Name : Irfanullah

ID : 15431

Final-Term : Assignment

Subject : Digital Logic & Design

Teacher : Muhammad Amin

Program : BS (CS)

Q.1 Draw and explain the logic diagram for each of the following:

a) A circuit for adding or subtracting two 4-bit numbers

b) 4-bit active low decoder

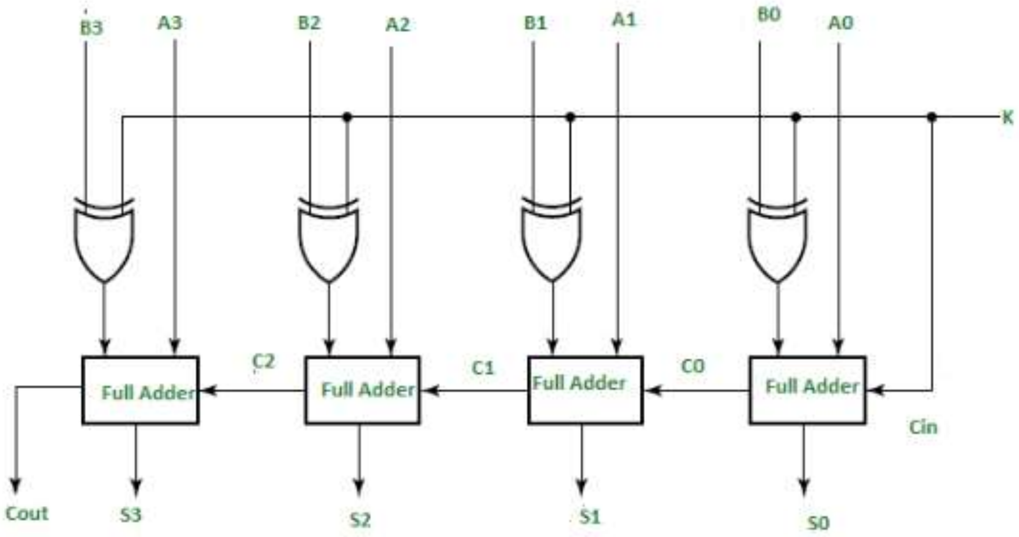
c) Decimal to BCD encoder

d) Frequency divider (Use 3 J-K flip-flops and assume 16 kHz frequency of the initial wave-form.)

(a)

A circuit for adding or subtracting two 4-bit numbers

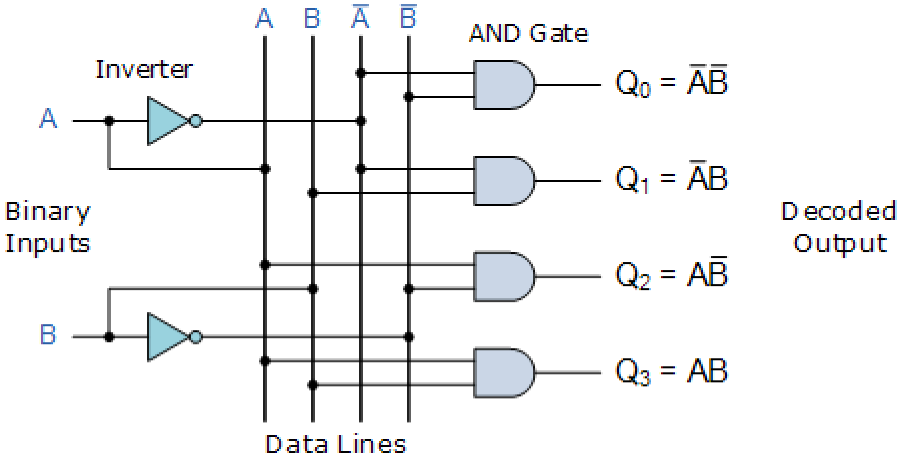
The circuit consists of 4 for bits since we are performing operation on 4-bit numbers. There is a control line K that holds a binary value of either 0 or 1 which determines that the operation being carried out is addition or subtraction.

