Chapter 7 Counter Practice

7-1 Practice purpose

1. To understand the binary counter's structure
2. To understand synchronous and asynchronous counter
3. To understand N-MODE counter's design
4. To understand jumping counter's design

7-2 Practice theory

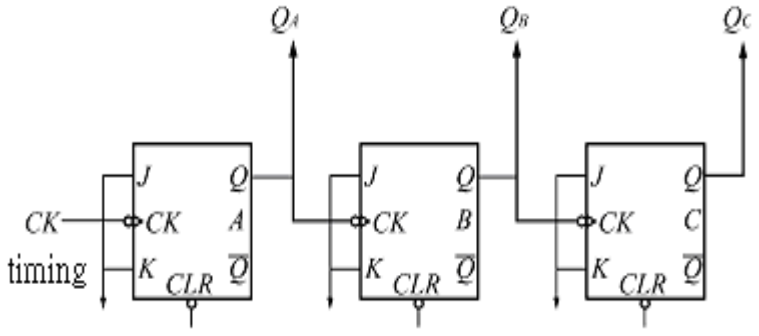
A counter is made by a flip-flop and logic gate which plays a very important role in the digital system. This chapter discusses counters' design technique and when you have read this chapter, you will easily understand how counters are designed. We will talk about IC counter(shift register) in the next chapter.

A counter is an order logic circuit. It is made by flip-flop and logic gate. When there is an input pulse entering the flip-flop, it converts its output state by order. As for the input pulse, we call it a counter clock. Counters play very important roles in the digital system. They are used to manage wave counts, frequency cutting, A/D, D/A switch and frequency counter and so forth.

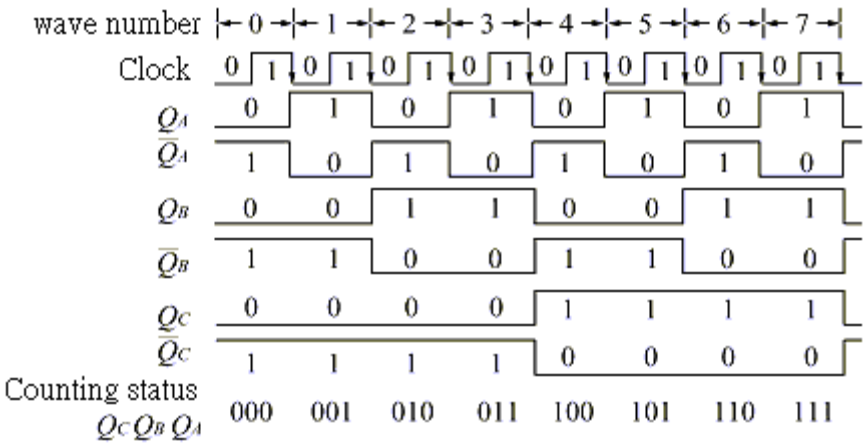
This chapter is talking about counters' categories, designs.

7-2-1 Asynchronous counter

Figure 7-1 is a three stage Binary ripple counter connected with three JK flip-flops.
(1) A counter is made by 3 JK flip-flops.
(2) Each flip-flop's Q output is connected to the next stage clock input. Therefore each clock is the previous stage Q's output, meaning that this stage's output is next stage's clock input.
(3) Each flip-flop has CLR input.
(4) Each flip-flop's CK input has a small circle to point out where the trigger edge is.



(a)



(b)

Figure 7-1 Three stage binary ripple counter (a) circuit (b) wave shape

Negative edge trigger edge has many other names, such as back edge trigger, lower edge trigger, and they all mean the same. When input pulse changes from 1 to 0, the flip-flop's output state changes, in other words, if the input pulse changes from 0 to 1, the flip-flop's output state changes, which we call positive converting state, front edge trigger, outer edge trigger, etc.

(5) To make all the Q positions as 0, we use CLR input to clear all the flip-flops.

(6) A periodical square wave is added to A flip-flop's input, as Figure 7-1(b) shows.

(7) When A flip-flop's Q_A negative edge triggers, B flip-flop changes its state, as Figure 7-1(b).

(8) When B flip-flop's Q_B negative edge triggers, C flip-flop changes its state, as Figure 7-1(b).

