Register and Counters

Q9) Design a 3 bit asynchronous ripple counter with timing diagram

=> Logic1 :

CLK=

~~Logic2~~

Logic2

Q1 (LSB)

Q2

$\overline{Q3}$ (MSB)
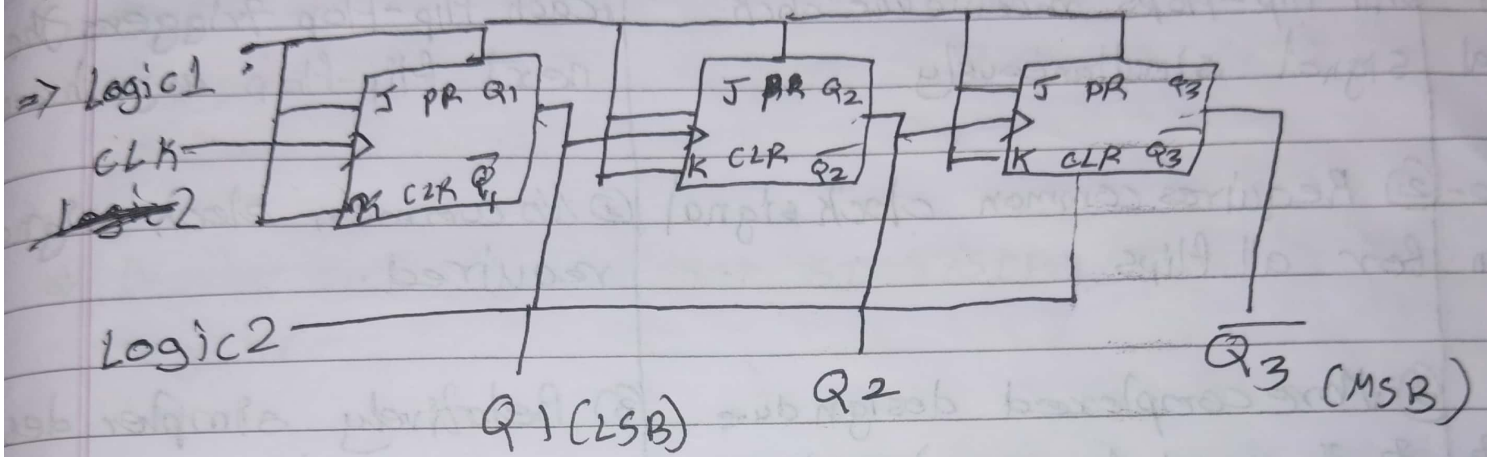
CLK	Q_3	Q_2	Q_1	state	D.E
	0	0	0	1	0
↑	0	0	1	2	1
↑	0	1	0	3	2
↑	0	1	1	4	3
↑	1	0	D	5	4
↑	1	0	1	6	5
↑	1	1	0	7	6
↑	1	1	1	8	7

Q10) Design Modulo 5 ripple counter using a 3bit ripple counter

Q13) Compare Synchronous and asynchronous ripple counter.

Features	Synchronous	Asynchronous
Clock- Signal	① All flip-flops receive the clock signal simultaneously	① Each flip-flop triggers the next flip-flop asynchronously
Synchro-nization	② Requires common clock signal for all flips	② No common clock signal required.
Design Complexity	③ More complexed design due to the need for synchronized Clock signal	③ Relatively simpler design
Speed	④ Faster operation due to synchronized clock design	④ Slower operation due to Asynchronous Triggering
Timing-Analysis	⑤ Easier to perform timing Analysis	⑤ More challenging to perform Timing Analysis
Application	⑥ Used in applications requiring precise timing and synchronous operation	⑥ Commonly used In simple application where speed is not critical

Q14) State all possible mode of operation of registers.

➡ ① SISO:-

➡ ① Serial In Serial Out (SISO):- Data is shifted in serially, one bit at a time, usually synchronous with a clock signal

② Serial In - Parallel Out (SIPO) :- Data is shifted in
serially and then output in parallel.
③ Parallel In / Serial out (PISO) :- Dat is shifted in
parallel and then sifted out serially.
④ Parallel In / Parallel out (PIPO) :- Data is shifted
in parallel and then ~~sifted shifted out parallel~~.
output in parallel

5) Describe the function of 3bit SISO/SIPO/PIPO/PISO
register :-

→ ① A 3 bit Serial - In - Serial - Out (SISO) shift register is a
digital circuit that can store and shift a 3-bit binary
number, one bit at a time.

Ⅱ) Working :-
1] Serial - In :- ⓘ The 3 bit binary number is input serially one
bit at a time, into the shift register. Each Iinput bit is
loaded in the first flip-flop (LSB) on the rising edge of the
clock signal.
2] Shift :- i) Once the 3-bits are loaded into the shift
register, they can be shifted to right (or left) serially.
ii) Shifting is done by applying clock ~~signal~~ pulse.
iii) Each clock pulse causes the bit to move position to the
to right (or left), with the LSB moving to the next
flip-flop and MSB moving out of the register.
3] Serial -Out :- i) The output of the shift register is taken
from the last flip flop. As the bits are shifted, the MSB
changes to the reflect the current value of 3-bits
number stored in the register
~~4] Control signal :- In addition to clock signal~~

For SI-PO

I} For 3 bits SIPO

① {This point is same as last one first point }

Ⅱ} Working:-

1] Serial In :- { Its same as last one }

2] Shift :- { Its same as last one }

3) Parallel Out :- ① The output of shift registed is avalaible in parallel form, the outputs of all the Flip-Flops.
② Each flip-flop output corresponds to one bit of the 3-bit binary number stored in the register

4] Application:- ① Its used in application where serial data is to be converted to parallel data for further processing - 2) Application such as LED matrix control

Ⅲ} For PISO :-

I{Same as above just chage serial with parallel and parallel with serial }

Ⅱ} Working

1] Parallel-In :- The 3-bit binary number is input in parallel to the shift register. Each bit is loaded into a separate flip-flop (one for each bit). simultaneously.

2] Shift :- {Same as above }

3] Serial Out :- { Same as 1st (SISO) }

4] Applications :- 1} Used in applications where parallel data needs to be converted to serial data. 2) Applications like serial communication Interface, data communication on a single wire, and In some type of sensor.

Ⅳ} PIPO

Ⅲ} {Same as above just chang all to parralle }

Ⅰ} Working

I) Parallel In :- {Same as above }

Ⅱ) Parallel Out :- { Same as the top one}

III} Application:- i) Application ~~like~~ wher parallel data needs to be ~~st~~-stroed temporarily and then output is parallel.

ii) Applications like memory units, parallel data processing, interfacing between parallel and serial system