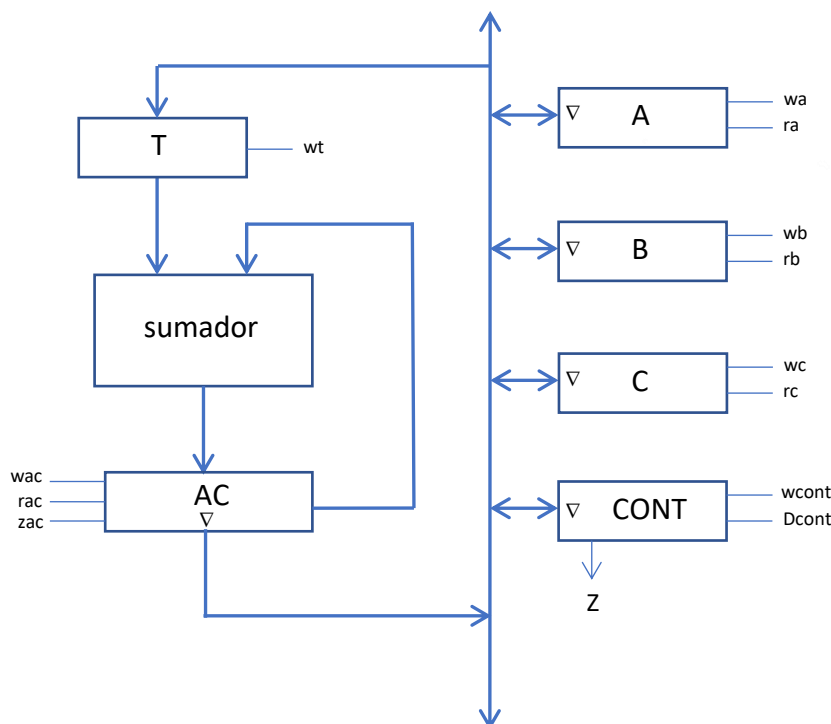


ALUMNO: _____

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.



1) Las entradas de la unidad de control; son Xs (señal de comienzo) y Z, que informa de cuando el contador CONT ha llegado a 0.

2) Registro A a nivel RT



wa ra	A ←	dat
0 0	A	HI
0 1	A	[A]
1 0	dat	entradas
1 1	proh	proh

3) Registro T en Verilog

```
module regT (input ck, wt, input [7:0] in,
             output [7:0] out);
```

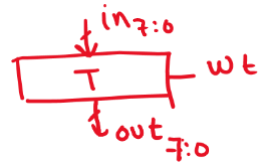
```
    reg [7:0] q;
```

```
    always (posedge ck)
```

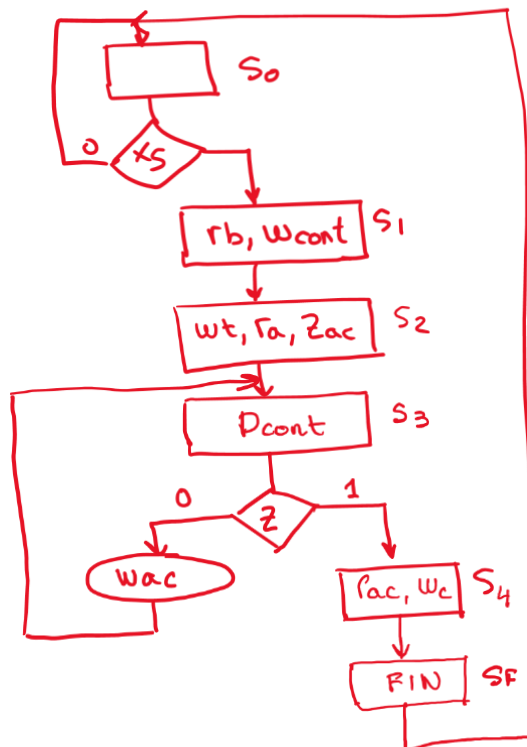
```
        if (wt)
            q <= in;
```

```
    assign out = q;
```

```
endmodule
```



4) Carta ASM



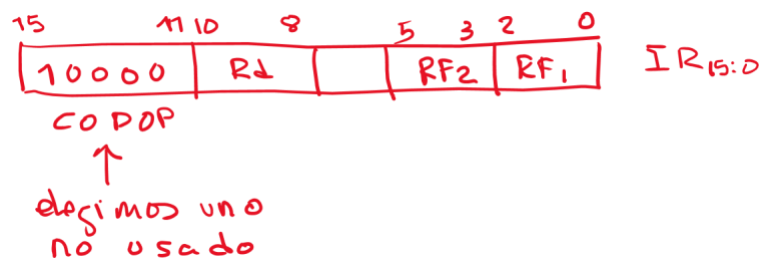
Ejercicio 2 (3 puntos)

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.

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

. En primer lugar, elegimos el formato:



Leer del registro RF1 que está codificado en IR2:0 o escribir en RD que está codificado en IR10:8 no será problema pues otras instrucciones lo hacen y está resuelto en la arquitectura, pero leer de RF2 que está codificado en IR5:3, requiere un cambio en la unidad de datos. Este cambio será multiplexar esos bits con IR10:8 para facilitar su acceso a las líneas SA.



