CHAPTER 2

COUNTER

Introduction

- Counter
  - A counter is a sequential logic circuit consisting of a set of flip-flops which can go through a sequence of states.

Counters

- Asynchronous counter
  - Also known as ripple counter
  - The first flip-flop is driven by external clock while the successive flip-flops by the output of preceding flip-flop

Synchronous counters

- Clock pulse is applied to each FF simultaneously

Counters are formed by connecting flip-flops together

Types of counter are;

- Asynchronous
  - Clock pulse is applied to each FF simultaneously
  - The output of one FF drives the input of the next one

High Speed

Slow speed
Introduction (continue)

- Synchronous
  - All flip-flops are simultaneously driven by common clock

- Each type of counter are classified by:
  - Sequence i.e up or down
  - Number of states i.e 2-bit will have 4 states ($2^N$)
  - Number of flip-flops i.e same as number of bits

Asynchronous counter

- Also known as ripple counter. Ripple counters are the simplest type of binary counters because they require the fewest components to produce a given counting operation.
- Each FF output drives the CLK input of the next FF.
- FFs do not change states in exact synchronism with the applied clock pulses.
- There is delay between the responses of successive FFs.
- It is also often referred to as a ripple counter due to the way the FFs respond one after another in a kind of rippling effect.

Up counter and down counter for negative edge clock
Up counter and down counter for positive edge clock

3-bit binary down counter

3-bit binary up counter

Asynchronous Counter Operation

- For example, 2-bit asynchronous binary counter using J-K FF
- CLK is only connected to 1st FF, LSB FF
- The 2nd FF clock is driven by \( Q_0 \) of 1st FF
- Both FF input are always HIGH
- \( Q_0 \) changes state at the positive-edge clock
- \( Q_1 \) change at the positive-edge of the \( Q_0 \)
- Note that the two FFs do not triggered at the same time because clock and \( Q_0 \) transitions do not occur at the same time

Asynchronous Counter Operation (continue..)

- Timing diagram for 2-bit asynchronous binary counter
- Four clock pulses are applied, assume initially all LOW
- \( Q_0 \) (LSB) is always toggle at positive-edge clock (J and K are HIGH)
- \( Q_0 \) is reciprocal of \( Q_0 \)
- \( Q_1 \) (MSB) is toggle at positive-edge of \( Q_0 \)
- At 4th clock pulse, the counter is recycle to its original state (both FF are LOW)

Asynchronous Counter Operation (continue..)

- Binary state sequence for 2-bit asynchronous binary counter
- The counter is in up sequence (\( Q_1 \) is MSB, \( Q_0 \) is LSB)
- Count from 0 to 3 in binary sequence
- The term ‘recycle’ refers to the transition from final state to original state
- Therefore, 2-bit asynchronous counter has four state and consists of two FF
A 3-bit Asynchronous Binary Counter

- Draw 3-bit asynchronous up counter using J-K FFs
- Sketch the timing diagram for 3-bit asynchronous up counter

**Disadvantages of asynchronous counter: Propagation Delay**

- Propagation delay in 3-bit asynchronous counter (ripple clocked) binary counter as shown below
- This effect 'ripples' the next FF resulting Q<sub>1</sub> delay some time from Q<sub>0</sub>
- The cumulative delay of asynchronous counter is the major disadvantage of this counter in many applications.
- It limits the rate at which the counter can be clocked and creates decoding problems.

**Conclusion**

- 3-bit asynchronous up counter consists of three J-K FFs and counts from 0 to 7 (8 states)

Disadvantages of asynchronous counter (continue)

- Asynchronous counters are not useful at very high frequencies, especially for counters with large number of bits.
- Another problem caused by propagation delays in asynchronous counters occurs when we try to electronically detect (decode) the counter’s output states.
Disadvantages of asynchronous counter (continue)

- Eg: If you look closely at the figure below, for a short period of time (50ns) right after state 011, you see that state 010 occurs before 100. This is obviously not the correct binary counting sequence and while the human eye is much too slow to see this temporary state, our digital circuits will be fast enough to detect it. These erroneous count patterns can generate what are called glitches in the signals that are produced by digital systems using asynchronous counters. In spite of their simplicity, these problems limit the usefulness of asynchronous counters in digital applications.

