

EdC-TI	3ª convocatoria	Diciembre 2020
Apellidos:		Nombre:

Ejercicio 1. (4 puntos)

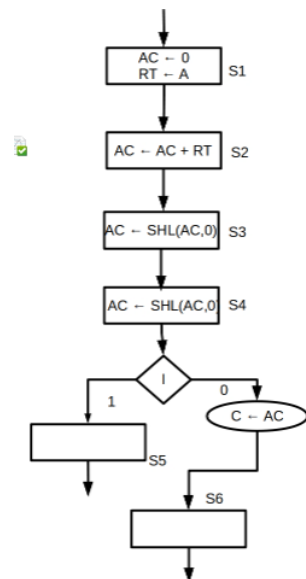
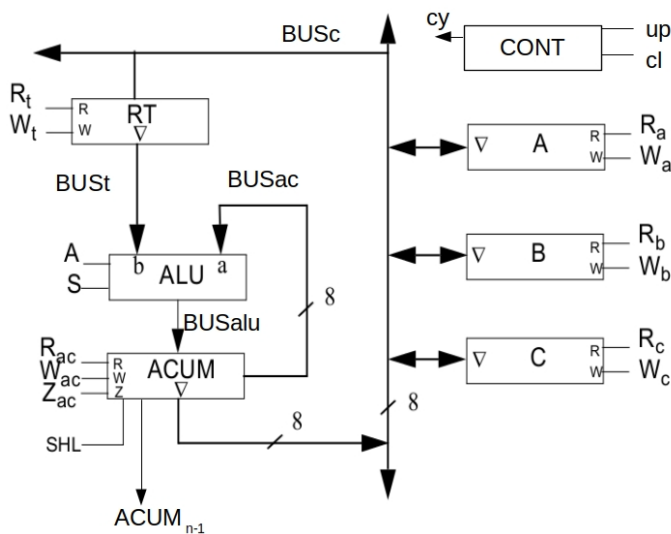
- Responda a las siguientes cuestiones:

 - Diga la diferencia entre las instrucciones CALL 127 y JMP 127
 - Para el CS3, obtenga el código máquina de cada una de las siguientes instrucciones y diga qué formato tienen:
CALL 127
LDI R7,160
 - Escriba las líneas de programa necesarias para hacer que el Timer1 trabaje en modo CTC y con frecuencia clk/8
- Considere el computador CS3. En la siguiente figura se muestra el contenido del fichero de registros, el PC, el SP y un trozo de la memoria DATMEM. Se quiere conocer el resultado de las instrucciones ST Z, R4 y SUB R1, R2. Para ambas instrucciones, conteste:

 - ¿Cuál es el registro o la posición de memoria afectada y su contenido?
 - Tras la ejecución de la instrucción, ¿se ha modificado el contenido del registro SREG?

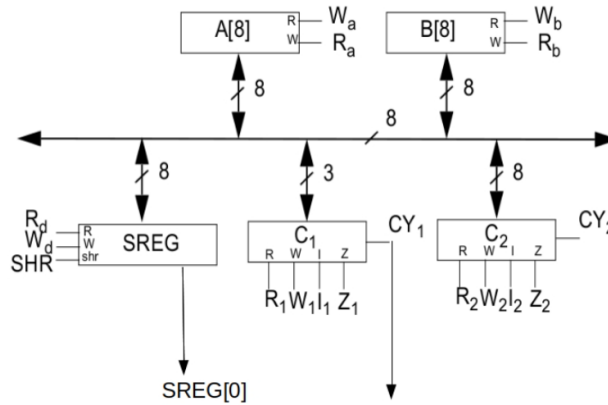
		DATMEM	
R0	0	10	0
R1	127	11	10
R2	-128	12	-10
R3	1	13	20
R4	-1	14	100
R5	12		
R6	12	Y	PC
R7	14	Z	SP

- Sobre la unidad de datos mostrada en la figura se va a realizar un conjunto de operaciones de las cuales sólo extraemos el segmento de la carta ASM mostrado.