From Figure 7-1, we can tell that when each signal goes through a flip-flop, the period is doubled and the frequency is reduced to half. Therefore, Figure 7-1's counter can be written as:

In Q_A period$=2\times$timing period

Frequency$=\frac{1}{2}\times$timing frequency

In Q_B $P_B = 2^{2\text{ or}}\times P_{\text{clock}}$

$$f_B = \frac{1}{2^2}\text{ or }\frac{1}{4}\times f_{clock}$$

In Q_C $P_C = 2^{3\text{ or}}8\times P_{clock}$

$$f_C = \frac{1}{2^3}\text{ or }\frac{1}{8}\times f_{clock}$$

Therefore, there is usually n class binary ripple counter, and we get

$$P = 2^n \times \text{timing period} \qquad (7\text{-}1a)$$

$$f = \frac{1}{2^n}\times \text{timing frequency} \qquad (7\text{-}1b)$$

Now we observe Figure 7-1(b)'s each clock Q *and* $\overline{Q}$ positions.

(1) Assume clock 0 is the fist one.

(2) List Q and $\overline{Q}$ at each timing and class's output, as table 7-1's $Q_{A,}\overline{Q}_{A,}Q_{B,}\overline{Q}_{B,}Q_{C,}\overline{Q}_{C.}$

(3) At each timing, Q_A output changes its state.

(4) When QA converts from 1 to 0, Q_B converts its state.

(5) When QB converts from 1 to 0, Q_C converts its state.

(6) Use binary $Q_CQ_BQ_A$ to present Q_C, Q_B, Q_A's states which is called count state. As table 7-1, E column.

(7) F column is E column's decimal effect. F column's decimal digital rises from 0 to 7 and then starts from 0 again. If we assign the first clock as 0m, then the clock digital and E column's binary and F column's decimal system are all the same.

(8) E column's counter is proceeding as binary form; therefore, Figure 7-1's counter is binary counter. The digital goes up to 7 and back to 0 and does the same thing over and over again.

Table 7-1 logic position: three class binary counter

Clock	Q_C	Q_C	Q_A	Counting state Q_C Q_B Q_A E	E column decimal digital F	Q_C	Q_B	Q_A	Q_C Q_B Q_A G	G column decimal digital H
	Weight					Weight				
	4	2	1			4	2	1		
0	0	0	0	000	0	1	1	1	111	7
1	0	0	1	001	1	1	1	0	110	6
2	0	1	0	010	2	1	0	1	101	5
3	0	1	1	011	3	1	0	0	100	4
4	1	0	0	100	4	0	1	1	011	3
5	1	0	1	101	5	0	1	0	010	2
6	1	1	0	110	6	0	0	1	001	1
7	1	1	1	111	7	0	0	0	000	0
8	0	0	0	000	0	1	1	1	111	7
					Etc					Etc

(9) To get the equal decimal digital, Q_C column's weight is 4, Q_B is 2 and Q_A is 1.

(10) Table 7-1 $\overline{Q}_{C,}\overline{Q}_{B,}$ and $\overline{Q}_A$ column is $Q_{C,}$ Q_B and Q_A column 's opposites. G column is E column's supplement which counts down from 7 to 0, then again from 7.

(11) F column counts upward, H column downward, but no matter what, F and H columns only go up to 7 which is the max counts.

It may sound confusing. Table 7-1's counting state and the binary digital symbols are Q_C, Q_B, Q_A, but figure 7-1's counting is drawn from the opposite direction. Unfortunately, circuit's normal drawing signal is usually from left to right. Therefore, most of the data manuals are drawn as the opposite of figure 7-1's circuit which means the signal input is on the right and it goes from right to left.

Three class binary counter's max. counting ability is 111(binary) or 2^3-1 = 7 (decimal). Usually, a N class binary counter's max. decimal digital is :

$$N\ stage\ max\ counting = n \text{ multiple } 1 = 111\cdots \qquad (7\text{-}2)$$
$$= 2^n\text{-}1$$

the counter can be written as

$$\text{counter} = B_n \times 2^{n-1} + B_{n-1} \times 2^{n-2} + B_{n-3} \times 2^{n-3} + \cdots\cdots B_1 \times 2^0 \qquad (7\text{-}3)$$

B is each class's binary position (1 or 0)。

In Figure 7-1's counter, the first class makes a lowest bit LSB's weight ≡1, and the last class's highest bit MSB's weight is ≡4. We have discussed the three stage counters, for 6 or 110's counting, B_3 = 1,B_2 = 1, and B_1= 0, therefore

$$\text{count} = 1\times 2^{3-1} + 1\times 2^{3-2} + 0\times 2^0$$
$$= 4+2+0=6$$

Assume counting state is 011, and it is added to the next clock, then the next counting state 100, three classes are changed. Therefore, can they all change at the same time? For IC, 0 position's output delay time is 16ns, and from timing to logic 1` position's output delay is about 25ns. Therefore, 011 to 100's conversion delay is

First class	1 to 0	$25ns$
Second class	1 to 0	$25ns$
Third class	0 to 1	$\frac{16ns}{66ns}$

It is very fast. But according to the modern standard, it might be just OK. Some people may probably think it takes too long. The counter's class is always calculated by the previous class, and its changing order is sending downwards or by clock through the counter. Figure 7-1's serial counter is called a ripple counter.