Exercise: A 4-bit Asynchronous Binary Counter

- Draw the timing diagram for 4-bit asynchronous up counter given below

A 4-bit Asynchronous Binary Counter (Continue)

- Each flip-flop has a propagation delay for 10ns. Determine the total propagation delay time from the triggering edge of a clock pulse until a corresponding change can occur in the state of Q3. Also determine the maximum clock frequency at which the counter can be operated.

Answer:
For the total delay time, the effect of CLK8 or CLK16 must propagate through 4 flip-flops before Q3 changes.

\[ t_{\text{prop}} = 4 \times 10 \text{ns} = 40 \text{ns} \]

The maximum clock frequency is

\[ f_{\text{max}} = \frac{1}{t_{\text{prop}}} = \frac{1}{40 \text{ns}} = 25 \text{MHz} \]

* The counter should be operated below this frequency to avoid problems due to the propagation delay.

Asynchronous MOD counter

- MOD number is generally equal to the number of states that the counter goes through in each complete cycle before it recycles back to its starting state.
- MOD number can be increased simply by adding more FFs to counter. MOD number = \( 2^N \)
Asynchronous MOD counter
(continue)

Example
- A photocell and light source combination is used to generate a single pulse each time an item crosses its path. The counter must be able to count as many as 1000 items. How many FFs are required?
  
  ans.) \[ 2^n = 1000 \]
  
  \[ n = \log_{10}1000 / \log_{10}2 \]
  
  = 10 FFs

Changing the MOD number
Determine the MOD number of the counter in Fig. 6-6(a). Also determine the frequency at the D output.

Ans.) Mod-14 ripple counter, \[ 30/14 = 2.14 \text{ kHz} \]

Changing the MOD number
Construct a MOD-10 counter that will count from 0000 through 1001.

Frequency Division
- Each flip-flop provides an output waveform that is exactly half the frequency of the waveform at its CLK input.
- In any counter, the signal at the output of the last flip-flop (i.e., the MSB) will have a frequency equal to the input clock frequency divided by the MOD number of the counter.
Frequency Division (continue)

- E.g: in a MOD-16 counter, the output from the last FF will have a frequency of 1/16 of the input clock frequency. Thus, it can also be called a divide-by-16 counter. Likewise, a MOD-8 counter has an output frequency of 1/8 the input frequency; it is a divide-by-8 counter.

Example

60-Hz signal is fed into a Schmitt-trigger, pulse-shaping circuit to produce a square wave. 60 Hz square wave is then put into a MOD-60 counter, which is used to divide the 60-Hz frequency by exactly 60 to produce a 1-Hz waveform. 1-Hz waveform is fed to a series of counters, which then count Ss, Ms, Hs, and so on. How many FFs are required for the MOD-60 counter? 6 FFs

Asynchronous Decade Counter

- Counters can be designed to have a number of states in their sequence that is less than the maximum of 2^N. This type of sequence is called a truncated sequence.
- For example, asynchronous modulus ten (MOD-10) counter or decade counter
- CLR is produced and then reset all FFs to recycle
- The counter count again

Asynchronous Decade Counter (continue..)

- Timing diagram and binary state sequence for decade counter
- Note that 10 is 1010 which is Q_3 AND Q_1 are HIGH
- CLR is produced and then reset all FFs to recycle
- The counter count again
Asynchronous Decade Counter Exercise

Modify MOD-10 asynchronous counter to have MOD-12 and draw the timing diagram (1100)

Exercise

1. What is the difference of operation between asynchronous and synchronous counter?

Exercise (continue)

2. Draw the circuit for asynchronous counter according to these attributes:
   - MOD 13 counter using JK flip-flops.
   - Negative edge triggered
   - Down counter
   - Active low preset and clear input

Answer:

Counters

- Asynchronous counter (Ripple counter): The output of one FF drives the input of the next one.
- Synchronous counters: Clock pulse is applied to each FF simultaneously.

- Slow speed
- High Speed
**Synchronous counter**

- Also known as parallel counter.
- Synchronous counters eliminate the propagation delay problem because all the clock inputs ($C_p$) are tied to a common clock.
- Can operate at higher clock frequencies. Asynchronous counters are not useful at very high frequencies, especially for large number of bits.
- Requires more circuitry than the asynchronous counterpart.
- The design starts with
  - State diagram
  - Truth table
  - K-map & equation
  - Circuit

**Excitation table**

- The flip-flop inputs are based on excitation table.