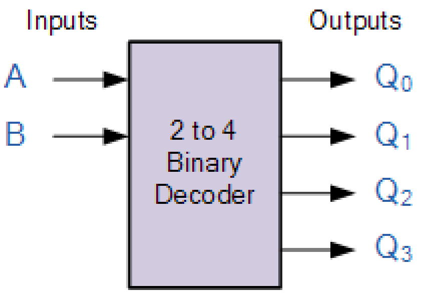


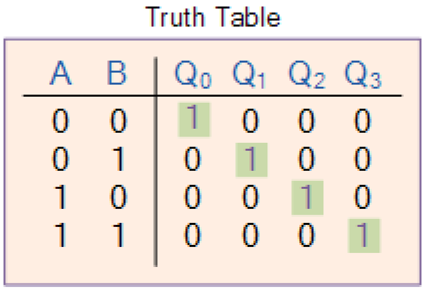
(b)

4-bit active low decoder

4 bit decoder:





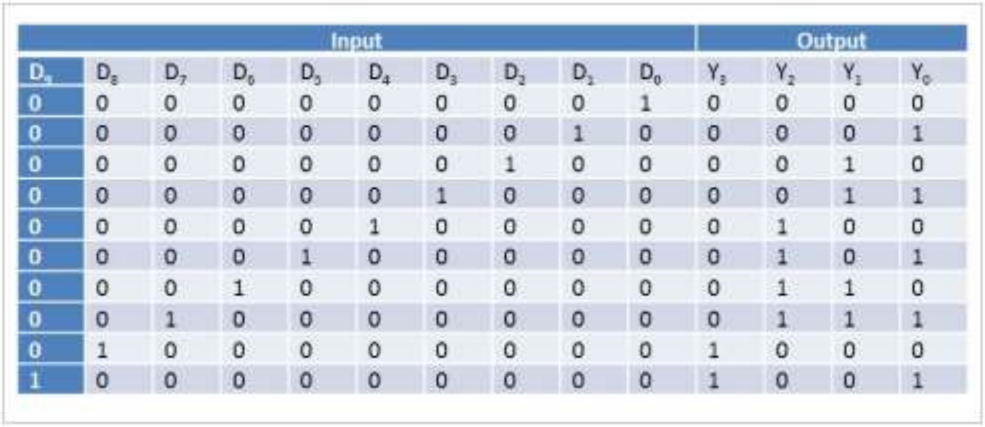

Truth Table

A	B	Q_0	Q_1	Q_2	Q_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

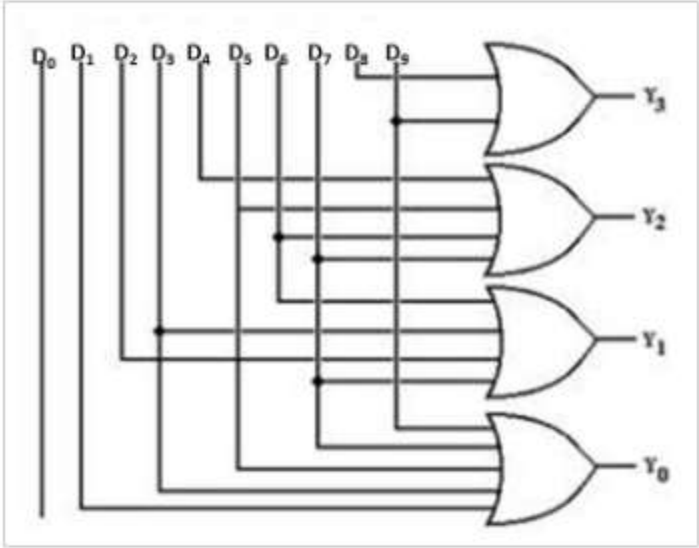
(C) Decimal to BCD encoder

Decimal To BCD Encoder

A decimal to BCD has 10 input lines D_0 to D_9 and 4 output lines Y_0 to Y_3. Below is the **truth table for a decimal to bcd encoder**.



