UNIT 7

SEQUENTIAL LOGIC

COUNTERS AND REGISTERS

2ND SESSION

Asynchronous and synchronous counter

Counter:

- Counters are sequential logic circuits that proceed through a well defined sequence of states after application of clock pulses.
- Special type of registers (Combination of Flip-flops) with a capability of counting.
- Counters are used for a counting pulses.
- Counters are constructed using Flip-flops and logic gates.
- One of the most useful and versatile subsystem in digital system.
- Sequential ckt. that passes through predefined no. of states.

- Two types:
  i. **Asynchronous Counter**
  ii. **Synchronous counter**

Comparison Basis	Synchronous Counter	Asynchronous Counter
Also called	Parallel Counter	Serial Counter
Principle of operation	Each flip flop is triggered with same clock signal at the same time.	Each flip flop is triggered with different clock signal at different instant of time.
Decoding errors	Not produced	Produced
Operating speed	Fast	Comparatively slow
Design	Complex	Simple
Signal propagation	Very Low	Comparatively high
Count sequence	Not Fixed	Fixed
Response to clock signal	Each flip-flop changes its state simultaneously.	There is no simultaneous change in the state of all flip flops with change in clock input.
Applications	In moving machine controlling, alarms clocks, multiplexing circuits, etc.	In ring and johnson counters, frequency dividers, etc.

Asynchronous or ripple counters

- Clock 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.
- An n-bit **Asynchronous counter can have 2^n** possible counting states e.g. MOD-8 for a 3-bit counter have (0-7) states.
- But it is also possible to use the basic asynchronous counter configuration to construct special counters with counting states less than their maximum output number.

- This is achieved by forcing the counter to reset itself to zero at a pre-determined value producing a type of asynchronous counter that has truncated sequences.
- Then an n-bit counter that counts up to its maximum modulus ($\mathbf{2^n}$) is called a **full sequence counter** and a n-bit counter whose modulus is less than the maximum possible is called a **truncated counter.**

1. 2-bit Asynchronous counter (MOD-4 ripple counter):

- Use two T flip-flops.
- we can also use the JK flip flop by setting both of the inputs to 1 permanently.
- The external clock pass to the clock input of the first flip flop, i.e., FF-A and its output, i.e., is passed to clock input of the next flip flop, i.e., FF-B.