<table>
<thead>
<tr>
<th>$Q_1$</th>
<th>$Q_{c+1}$</th>
<th>$S$</th>
<th>$R$</th>
<th>$J$</th>
<th>$K$</th>
<th>$D$</th>
<th>$T$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>X</td>
<td>0</td>
<td>X</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>X</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>X</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>X</td>
<td>0</td>
<td>X</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
Types of synchronous counter

- Up counter. Eg: 0 → 1 → 2 → 3
- Down counter. Eg: 3 → 2 → 1 → 0
- Irregular binary sequence counter. Eg: 0 → 3 → 4 → 7
- Synchronous mod-counter
- Up/down counter or bidirectional counter (a control input is required for selection of modes).
- Up counter or down counter with asynchronous inputs (active high or active low preset and clear).

Design step for synchronous up counter

Example: Design a 2 bit counter using D, T and JK flip-flop based on the sequence 0 → 1 → 2 → 3.

Step 1: Draw the state diagram

![State Diagram]

Step 2: Fill in the truth table

<table>
<thead>
<tr>
<th>Present state</th>
<th>Next state</th>
<th>Flip-flop inputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1, Q0</td>
<td>Q1, Q0</td>
<td>D1, D0</td>
</tr>
<tr>
<td>0 0</td>
<td>0 1</td>
<td>0 1</td>
</tr>
<tr>
<td>0 1</td>
<td>1 0</td>
<td>1 0</td>
</tr>
<tr>
<td>1 0</td>
<td>1 1</td>
<td>1 1</td>
</tr>
<tr>
<td>1 1</td>
<td>0 0</td>
<td>0 0</td>
</tr>
</tbody>
</table>

# The flip-flop inputs are based on the excitation table

<table>
<thead>
<tr>
<th>Q0, Q0</th>
<th>S</th>
<th>R</th>
<th>J</th>
<th>K</th>
<th>D</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0</td>
<td>X</td>
<td>X</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0 1</td>
<td>X</td>
<td>X</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1 0</td>
<td>X</td>
<td>X</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1 1</td>
<td>X</td>
<td>X</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Step 3: Generate k-map

Step 4: Draw logic circuit

![Logic Circuit]
Up counter using T flip-flop

- Using T flip-flop

Up counter using JK flip-flop

- Using JK flip-flop

Design of Synchronous Counter Exercise 1

Design a counter to produce 3-bit binary counter using J-K FF

- State diagram
- State and excitation tables

Design of Synchronous Counter Exercise 1 (continue..)

Design a counter to produce 3-bit binary counter using J-K FF

- K-maps
- Counter implementation
Design of Synchronous Counter Exercise 2

- Design a counter to produce 3-bit binary counter using D FF
  - State diagram
  - State and excitation tables

<table>
<thead>
<tr>
<th>PRESENT STATE</th>
<th>NEXT STATE</th>
<th>FLIP-FLOP INPUTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q2, Q1, Q0</td>
<td>Q2, Q1, Q0</td>
<td>D2, D1, D0</td>
</tr>
<tr>
<td>0 0 0</td>
<td>0 0 1</td>
<td>0 0 1</td>
</tr>
<tr>
<td>0 0 1</td>
<td>0 1 0</td>
<td>0 1 0</td>
</tr>
<tr>
<td>0 1 0</td>
<td>0 1 1</td>
<td>0 1 1</td>
</tr>
<tr>
<td>0 1 1</td>
<td>1 0 0</td>
<td>1 0 0</td>
</tr>
<tr>
<td>1 0 0</td>
<td>1 0 1</td>
<td>1 0 1</td>
</tr>
<tr>
<td>1 0 1</td>
<td>1 1 0</td>
<td>1 1 0</td>
</tr>
<tr>
<td>1 1 0</td>
<td>1 1 1</td>
<td>1 1 1</td>
</tr>
</tbody>
</table>

Design of Synchronous Counter Exercise 2 (continue..)

- Design a counter to produce 3-bit binary counter using D FF
  - K-maps
  - Design a counter with the irregular binary count sequence 0 → 3 → 4 → 7 using D flip-flop.
  - Answer:

Irregular Binary Counter

- Counting without according to regular sequence.
  - Eg: Design a counter with the irregular binary count sequence 0 → 3 → 4 → 7 using D flip-flop.
  - Answer:

- Counter implementation
Synchronous mod-counter

- Eg: Design a mod-5 synchronous counter using D flip-flop. How many states does this counter have? What is the minimum number of flip-flop required?
- Answer: The counter will count from 0→4. Therefore there are 5 states. 3 flip-flop are required.

