Contents

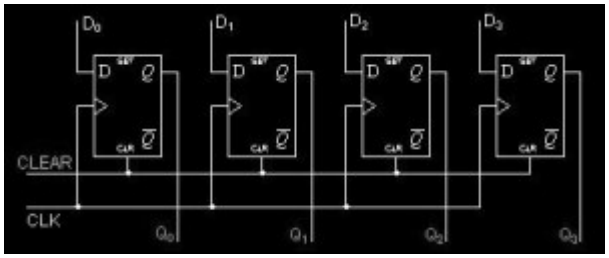


## Shift Register

VHDL Code for shift register can be categorised in serial in serial out shift register, serial in parallel out shift register, parallel in parallel out shift register and parallel in serial out shift register.

## Parallel In – Parallel Out Shift Registers

For parallel in – parallel out shift registers, all data bits appear on the parallel outputs immediately following the simultaneous entry of the data bits. The following circuit is a four-bit parallel in – parallel out shift register constructed by D flip-flops.




The D's are the parallel inputs and the Q's are the parallel outputs. Once the register is clocked, all the data at the D inputs appear at the corresponding Q outputs simultaneously.

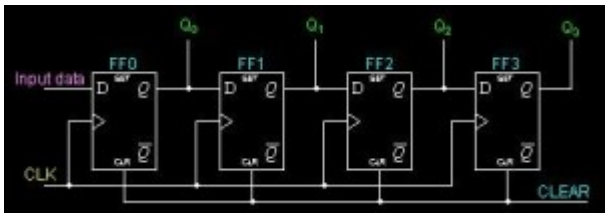
## VHDL code for Parallel In Parallel Out Shift Register

```
library ieee;
use ieee.std_logic_1164.all;
```

```
entity pipo is
 port(
 clk : in std_logic;
 D: in std_logic_vector(3 downto 0);
 Q: out std_logic_vector(3 downto 0)
 );
end pipo;

architecture arch of pipo is

begin

 process (clk)
 begin
 if (CLK'event and CLK='1') then
 Q <= D;
 end if;
 end process;

end arch;
```

## Serial In – Parallel Out Shift Registers

For Serial in – parallel out shift registers, all data bits appear on the parallel outputs following the data bits enters sequentially through each flipflop. The following circuit is a four-bit Serial in – parallel out shift register constructed by D flip-flops.



## VHDL Code for Serial In Parallel Out Shift Register

```
library ieee;</pre>
<pre>use ieee.std_logic_1164.all;

entity sipo is
 port(
 clk, clear : in std_logic;
 Input_Data: in std_logic;
 Q: out std_logic_vector(3 downto 0) );
end sipo;

architecture arch of sipo is

begin

 process (clk)
 begin
 if clear = '1' then
 Q <= "0000";
 elsif (CLK'event and CLK='1') then
 Q(3 downto 1) <= Q(2 downto 0);
 Q(0) <= Input_Data;
 end if;
 end process;
end arch;
```