# CS 221 DIGITAL DESIGN – 10 Oct 2013

Submitted by –

Abhinav Mittal 120101002

Animesh Karmakar 120101010

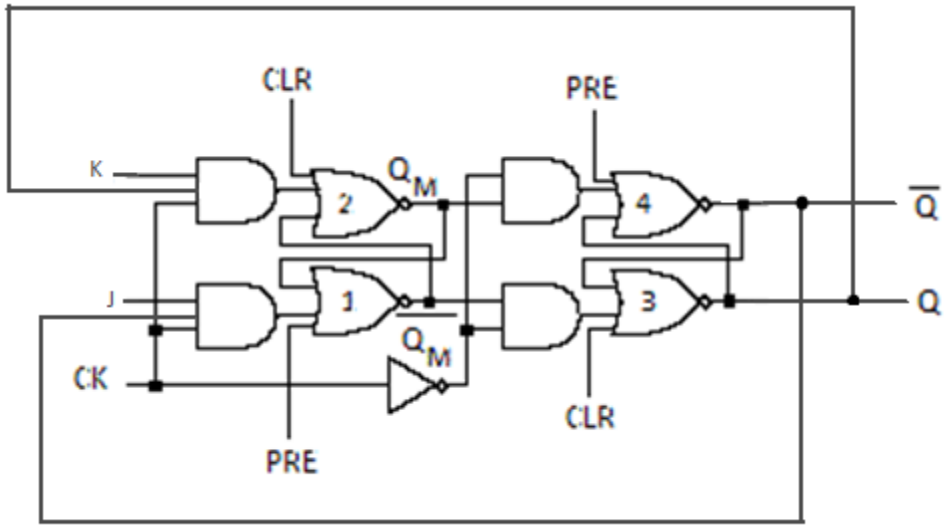
Bharat N. Gupta 120101017

**Synchronous Input:** The flip flop reads the input states on either the rising or falling edge of the clock pulse.

**Asynchronous Input:** These inputs override the synchronous inputs and can be activated any time i.e. independent of the clock pulse.

# J K Flip Flop with Clear and Preset inputs

Clear (CLR) and Preset (PRE) inputs are asynchronous inputs




When Clear = 0, Preset = 1 Then $\bar{Q}$=0, Q =1 (Asynchronous Set Condition)

Since PRE is 1, so Outputs of Gates 1 and 4 are 0. Therefore, the inputs for gate 3 are 0, CLR =0 and $\bar{Q}$=0 so, gate 3 acts as inverter and thus Q = 1

When Clear = 1, Preset = 0 Then $\bar{Q}$=1, Q=0 (Asynchronous Reset Condition)

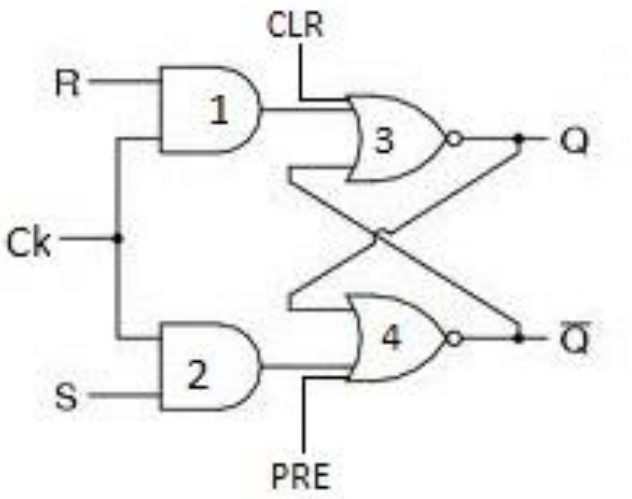
Since CLR is 1, so Outputs of Gates 2 and 3 are 0. Therefore, the inputs for gate 4 are 0, PRE =0 and Q =0 so, gate 4 acts as inverter and thus $\bar{Q}$ = 1

When Clear = 0 and Preset =0, it acts as normal J K Flip Flop.