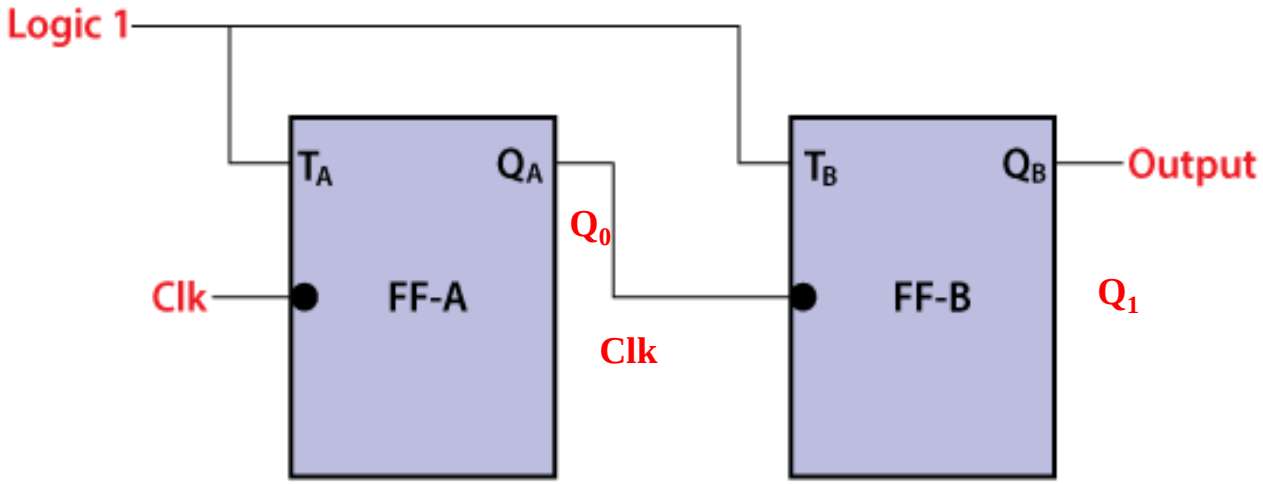


Fig: Block Diagram

Clock	Counter output		State number	Deciimal Counter output
	Q_B	Q_A		
Initially	0	0	—	0
1st	0	1	1	1
2nd	1	0	2	2
3rd	1	1	3	3
4th	0	0	4	0

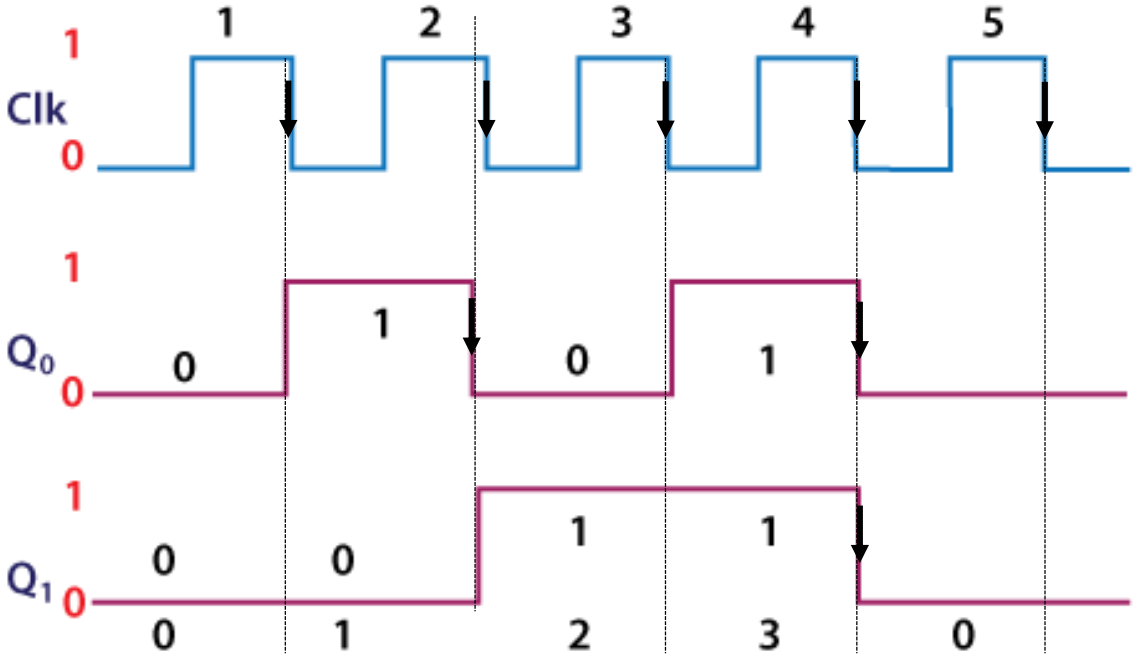


Fig: Signal/Timing Diagram

CLK	Q1	Q0
0	0	0
1	0	1
2	1	0
3	1	1

Truth Table

Operation

1. **Condition 1:** When both the flip flops are in reset condition.
   Operation: The outputs of both flip flops, i.e., Q_A Q_B, will be 0.
2. **Condition 2:** When the first negative clock edge passes.
   Operation: The first flip flop will toggle, and the output of this flip flop will change from 0 to 1. The output of this flip flop will be taken by the clock input of the next flip flop. This output will be taken as a positive edge clock by the second flip flop. This input will not change the second flip flop's output state because it is the negative edge triggered flip flop. So, $Q_A = 1$ and $Q_B = 0$
3. **Condition 3:** When the second negative clock edge is applied.
   Operation: The first flip flop will toggle again, and the output of this flip flop will change from 1 to 0. This output will be taken as a negative edge clock by the second flip flop. This input will change the second flip flop's output state because it is the negative edge triggered flip flop. So, $Q_A = 0$ and $Q_B = 1$.

