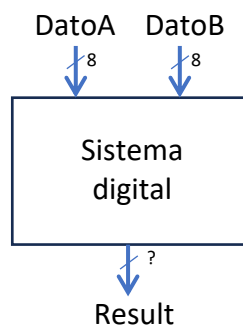


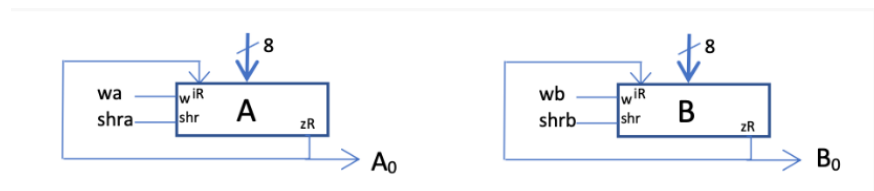
ALUMNO: _____

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.

Hay que añadir dos contadores:

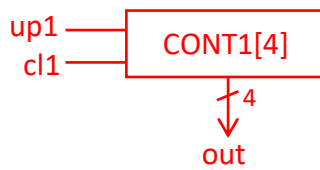
CONT1: contador de 4 bits para contar los bits coincidentes en los registros A y B, tiene que ser de 4 bits ya que hay 9 posibilidades desde que no coincida ninguno hasta que coincidan todos (0,1,...,8).

Este contador tendrá bus de salida que será el que se conecte a la salida Result del sistema digital que, por tanto, será de 4 bits

CONT2: contador de 3 bits para llevar la cuenta de los bits ya comparados.

Este contador tendrá salida de acarreo que será la que informará de que ya se han comparado los 8 bits de los registros A y B.

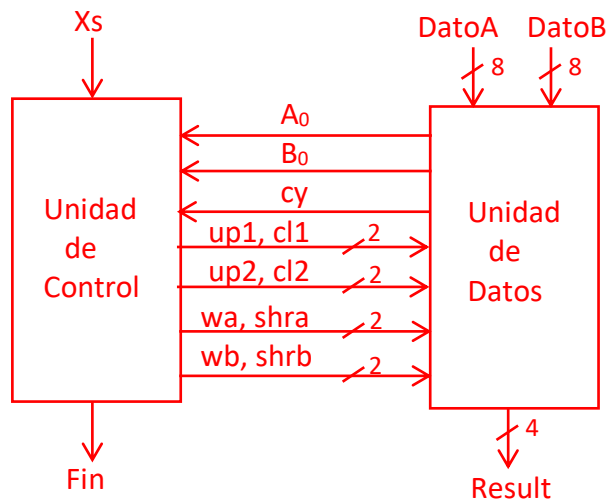
2) Describir a nivel RT los elementos secuenciales que ha añadido a la unidad de datos.



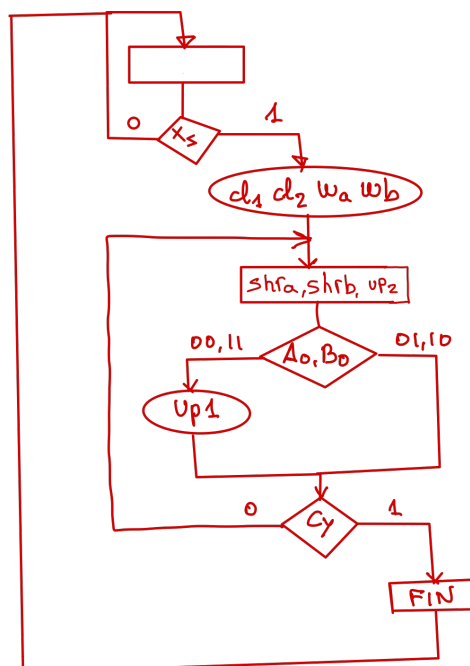
up1 cl1	CONT1 ←	out =
0 0	CONT1	[CONT1]
0 1	0	[CONT1]
1 -	CONT1 + 1	[CONT1]

up2 cl2	CONT2 ←	cy
0 0	CONT2	=1 sii
0 1	0	[CONT2]=7
1 -	CONT2 + 1	

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)