Clear =1 and Preset = 1 are forbidden inputs since Q = $\bar{Q}$ = 0

| PRE | CLR | CK | J | K | Q | $\bar{Q}$ |
|---|---|---|---|---|---|---|
| 1 | 0 | X | X | X | 1 | 0 |
| 0 | 1 | X | X | X | 0 | 1 |
| 0 | 0 | Valid J K Values | | | Normal J K Flip Flop | |
| 1 | 1 | X | X | X | Not Allowed (Q = 0,$\bar{Q}$ = 0) | |

# GATED RS LATCH



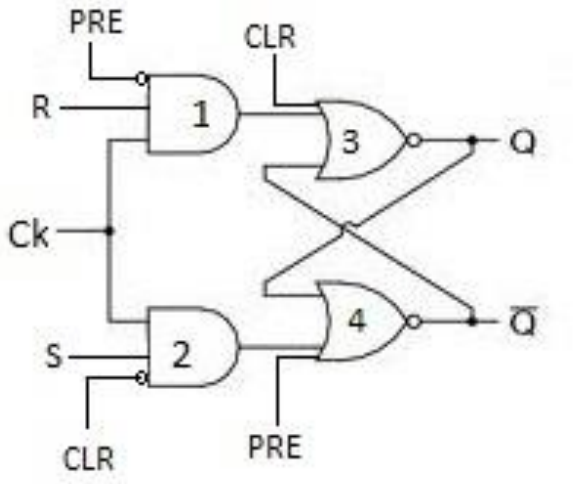

When PRE= 1 , CLR=0 Then Q= 1 (Whenever PRE= 1 , CLR has to be 0)

To get Q = 1 , input R =0 Or CLR or $\overline{PRE}$ should be connected at gate-1 .

When CLR= 1 , PRE= 0 Then Q= 0

To get Q = 0 , input S =0 Or PRE or $\overline{CLR}$ should be connected at gate-2.

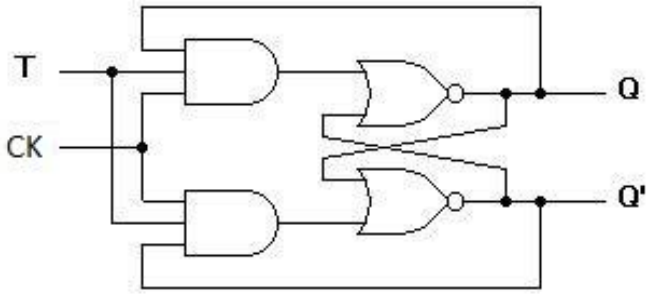
PRE and CLR are External Asynchronous inputs ( independent of clock signal) . We want to overwrite the effect of R and S inputs so we are using PRE and CLR .

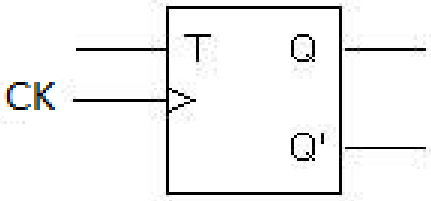



The above arrangement works at PRE =0 , CLR= 0 as normal RS Latch but does not work at PRE=1, CLR= 1 (Since Q = $\overline{Q}$ = 0)

| PRE | CLR | CK | R | S | Q | $\bar{Q}$ |
|---|---|---|---|---|---|---|
| 1 | 0 | X | X | X | 1 | 0 |
| 0 | 1 | X | X | X | 0 | 1 |
| 0 | 0 | Valid R S Values | | | Normal R S Latch | |
| 1 | 1 | X | X | X | Not Allowed (Q = 0, $\bar{Q}$ = 0) | |

## T Flip Flop

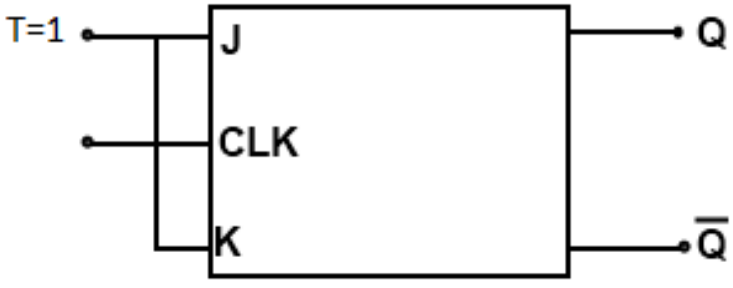



The T or Toggle Flip Flop changes its output on each clock edge, giving an output which is half the frequency of the signal of the T input.

