Counter Circuits

Objectives:

To construct and study the operations of the following circuits:

(i) A 4-bit binary ripple Up-counter
(ii) A 4-bit binary ripple Down-counter
(iii) A Mod-12 counter
(iv) A 4-bit binary ripple Up/Down-counter
(v) A Ring counter

Overview:

Binary Counters are one of the applications of sequential logic using flip-flops. A counter is a device which stores (and sometimes displays) the number of times a particular event or process has occurred, in form of a clock pulse. Counters can be formed by connecting individual flip-flops together. On application of pulses, the flip-flops in the counter undergo a change of state in such a manner that the binary number stored in the flip-flops represents the number of pulses applied at input. When clock pulses are applied to a counter, the counter progresses from one state to another and the final output of the flip-flop in the counter indicates the pulse count. If all the flip-flops are not clocked at the same time, the counter is *asynchronous (or Ripple)* and if they are clocked simultaneously, the counter is *synchronous*. In practice, there are two types of counters:

- up counters, for increment in value
- down counters, for decrement in value

Frequency Division: For frequency division, toggle mode flip-flops are used in a chain as a divide by two- counter. One flip-flop will divide the input clock frequency by 2, two flip-flops will divide it by 4 (and so on). One benefit of using toggle flip-flops for frequency division is that the output at any point has an exact 50% duty cycle. The final output clock signal will have a frequency value equal to the input clock frequency divided by the MOD number of the counter. Such circuits are known as "divide-by-n" counters, where "n" is the number of counter stages used.

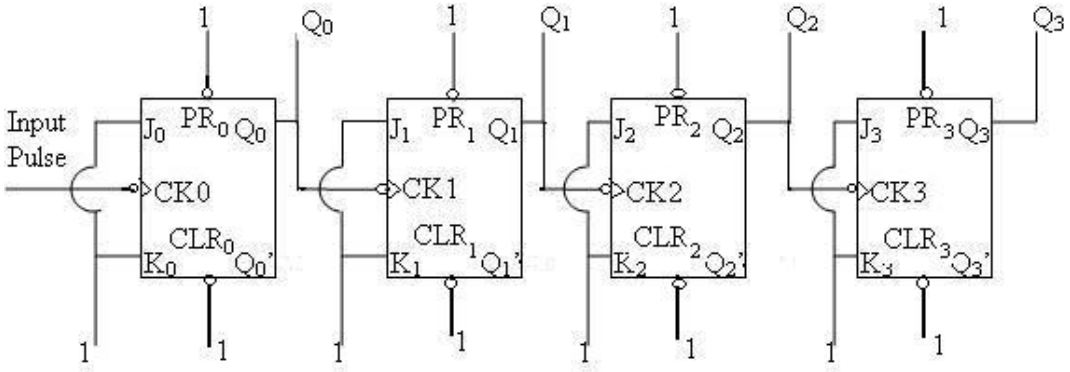
Binary ripple Up-counter:

We will consider a basic *4-bit binary up counter*, which belongs to the class of *asynchronous counter* circuits and is commonly known as a *ripple counter*. Since a flip-flop has two states, a counter having n flip-flops will have 2^n states. Hence, in this case the counter will have 2^4 or 16 states.

The schematics below shows a 4-bit up-counter implemented with four JK flip-flops. It can be noticed that the normal output of each flip-flop is connected to the clock input of next flip-flop. Please recall that in case of JK flip-flop, with J=K=1, if an input clock pulse is supplied, the output toggles during the positive or negative (which is the case here, i.e. transition of pulse from 1 to 0) edge of the pulse. The count held by this counter is read in the reverse order from the order in which the flip-flops are triggered. Thus, output Q_3 is the highest order of the count, while output Q_0 is the lowest order. The binary count held by the counter is then $Q=Q_3Q_2Q_1Q_0$.

Let us start from the reset condition of all the flip-flops so that counter reads 0000 (decimal 0). When the trailing or negative end of the first pulse arrives, the first flip-flop gives an output $Q_0=1$, which does not affect the second flip-flop and the counter reads 0001. The counting sequence corresponding to each input pulse is summarized in the table below. On supplying 15th pulse the counter reads 1111 (decimal 15). The next clock pulse after count 1111 will cause the counter to try to increment to 10000 (decimal 16). However, that 1 bit is not held by any flip-flop and is therefore lost. As a result, the counter actually reverts to 0000, and the count begins again.

