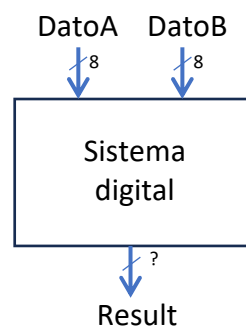


ALUMNO: _____

Ejercicio 1 (4 puntos)

Se desea comparar dos palabras DatoA y DatoB de 8 bits para saber cuántos de sus bits son coincidentes. Por ejemplo: si las palabras fueran 11010010 y 01001011, la respuesta sería 4 ya que los bits 1, 2, 5 y 6 son iguales en ambas palabras. Diseñe un sistema digital que lleve a cabo esta tarea.



El sistema debe contener al menos los elementos que se muestran en la siguiente figura. Los registros A y B, son registros de entrada paralelo y salida serie, cuando realizan la operación de desplazamiento el bit que sale es reintroducido por la entrada serie para que el dato no se pierda. Al comenzar la operación deben introducirse las entradas (DatoA y DatoB) en los registros A y B respectivamente.



Se pide:

- 1) Completar la unidad de datos. Debe añadir los elementos necesarios, pero no puede modificar los registros A y B.
- 2) Describir a nivel RT los elementos secuenciales que ha añadido a la unidad de datos.
- 3) Un diagrama de bloques donde muestre claramente las entradas y salidas de las unidades de datos y de control.
- 4) La carta ASM de la unidad de control.

Ejercicio 2 (4 puntos)

Se desea añadir al CS3 una nueva instrucción, XCHG, cuya sintaxis es: XCHG Y,Z

Dicha instrucción intercambia dos datos en memoria: MEMDAT(Y) $\leftrightarrow$ MEMDAT(Z)

- Asigne a la nueva instrucción el código de operación y el formato de código máquina que considere oportunos.
- Describa, si son necesarias, las modificaciones que se deben realizar en la unidad de datos para poderla implementar.
- Obtenga la secuencia de microoperaciones de XCHG (transferencias RT y señales a activar).
- Obtenga un programa que, haciendo uso de XCHG, intercambie los datos de las posiciones \$80 y \$90 de la memoria de datos. Proporcione también el código máquina de dicho programa.

Ejercicio 3 (2 puntos)

3.1 ¿Qué elemento describe el siguiente módulo Verilog?

```
module incognita #(parameter n=16)(
    input [n-1:0] in, input a, input b, ck,
    output [n-1:0] out);

    reg [n-1:0] q;
    always @(negedge ck)
        if (a)
            q <= in;
        else if (b)
            q = {q[n-2:0],q[n-1]};
    assign out = q;
endmodule
```

3.2 Dibuje las ondas correspondientes al siguiente testbench:

```
module incognita_tb;
    reg [7:0] in;
    reg a, b, ck;
    wire [7:0] out;

    incognita #(8) uut (.in(in),.a(a),.b(b),.ck(ck),.out(out));

    always #10 ck = ~ck;

    initial begin
        ck=0; a=0; b=0; in = 8'h8a;
        @(posedge ck) a=1;
        @(posedge ck) a=0;
        @(posedge ck) b=1;
        repeat (3) @(posedge ck);
        b=0;
        @(posedge ck);
        @(posedge ck);
        $finish;
    end
endmodule
```