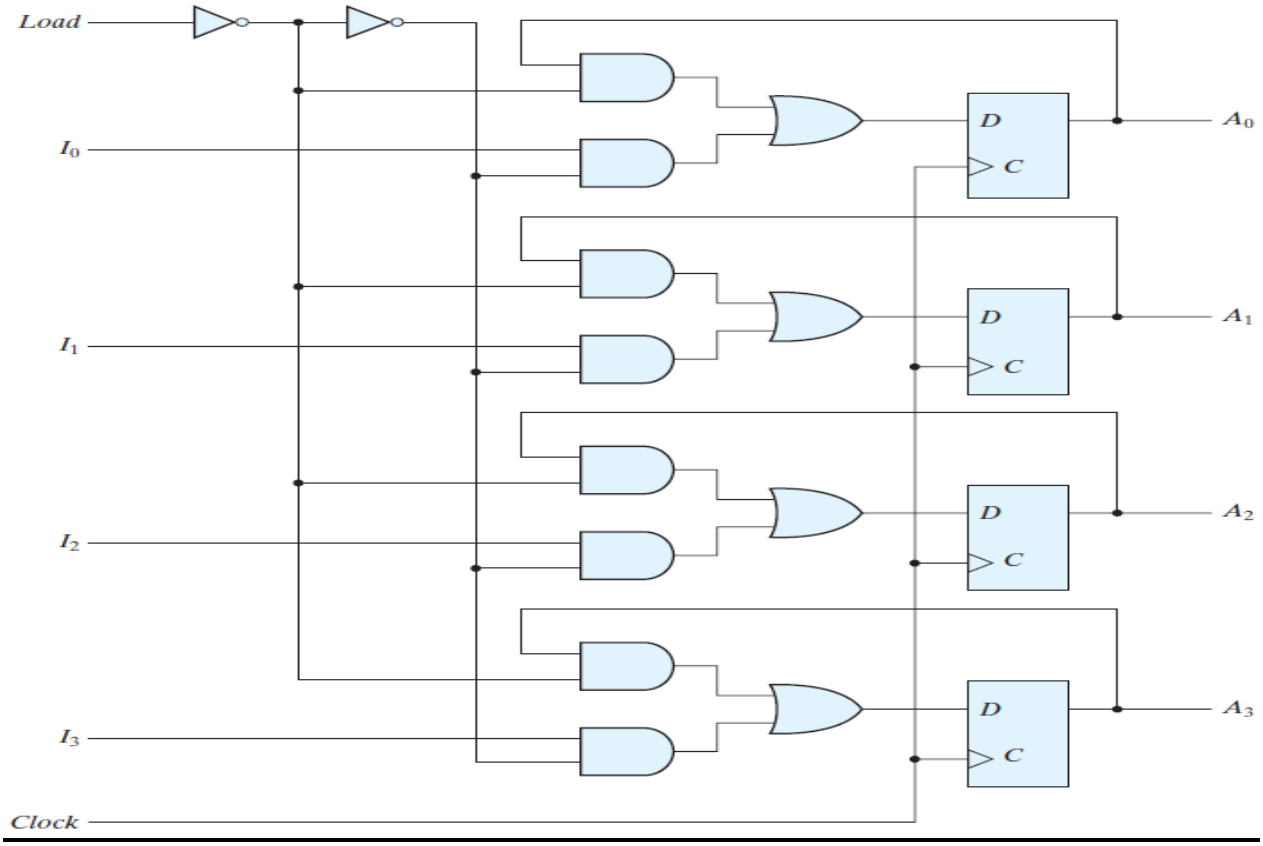
Sheet 5

Q1-Include a synchronous clear input to the register of Fig.The modified register will have a parallel load capability and a synchronous clear capability.The register is cleared synchronously when the clock goes through a positive transition and the clear input is equal to 1.Write and verify a structural HDL model for this register.