Input										Output			
D_9	D_8	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0	1
0	0	0	0	0	0	0	1	0	0	0	0	1	0
0	0	0	0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	0	1	0	0	0	0	0	1	0	0
0	0	0	0	1	0	0	0	0	0	0	1	0	1
0	0	0	1	0	0	0	0	0	0	0	1	1	0
0	0	1	0	0	0	0	0	0	0	0	1	1	1
0	1	0	0	0	0	0	0	0	0	1	0	0	0
1	0	0	0	0	0	0	0	0	0	1	0	0	1



Limitations With Decimal To BCD Encoder

A decimal to BCD encoder has limitations similar to octal to binary encoder.

1. For 10 bits input, there can be 2^{10}possible combinations, out of which only 10 are used using 4 output lines.
2. The discrepancy of 0 output due to all inputs being 0 or D_0 being 0 is resolved by using additional input known as Enable
3. Only one input can be active at any given time.

(d) Frequency divider (Use 3 J-K flip-flops and assume 16 kHz frequency of the initial wave-form.)

If the circuit of 3-bit counter

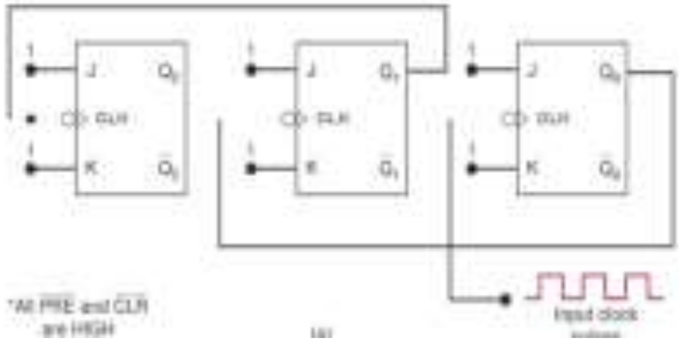
– Determine the counter's MOD number
- MOD number = 2 64 6 =

– Determine the Frequency of the last FFs (Q5) when the input frequency is 1 MHz
- f = 1MHZ/64 = 15.625 kHz

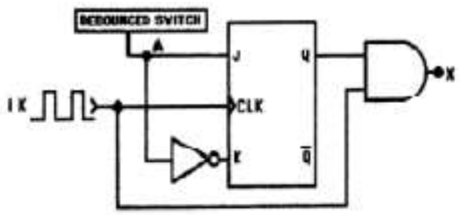
– What is the range of counting states for this counter?
- 000000 to 111111 (0 to 63), number of states = 64

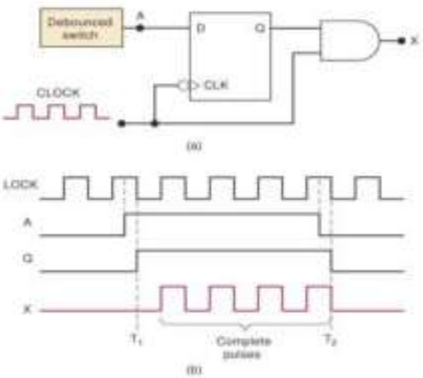
– Assume a starting state (000000). What will be the counter's state after 129 pulses? Slide - EET2411

DIGITAL ELECTRONICS 198
- Counter will be back to its starting state every 64 pulses. So the 129th Pulse will bring the counter to state (000001)



Modify the circuit to use a 3j -k instead





Q.2

For the 4-input multiplexer, data inputs are given as:

D0 = 0, D1 = 1, D2 = 0, D3 = 1

Find the output Y if the select inputs are given as:

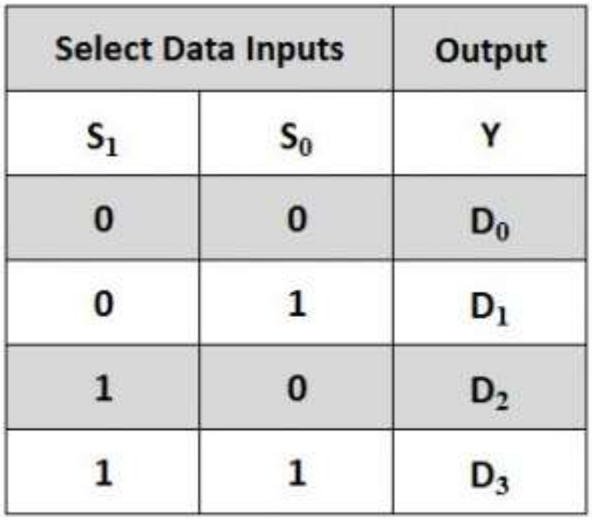
a) S0 = 1, S1 = 0

b) S0 = 0, S1 = 1

c) S0 = 1, S1 = 1

d) S0 = 0, S1 = 0

The truth table of a 4-to-1 multiplexer is shown below in which four input combinations 00, 10, 01 and 11 on the select lines respectively switches the inputs D0, D2, D1 and D3 to the output. That means when S1=0 and S0 =0, the output at Y is D0, similarly Y is D1 if the select inputs S1=0 and S0= 1 and so on.



Select Data Inputs		Output
S_1	S_0	Y
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

From the above truth table, we can write the output expressions as

If S1=0 and S0=0 then Y = D0

Therefore, Y = D0 (S1)¯(S0)¯

If S1= 0 and S0=1, the Y = D1

Therefore, Y = D1 (S1)¯S0

If S1=1 and S0=0, then Y = D2

Therefore, Y = D2 S1 (S0)¯

If S1=1 and S0=1 the Y = D3

Therefore, Y = D3 S1 S0

To get the total data output from the multiplexer, all these product terms are to be summed and then the final Boolean expression of this multiplexer is given as

$$\mathbf{Y = D0\,\overline{S1}\,\overline{S0} + D1\,\overline{S1}\,S0 + D2\,S1\,\overline{S0} + D3\,S1\,S0}$$

Q.3

Timing diagram in Figure 01 shows inputs to a 9-bit parity checker. Draw the ∑ Even and ∑ Odd output for the even parity checking.

9-bit odd/even parity generator/checker

PIN DESCRIPTION

PIN NO.	SYMBOL	NAME AND FUNCTION
8, 9, 10, 11, 12, 13, 1, 2, 4	I_0 to I_8	data inputs
5, 6	Σ_E, Σ_O	parity outputs
7	GND	ground (0 V)
14	V_{CC}	positive supply voltage

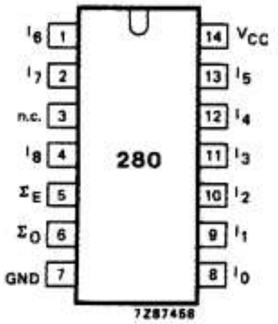


Fig.1 Pin configuration.

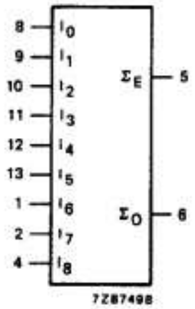


Fig.2 Logic symbol.

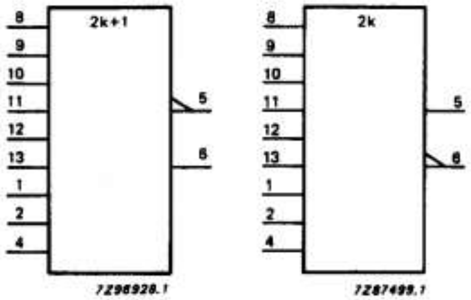


Fig.3 IEC logic symbol.