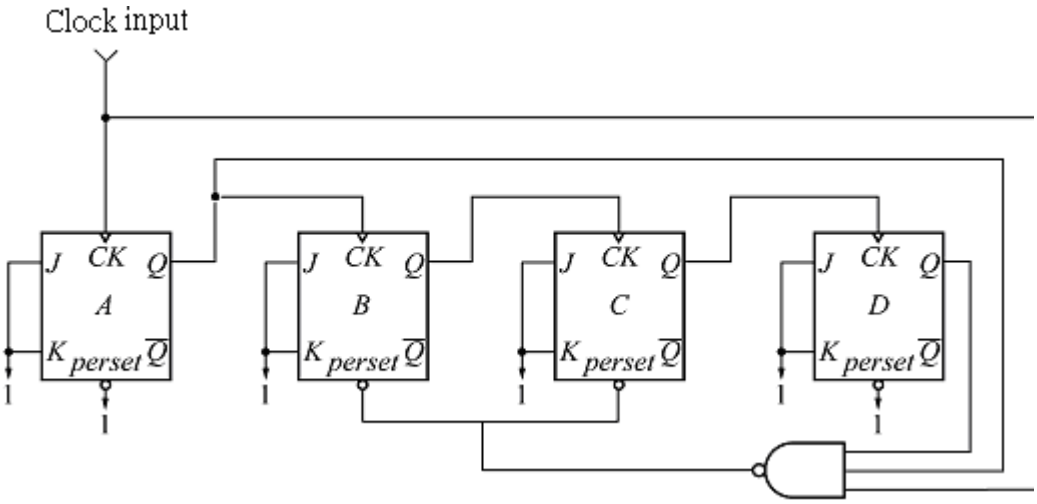
7-2-2 Ripple counter

We can design a asynchronous counter according to our own needs. There are three of them: (1) connect the signal to preset (2) connect the control signal to clear (3) auto stop design.

(1) the first one:

Figure 7-2 is 1 divided by 10 BCD counter, it repeats each 10 times. The counter goes back to 0 state. It uses the Nth (10^{th} in here) positive timing to push all the flip-flops back

to 1 state and when it is negative edge all the flip-flops go back to 0 state. So the counter starts all over again.
JK flip-flops always have preset and CLR installation, pushing preset is to make flip-flop 1 state, pushing CLR is to make it 0 state (output). As for 1 or 0 state to push preset or CLR, it depends on the symbol. If there is a small circle or half circle in front of the preset and CLR which means low state is pushing it, or else, high state.



State	Q_D	Q_C	Q_B	Q_A
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
0	0	1/0	1/0	0

Figure 7-2 BCD decimal system ripple counter

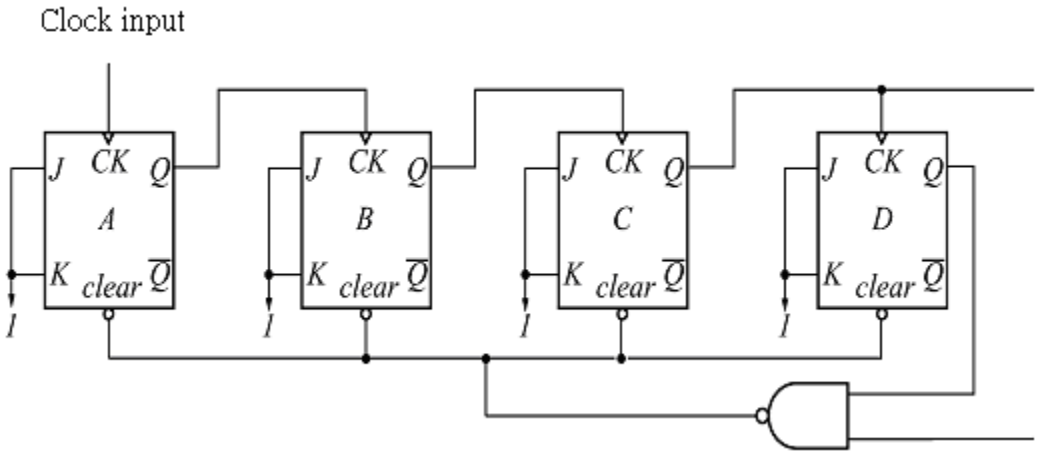
(2) The second one:
There is no preset in most of the IC flip-flops, but only clear. Figure 7-3 is to use common clear line to divide 12 counters. The steps are as follows:

(1) Solve needed flip-flop numbers, n, which has to meet the following need.

$$2^{n-1} \le N \le 2^n$$

(2) Connect all the flip-flops to ripple counter.
(3) Find the binary system number N.
(4) When the counter counts to N, Q=1's flip-flop output to NAND gate's input and NAND gate's output connects to flip-flop's clear.