ANS

(a) *Block diagram:-*



(b) Code:-

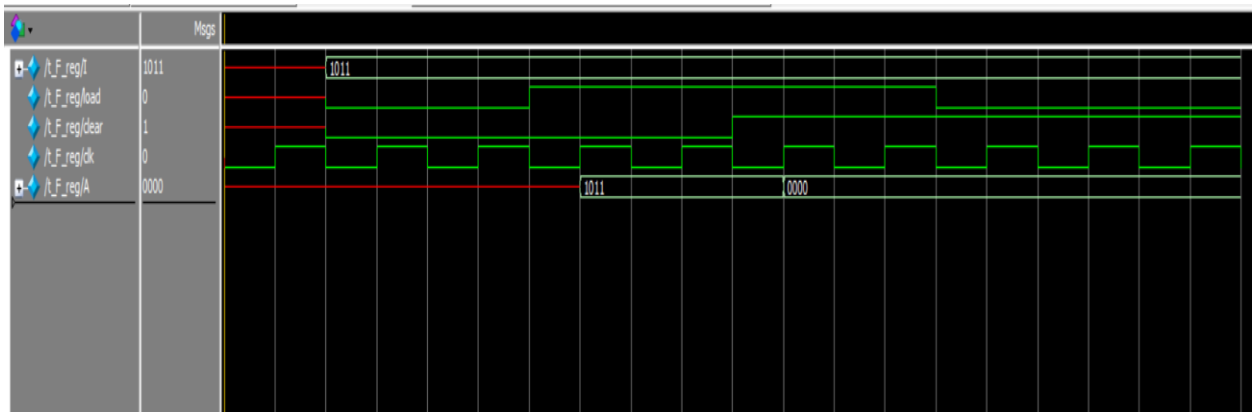
```
module D_flip_flop (
   output reg Q,
   input D,
   input CLK);

always @(posedge CLK)
  Q<=D;
endmodule
```

```
module Register_1 (
    output A,
    input load,clear,I,clk);

wire notload1,notload2,notclear,C0,C1,C2;

not n0(notload1,load);
not n1(notload2,notload1);
not n2(notclear,clear);
and n3(C0,A,notload1,notclear);
and n4(C1,I,notload2,notclear);
or n5(C2,C0,C1);

D_flip_flop w (A,C2,clk);

endmodule


module Register_4 (
  output [3:0] A,
  input [3:0] I,
  input load,clk,clear);

Register_1 W0 (A[0],load,clear,I[0],clk);
Register_1 W1 (A[1],load,clear,I[1],clk);
Register_1 W2 (A[2],load,clear,I[2],clk);
Register_1 W3 (A[3],load,clear,I[3],clk);

endmodule

`timescale 1ns/1ps
module Register_4_ts ();

wire [3:0] A;
reg [3:0] I;
reg load,clk,clear;

Register_4 W4 (A,I,load,clk,clear);

initial
begin
clk = 0;
forever
#5 clk = ~clk;
end
initial
begin
#10 clear=0;
#10 load=1;
#10 I=4'b0101;
#10 load=0;
#10 I=4'b1001;
```

```
#10 load=1;
#10 I=4'b1001;
#10 clear=1;
end
endmodule


module Register_4_bh ( output reg D,
                       input load,clr,clk,I);

always @ (posedge clk)
begin
if ( load ==1 && clr==0)
D = I;
else if ( clr == 1)
D = 0;

end
endmodule


`timescale 1ns/1ps
module Register_4_ts_2 ();

wire [3:0] A;
reg [3:0] I;
reg load,clk,clear;

Register_4_bh W5 (A,I,load,clk,clear);

initial
begin
clk = 0;
forever
#5 clk = ~clk;
end
initial
begin
#10
I = 4'b1011;
clear = 1'b0;
load = 1'b0;
#20
clear = 1'b0;
load = 1'b1;
#20
clear = 1'b1;
load = 1'b0;
#20
clear = 1'b1;
load = 1'b1;
end
```

