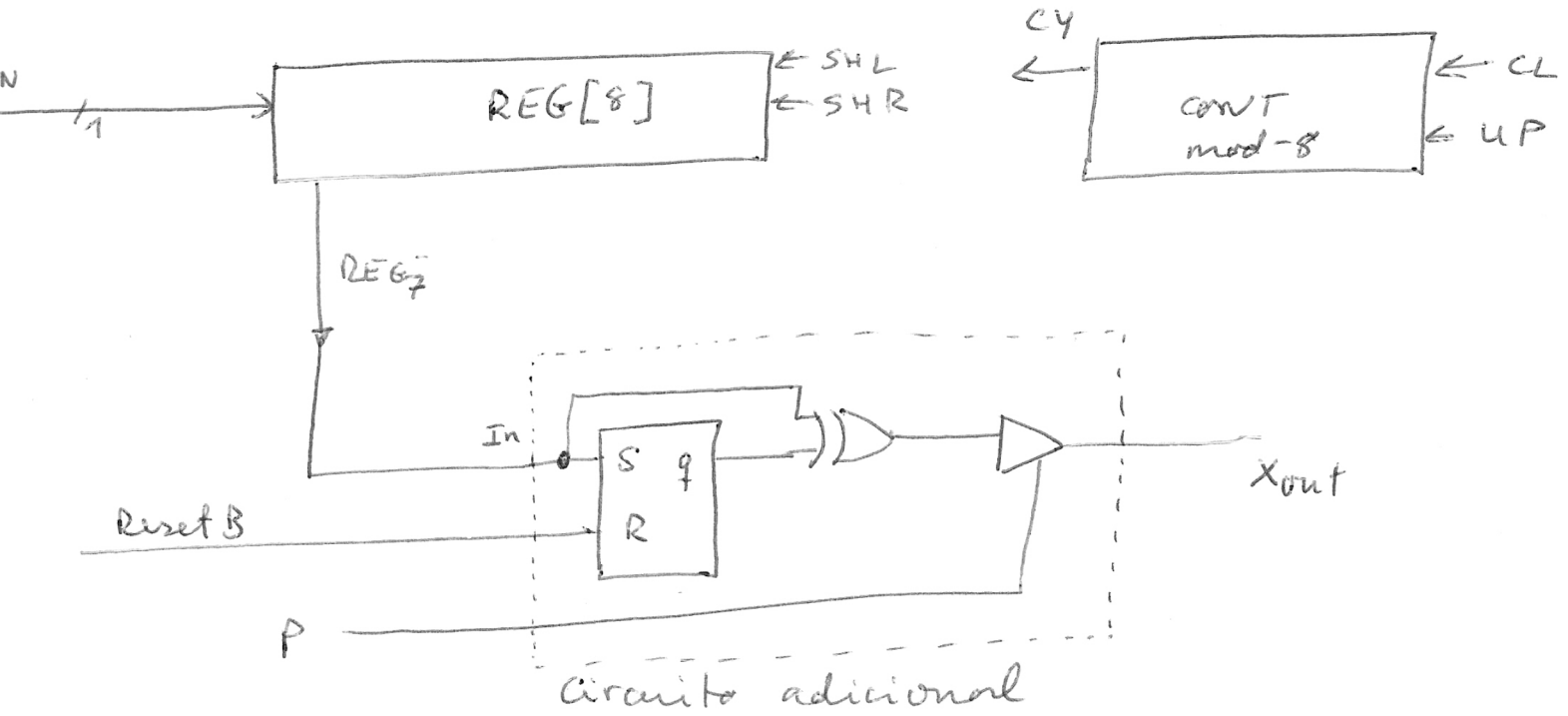
1) La unidad de datos que permite realizar lo descrito en el enunciado se muestra a continuación:



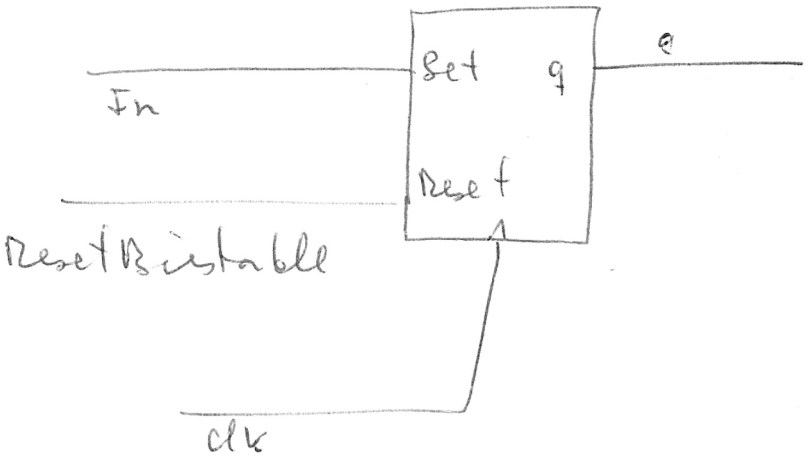
I) Registro de 8 bits con desplazamiento a derecha e izquierda. La entrada de desplazamiento a la derech se conecta con X in. Durante 8 ciclos de reloj, los datos (bits) de entrada se introducen en él. Obsérvese que el bit más significativo (que es el 1º que se recibe) se almacenonia en la posición REG0 una vez transcurridos los 8 ciclos de reloj. A continuación, los bits almacenados deben salir en orden inverso, por lo que la salida REG7 y el desplazamiento a la izquierda hacen o producen dicho efecto.

II) El contador módulo-8 permite determinar los 8 ciclos de reloj necesarios para la entrada y salid del dato.

III) La descripción Verilog del biestable

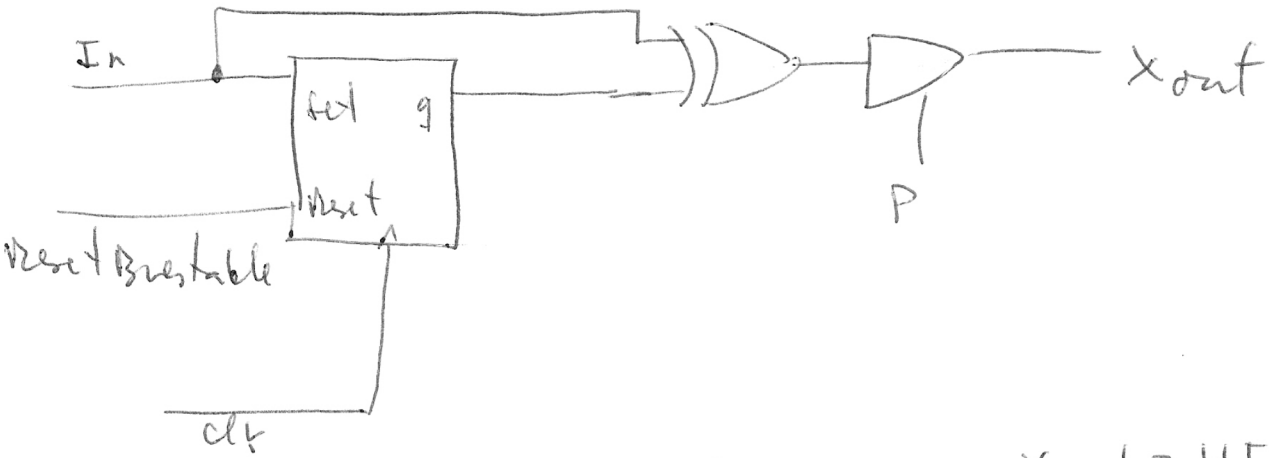
biestable SR (.Set(In), .Reset(ResetBiestable), .Clk(
, .q(q));

indica que:



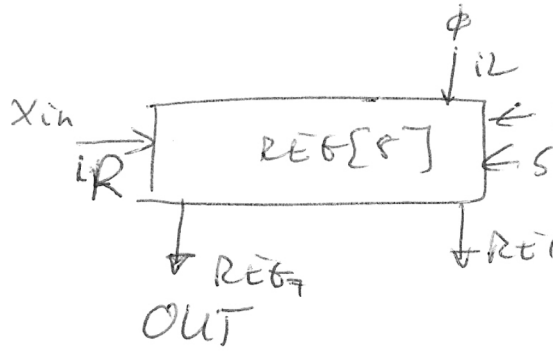
Por otro lado

Xout = p ? (In ^ q) : 1'bz ;

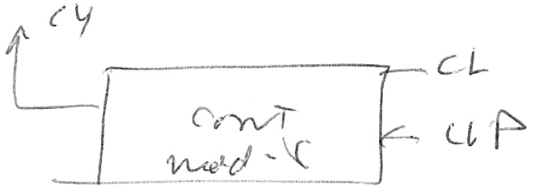


Este circuito adicional hace que Xout = HI si p=0, o que muestre el Cal de In si p=1. Eso sí, es importante poner a 0 el biestable antes de recibir el primer bit de In. Tan sólo se necesita conectar In a REG7

c) Descripción RT de los componentes de la UD (registro y contador)