4. **Condition 4:** When the third negative clock edge is applied. **Operation:** The first flip flop will toggle again, and the output of this flip flop will change from 0 to 1. This output will be taken as a positive edge clock by the second flip flop. This input will not change the second flip flop's output state because it is the negative edge triggered flip flop. So, $Q_A = 1$ and $Q_B = 1$
5. **Condition 5:** When the fourth negative clock edge is applied. **Operation:** The first flip flop will toggle again, and the output of this flip flop will change from 1 to 0. This output will be taken as a negative edge clock by the second flip flop. This input will change the output state of the second flip flop.

   So, $Q_A = 0$ and $Q_B = 0$

3-bit Ripple Up Counter (MOD-8)

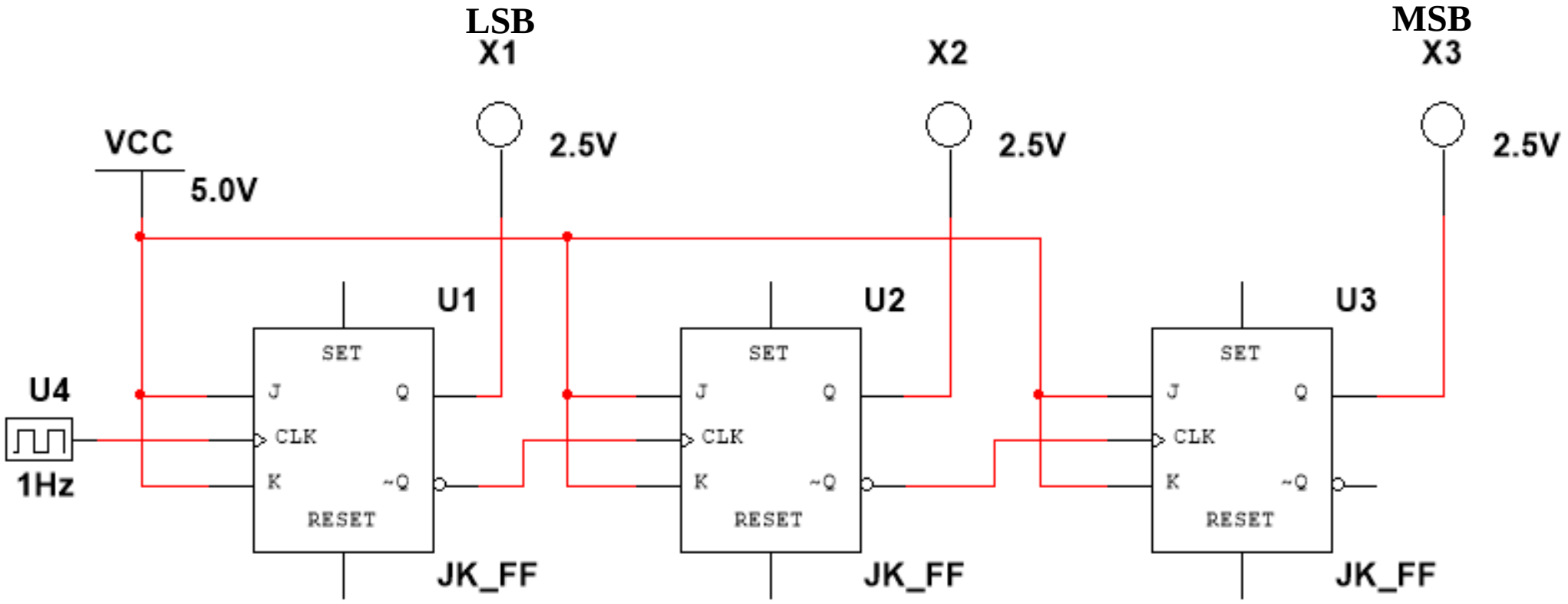


Fig: 3-bit Ripple up counter Logic Diagram

- Constructed using JK FF.
- JK inputs are tied up together to +5 VCC such at each +ve rising edge transition of clock input, the FF toggles its states.
- The complemented output of first FF act as clock for the next FF and so on.