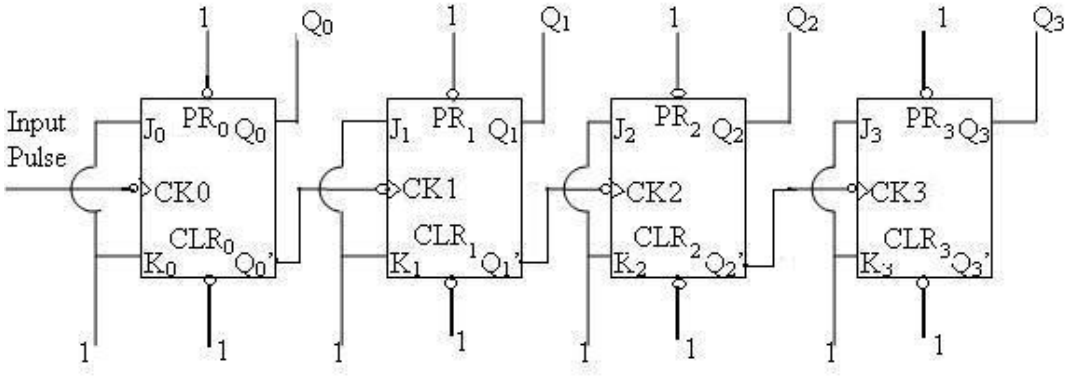
Circuit Diagram:



Binary ripple Down-counter:

The binary ripple down-counter decreases the count by one each time a pulse occurs at the input. The only difference it has from the up-counter is that the complement output of one flip-flop is connected to the clock input of the subsequent flip-flop. Here the complement output toggles at each negative edge of the clock pulse (1 to 0 transition), which is equivalent to a normal output toggling for positive edge of the clock pulse (0 to 1 transition). The counter starts from 1111 with the first pulse after it is reset and reverts back to 0000 after 15 pulses.

Circuit diagram:



Characteristic Tables:

UP COUNTER

Input pulse	Binary count				Decimal count
	Q_3 2^3	Q_2 2^2	Q_1 2^1	Q_0 2^0	
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	2
3	0	0	1	1	3
4	0	1	0	0	4
5	0	1	0	1	5
6	0	1	1	0	6
7	0	1	1	1	7
8	1	0	0	0	8
9	1	0	0	1	9
10	1	0	1	0	10
11	1	0	1	1	11
12	1	1	0	0	12
13	1	1	0	1	13
14	1	1	1	0	14
15	1	1	1	1	15
16	0	0	0	0	0 (RESET)

DOWN COUNTER

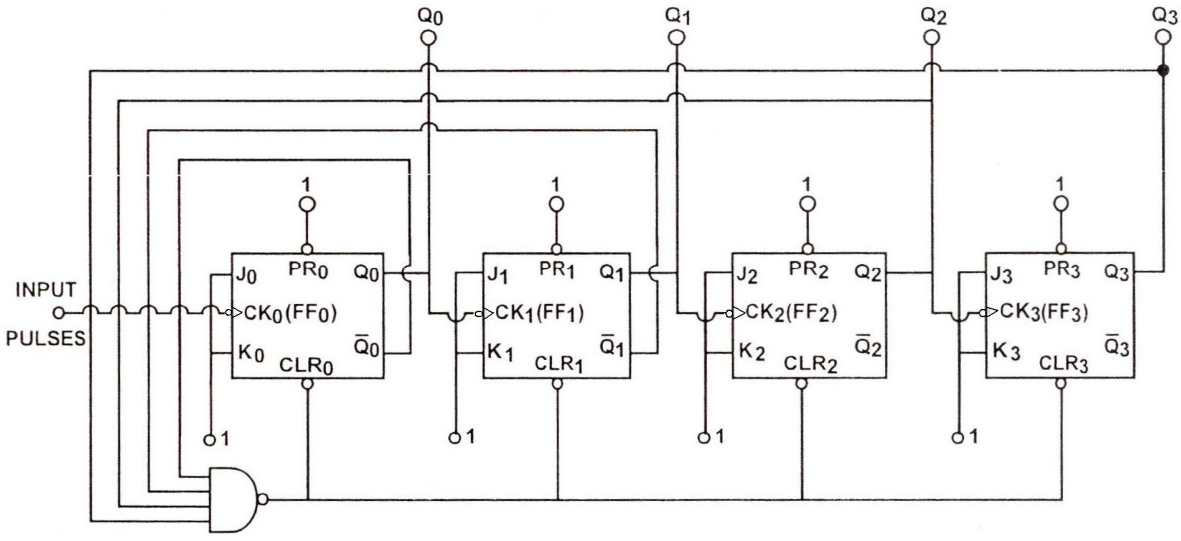
Input pulse	Binary count				Decimal count
	Q_3	Q_2	Q_1	Q_0	
0	0	0	0	0	0
1	1	1	1	1	15
2	1	1	1	0	14
3	1	1	0	1	13
4	1	1	0	0	12
5	1	0	1	1	11
6	1	0	1	0	10
7	1	0	0	1	9
8	1	0	0	0	8
9	0	1	1	1	7
10	0	1	1	0	6
11	0	1	0	1	5
12	0	1	0	0	4
13	0	0	1	1	3
14	0	0	1	0	2
15	0	0	0	1	1
16	0	0	0	0	0 (RESET)

