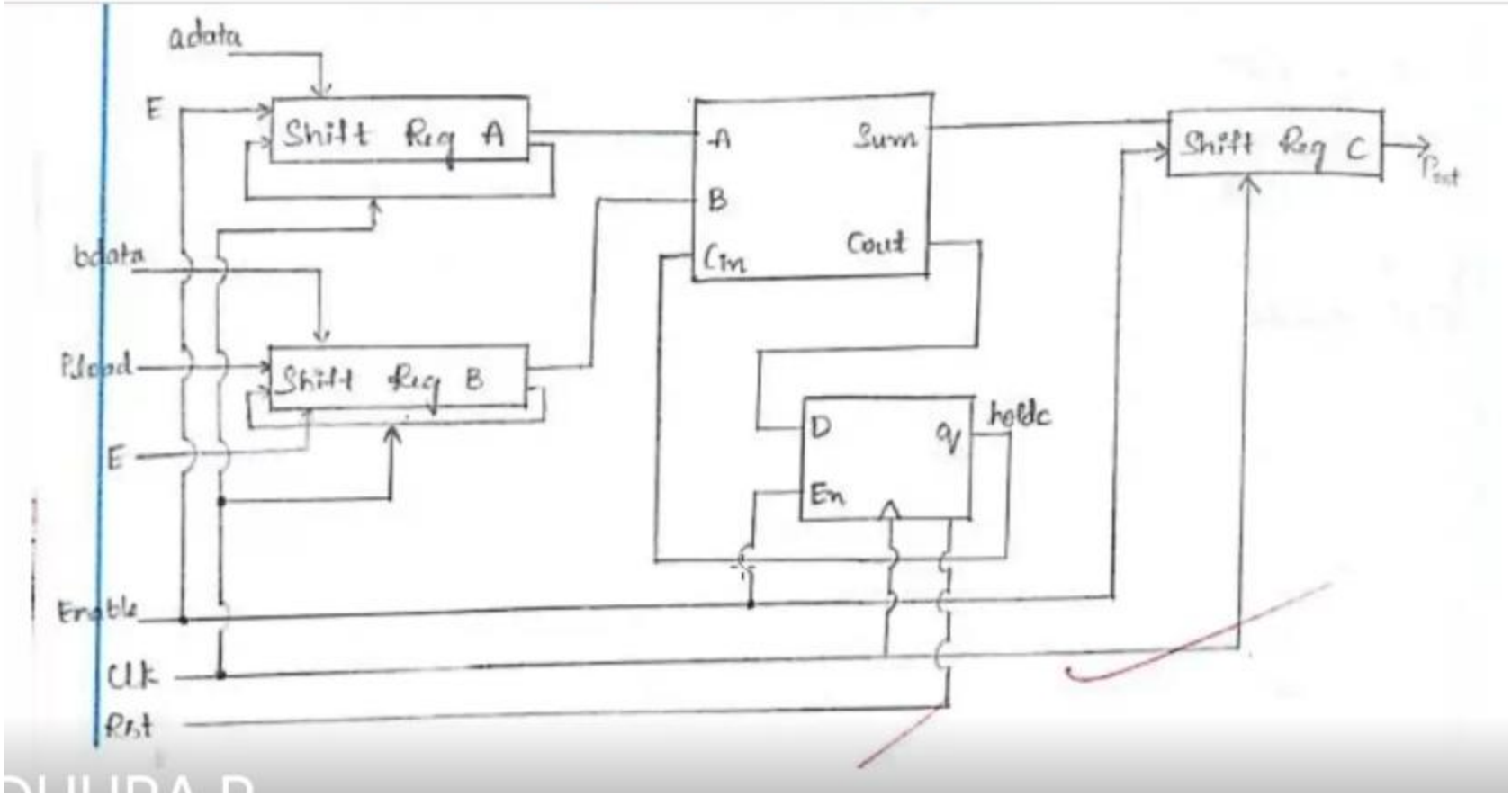
```
module serialadder(clk,rst,pload,adata,bdata,enable,pout);
  input clk,rst,pload,enable;
  input [7:0] adata, bdata;
  output [7:0] pout;

  wire shiftrega_lsb,  shiftregb_lsb;
  reg [7:0] shiftrega, shiftregb, shiftregc;

  wire sum,cout;
  reg holdc;

// instantiated the full adder
full_adder_1bit bit_adder_inst(shiftrega[0],shiftregb[0],holdc,sum,cout);


assign pout=shiftregc;

always@(posedge clk or rst)
 begin
   if (rst) begin
     shiftrega<=8'd0;
     shiftregb<=8'd0;
     shiftregc<=8'd0;
   end else
   if(pload)   begin
     shiftrega<=adata;
     shiftregb<=bdata;
     shiftregc<=8'b0;
   end else if(enable) begin
```

```
assign co =(a&b)|(b&ci)|(ci|a);
endmodule
module serialadder_tb;
  reg clk,rst,pload,enable;

  reg [7:0] adata,bdata;
  wire [7:0]pout;

  // DUT serial adder instance
  serialadder DUT_serial_adder(clk,rst,pload,adata,bdata,enable,pout);

  // clock generation 100 Mhz frequency
  always
    #5 clk=~clk;

  initial
  begin
   clk=1'b0;
   rst=1'b1;
   pload=1'b0;
   enable = 1'b0;

   #10
   rst=1'b0;
   #10
   pload=1'b1; enable=1'b0;
   $display ($time, "   On Reset : adata=%0h, bdata=%0h, pout=%0h",  adata, bdata, pout);
   adata=8'd1;bdata=8'd2;
   $display ($time, "   First Data - Data Inputs loaded: adata=%0h, bdata=%0h, pout=%0h",  adata, bdata, pout);
   #10
```


adata
E
Shift Reg A
A
Sum
B
Cin
Cout
Shift Reg C
Pout
bdata
Pload
Shift Reg B
D
q
En
holdc
E
Enable
Clk
Rst

Shiftreg a		Shiftreg b		Shiftreg c	
0000 0001	01	0000 0010	02	0000 0000	
1000 0000	80	0000 0001	01	1000 0000	80
0100 0000	40	1000 0000	80	1100 0000	C0
0010 0000	20	0100 0000	40	0110 0000	60
0001 0000	10	0010 0000	20	0011 0000	30
0000 1000	08	0001 0000	10	0001 1000	18
0000 0100	04	0000 1000	8	0000 1100	0C
0000 0010	02	0000 0100	4	0000 0110	06
0000 0001	01	0000 0010	2	0000 0011	03

MADHURA.R