Truth Table

CLK	Q2	Q1	Q0
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

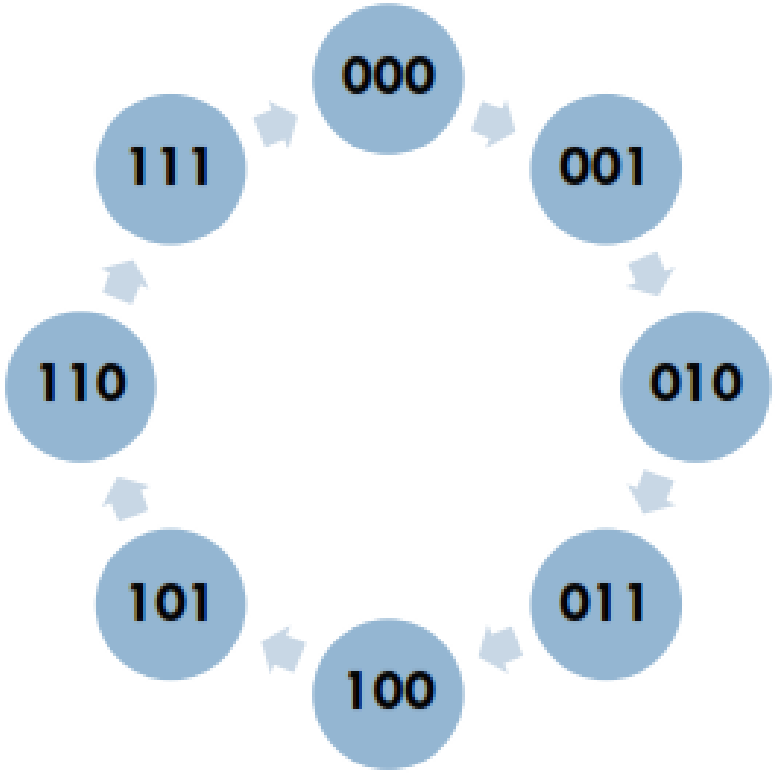
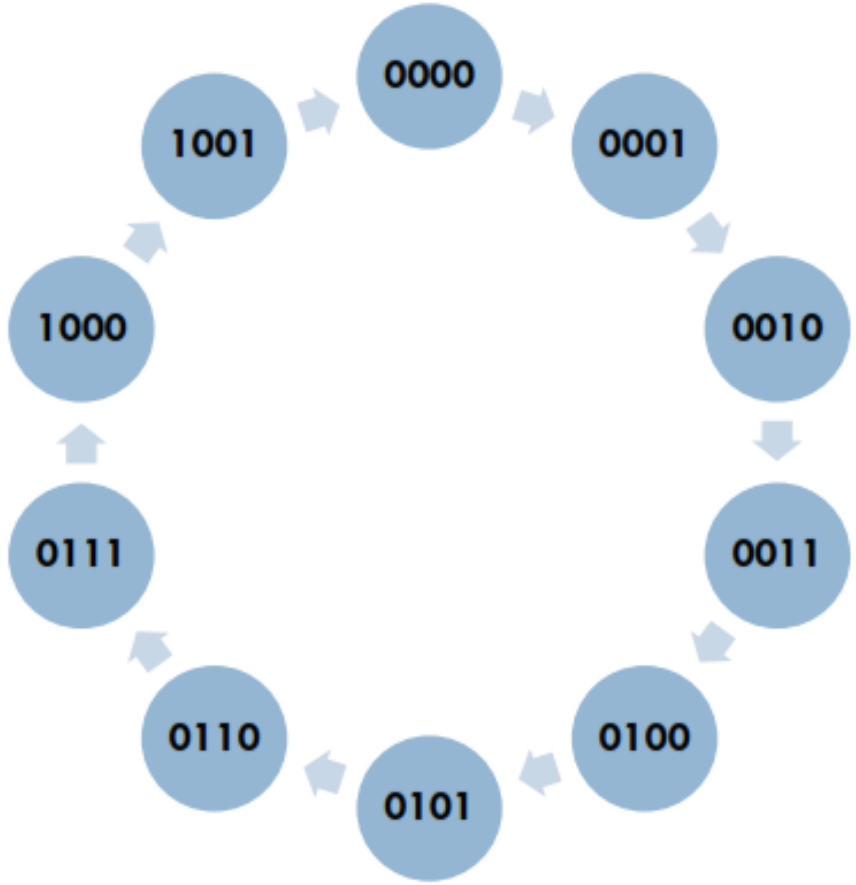