Design of Synchronous Counter Exercise 3

<table>
<thead>
<tr>
<th>Present State</th>
<th>Next State</th>
<th>Flip-Flop Inputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q2 Q1 Q0</td>
<td>D2 D1 D0</td>
<td></td>
</tr>
<tr>
<td>0 0 0</td>
<td>0 0 1</td>
<td>0 0 1</td>
</tr>
<tr>
<td>0 0 1</td>
<td>0 1 0</td>
<td>0 1 0</td>
</tr>
<tr>
<td>0 1 0</td>
<td>1 0 0</td>
<td>1 0 0</td>
</tr>
<tr>
<td>1 0 0</td>
<td>0 0 1</td>
<td>0 0 1</td>
</tr>
<tr>
<td>1 0 1</td>
<td>0 1 0</td>
<td>0 1 0</td>
</tr>
<tr>
<td>1 1 0</td>
<td>1 0 0</td>
<td>1 0 0</td>
</tr>
<tr>
<td>1 1 1</td>
<td>0 0 1</td>
<td>0 0 1</td>
</tr>
</tbody>
</table>

Design of Synchronous Counter Exercise 3 (continue..)

- Design MOD5 synchronous counter using D FF
  - State diagram
  - State and excitation tables

D2 map:

D1 map:

D0 map:

D2 = Q1 Q2
D1 = Q1 Q2 + Q1 Q0
D0 = Q2 Q0
Design of Synchronous Counter Exercise 3 (continue..)

- Design MOD5 synchronous counter using D FF
  - Counter implementation

Design of Synchronous Counter Exercise 4

- Design MOD5 synchronous counter using T FF
  - State diagram
  - State and excitation tables

<table>
<thead>
<tr>
<th>PRESENT STATE</th>
<th>NEXT STATE</th>
<th>FLIP-FLOP INPUTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q2 Q1 Q0 T2</td>
<td>Q2 Q1 Q0 T1</td>
<td>T0 T1 T2</td>
</tr>
<tr>
<td>0 0 0 0</td>
<td>0 0 1 0</td>
<td>0 0 1</td>
</tr>
<tr>
<td>0 0 1 0</td>
<td>0 1 0 0</td>
<td>0 1 1</td>
</tr>
<tr>
<td>0 1 0 0</td>
<td>0 1 1 0</td>
<td>1 0 0</td>
</tr>
<tr>
<td>0 1 1 0</td>
<td>0 0 0 1</td>
<td>0 0 1</td>
</tr>
<tr>
<td>1 0 0 0</td>
<td>1 1 0 1</td>
<td>1 1 1</td>
</tr>
<tr>
<td>1 0 1 0</td>
<td>1 1 1 0</td>
<td>0 1 0</td>
</tr>
<tr>
<td>1 1 0 0</td>
<td>1 0 0 1</td>
<td>1 0 1</td>
</tr>
<tr>
<td>1 1 1 0</td>
<td>0 1 1 0</td>
<td>1 1 1</td>
</tr>
</tbody>
</table>

Design of Synchronous Counter Exercise 4 (continue..)

- Design MOD5 synchronous counter using T FF
  - K-maps
  - Counter implementation

\[ T_2 = Q_1 \cdot Q_0 \]
\[ T_1 = \overline{Q}_1 Q_0 + \overline{Q}_0 Q_1 + \overline{Q}_0 \]
\[ T_0 = Q_1 + Q_0 \]
Exercise

Design a 3 bit synchronous counter for the sequence 0→6→4→2 using
a) D flip-flop
b) T flip-flop
c) JK flip-flop

Topic:

- Up/down counter or bidirectional counter
- Cascaded counter
  - Asynchronous cascaded counter
  - Synchronous cascaded counter
- Counter decoding
  - Decoding glitches
  - Strobing technique

Up/Down Synchronous Counter (bidirectional counter)

- Bidirectional counters, also referred to as UP/DOWN counters, are capable of progressing in either direction through any given count sequence. Recall that in general, bidirectional counters can be reversed at any point in their count sequence.
- Capable to count in either direction through a certain sequence
- For example 3-bit up/down synchronous counter
  - Able to count from 0 to 7 or 7 to 0

Up/Down Synchronous Counter design procedure

- Eg: Design a 2 bit up/down counter using T flip-flop based on the state diagram below. Assume up = 1 and down = 0.
  Answer:

<table>
<thead>
<tr>
<th>Present state Q1 Q0</th>
<th>Next state Q1 Q0</th>
<th>Y</th>
<th>Flip/flop inputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0</td>
<td>1 1</td>
<td>0</td>
<td>1 1</td>
</tr>
<tr>
<td>0 1</td>
<td>0 0</td>
<td>0</td>
<td>0 0</td>
</tr>
<tr>
<td>1 0</td>
<td>1 1</td>
<td>0</td>
<td>1 1</td>
</tr>
<tr>
<td>1 1</td>
<td>0 0</td>
<td>0</td>
<td>0 0</td>
</tr>
</tbody>
</table>

Diagram showing the state transition and flip/flop inputs for the 2-bit up/down synchronous counter.
Cascaded Counters

- Counters can be connected to achieve higher modulus operation.
- Cascading means that the last stage output of one counter drives the input of the next counter.
- A mod-\( M \) and a mod-\( N \) counter in cascade give a mod-\( MN \) counter.
- 2 types of cascading: Asynchronous cascading and synchronous cascading.

Cascaded Counters (continue)

Asynchronous cascading

Two asynchronous counters connected in cascade for a 2 bit and a 3 bit ripple counter. The overall modulus of the two cascaded counters is \( 4 \times 8 = 32 \); that is they act as a divide-by-32 counter.

Synchronous cascading

- In synchronous cascaded counter, it is necessary to use the count enable (CTEN) and the terminal count (TC) functions to achieve higher modulus operation.
- Terminal count (TC) is analogous to ripple clock or ripple carry out (RCO) on some IC counters.

Example 1: The figure below shows a mod-10 counter and mod-8 counter connected in cascade. What is the overall modulus of these two cascaded counter? Determine the frequency at B if \( f_{\text{in}} \) is 20kHz.

Answer:
Overall modulus = 10 \times 8 = 80 = \text{mod-80 counter}
Frequency at B = \( f_{\text{in}}/80 = 250 \text{Hz} \)
Cascaded Counters (continue)

Example 2:
How many decade counters are required to convert a clock of 1 MHz to 1 Hz? Draw the circuit.

Answer: \( f_{\text{out}} = \frac{f_{\text{in}}}{10^n} \)

\[ 1 = \frac{1 \times 10^6}{10^n} \]

\[ n = \log \left( \frac{1 \times 10^6}{10} \right) / \log 10 \]

\[ n = 6 \text{ decade counter} \]

Decade Counters/BCD counters

- Decade counter
  - Any counter has 10 distinct states, no matter what the sequence.

- BCD counter
  - A decade counter counts in sequence from 0000 (zero) through 1001 (decimal 9).

Cascaded Counters (continue)

Example 3: Determine the overall modulus of the two cascaded counter for (a) and (b)

Answer:
(a) the overall modulus for the 3 counter configuration is 8 x 12 x 16 = 1536 = mod-1536
(b) the overall modulus for the 4 counter configuration is 10 x 4 x 7 x 5 = 1400 = mod-1400

Counter Decoding (decoding a counter)

- Mentally decoding the binary states of the LEDs
  - Becomes inconvenient as the size of the counter increases

- Electronically decoding
  - To determine when the counter is in a certain binary states in its sequence using decoders or logic gates.

- 3 types of decoding:
  1. Active-High Decoding (AND gate)
  2. Active-Low Decoding (NAND gate)
  3. BCD counter decoding
Counter Decoding (decoding a counter)

Example: to decode binary state 6 (110) of a 3 bit binary counter. When Q2=1, Q1=1 and Q0=0, a HIGH appears on the output of the decoding gate.

Counter Decoding (decoding a counter)

Example: A 3-bit counter with active-HIGH decoding of count 2 and count 7.

Decoding glitches

- The decoding process may resort to glitches. What is glitch?
  Glitch is an erroneous count patterns or unwanted output voltage caused by the propagation delay effect.

- Occurs to
  (a) Asynchronous counter – propagation delay due to ripple effect
  (b) Synchronous counter – propagation delay from the clock to the Q output of every flip-flop

Example of glitches

Figure: a basic BCD/decade counter

Figure: Outputs with glitches
Solution to eliminate glitches

- **Strobing** - enable the decoded outputs at a time after the glitches have had time to disappear.
- Accomplished in the case of an active high clock by using the low level of the clock to enable the decoder.

Example without glitches

Exercise 1

- Design a 3 bit up/down counter using JK flip-flop based on the state diagram below. Assume up = 1 and down = 0.