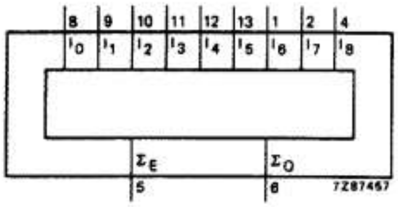


Fig.4 Functional diagram.

FUNCTION TABLE

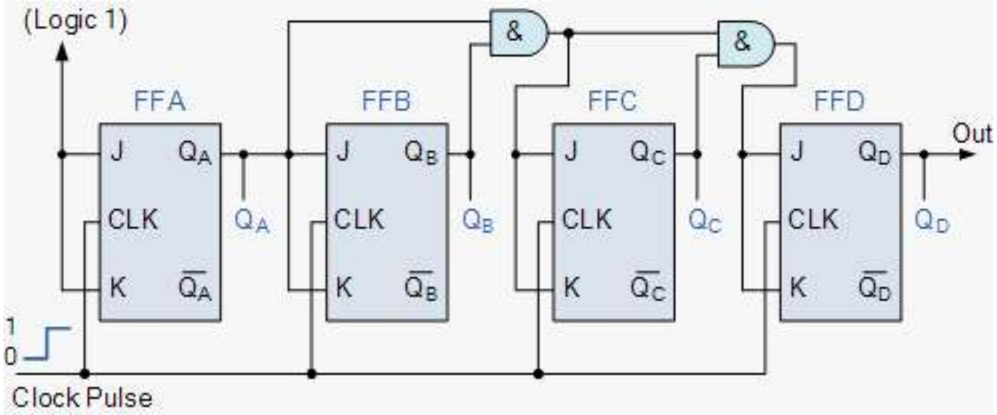
INPUTS	OUTPUTS	
number of HIGH data inputs (I_0 to I_8)	Σ_E	Σ_O
even odd	H L	L H

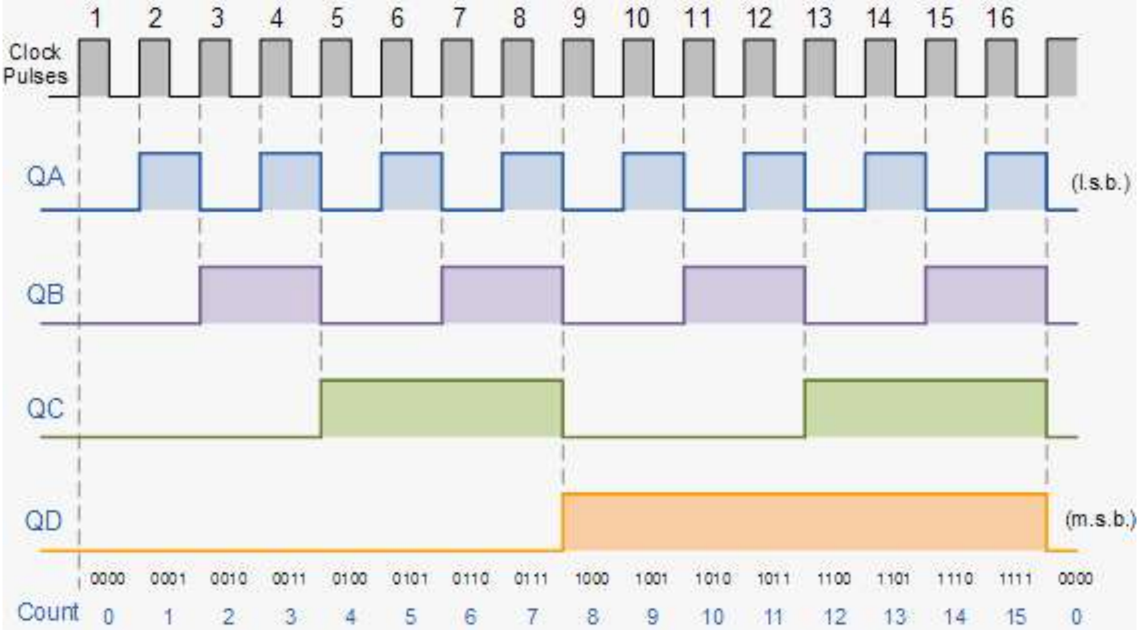
Note

1. H = HIGH voltage level
   L = LOW voltage level

Q.6

Draw the logic diagram and timing diagram for the 4-stage synchronous binary counter. Verify that the waveforms of the Q outputs represent the proper binary number after each clock pulse.