In this way, when the counter counts to N, NAND gate's output is 0 and all the flip-flops go back to 0 state.
Asynchronous counter is easy, but there are still some disadvantages; the slow operation is one of them. Assume all the flip-flops' delay is 50ns, then 4 flip-flops' delay is 200ns, which means the movement frequency might get affected to operate at very slow speed.



State	Q_D	Q_C	Q_B	Q_A
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
0	1/0	1/0	0	0

Figure 7-3 Divided by 12's ripple counter

(3) The third one:
Figure 7-2's counter doest not stop itself. As long as there is timing input, its counting value just keeps increasing until it has the biggest value 1111(15), then it goes back to 0000(0), then it starts all over again. If we want it to stop at 1001(9) we only need to make some adjustment (increase NAND gate). As Figure 7-4 shows, this is called automatically stopping ripple counter.

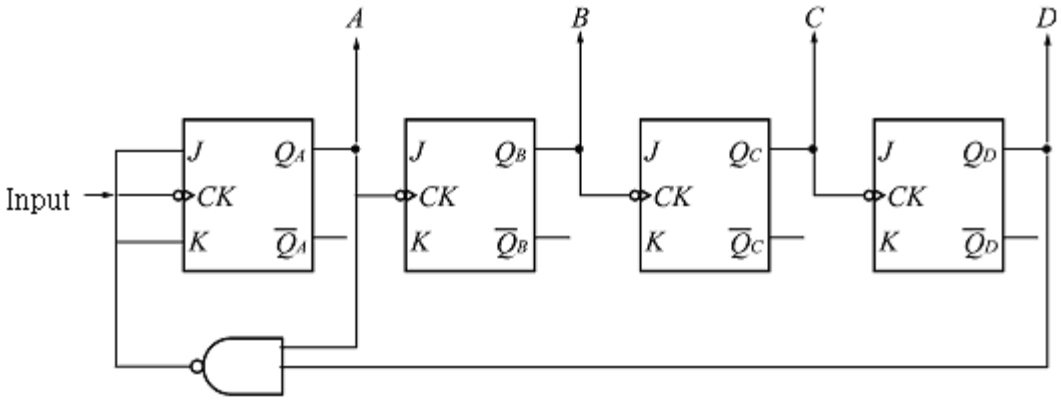


Figure 7-4 automatically stopping ripple counter stops at 1001.

When input timing, the counter counts normally and after the 9th input, the counter shows 1001. At this point, A and D's outputs are both Hi, therefore NAND gate output is low to make $JA=K_A=0$. Because J=K=0, FFA will not convert to another state. AFF output does not either, which makes B, C and DFF stay the same, the counter stays at 1001. If we want to make the counter start counting, we only need to add clear clock to AFF and DFF to make $Q_A=0$, $Q_D=0$.

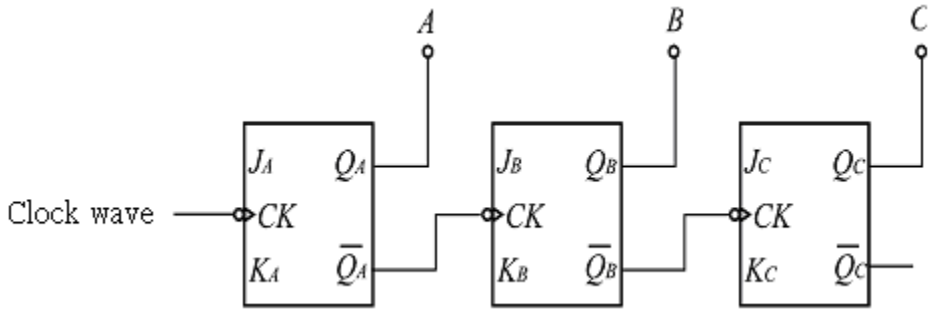
Down counter

The discussion above is about a asynchronous up counter which means to count from 0. We can design a down counter to let the counter count from the max. value back to 0. As follows:

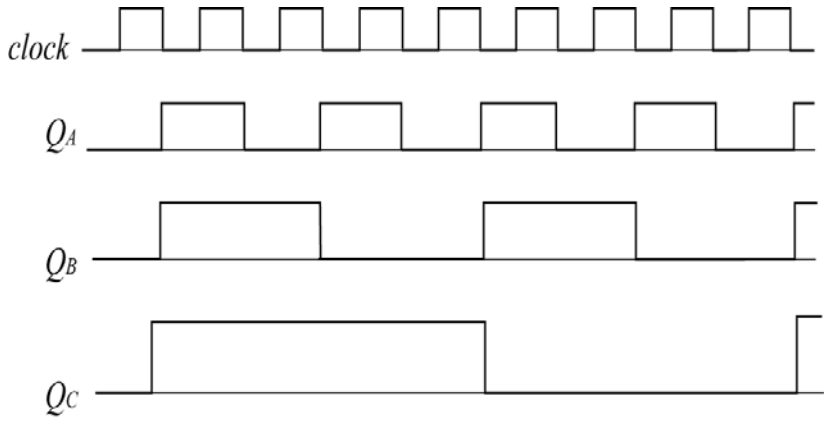
	CBA	CBA	CBA
(7)	111	111	111
(6)	110	110	110
(5)	101	101	101
(4)	100	100	100
(3)	011	011	011
(2)	010	010	010
(1)	001	001	001
(0)	000	000	000

A, B and C present FF's output state, so that we can tell that AFF's lowest LSB changes state every time, just like the up counter. BFF changes its state when A converts from low

to Hi. C converts its state when B converts from low to Hi. Therefore, if we use previous FF $\overline{Q}$ output as later FF CK input, then we can meet the low count's need. Figure 7-5 shows MOD-8 's low count and wave Figure.



(a)BINARY DOWN COUNTER



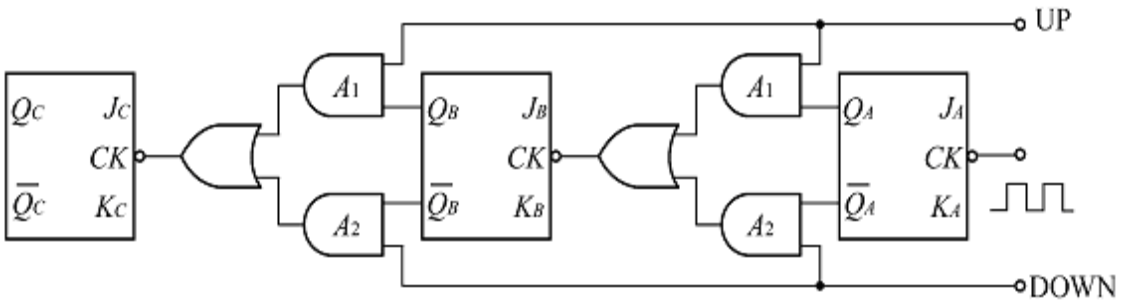
(b) Timing Chart

7-2-3 Asynchronous up/down counter

Figure 7-6 is an up/down counter. Its up and down are the two inputs controlling the counters going up or down.

When up is 1, down is 0 and A1's changes follow Q, A2's output is 0 for certain. Therefore it is up counter, vice versa. When up is 0, down is1, A2's changes follow $\overline{Q}$, A1's output is 0 for certain, so it is low counter. If up=down =0, there is no counting function, because OR gate's output is definitely 1 or 0.

All the JK inputs are 1.



Up	Down	Function
0	0	No counting
0	1	Downcounting
1	0	Up counting
1	1	No counting