Modulus-12 Counters:

The modulus of a counter is the number of discrete states a counter can take up. A counter with n no. of flip flops will have 2^n number of possible states. So counters with modulus, for example, 2, 4, 8, 16, can be built up using 1, 2, 3, 4 flip flops. It is quite often desirable to construct a counter having a modulus of 5, 9 or 12 etc. To design counters of modulus-12 (say), one has to use a modulus 16 counter and to arrange the circuit in such a way that it skips some of its natural states restricting it to12. The simplest way of doing this is the direct clearing method, where a gate circuit is used to clear all the flip flops as the desired count is reached. Thus, for a modulus N counter, the number n of flip-flops should be such that n is the smallest number for which 2^n>N and then to skip the surplus states with some rearrangements of the circuit.

The circuit diagram for a Mod-12 counter is shown below. It is obvious that a mod-12 counter will require 4 flip-flops which when connected as a counter, will provide 16 states. This counter counts 0, 1, 2, .., 15 and then it resets to 0. For a mod-12 counter, one may skip state 12 and return to state 0 from state 11 and the cycle should continue this way. For this an additional combinational logic circuit, i.e. a 4-input NAND gate is required, whose output is connected to clear terminal of all the flip flops. This will feed a reset pulse to the counter during state 12 (1100) and immediately after state 11 (1011). The flip-flops are reset and the counter starts counting again.

Circuit diagram:



Characteristic Table:

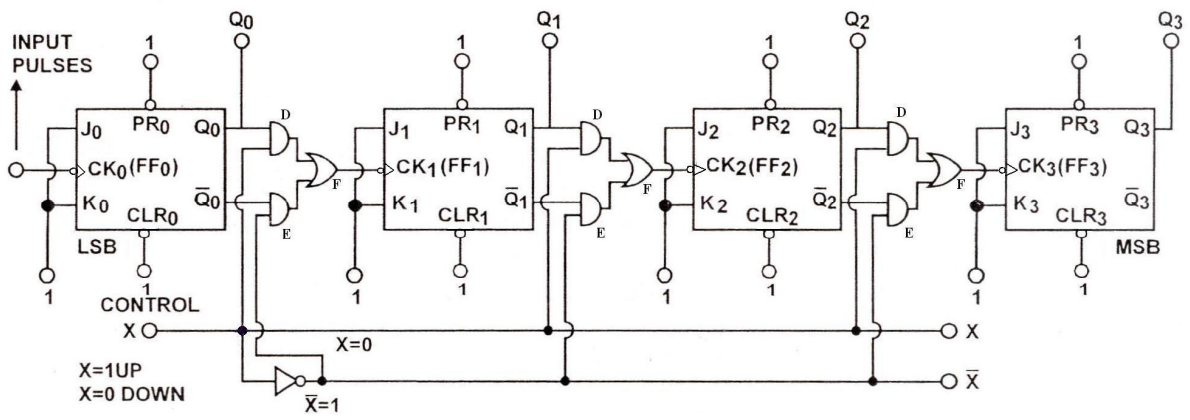
No. of Input pulses	Binary states				Decimal count
	Q_3	Q_2	Q_1	Q_0	
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	2
3	0	0	1	1	3
4	0	1	0	0	4
5	0	1	0	1	5
6	0	1	1	0	6
7	0	1	1	1	7
8	1	0	0	0	8
9	1	0	0	1	9
10	1	0	1	0	10
11	1	0	1	1	11
12	0	0	0	0	0

Up-Down counter:

The counters considered so far are capable of only count up or down. With some modification of the circuitry, one can use the same circuit to count either up or down. This can be achieved by connecting the normal and complement output of flip-flops to two AND gates, D and E, and the output of theses AND gates is fed to the clock input of the next flip-flop via an OR gate F, as shown in the circuit diagram. The rest two terminals of the AND gates are connected to an up-down control X.