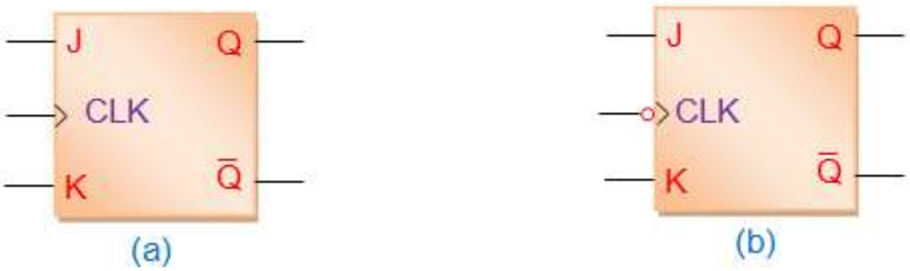


Q.4

The waveforms in Figure 02 are applied to the J, K, CLK, $\overline{PRE}$, and $\overline{CLR}$ inputs as indicated. Determine the Q output, if the flip-flop is initially RESET.

JK Flip Flop:

A JK flip-flop is a sequential bi-state single-bit memory device named after its inventor by Jack Kil. In general it has one clock input pin (CLK), two data input pins (J and K), and two output pins (Q and $\bar{Q}$) as shown in . JK flip-flop can either be triggered upon the leading-edge of the clock or on its trailing edge and hence can either be positive- or negative- edge triggered, respectively.



(a) (b)

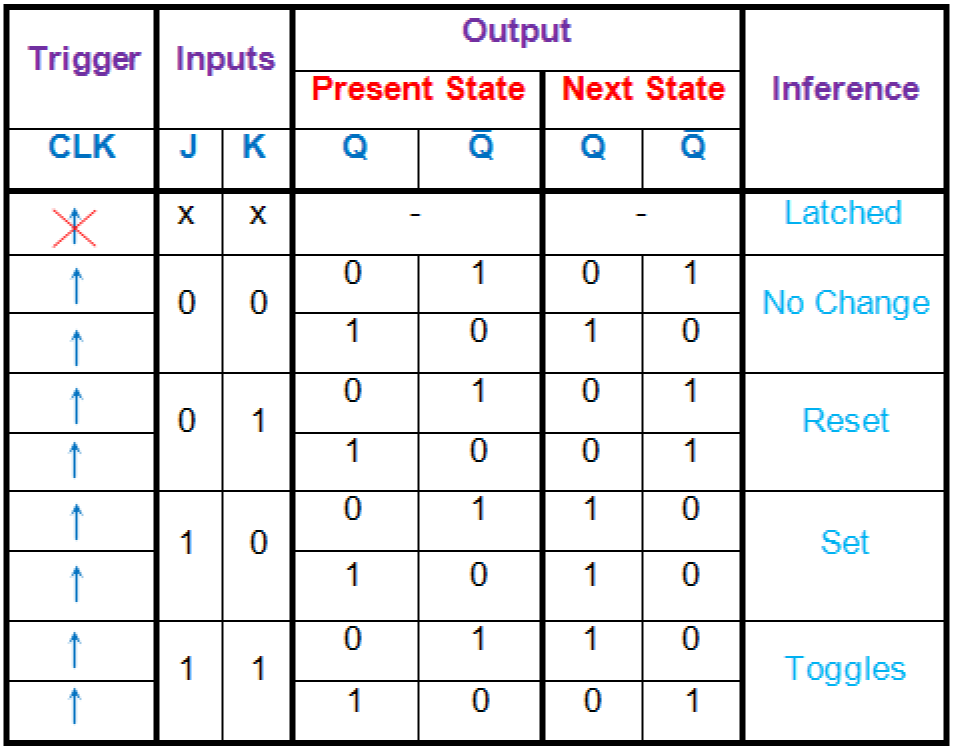
However if one considers the initial states to be J = K = 0, Q = 1 and $\bar{Q}$ = 0, then $X_1 = X_2 = 0$ which results in Q = 1 and $\bar{Q}$ = 0. This indicates that the state of flip-flop outputs Q and $\bar{Q}$ remains unchanged for the case of J = K = 0.