endmodule

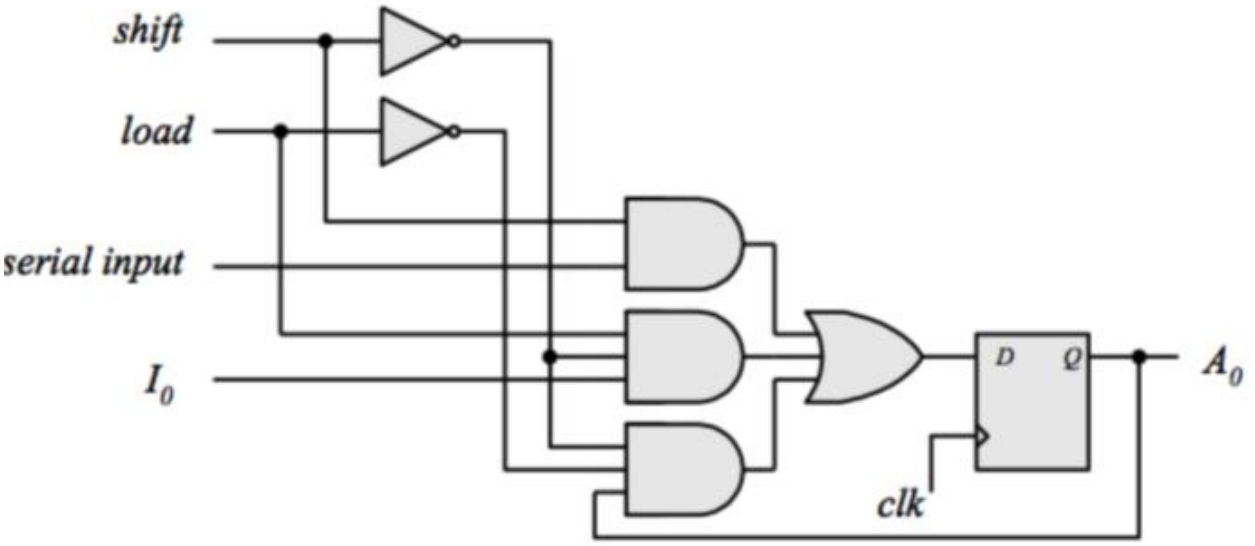
(c) Simulation:-



Q1-Design a four-bit shift register with parallel load using D flip-flops.There are two control inputs:shift and load.When shift=1,the content of the register is shifted by one position.New data are transferred into the register when load=1 and shift=0.If both control inputs are equal to 0,the content of the register dose not change.Write and verify a behavioral HDL model for this register.

ANS

(a) Block diagram:-



(a)Code:-

```
module shiftregister_bh ( output reg [3:0] Q,
                          input [3:0] I,
                          input load,clk,shift);
always @ (posedge clk)
begin
if (shift)
begin
Q[3:1] <= Q[2:0];
Q[0] <= 1'b0;
end
else if (load)
Q <= I;

end
endmodule

`timescale 1ns/1ps

module shiftregister_bh_ts();
wire [3:0] Q;
reg shift,clk,Load;
reg [3:0] I;
shiftregister_bh P2 (Q,I,Load, CLK, shift);
initial
begin
clk=0;
forever
#2 clk=~clk;
end
initial
begin
I=4'b1011;shift=0;Load=0;
#5 shift=0;Load=1;
#5 shift=1;Load=0;
#5 shift=0;Load=0;
#5 shift=1;Load=0;
#5 I=4'b1010;shift=0;Load=0;
#5 shift=0;Load=1;
#10 shift=1;
end
endmodule

//////////////////////////////////////////////////

module D_FF (output reg Q,
             input D, CLK);
always@(posedge CLK)
Q<=D;
endmodule
```

```
module shift_1 (input load, shift,serial_in, I0, CLK,
                output A0);
wire load_bar,shift_bar, O1, O2, O3, FF_in;
not (load_bar, load);
not (shift_bar, shift);
and (O1,A0,load_bar,shift_bar);
and (O2,I0,load,shift_bar);
and (O3,serial_in,shift);
or  (FF_in, O1,O2,O3);
D_FF FF1(A0,FF_in,CLK);
endmodule

module shift_4 (output [3:0] Q,
                input [3:0] I,
                input load, CLK, shift,serial_in);
shift_1 R0 (load, shift,serial_in, I[0], CLK, Q[0]);
shift_1 R1 (load, shift,Q[0], I[1], CLK, Q[1]);
shift_1 R2 (load, shift,Q[1], I[2], CLK, Q[2]);
shift_1 R3 (load, shift,Q[2], I[3], CLK, Q[3]);
endmodule

`timescale 1ns/1ps

module shift_4_ts();
reg [3:0] I;
reg shift,serial_in,CLK,Load;
wire [3:0] Q;
shift_4 P1 (Q,I,Load, CLK, shift,serial_in);
initial
begin
CLK=0;
forever
#2 CLK=~CLK;
end
initial
begin
I=4'b1011;shift=0;Load=0;serial_in=0;
#5 shift=0;Load=1;
#5 shift=1;Load=0;
#5 shift=0;Load=0;
#5 shift=1;Load=0;
#5 I=4'b1010;shift=0;Load=0;
#5 shift=0;Load=1;
#10 shift=1;
end
endmodule
```

(b)Simulation:-