Figure 7-6 UP/DOWN COUNTER CIRCUIT

7-3 Practice items

7-3-1

Experiment steps

1. four bit binary up asynchronous counter circuit, as Figure7-7

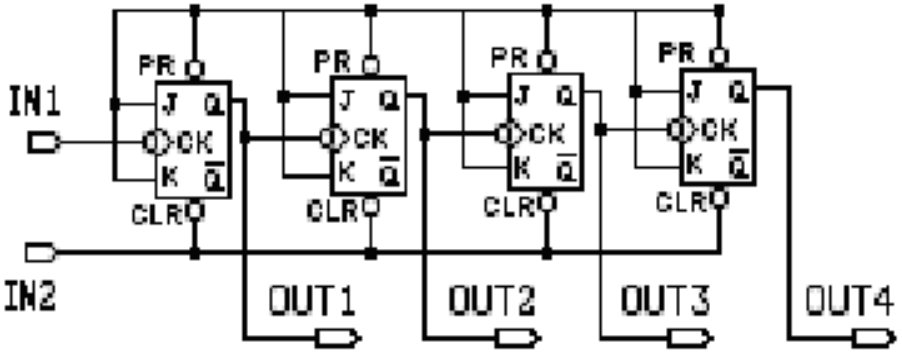


Figure 7-7

2. Input: Connect CON5 CLK 10HZ to CON1 IN1
Connect CON7 S1 to CON1 IN2
Output: Connect CON4 OUT1 to LED DISPLAY CON15 Q1
Connect CON4 OUT2 to LED DISPLAY CON15 Q2
Connect CON4 OUT3 to LED DISPLAY CON15 Q3
Connect CON4 OUT4 to LED DISPLAY CON15 Q4

3. Switch S1 after finished connecting and record LED changes in table 7-1.

CLR	Clock	OUT4	OUT3	OUT2	OUT1	Equal value decimal system
0	0	0	0	0	0	0
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						

Table 7-1

7-3-2

Experiment steps

1. Four bit binary down counter asynchronous counter circuit, as Figure 7-8.

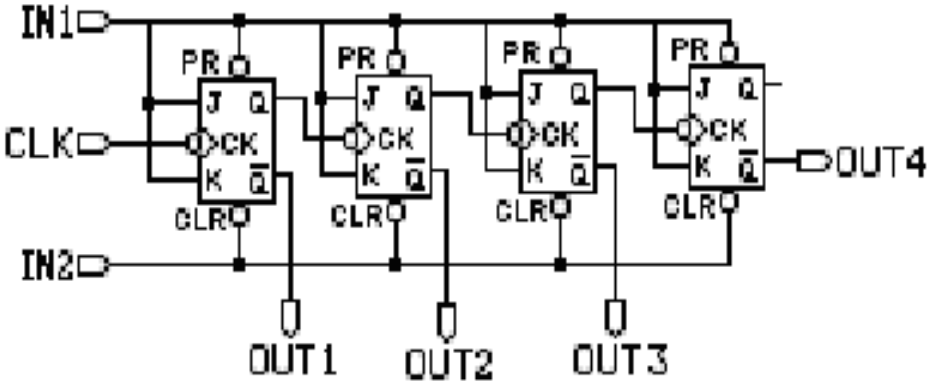


Figure 7-8

2. Input: Connect CON5 CLK 10HZ to CON1 IN1
Connect CON7 S1 to CON1 IN2
Output: Connect CON4 OUT1 to LED DISPLAY CON15 Q1
Connect CON4 OUT2 to LED DISPLAY CON15 Q2
Connect CON4 OUT3 to LED DISPLAY CON15 Q3
Connect CON4 OUT4 to LED DISPLAY CON15 Q4

3. Switch S1 after finished connecting and record LED changes in table 7-2.

LR C	Clock	OUT4	OUT3	OUT2	OUT1	Equal value decimal system
0	0	0	0	0	0	0
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						

table 7-2

7-3-3

Experiment steps

1. 0-9 counter circuit, as Figure 7-9

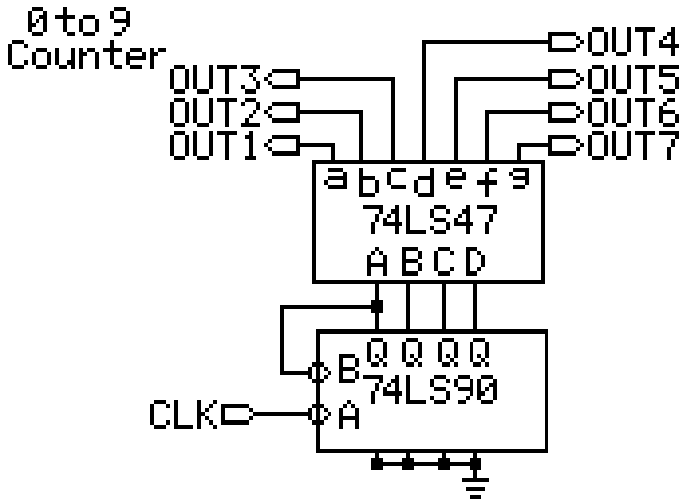


Figure 7-9

2. Input: Connect CON5 CLK 10HZ to CON1 IN1
   Output: Connect CON4 OUT1 to 7-SEGMENT CON10 LED 1_a
   Connect CON4 OUT2 to 7-SEGMENT CON10 LED 1_b
   Connect CON4 OUT3 to 7-SEGMENT CON10 LED 1_c
   Connect CON4 OUT4 to 7-SEGMENT CON10 LED 1_d
   Connect CON4 OUT5 to 7-SEGMENT CON10 LED 1_e
   Connect CON4 OUT6 to 7-SEGMENT CON10 LED 1_f
   Connect CON4 OUT7 to 7-SEGMENT CON10 LED 1_g

3. Record 7-SEGMENT changes in table 7-3.

0	1	2	3	4	5	6	7	8	9

Table 7-3

4. Observe 7-SEGMENT CHANGES and see if they are 0→1→2→……→9?

7-3-4

Experiment steps

1. 0-59 counter circuit, as Figure 7-10.