Now assume that J = 0, K = 1, Q = 0 and $\bar{Q}$ = 1. Analyzing on the same grounds, one gets $X_1 = X_2 = 0$ which further results in Q = 0 (and hence $\bar{Q}$ = 1). For the same case if Q and $\bar{Q}$ were 1 and 0, respectively, then $X_1 = 1$ and $X_2 = 0$ which would result in Q = 0 (and hence $\bar{Q}$ = 1).

This implies that if J = 0 and K = 1, then the flip-flop resets (Q = 0 and $\bar{Q}$ = 1).
Next if J = 1, K = 0, Q = 1 and $\bar{Q}$ = 0, then $X_1 = X_2 = 0$ which results in Q = 1 (and thus $\bar{Q}$ = 0). For the same case if Q = 0 and $\bar{Q}$ = 1, then $X_1 = 0$, $X_2 = 1$ which leads to $\bar{Q}$ = 0 and hence Q is forced to value 1. This means that for the case of J = 1 and K = 0, flip-flop output will always be set i.e. Q = 1 and $\bar{Q}$ = 0.

Similarly for J = 1, K = 1, Q = 1 and $\bar{Q}$ = 0 one gets $X_1 = 1$, $X_2 = 0$ and Q = 0 (and hence $\bar{Q}$ = 1); and if Q changes to 0 and $\bar{Q}$ to 1, then $X_1 = 0$, $X_2 = 1$ which forces $\bar{Q}$ to 0 and hence Q to 1. This indicates that for J = K = 1, flip-flop outputs toggle meaning which Q changes from 0 to 1 or from 1 to 0, and these changes are reflected at the output pin $\bar{Q}$ accordingly.

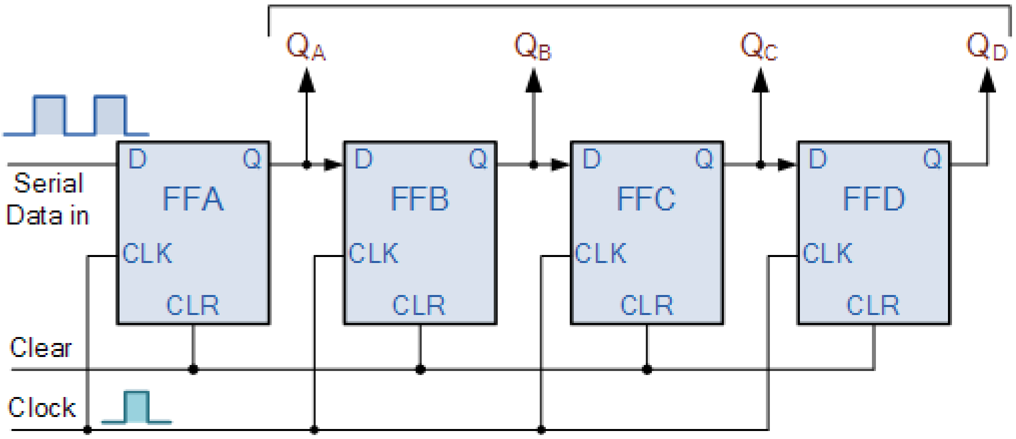
Truth Table



Trigger	Inputs		Output				Inference
			Present State		Next State		
CLK	J	K	Q	$\bar{Q}$	Q	$\bar{Q}$	
✗	x	x	-		-		Latched
↑	0	0	0	1	0	1	No Change
↑			1	0	1	0	
↑	0	1	0	1	0	1	Reset
↑			1	0	0	1	
↑	1	0	0	1	1	0	Set
↑			1	0	1	0	
↑	1	1	0	1	1	0	Toggles
↑			1	0	0	1	

