the counter state back to the loop, after maximum 2 clock cycles.

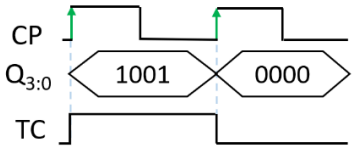
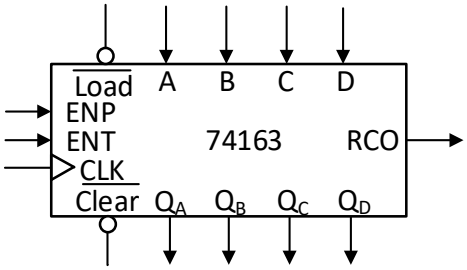



Figure 9. 10 The output TC signals the presence of the upper bound for 74162

### 9.2.7 The synchronous count-up binary counter 74163

The counter 74163 is similar to the decade counter 74162 presented in the previous section, except that the counting loop is $0 \div 15$. According to its symbol, highlighted in Figure 9. 11, the terminals of the counter have different names, but their functionality is preserved, as described in the table below.




| $\overline{\text{Clear}}$ | $\overline{\text{Load}}$ | ENP | ENT | CLK | $Q_{D:A}^{t+1}$ |
|---|---|---|---|---|---|
| 0 | X | X | X | ↑ | $(0000)_2$ |
| 1 | 0 | X | X | ↑ | $(DCBA)_2$ |
| 1 | 1 | 1 | 1 | ↑ | $Q_{D:A}^{t}+1$ |
| 1 | 1 | 0 | X | X | $Q_{D:A}^{t}$ |
| 1 | 1 | X | 0 | X | $Q_{D:A}^{t}$ |

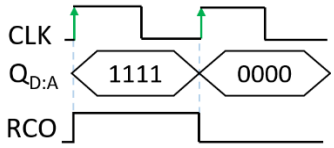



Figure 9. 11 Counter 74163, the function table and the signaling of the upper bound

The active low commands $\overline{\text{Clear}}$ and $\overline{\text{Load}}$, reset and load the binary value $DCBA_2$, respectively. Both have priority over counting. To enable the counting, ENP and ENT must be activated (ENP=ENT=1). The current value of the counter can be read on outputs $Q_{D:A}$. The output RCO (Ripple Carry Output) signals when reaching 15, the maximum value. The behavior is described in the timing diagram from Figure 9. 11. The expression for RCO is:

$$\text{RCO} = Q_D \cdot Q_C \cdot Q_B \cdot Q_A \cdot \text{ENT} \tag{9. 5}$$

## 9.3 Assignments

1. Implement on the board counter 74192. Test the reset and load commands, the up-down counting, the self-correction, and the signaling of the boundaries.
2. Implement on the board counter 74162. Test the reset and load commands, the counting, the self-correction, and the signaling of the upper bound.
3. Implement in Logisim counter 74193. Test the reset and load commands, the up-down counting, and the signaling of the boundaries.
4. Implement in Logisim counter 74163. Test the reset and load commands, the counting, and the signaling of the upper bound.
5. Implement in Logisim a 3-bit asynchronous count-up binary counter, using 7473 JK flip-flops. Test the counting loop.
6. Implement in Logisim a 4-bit synchronous count-up serial binary counter, using 7473 JK flip-flops. Test the counting loop.