When the up-down control is at binary 1 state, gates D and F are enabled and gate E is inhibited due to inverted input. Thus the normal output of each flip-flop is coupled via OR gate F to the clock input of next flip-flop and the counter counts up. Similarly, when the up-down control is at binary 0 state, gate D is inhibited and gates E and F are enabled. Hence, the complement output of each flip-flop is connected to the clock input of next flip-flop and the counter counts down.

Circuit diagram:



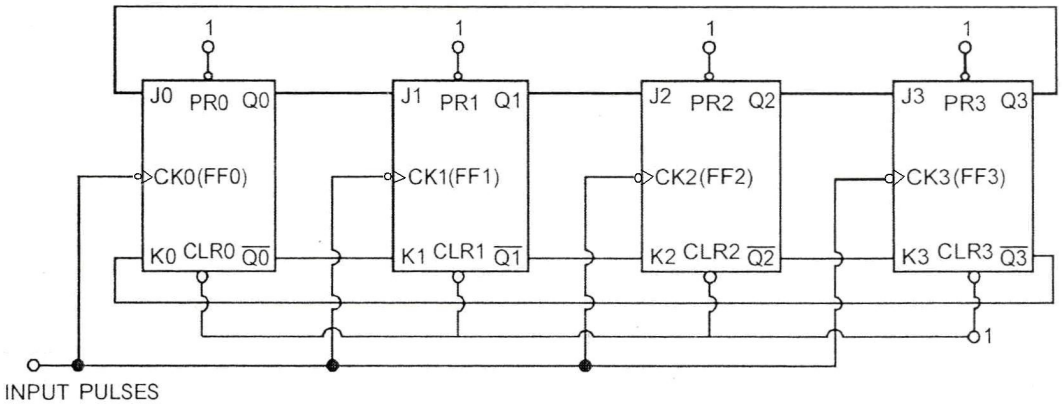
Characteristic Table: Same as Up- and Down-counter, for X=1 and 0, respectively.

Ring Counter:

Ring counters provide a sequence of equally spaced timing pulses and hence find considerable application in logic circuits which require such pulses for setting in motion a series of operations in a predetermined sequence at precise time intervals. A ring counter consists of an array of coupled flip-flops and the last flip-flop is coupled back to the first as shown in the circuit diagram. If one of the flip-flops is in the SET (or 1) state and the others are in the RESET (or 0 state) and then applying clock pulses, the logic 1 will advance by one flip-flop around the ring for each pulse. The sequence of operation of the ring counter is summarized in the characteristic table. The logic 1 will return to the original flip-flop after exactly 4 clock pulses (shown in shades) for a 4-bit ring counter.

Ring counter is extremely fast but it is uneconomical in the number of flip-flops (A simple mod-8 counter requires 4 flip-flops whereas a mod-8 ring counter needs 8!!). This is overcome by a modified circuit known as a Johnson counter or switchtail ring counter or twisted counter, where the outputs of the last flip-flop are crossed over and then fed back to the first flip-flop. That is the normal and complement output of the last flip-flop are connected to the K and J inputs of the first flip-flop respectively.

Circuit diagram:



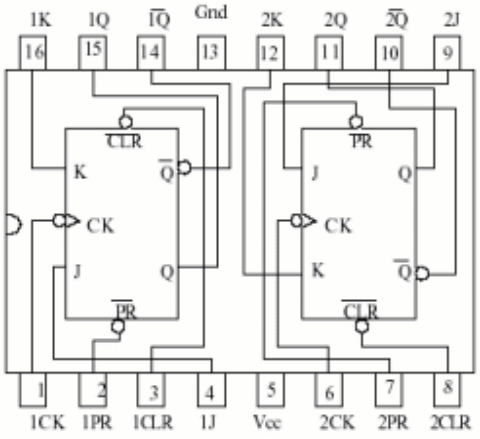
Characteristic Table:

No. of Input pulses	Binary states			
	Q_3	Q_2	Q_1	Q_0
0	0	0	0	1
1	0	0	1	0
2	0	1	0	0
3	1	0	0	0
4	0	0	0	1
5	0	0	1	0
6	0	1	0	0
7	1	0	0	0
8	0	0	0	1