Q.5

Use the waveforms in Figure 03 to draw the timing diagram for the parallel outputs(Q1, Q2, Q3, Q4) for the shift register. Assume that register is initially cleared.



The operation is as follows. Lets assume that all the flip-flops (FFA to FFD) have just been RESET (CLEAR input) and that all the outputs Q_A to Q_D are at logic level “0” ie, no parallel data output.

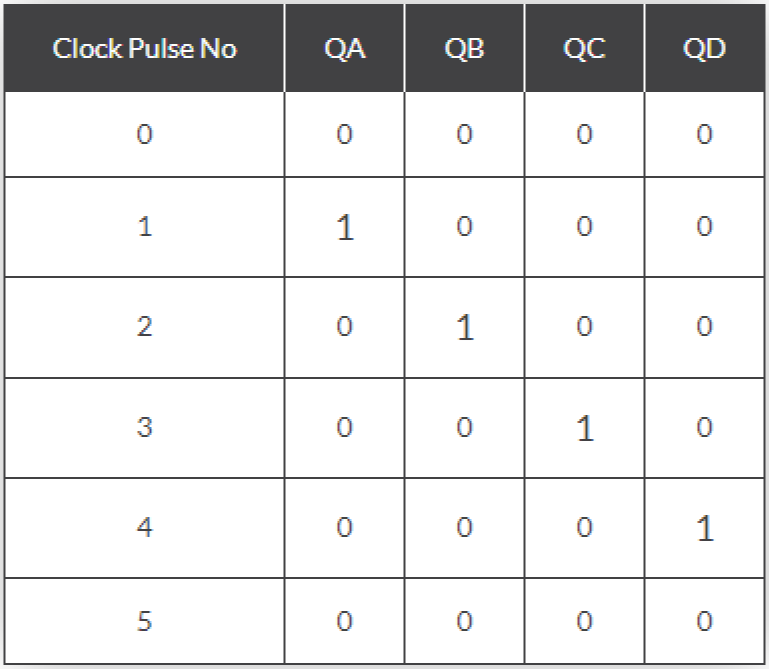
If a logic “1” is connected to the DATA input pin of FFA then on the first clock pulse the output of FFA and therefore the resulting Q_A will be set HIGH to logic “1” with all the other outputs still remaining LOW at logic “0”. Assume now that the DATA input pin of FFA has returned LOW again to logic “0” giving us one data pulse or 0-1-0.

The second clock pulse will change the output of FFA to logic “0” and the output of FFB and Q_B HIGH to logic “1” as its input D has the logic “1” level on it from Q_A. The logic “1” has now moved or been “shifted” one place along the register to the right as it is now at Q_A.

When the third clock pulse arrives this logic “1” value moves to the output of FFC (Q_C) and so on until the arrival of the fifth clock pulse which sets all the outputs Q_A to Q_D back again to logic level “0” because the input to FFA has remained constant at logic level “0”.

The effect of each clock pulse is to shift the data contents of each stage one place to the right, and this is shown in the following table until the complete data value of 0-0-0-1 is stored in the register. This data value can now be read directly from the outputs of Q_A to Q_D.

Then the data has been converted from a serial data input signal to a parallel data output. The truth table and following waveforms show the propagation of the logic “1” through the register from left to right as follows.



Clock Pulse No	QA	QB	QC	QD
0	0	0	0	0
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1
5	0	0	0	0