

**Ingeniería Informática. Tecnologías Informáticas.  
Estructura de Computadores. Grupo 2. Curso 2016/2017**

1. Se desea disponer de un sistema que permita realizar las siguientes operaciones:

$A \leftarrow 4 * A$   
 $B \leftarrow$  numero de 1's de A  
 $A \leftarrow A + B$   
 $A \leftarrow A - B$

Los datos son de 8 bits y se cargan en los registros A y B tras activar la señal de comienzo. El sistema generará una señal de FIN tras completar la operación.

a) Diseñe la unidad de datos del sistema digital, describiendo adecuadamente sus componentes. (2.5 puntos)

b) Dibuje un diagrama de bloques del sistema digital que incluya la unidad de datos, la de control y la interconexión entre ambas. (1.5 puntos)

c) Diseñe la carta ASM de la unidad de control del sistema incluyendo en ella las transferencias entre registros y las señales a activar. (2 puntos)

2. A continuación se muestra la descripción verilog de la unidad de datos y la unidad de control de un sistema digital.

### Unidad de Datos

```
module UnidadDatos( input Clk, Wx, Wy, Clp, Dx, Dy, Up,
output Bwx, Bwy );

parameter tam = 8;
wire [tam-1:0] Inx = 'd12;
wire [tam-1:0] Iny = 'd3;
contadordescendente #(tam) X (Clk, Dx, Wx, Inx, Bwx);
contadordescendente #(tam) Y (Clk, Dy, Wy, Iny, Bwy);
contadorascendente #(2*tam)P (Clk, Up, Clp);
endmodule

module contadordescendente #(parameter tam=8)( input Clk, input Down, input W, input [tam-1:0] entrada,
output Bw );

reg [tam-1:0] q;
always @(posedge Clk)
begin
if (W)
q<=entrada;
else if (Down)
q<=q-1;
end
assign Bw = (q==0);
endmodule

module contadorascendente #(parameter tam=8)( input Clk, input Up, input Cl );

reg [tam-1:0] q;
always @(posedge Clk)
begin
if (Cl)
q<=0;
else if (Up)
q<=q+1;
end
endmodule
```

## Unidad de Control

```
module UnidadControl( input Xs,Clk, reset, Bwx,Bwy, output reg Wx, Wy, Clp, Dx, Dy, Up,Fin );

    parameter S0 = 2'b00, S1 = 2'b01, S2 = 2'b10, S3 = 2'b11;

    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,Bwx,Bwy)
    begin
        {Wx,Wy,Clp,Dx,Dy,Up,Fin}=7'd0;
        case (current_state)
        S0:if(Xs)
            begin
                next_state=S1;
                {Wx,Wy,Clp}=3'b111;
            end
            else
                next_state=S0;

        S1:begin
            if (Bwy)
                next_state=S2;
            else
                begin
                    next_state=S1;
                    if (Bwx == 0)
                        {Dx,Up}=2'b11;
                    else
                        {Wx,Dy}=2'b11;
                end
            end

        S2:begin
            Fin = 1;
            next_state=S0;
        end

        default:
            next_state=S0;
        endcase
    end
end

endmodule
```

- Muestre el diagrama de la unidad de datos y describa a nivel RT sus componentes. (1 punto)
- Realice la carta ASM (datos y control) y explique verbalmente la operación que realiza el sistema. (2 puntos)
- Dibuje un diagrama de bloques del sistema digital completo con sus dos bloques principales (Unidad de datos y Unidad de Control) y escriba la descripción verilog del mismo instanciando las unidades de datos y control ya proporcionadas. (1 punto)