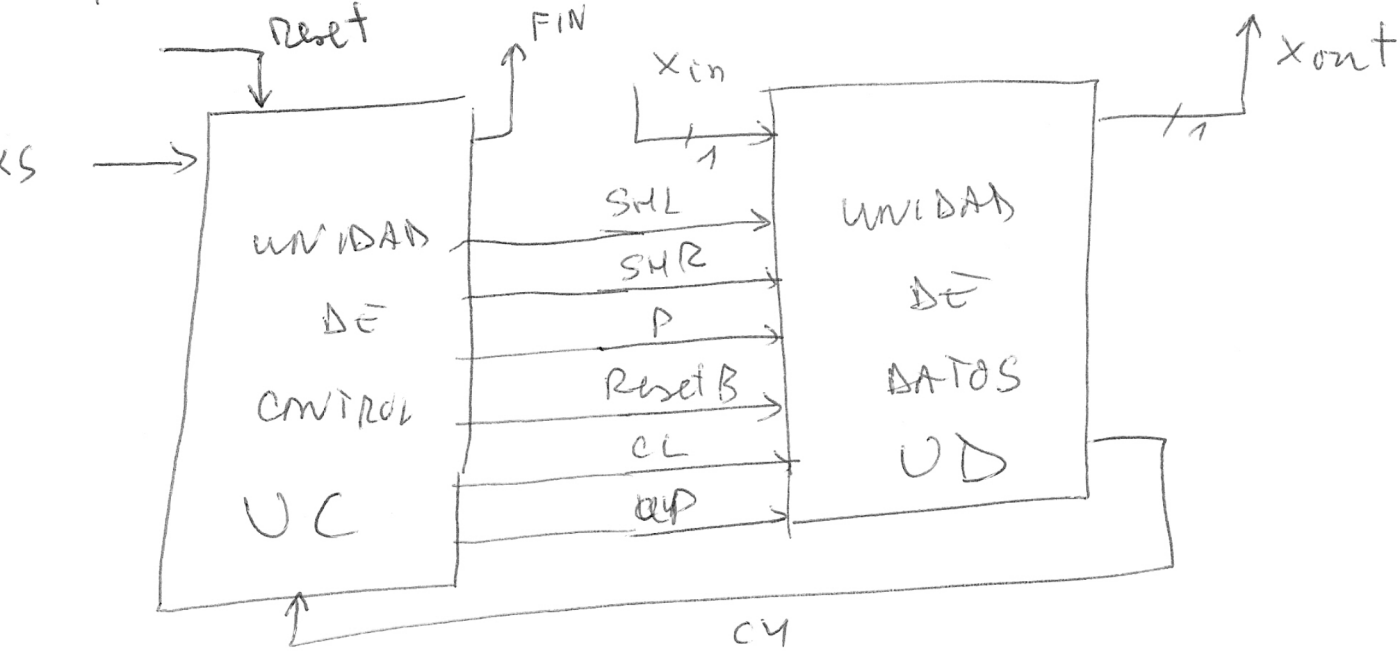
SHL	SHR	REG←	OUT=
0	0	REG	REG_7
0	1	SHR(REG, Xin)	REG_7
1	0	SHL(REG, 0)	REG_7
1	1	—	—



