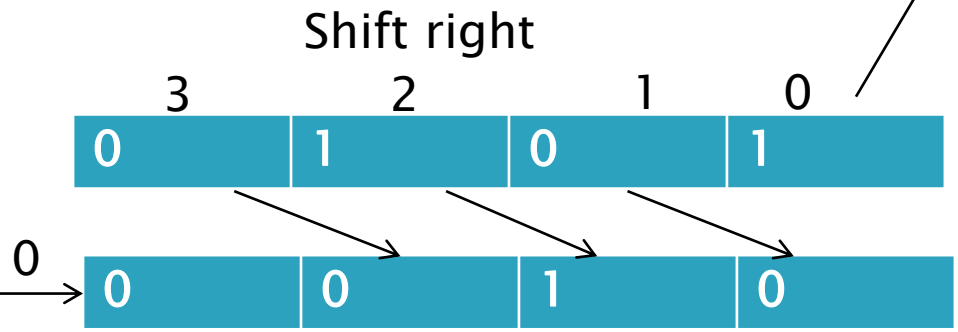
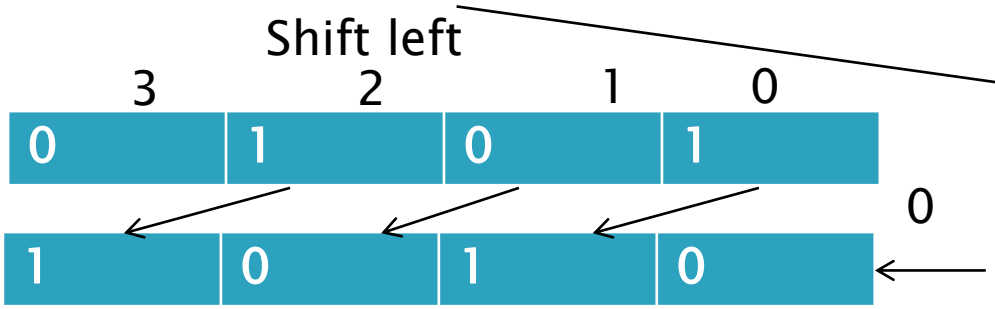
Shift Register

Dr. Fatma Elfouly

Shift Register

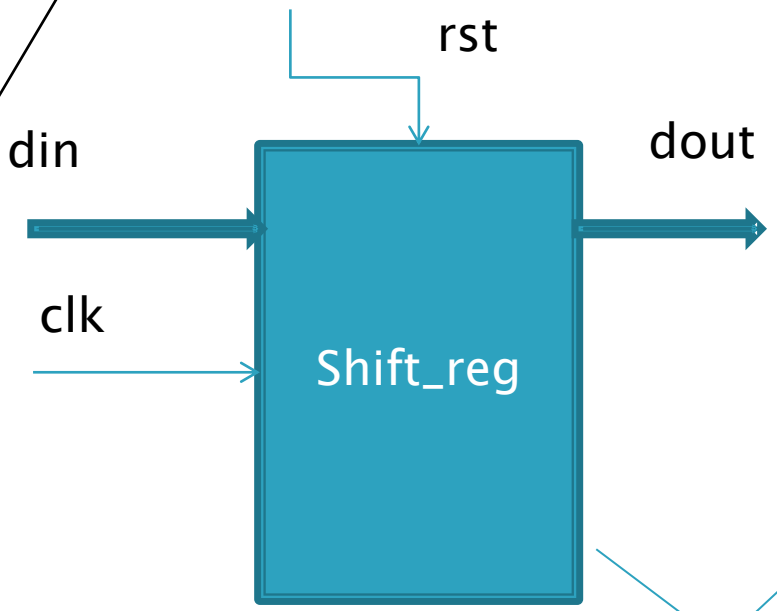
- load data
- Shift right/ shift left





dout<=‘0’&dout(3 downto 1)

d_temp<=‘0’&d_temp(3 downto 1)



dout<=dout(2 downto 0) &’0’

d_temp<=d_temp(2 downto 0) &’0’

Shift Register

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
entity reg_par_shift is
port (clk, rst, load: in std_logic;
din: in std_logic_vector(7 downto 0);
dout:out std_logic_vector(7 downto 0));
end entity reg_par_shift;
architecture rtl of reg_par_shift is
signal d_temp: std_logic_vector(7 downto 0);
begin
process (rst, clk)
begin
if (rst = '1') then
d_temp <= (others => '0');
elsif (rising_edge(clk)) then
If (load ='1') then
d_temp <= din;
else
d_temp <= '0' & d_temp(7 downto 1);
end if;
end if;
end process;
dout <= d_temp;
end architecture rtl;
```

Shift Register

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
entity reg_par_shift is
port (clk, rst: in std_logic;
load : in std_logic_vector(1 downto 0);
din: in std_logic_vector(7 downto 0);
dout:out std_logic_vector(7 downto 0));
end entity reg_par_shift;
architecture rtl of reg_par_shift is
signal d_temp: std_logic_vector(7 downto 0);
begin
process (rst, clk)
begin
if (rst = '1') then
d_temp <= (others => '0');
elsif (rising_edge(clk)) then
If (load ="00") then
reg_temp <= din;
elsif(load="01") then
d_temp <= '0' & d_temp(7 downto 1);
elsif(load ="10") then
d_temp <= 'd_temp(6 downto 0)&'0';
end if;
end if;
end process;
dout <= d_temp;
end architecture rtl;
```