| $Q_N$ | T | $Q_{N+1}$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

### Toggling using J K Flip Flop

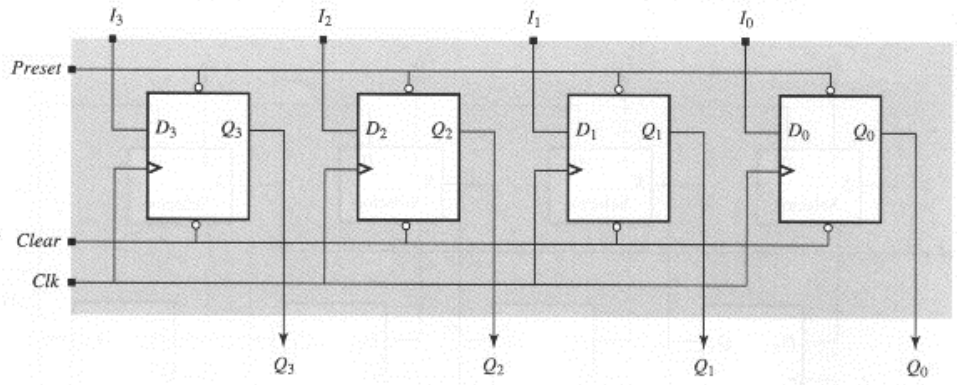


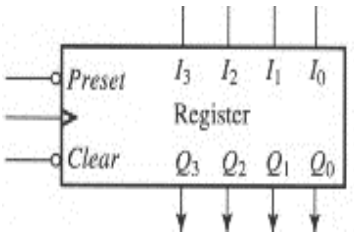

| T | $Q^+$ |
|---|---|
| 1 | $\bar{Q}$ |
| 0 | Q |

# 4 Bit Storage Register

Note: We generally use D Flip Flop to store data.

## Parallel input – Parallel output

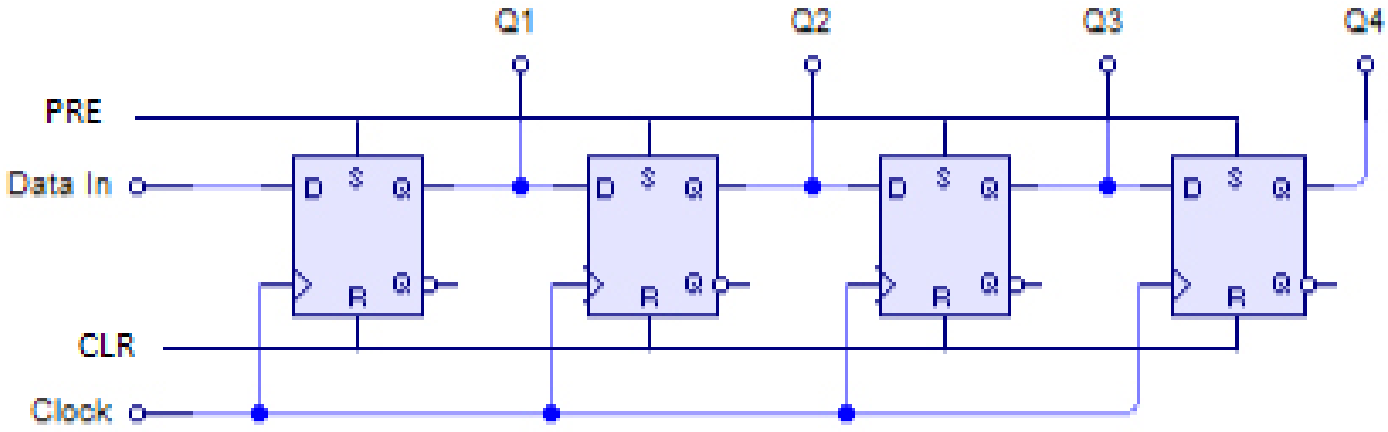



I3, I2, I1, I0-> inputs Q3, Q2, Q1, Q0 -> outputs

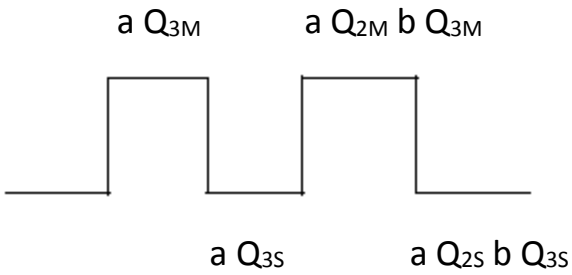
On giving inputs d, c, b, a in inputs I3, I2, I1, I0 ,we get the same outputs in Q3, Q2, Q1, Q0 respectively.