CL	UP	CONT←
0	0	CONT
0	1	CONT + 1
1	0	Φ
1	1	—

CY = 1 Si [CONT] = 7

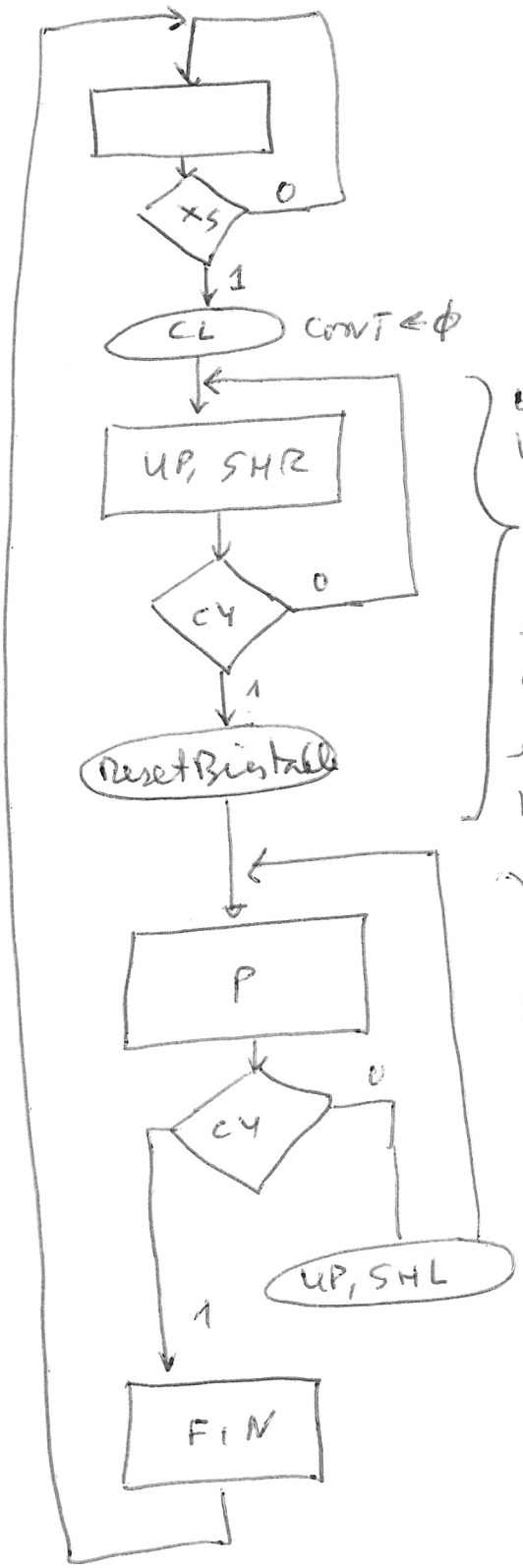
d) Esquemáticamente, la unidad de datos y control quedaría de la siguiente forma



Entradas de la UC: XS, CY
Salidas de la UC: FIN, SHL, SHR, P, ResetB, CL, CUP

Carta ASM de la UC

Esperamos que Xs
a 1 y, entonces,
ponemos a ϕ el
contador



En este bloque se reciben los 8
bits del dato que aparecen
Xin. Es importante que
aparezca en la caja de est
en caso contrario no se ase
que los 8 bits del dato se c
en el registro. Al final re
mos a ϕ el biestable

Habilitamos la salida
durante 8 ciclos de reloj
En este bloque, la orden
desplazamiento e incre
to del contador puede
en una caja de acci
condicional.

) Código Verilog de la UD.

```
module unidaddatos (input clk, xin, SHL, SHR,
        p, ResetBiestable, CL, UP, output xout, C4 );

    wire in, q;
    contador  CONT (.clk(clk), .cl(CL), .up(UP), .C4(C
    registro  REG(.clk(clk), .shR(shR), .shl(shl),
                  .out(in), .iR(xin));
    biestable SR(.set(in), .reset(resetbiestable), .clk(clk
                  .q(q));
    Xout = p ? (In^q) : 1'bz;
endmodule

module contador (input clk, cl, up, output C4);
    :
endmodule
module registro (input clk, shr, shl, iR, output ou
    :
endmodule
```

5) Código Verilog de la UC.

```
module unidadControl ( input clk, xs, c4, Reset,
    output Fin, shl, shr, ResetBiestable, cl, up);

parameter S0 = 2'b00, S1 = 2'b01, S2 = 2'b10, S3 = 2'b1
reg [1:0] current_state, next_state;
always @ ( posedge clk or posedge Reset )
begin
      if (Reset) current_state <= S0;
      else current_state <= next_state;
end
always @( current_state, xs, c4 )
 begin
      {Fin, shl, shr, p, ResetBiestable, cl, up} = 7'd0
      case ( current_state )
        S0: if (xs)
              begin
                cl=1; next state = S1;
              end
        S1: begin
              up=1; shR=1;
              if (c4)
              begin
                ResetBiestable=1; next_state <= S
              end
            end
```

```
        S2 : begin
             P = 1
             if (CA == 0) begin
                  UP = 1; SHL = 1;
             end
             else
                next_state = S3;
             end

        S3 : begin
                FIN = 1;
                next_state = S4;
             end

    endcase
end
endmodule
```