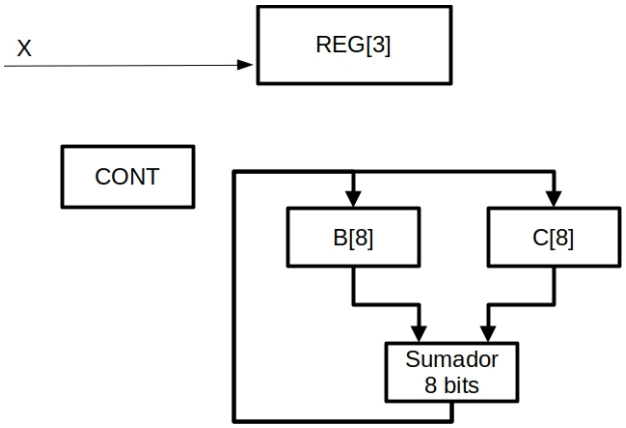
Estructura de Computadores - Tecnologías Informáticas – Segunda Convocatoria Julio 2023

Apellidos (en **MAYÚSCULA**):	Nombre (en **MAYÚSCULA**):

Problema 1 (Alumnos con todo suspenso o con primer parcial suspenso)

Diseñar un Sistema Digital (SD) que, entre otros elementos, dispone de una entrada X, de un bit, por la que se recibe, en serie, un número de tres bits que se almacena en el registro REG. En concreto, tras activarse la señal de inicio, Xs, por la entrada X, sincronizada con la señal de reloj, se recibe, bit a bit, el número que se almacena en el registro. A continuación, el SD deberá determinar si el contenido de REG es mayor o igual a 4. Si lo es, deberá realizar la operación B← 2(B+C), donde B y C son registros de 8 bits. En caso contrario, para REG menor que 4, la operación a realizar es C←3B.

1. Complete la Unidad de Datos de la figura añadiendo las señales de control necesarias para su correcto funcionamiento.
2. Dibuje los bloques del SD (unidad de datos y de control) mostrando sus señales.
3. Determine la carta ASM.



Problema 2 (Alumnos con todo suspenso o con primer parcial suspenso)

Se desea añadir al CS3 la instrucción STIP , cuya sintaxis es la siguiente: `STIP Z+,dato`

Dicha instrucción almacena una palabra de 8 bits (denominada dato en la sintaxis) en la dirección de la memoria de datos indicada por el puntero Z y posteriormente incrementa dicho puntero.

Se pide:

a. El formato de la nueva instrucción y asignar un código de operación disponible. Posteriormente, como ejemplo, escribir el código máquina de la instrucción STIP Z+, 127.

b. Indicar qué cambios habría que realizar en la unidad de datos para poder implementar la instrucción en el caso de que fuera necesario.

c. La secuencia de microoperaciones, junto con las señales a activar por la UC tanto para el ciclo de búsqueda como para el de ejecución.

d. El código máquina de un programa que, haciendo uso de STIP, y las instrucciones que necesite del CS3, inicialice la memoria de datos a partir de la dirección $C0 con los valores:

-1,+25 y -128.

Problema 3 (Alumnos con todo suspenso o con segundo parcial suspenso)

```
.include "m328pdef.inc"
.dseg
.org $100
.equ N = 200
Tabla_numeros:   .byte N
Tabla_negativos: .byte N
Tabla_pares:     .byte N
Negativos:       .byte 1
Pares:           .byte 1
.cseg
.org 0
```

En las líneas superiores se muestra el contenido inicial de un fichero en ensamblador. Como puede ver, hay definidas una serie de etiquetas que debe utilizar cuando diseñe el programa principal, que es lo que falta en el fichero y se le pide que realice.

Debe suponer que la tabla referenciada por la etiqueta **Tabla_numeros** está ya creada y contiene **N** números de **8 bits con signo**.
El programa pedido debe recorrer la tabla referenciada por **Tabla_numeros**, creando dos tablas nuevas, referenciadas por las etiquetas **Tabla_negativos** y **Tabla_pares**.
En la tabla **Tabla_negativos** el programa irá copiando los datos de la tabla **Tabla_numeros** que sean negativos. De modo similar, en la tabla **Tabla_pares** se irán copiando los datos de la tabla **Tabla_numeros** que sean pares.
Tenga en cuenta que es posible que un número sea par y negativo a la vez y deberá copiarse en las dos tablas.
Al finalizar el programa, en la dirección **Negativos** se debe escribir el número de valores negativos que se han copiado en la tabla **Tabla_negativos**. Del mismo modo, en la dirección **Pares** se debe escribir el número de valores pares de la tabla **Tabla_pares**.

NOTA: todos los problemas tienen el mismo peso en las puntuaciones.