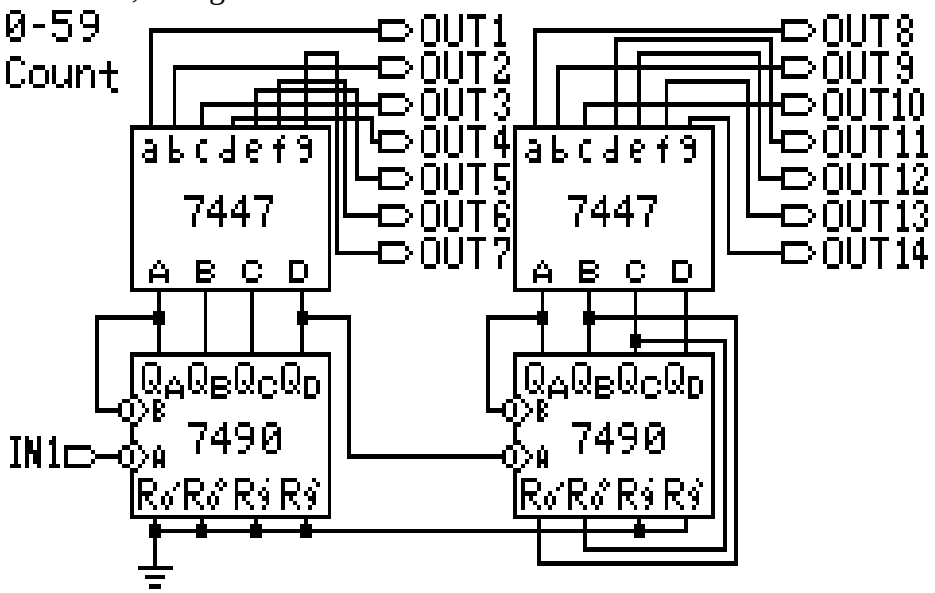


Figure 7-10

2. Input: Connect CON5 CLK 10HZ to CON1 IN1
Output: Connect CON4 OUT1 to 7-SEGMENT CON10 LED 1_a
Connect CON4 OUT2 to 7-SEGMENT CON10 LED 1_b
Connect CON4 OUT3 to 7-SEGMENT CON10 LED 1_c
Connect CON4 OUT4 to 7-SEGMENT CON10 LED 1_d
Connect CON4 OUT5 to 7-SEGMENT CON10 LED 1_e
Connect CON4 OUT6 to 7-SEGMENT CON10 LED 1_f
Connect CON4 OUT7 to 7-SEGMENT CON10 LED 1_g
Connect CON4 OUT8 to 7-SEGMENT CON10 LED 1_a
Connect CON4 OUT9 to 7-SEGMENT CON10 LED 1_b
Connect CON4 OUT10to 7-SEGMENT CON10 LED 1_c
Connect CON4 OUT11to 7-SEGMENT CON10 LED 1_d
Connect CON4 OUT12to 7-SEGMENT CON10 LED 1_e
Connect CON4 OUT13to 7-SEGMENT CON10 LED 1_f
Connect CON4 OUT14to7-SEGMENT CON10 LED 1_g