- Si AC = 2, A = 4 y C=0, determine el valor del registro C en S5
- Complete sobre la plantilla el código Verilog de los estados S3 y S4

4. En la siguiente figura se muestra la unidad de datos de un SD cuyo código Verilog se muestra parcialmente debajo. Se trata de completar el módulo Unidad_Datos que se acompaña, pues está incompleto. Cree las instancias que se indican dentro del módulo Unidad_Datos



```

module registro #(parameter n=8) (input clk, r, w, shr, inout [n-1:0] bus, output o);
    reg [n-1:0] q;
    ..... // No hay que completar este módulo
    assign o = q[0];
endmodule

module contador #(parameter n=8) (input clk, r, w, i, z, inout [n-1:0] bus, output cy);
    reg [n-1:0] q;
    ..... // No hay que completar este módulo
    assign cy = &q;
endmodule

module Unidad_Datos #(parameter n=8)
    (input clk, wa, ra, wb, rb, wd, rd, shr, r1, w1, i1, z1, r2, w2, z2, i2, cy1, sreg0 );

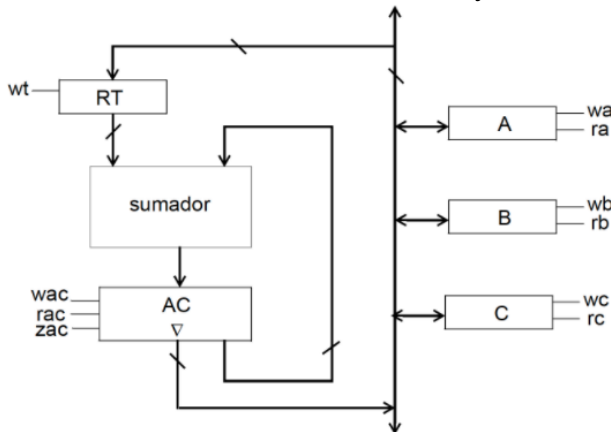
    // Completar: debe contener las instancias del reg. de desplazamiento SREG y los cont. C1 y C2
    wire [7:0] BUS;
    .....
endmodule

```

Ejercicio 2. (3 puntos)

Mediante la unidad de datos de la figura se desea realizar dos operaciones diferentes en función del valor de una variable S. Cuando S = 0 se realizará $B \leftarrow 3(A + C)$ y cuando S=1 se realizará $B \leftarrow 4(A + C)$. Se pide:

1. Describir a nivel RT el registro AC.
2. Un diagrama de bloques donde muestre claramente las entradas y salidas de las unidades de datos y de control.
3. La carta ASM de la unidad de control y su descripción en Verilog.



Ejercicio 3. (3 puntos)

Se desea diseñar un programa para el ATmega328P que permita el control de las luces de un árbol de Navidad. En concreto, el sistema se pondrá en marcha tras accionar un interruptor conectado a PD0, cada pulsación cambia el tiempo de intermitencia de las luces dando lugar a tres modos de funcionamiento: 1sg de intermitencia con la primera pulsación, 3 sg con la segunda o 6 sg con la tercera, la cuarta vuelve a 1sg y así se repite de forma cíclica. Se debe de utilizar una variable para escribir el modo de funcionamiento y otra que contabilice los segundos transcurridos (mod. 6) desde el último cambio del estado de las luces. Las luces se encienden poniendo a 1 un actuador conectado a PD1, y se apagan si dicho actuador está a cero. Se contabiliza 1 sg mediante interrupciones del temporizador.

- a) Escriba la rutina de configuración de entrada salida. Esta rutina se encargará de configurar adecuadamente el puerto D. También deberá de configurar el temporizador para que se generen interrupciones cada segundo, sabiendo que el reloj trabaja a una frecuencia de 1 MHz.
- b) Escriba la rutina de interrupción que en función del modo de funcionamiento active el apagado y encendido de las luces.
- c) Escriba el programa principal. Este se encargará de definir e inicializar adecuadamente todas las variables. Instalará la rutina de interrupción. Posteriormente gestionará la secuencia de pulsaciones guardando el modo de funcionamiento en la variable que luego consultará la rutina de interrupción.