```
module unidad_de_control (
 input xs,s,ck,reset,
 output reg cld,wd,wa,ra,wb,rb,wc,FIN);

 parameter S0 = 0, S1 = 1, S2 = 2, S3 = 3, SF = 4;
 reg [2:0] estado_presente, proximo_estado;

 always @(posedge ck,reset)
  if (reset)
          estado_presente=S0;
 else
          estado_presente=proximo_estado;

 always @(xs,s,estado_presente)
  begin
   {cld,wd,wa,ra,wb,rb,wc,FIN}=0;
   proximo_estado=S0;
   case(estado_presente)
    S0: if(xs)
            proximo_estado=S1;
    S1:
     begin
          cld=1;
          wc=1;
          ra=1;
          proximo_estado=S2;
     end
    S2:
      begin
         wd=1;
          if (s)
```

```
                    proximo_estado=SF;
          else
            begin
                    wc=1; rb=1;
                    proximo_estado=S3;
            end
        end
       S3:
        begin
                  wd=1;
                    proximo_estado=SF;
        end
       SF: FIN=1;
      endcase
     end
  endmodule


module registroX (input wx,rx,ck,
              inout [N-1:0] dat);
  parameter N=8;
  reg [N-1:0] q;

  initial q=25;
  always @(posedge ck)
     if (wx) q<=dat;

  assign dat=rx?q:'bz;

endmodule
```

```
module registroC (input wc, ck, input [N-1:0] in,
                  output [N-1:0] out);

 parameter N=8;
 reg [N-1:0] q;
 always @(posedge ck)
  if(wc) q<= in;
 assign out =q;

endmodule

module registroD (input cld, wd, ck, input [N-1:0] in,
                  output [N-1:0] out);

 parameter N=8;
 reg [N-1:0] q;
 always @(posedge ck)
  if(cld) q<= 0;
                  else if(wd) q<= in;
 assign out =q;

endmodule

module sumador (input [N-1:0] a,b, output reg [N-1:0] s);
 parameter N=8;
 always @*
  s=a+b;
endmodule
```

```
module unidad_de_datos (input wa,wb,ra,rb,wc,cld,wd,ck,output [N-1:0] OUTD);

 parameter N=8;
 wire [N-1:0] BUS,OUTSUM,OUTC;
 registroX A(.dat(BUS),.wx(wa),.rx(ra),.ck(ck));
 registroX B(.dat(BUS),.wx(wb),.rx(rb),.ck(ck));
 registroC C(.in(BUS),.wc(wc),.ck(ck),.out(OUTC));
 registroD D(.in(OUTSUM),.cld(cld),.wd(wd),.ck(ck),.out(OUTD));
 sumador sum(.a(OUTC),.b(OUTD),.s(OUTSUM));

endmodule


module sistema_digital (input xs,s,ck,reset,output [N-1:0] OUTD, output FIN);

 parameter N=8;
 wire wa,wb,wc,wd,ra,rb,cld;
 unidad_de_control UC (.xs(xs),.s(s),.ck(ck),.reset(reset),.wa(wa),.wb(wb),.wc(wc),
                       .ra(ra),.rb(rb),.cld(cld),.wd(wd),.FIN(FIN));
 unidad_de_datos UD (.ck(ck),.wa(wa),.wb(wb),.wc(wc),
                     .ra(ra),.rb(rb),.cld(cld),.wd(wd),.OUTD(OUTD));
endmodule
```

```
module sistemadigital_tb;
 reg  xs,s,ck,reset;
 wire FIN;
 wire [7:0] OUTD;
 sistema_digital uut (.xs(xs),.s(s),.ck(ck),.reset(reset),.OUTD(OUTD),.FIN(FIN));
 always      // Reloj de frecuencia 50 Mhz (periodo 20ns)
  begin
    #10;
    ck = ~ck;
  end
 initial begin
    xs=0;  ck=0; s=1; reset=1;
  $monitor("tiempo=%0d,xs=%b, s=%b,estado=%d, registroD=%d, fin=%d",$time,xs,s,
sistemadigital_tb.uut.UC.estado_presente, OUTD,FIN);
  $dumpvars(0, sistemadigital_tb);
  @(negedge ck);
    reset=0;
  @(negedge ck);
    xs=1;
  @(negedge ck);
    xs=0;
  repeat (5) @(negedge ck);
    s=0;
  @(negedge ck);
    xs=1;
  @(negedge ck);
    xs=0;
  repeat (5) @(negedge ck);
  $finish;
 end
    endmodule
```