Fig: State Diagram

Ripple Decade Counter (Truncated)

- **Decade Counter (MOD-10):**
- Also called **BCD counter**
- For a counter to count from ***0000 to 1001***, ***four flip flops are required***. But we need to mechanism to restrict the count to 1001 and thereafter reset the counter to 0000 again otherwise our counter (with 4 flip flops) will continue to 1111 making it MOD-16 counter instead of MOD-10.



When the counter counts $Q_DQ_CQ_BQ_A$=1010, then the outputs of NAND gate is low, which clears all the FFs.

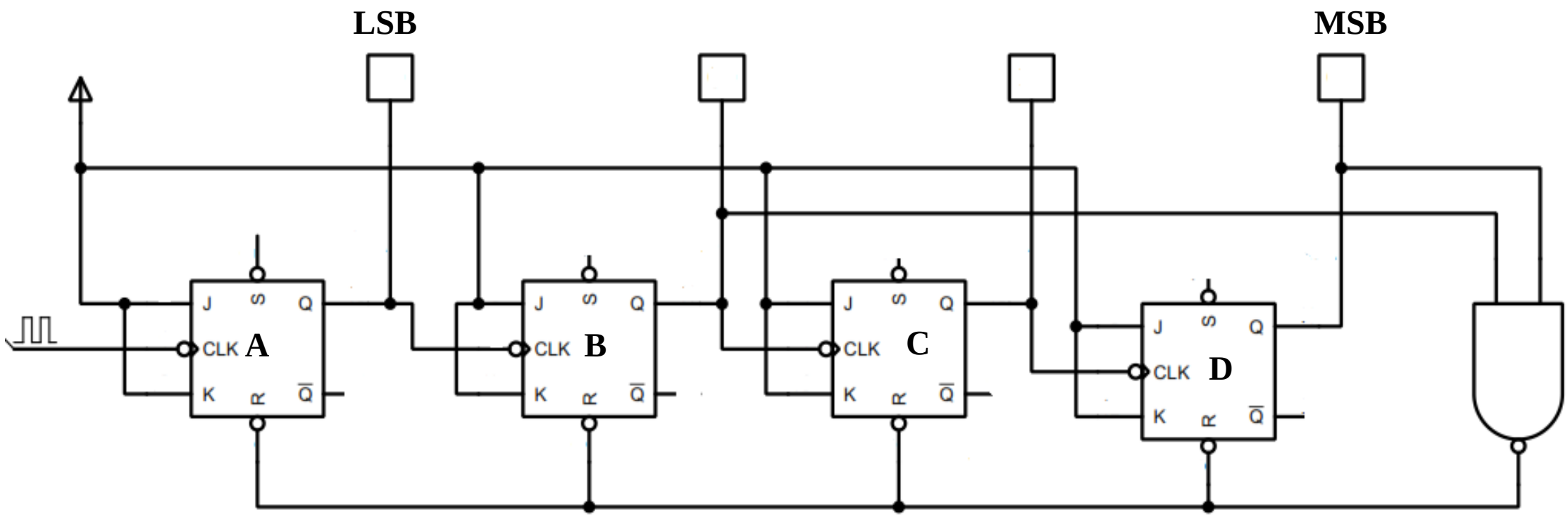


Fig: MOD-10 Ripple up counter Logic Diagram

Exercise 1

a) Design and explain MOD-16 Asynchronous UP counter.
b) Design and explain MOD-8 Asynchronous UP counter using D Flip-flops and PLA.
c) Design and explain MOD-5Asynchronous UP counter.