## Serial input – Parallel output




Data In –> Serial Input Q3, Q2, Q1, Q0 -> Outputs

On providing input dcba where a is LSB following values are stored according to clock cycle:



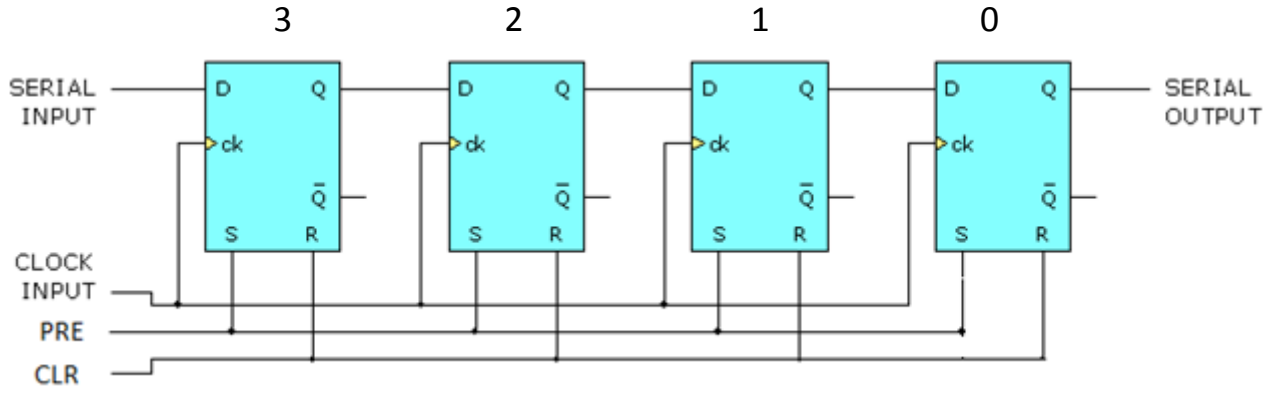
# Serial Input – Serial Output:

For 4 cycles there will be no output because the first bit of input (a) will reach the 4th 1- bit register during the 4th cycle so we will get output from 5th cycle onwards.

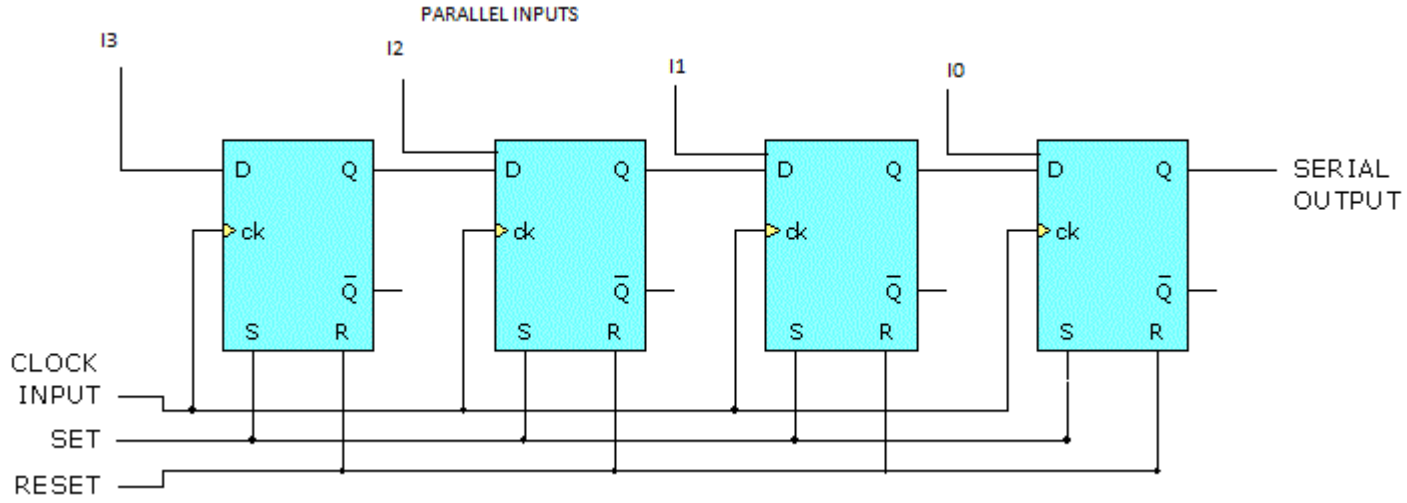
dcba -> input

Output ->

| $Q_3$ | $Q_2$ | $Q_1$ | $Q_0$ | Output | Cycle no. |
|---|---|---|---|---|---|
| X | d | c | b | a | 5 |
| X | X | d | c | b | 6 |
| X | X | X | d | c | 7 |
| X | X | X | X | d | 8 |

