

# ESTRUCTURA DE COMPUTADORES – TI (G3)

## Primera prueba de evaluación continua



APELLIDOS Y NOMBRE: \_\_\_\_\_

**P1.-** Defina los siguientes términos: RISC, SISD, SIMD, Hardware, Firmware (1 punto).

**P2.-** Se desea diseñar un Sistema Digital (SD) que permita calcular el complemento a 2 de un número de 8 bits que se recibe por **una ÚNICA línea** de entrada llamada **Xin**.

**RECEPCIÓN DEL NÚMERO:** Tras la activación de la señal **Xs**, el SD, recibe, por **Xin**, dicho número en serie (bit a bit) empezando por el bit más significativo. Durante la recepción del número (8 ciclos de reloj), la salida **Xout** permanece en alta impedancia.

**SALIDA DEL CA2 DEL NÚMERO:** Una vez finalizada la recepción, comenzará a salir, por la **LÍNEA Xout**, el Ca2 del número recibido también de forma serie, pero ahora, empezando por el bit menos significativo.

**COMPONENTES DE LA UNIDAD DE DATOS:** Para el diseño de la unidad de datos dispone de un **registro de desplazamiento**, un **contador** y un **circuito adicional** que se describe en el código Verilog de la unidad de datos (en el reverso de la hoja).

**DESCRIPCIÓN DEL CIRCUITO ADICIONAL:** La circuitería adicional gestiona la salida **Xout** mediante tres entradas: **p**, **ResetBiestable** e **In**. La entrada **p** hace que la salida esté, o no, en HI; por la entrada **In** se recibe el número de forma serie cuyo Ca2 se desea calcular (empezando por el menos significativo). El biestable del circuito que genera el Ca2 debe estar a 0 en el ciclo de reloj previo a la recepción del primer bit del número, para ello usará la entrada **ResetBiestable**.

- Diseñe la unidad de datos incluyendo los componentes del circuito que están descritos en Verilog (3 puntos).
- Describa a nivel RT los componentes de la unidad de datos (1 punto).
- Obtenga la carta ASM de la unidad de control (3 puntos).
- Complete el código Verilog de la unidad de datos que se muestra (1 punto).
- Complete el código Verilog de la unidad de control (1 punto).

```

module UnidadDatos( input Clk, Set, ResetBiestable,...
    //Aquí declaración de entradas y salidas
    .....
);
    wire q, In; //Cables internos a la unidad de datos
    //Aquí se instancian los módulos que faltan (contador y registro)
    .....
    biestable SR(.Clk(Clk),.Reset(ResetBiestable),.Set(In),.q(q)); //
    assign Xout = p ? (q ^ In) : 'bz ; // p ? A : B implica que para p=1 se elige A y B en otro caso
endmodule

```

```

module UnidadControl(
    //Aquí declaración de entradas y salidas
    .....
);
    parameter S0 = , S1 =... ; //Complete la lista de parámetros
    reg [?:0] current_state, next state; //Sustituya ? por el valor numérico que
    corresponda.

    always @(posedge Clk or posedge Reset)
    begin
        if(Reset)
            current_state <= S0;
        else
            current_state <= next_state;
    end

    always @(current_state, Xs, Cy)
    begin
        //Inicialización de salidas
        .....
        case (current_state)
        S0: .....
        S1: .....
        endcase
    end
endmodule

```