Synchronous Counter (Parallel counter)

- All the FFs are clocked together at the same time by same clock.
- No propagation delay like in asynchronous counter.
- The result of this ***synchronization*** is that all the individual output bits changing state at ***exactly the same time*** in response to the common clock signal with ***no ripple effect*** and therefore, ***no propagation delay***.
- For n-bit counter 2^n possible states with n-FFs.

Counter design synthesis

Following are the steps:

i. Draw the state diagram from the given word description problem.
ii. Draw the next state table and find the number of FFs required. Number of FFs =Number of bits used.
iii. Decide the type of FF to be used for the design, then determine the FF excitation table based on the transition of present state **(Q_n)** to next state **(Q_{n+1}).**
iv. Prepare the k-map for each FF input and simplify.
v. Connect the ckt. Using FF and the other gates according to the minimized expression.

Binary 3-bit Synchronous Up Counter

- Also called MOD-8 Synchronous counter

Step 1: State diagram

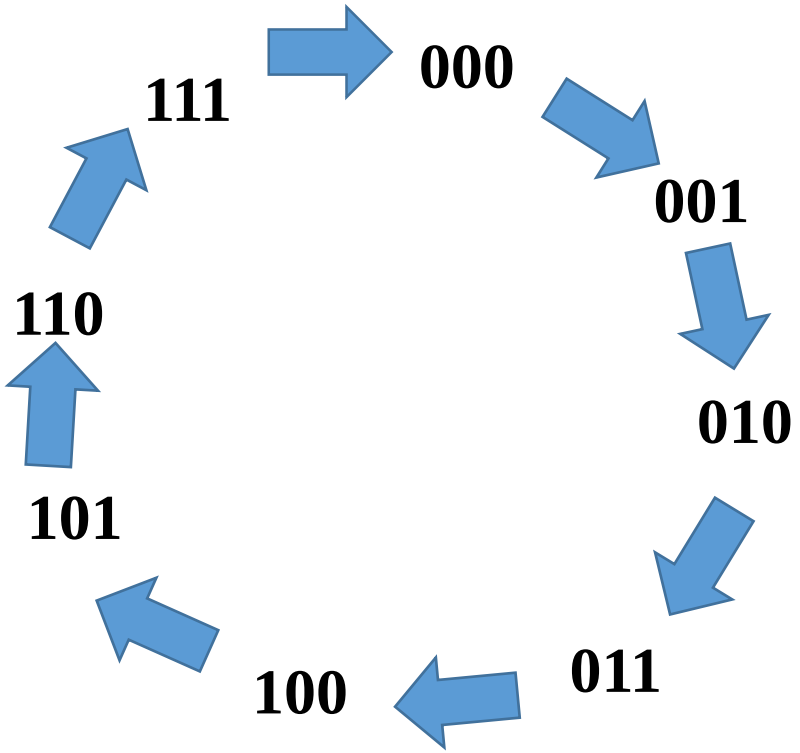


Fig: state diagram

Step 2 and 3: Next state table and excitation table

Present state(Q_n)			Next state(Q_{n+1})			FF excitation					
Q_{An}	Q_{Bn}	Q_{Cn}	Q_{An+1}	Q_{Bn+1}	Q_{Cn+1}	J_A	K_A	J_B	K_B	J_C	K_C
0	0	0	0	0	1	0	X	0	X	1	X
0	0	1	0	1	0	0	X	1	X	X	1
0	1	0	0	1	1	0	X	X	0	1	X
0	1	1	1	0	0	1	X	X	1	X	1
1	0	0	1	0	1	X	0	0	X	1	X
1	0	1	1	1	0	X	0	1	X	X	1
1	1	0	1	1	1	X	0	X	0	1	X
1	1	1	0	0	0	X	1	X	1	X	1

Step 4: Prepare the k-map for J and K input from transition table in step 3.