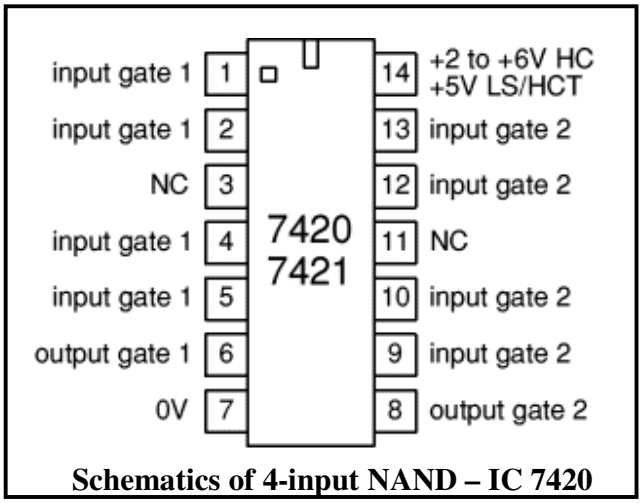
Circuit components/Equipments:

1. Resistors (1KΩ, 5 Nos)
2. ICs [JK FF-7476, 2 Nos; OR-7432,1 No; 2-input AND- 7408, 2 Nos; 4-input NAND -7420, 1 No; NOT-7404, 1 No]
3. A Surface mount dip switch
4. D.C. Power supply (5V)
5. Function Generator
6. Oscilloscope
7. Red/Green LEDs (4 Nos)
8. Connecting wires
9. Breadboard

Circuit Diagrams: Already provided with text.



Schematics of IC 7476



Schematics of 4-input NAND – IC 7420

Procedure:

1. Assemble the circuits one after another on your breadboard as per the circuit diagrams. Circuit diagrams given here do not show connections to power supply and LEDs assuming that you are already familiar with it from your previous lab experience. Here, all the LEDs are connected to the normal output of each flip flop. You will also use the oscilloscope to compare the timing diagrams of each of the output terminal and the input.
2. Connect the ICs properly to power supply and ground following the schematics for ICs given above.
3. Using dip switch and resistors, facilitate the required inputs from the power supply to the J, K, Pr and Cr terminals of the IC.
4. Use the function generator to facilitate clock pulse input to the circuit.
5. Turn on power to your experimental circuit.
6. **Reset** the circuit before applying pulse. **For ring counter preset the first flip-flop to give 1 at its normal output before applying pulse.**
7. Set the function generator in "Pulse" mode by pressing the "Function" button. Set the frequency at a very low value (~ 1 Hz, amplitude ~ 5V) so that you can notice the logic states of the normal outputs indicated by the LEDs (ON = 1; OFF = 0).
8. The logic states of the J, K inputs must not be allowed to change when clock is high.
9. Record the normal output states of all the flip flops in a table for every pulse applied. Compare your results with the characteristic table for operation.
10. **Feed the input and each of the output of the up-counter to oscilloscope.** Save the waveforms (Timing diagram) and compare their frequencies.
11. When you are done, turn off the power to your experimental circuit.

Observations:

Table for ripple Up-counter: ___ Timing Diagram for up-counter:_____

Table for ripple Down-counter: ____ Table for Mod-12 counter:_____

Table for ripple Up/Down-counter: ____ Table for ring counter : _____

Discussions:

Precautions:

1. Watch out for loose connections.
2. The logic states of the J, K inputs must not be allowed to change when clock is high.

<u>Appendix:</u> IC 7476 datasheet

Inputs					Outputs	
PR	**CLR**	**CLK**	**J**	**K**	**Q**	$\overline{Q}$
L	H	X	X	X	H	L
H	L	X	X	X	L	H
L	L	X	X	X	H (Note 1)	H (Note 1)
H	H	⎍	L	L	Q_0	$\overline{Q}_0$
H	H	⎍	H	L	H	L
H	H	⎍	L	H	L	H
H	H	⎍	H	H	Toggle	

H = High Logic Level
L = Low Logic Level
X = Either Low or High Logic Level
⎍ = Positive pulse data. The J and K inputs must be held constant while the clock is high. Data is transfered to the outputs on the falling edge of the clock pulse.
Q_0 = The output logic level before the indicated input conditions were established.
Toggle = Each output changes to the complement of its previous level on each complete active high level clock pulse.

Note 1: This configuration is nonstable; that is, it will not persist when the preset and/or clear inputs return to their inactive (high) level.