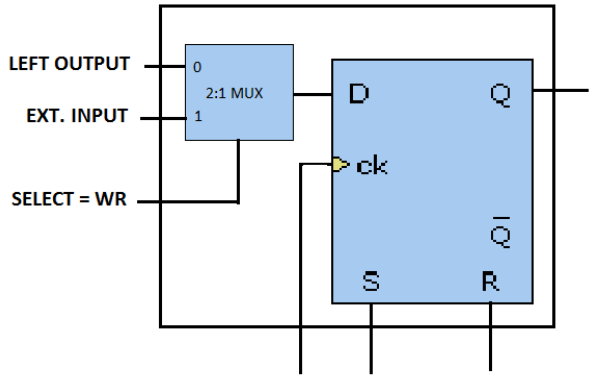



# Parallel input – Serial output:




But, there is a problem that how to decide whether to take the previous register's output as input or take external input.

To overcome this problem we modify each register by adding a 2:1 MUX as follows:

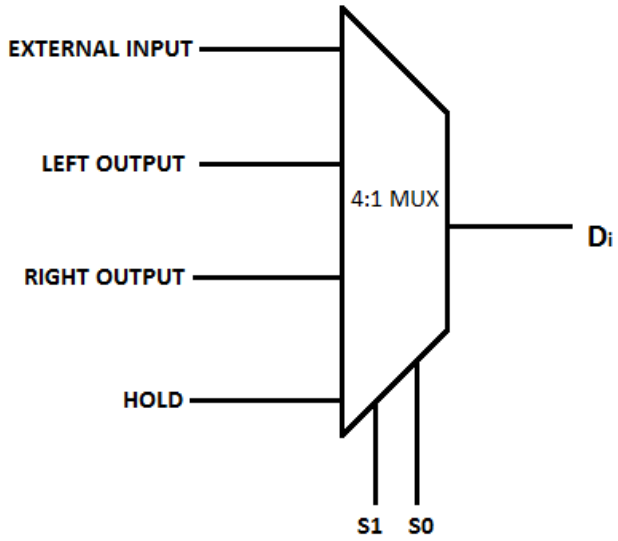



When WR is 0, D will be equal to LEFT OUTPUT (READ condition) and

When WR = 1, D will be equal to EXTERNAL INPUT (WRITE condition)

# Universal Shift Register (synchronous Clear input)

For Universal Shift Register, we require a 4:1 MUX as follows:

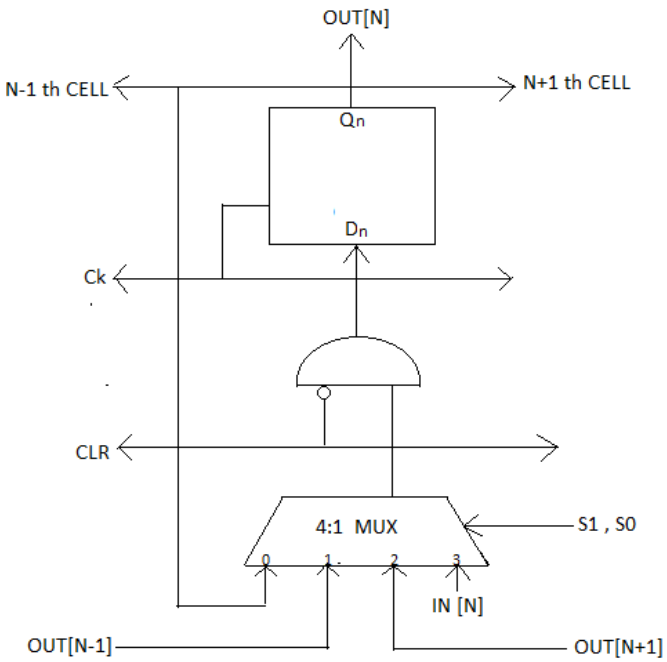



The corresponding table for this MUX is:

| S1 | S0 | |
|---|---|---|
| 0 | 0 | HOLD |
| 0 | 1 | SHIFT RIGHT |
| 1 | 0 | SHIFT LEFT |
| 1 | 1 | EXTERNAL INPUT |

If CLEAR = 1 then D = 0 else D is decided by above MUX.

Thus the Universal Shift Register is as shown below




UNIVERSAL SHIFT REGISTER

With Universal Shift Register we can shift the bits towards right and we can provide the output of LSB or its complement as an input at MSB or new external input at MSB. For Left shift, we can provide the output of MSB or its complement as an input at LSB or new external input at LSB.

e.g. – Shift right with output as input:

1001 -> 1100 -> 0110 -> 0011 -> 1001 ….