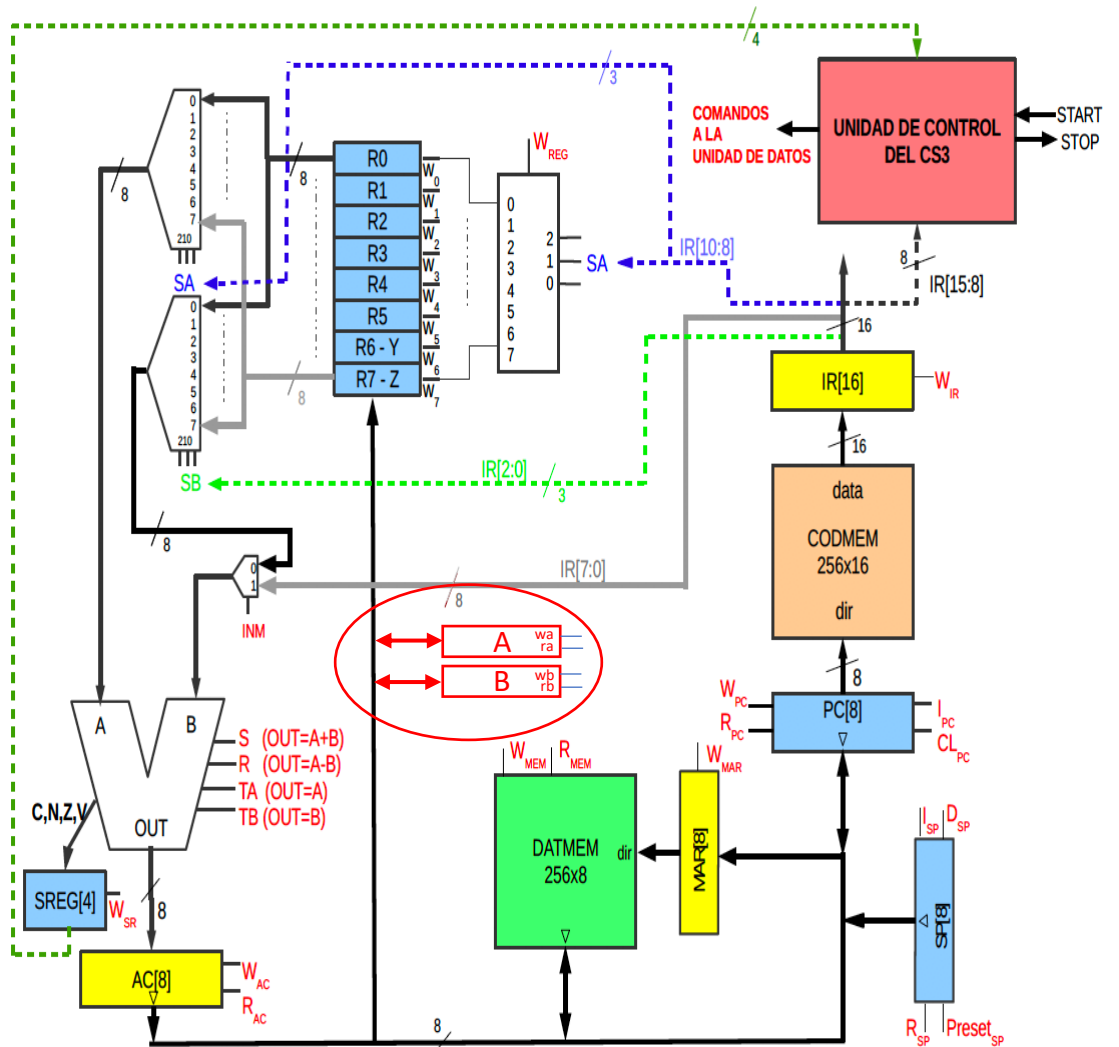
- a. Asigne a la nueva instrucción el código de operación y el formato de código máquina que considere oportuno.

El código de operación: 10000 y el formato el A

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1 0 0 0 0					Y			0 0 0 0 0					Z		

- b. Describa, si son necesarias, las modificaciones que se deben realizar en la unidad de datos para poderla implementar.

Introducimos dos registros A y B conectados al bus común para almacenar temporalmente los datos que se van a intercambiar.



- c. Obtenga la secuencia de microoperaciones de XCHG (transferencias RT y señales a activar).

1. $AC \leftarrow R(IR_{10:8}),$	W_{AC}, T_A
2. $MAR \leftarrow AC, AC \leftarrow R(IR_{2:0}),$	$W_{MAR}, R_{AC}, W_{AC}, T_B$
3. $A \leftarrow DATMEM(MAR),$	W_A, R_{MEM}
4. $MAR \leftarrow AC, AC \leftarrow R(IR_{10:8}),$	$W_{MAR}, R_{AC}, W_{AC}, T_A$
5. $B \leftarrow DATMEM(MAR),$	W_B, R_{MEM}
6. $DATMEM(MAR) \leftarrow A,$	R_A, W_{MEM}
7. $MAR \leftarrow AC,$	W_{MAR}, R_{AC}
8. $DATMEM(MAR) \leftarrow B,$	R_B, W_{MEM}

- d. 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.

LDI R6,\$80	11111 110 1000 0000	\$FE80
LDI R7,\$90	11111 111 1001 0000	\$FF90
XCHG Y,Z	10000 110 0000 0111	\$8607
STOP	10111 000000000000	\$B800

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
```

Se trata de un registro de desplazamiento a la izquierda circular, es decir, la salida serie se conecta a la entrada serie. Es un registro parametrizable de n bits. Posee un buses de entrada y de salida de n bits. El bus de salida es de lectura incondicional.

Las señales a y b son las señales de control. La señal a es la señal de escritura (del dato de n bits) y la señal b es la señal de desplazamiento a la izquierda.

Es un dispositivo disparado por flanco de bajada.

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
```

