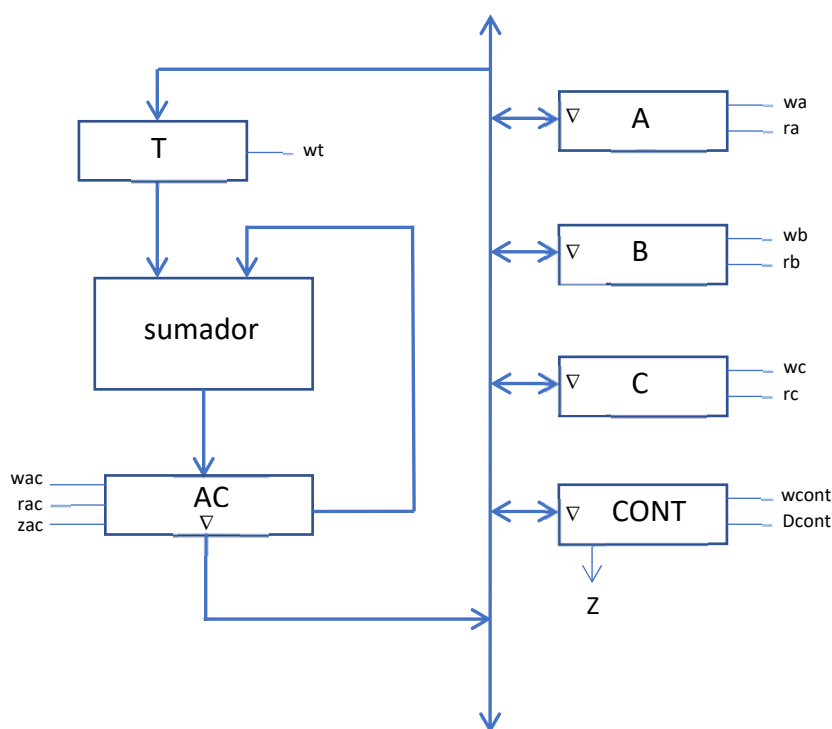


ALUMNO: _____

Ejercicio 1 (4 puntos)

Se tiene la unidad de datos de un sistema digital que se muestra en la siguiente figura. El contador, CONT, cuenta hacia abajo y su salida Z indica que ha llegado a 0. Todos los componentes: registros, sumador y buses, son de 8 bits. Se desea multiplicar dos palabras almacenadas en A y B mediante el algoritmo de sumas sucesivas, es decir, sumaremos A tantas veces como indica B. El resultado final se almacenará en C. Considere que no hay problemas de desbordamiento. 1) Enumere las entradas de la unidad de control. 2) Describa el registro A a nivel RT. 3) Describa el registro T en Verilog. 4) Obtenga la carta ASM de la unidad de control para llevar a cabo la tarea pedida.



Ejercicio 2 (3 puntos)

Se desea añadir al CS3 una nueva instrucción: ADD3, cuya sintaxis es ADD3 RD,RF1,RF2. Dicha instrucción es parecida a la instrucción ADD salvo que opera con tres registros, uno de ellos como destino y los otros dos como fuente, siendo su efecto $RD \leftarrow RF1 + RF2$

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 micro-operaciones de ADD3 para los ciclos de búsqueda y ejecución (transferencias RT y señales a activar por la unidad de control).
- Obtenga el código máquina (binario y hexadecimal) de la instrucción ADD3 R4,R7,R2

Ejercicio 3 (3 puntos)

Dibuje la carta ASM correspondiente al siguiente código Verilog. Esta carta se ejecuta sobre la unidad de datos del problema 1. Diga qué función realiza en función de la entrada *sel*.

```
module unidad_de_control (
    input xs,sel,ck,reset,
    output reg zac,wac,wa,ra,wb,rb,wt,FIN);

    parameter S0 = 0, S1 = 1, S2 = 2, S3 = 3, SF = 4;
    reg [2:0] estado_presente, proximo_estado;

    always @(posedge ck,reset)
        if (reset)
            estado_presente<=S0;
        else
            estado_presente<=proximo_estado;

    always @(xs,sel,estado_presente)
        begin
            {zac,wac,wa,ra,wb,rb,wt,FIN}=0;
            proximo_estado<=S0;
            case(estado_presente)
                S0: if(xs)
                    proximo_estado<=S1;

                S1: begin
                    {zac,wt,ra}=3'b111;
                    proximo_estado<=S2;
                end

                S2: begin
                    wac=1;
                    if (sel)
                        proximo_estado<=SF;
                    else
                        begin
                            wt=1; rb=1;
                            proximo_estado<=S3;
                        end
                end

                S3: begin
                    wac=1;
                    proximo_estado<=SF;
                end

                SF: FIN=1;
            endcase
        end
endmodule
```