3. Record 7-SEGMENT and see if they are00→01→⋯→59?

7-3-5

Experiment steps

1. BCD up/down counter circuit, as Figure 7-11

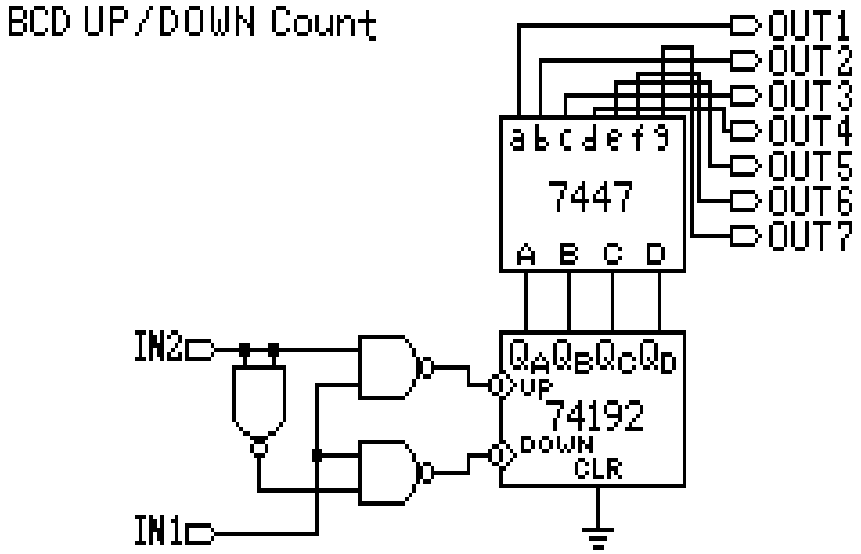


Figure 7-11

2. Input: Connect CON5 CLK 10HZ to CON1 IN1
Connect CON7 S1 to CON1 IN2
Output: Connect CON4 OUT1 to 7-SEGMENT CON10 LED 1_a
Connect CON4 OUT2 to 7-SEGMENT CON10 LED 1_b
Connect CON4 OUT3 to 7-SEGMENT CON10 LED 1_c
Connect CON4 OUT4 to 7-SEGMENT CON10 LED 1_d
Connect CON4 OUT5 to 7-SEGMENT CON10 LED 1_e
Connect CON4 OUT6 to 7-SEGMENT CON10 LED 1_f
Connect CON4 OUT7 to 7-SEGMENT CON10 LED 1_g

3. When U/D (S1) is 1, is counter up or down?_____ When U/D (S1) =0, is counter up or down? _____

7-3-6

Experiment steps

1. Johnson counter circuit, as Figure 7-12.

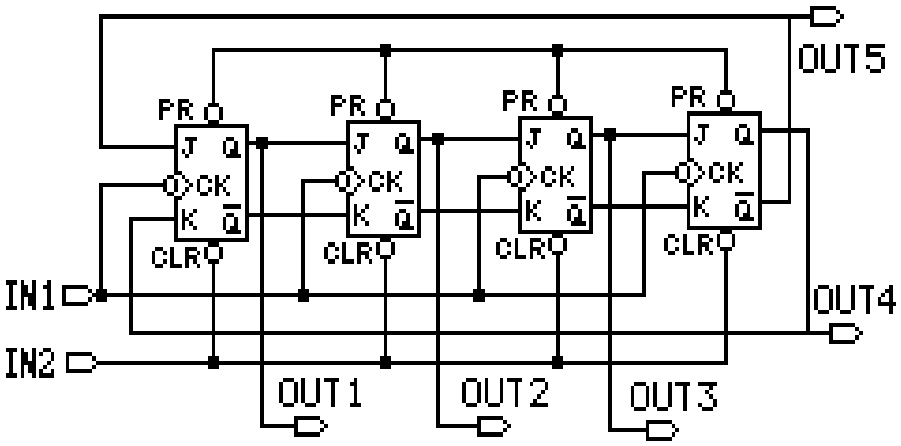


Figure 7-12

2. Input: Connect CON5 CLK 10HZ to CON1 IN1
Connect CON7 S1 to CON1 IN2
Output: Connect CON4 OUT1 to LED DISPLAY CON15 Q1
Connect CON4 OUT2 to LED DISPLAY CON15 Q2
Connect CON4 OUT3 to LED DISPLAY CON15 Q3
Connect CON4 OUT4 to LED DISPLAY CON15 Q4
Connect CON4 OUT5 to LED DISPLAY CON15 Q5

3.Switch S1, as table 7-4 after finished connecting. 0 means low logic, LED off, 1 means high logic, LED on.

4. Record LED changes to table 7-4.

CLR	Timing pulse wave	OUT4	OUT3	OUT2	OUT1	OUT5	Equal value decimal system
0	0	1	1	1	1		15
1	1						
1	2						
1	3						
1	4						
1	5						
1	6						
1	7						
1	8						
1	9						
1	10						
1	11						
1	12						
1	13						
1	14						
1	15						
1	16						
1	17						
1	18						
1	19						
1	20						
1	21						
1	22						
1	23						
1	24						
1	25						
1	26						
1	27						
1	28						
1	29						
1	30						
1	31						
1	32						

Table 7-4

7-4 Questions & Discussion

1. What is a counter?
2. How many F/F are needed when designing a divided 60 circuit?
3. Design a divided 9 circuit.
4. Design a divided 24 circuit.