For J_A

Q_A \ Q_BQ_C	00	01	11	10
0	0	0	*1*	0
1	*X*	*X*	*X*	*X*

$J_A=Q_BQ_C$

For K_A

Q_A \ Q_BQ_C	00	01	11	10
0	*X*	*X*	*X*	*X*
1	0	0	*1*	0

$K_A=Q_BQ_C$

For J_B

Q_A \ Q_BQ_C	00	01	11	10
0	0	1	*X*	*X*
1	0	1	*X*	*X*

$J_B=Q_C$

For K_B

Q_A \ Q_BQ_C	00	01	11	10
0	X	X	1	0
1	X	X	1	0

$K_B=Q_C$

For J_C

Q_A \ Q_BQ_C	00	01	11	10
0	1	*X*	*X*	1
1	1	*X*	X	1

$J_C=1$

For K_C

Q_A \ Q_BQ_C	00	01	11	10
0	*X*	1	1	*X*
1	X	1	1	X

$K_C=1$

Step 5:

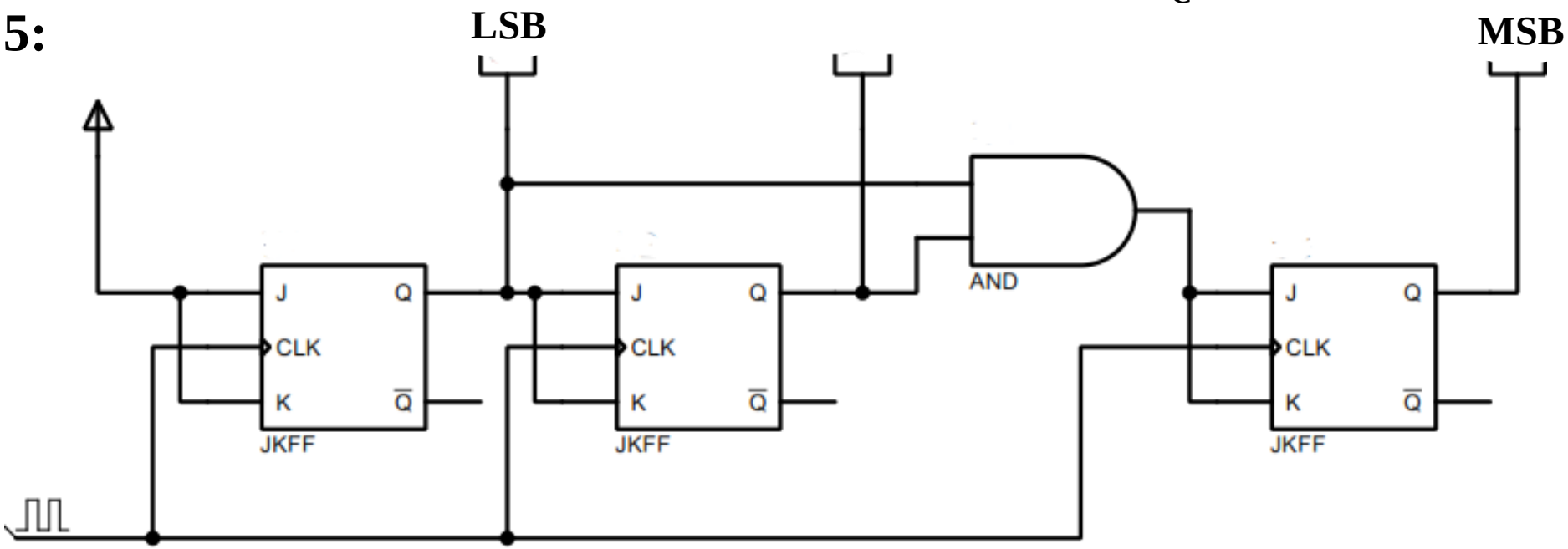


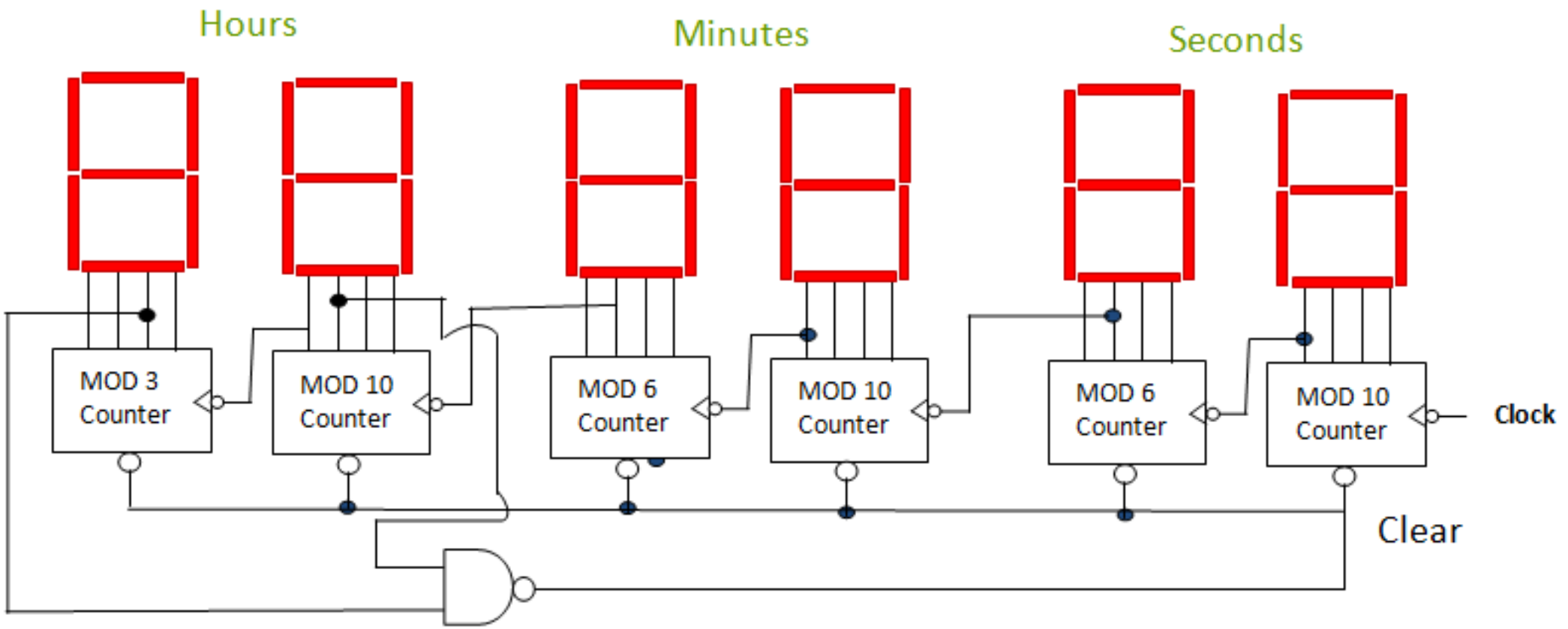
Fig: 3-bit Synchronous Mod-8 Counter logic diagram

Application of counters

a) Digital Watch
b) Frequency counters

a) Digital Watch

- Interesting application of counter and decoder.
- Ordinary clock: Hour, minutes and seconds
- ***Input:*** 60Hz,129V_{AC}
- T=1/f=1/60s
- For obtaining 1s pulse, f/60=1Hz=1s
- For obtaining one per minute pulse, 1Hz/60=1cycle.min.
- Dividing this signal by 60 provides a one-per hour wave form.


Hours
Minutes
Seconds
MOD 3 Counter
MOD 10 Counter
MOD 6 Counter
MOD 10 Counter
MOD 6 Counter
MOD 10 Counter
Clock
Clear

b) Frequency counters

- Used to measure the frequency of any periodic waveform.