- b. 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).

búsqueda: 1. $IR \leftarrow MEM(OP(PC), PC \leftarrow PC + 1$
 W_{IR}, IPC

ejecución 1. $AC \leftarrow R(IR_{5:3}) + R(IR_{2:0})$
 $SREG \leftarrow 2NVC(R(IR_{5:3}) + R(IR_{2:0}))$
 $W_{AC}, SELF2, S, W_{SR}$

2. $R(IR_{10:8}) \leftarrow AC$
 R_{AC}, W_{REG}

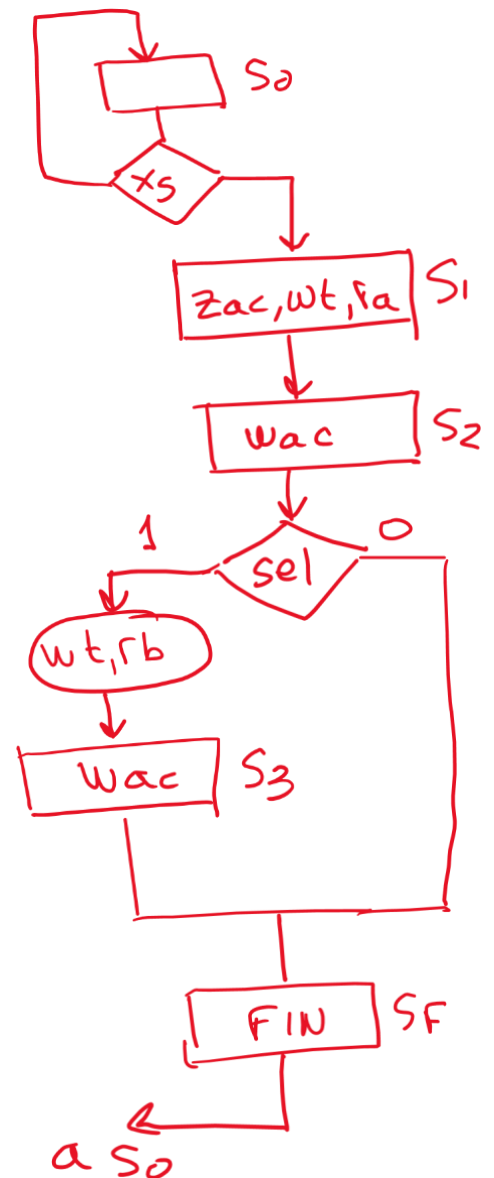
- c. Obtenga el código máquina (binario y hexadecimal) de la instrucción ADD3 R4,R7,R2

$\underbrace{10000}_{COPOP} \underbrace{10000}_{R_d} \underbrace{010}_{R_{f2}} \underbrace{111}_{R_{f1}} \rightarrow 0x8417$

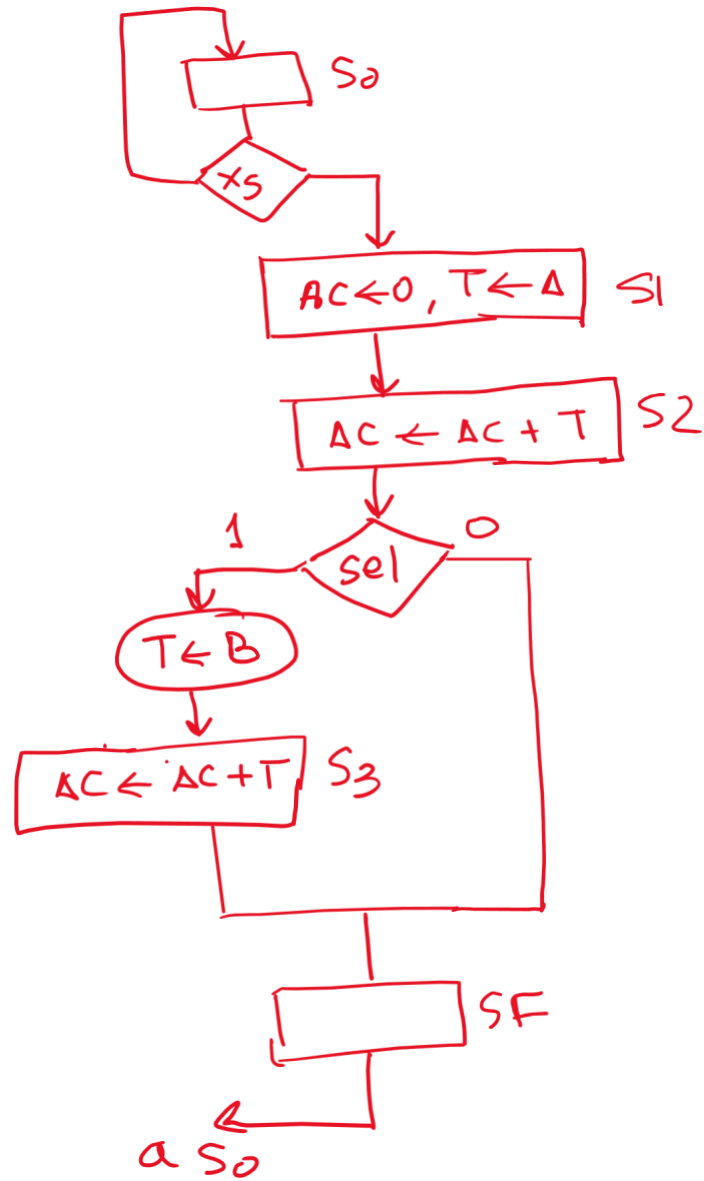
Ejercicio 3 (3 puntos)

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
```



Podemos también obtener la carta ASM de la unidad de datos:



la función es:

SEL	
0	$AC \leftarrow A + B$
1	$AC \leftarrow A$