

Apellido Bonifazi, Nombre Alba Bonifazi, DNI 12345678**Ejercicio 1**

Diseñe un **codificador Exceso-3** que tenga un número BCD a la entrada (número de 4 bits del 0 al 9) y produzca una salida BCD Exceso-3, (número de 4 bits del 3 al 12) como se muestra en la siguiente tabla.

Entrada	Salida
0	3
1	4
⋮	⋮
9	12

Se pide **minimizar** ecuaciones utilizando el método de **Karnaugh**. Implementar utilizando compuertas tipo AND, NOT y OR de cualquier número de entradas. No puede utilizar un sumador para resolver este problema, solo compuertas del tipo indicadas.

Ejercicio 2

Se pide diseñar un circuito secuencial para controlar un sistema de semáforos en una esquina *tipo T*, formada por una ruta que viene de Norte a Sur (mano única) y una calle lateral que viene de Oeste a Este (mano única). El sistema tiene un semáforo que mira al Norte, para los autos que vienen de Norte a Sur, y otro que mira al Oeste para los autos que vienen de Oeste a Este.

La luz verde dura 60 segundos, luego sigue el amarillo 10 segundos y a continuación el rojo 70 segundos, y luego el ciclo se repite. Este ciclo y estos tiempos son válidos para ambos semáforos. Obviamente mientras uno está en verde o amarillo, el otro está en rojo y viceversa. En el mismo instante cuando uno cambia de amarillo a rojo, el otro cambia de rojo a verde.

Ayuda: Utilizar un reloj con periodo de 10 segundos (un flanco ascendente cada 10 segundos).

Implementar con cualquier tipo de compuertas o circuitos combinacionales (muxes, decodificadores) que necesite.

Ejercicio 3

¿Qué hace el siguiente programa? Explicar y dar un ejemplo que apoye la hipótesis.

```
# $s0 contiene el valor inicial
# primera parte, resultado en $t0
srl $t0, $s0, 23
andi $t0, $t0, 255
addi $t0, $t0, 1
sll $t0, $t0, 23
```

```
# segunda parte, resultado en $t1
li $t1, 255
sll $t1, $t1, 23 → sll
nor $t1, $t1, $t1
and $t1, $s0, $t1
# junto resultados
or $s0, $t0, $t1
# $s0 contiene el valor final
```

Ejercicio 4

La memoria de una computadora MIPS32 se encontró este contenido:

Address	Contents
0x00400000	0x00104021
0x00400004	0x24090000

0x00400008	0x01314821
0x0040000c	0x2508ffff
0x00400010	0x1500fffd
0x00400014	0x00099021

1. Decidir si son **programas** o **datos**. Justificar.
2. En caso de ser programa: desensamblar, comentar el código y describir de la forma más abstracta posible lo que realiza.
3. Si son datos, darles alguna interpretación.

Ejercicio 5

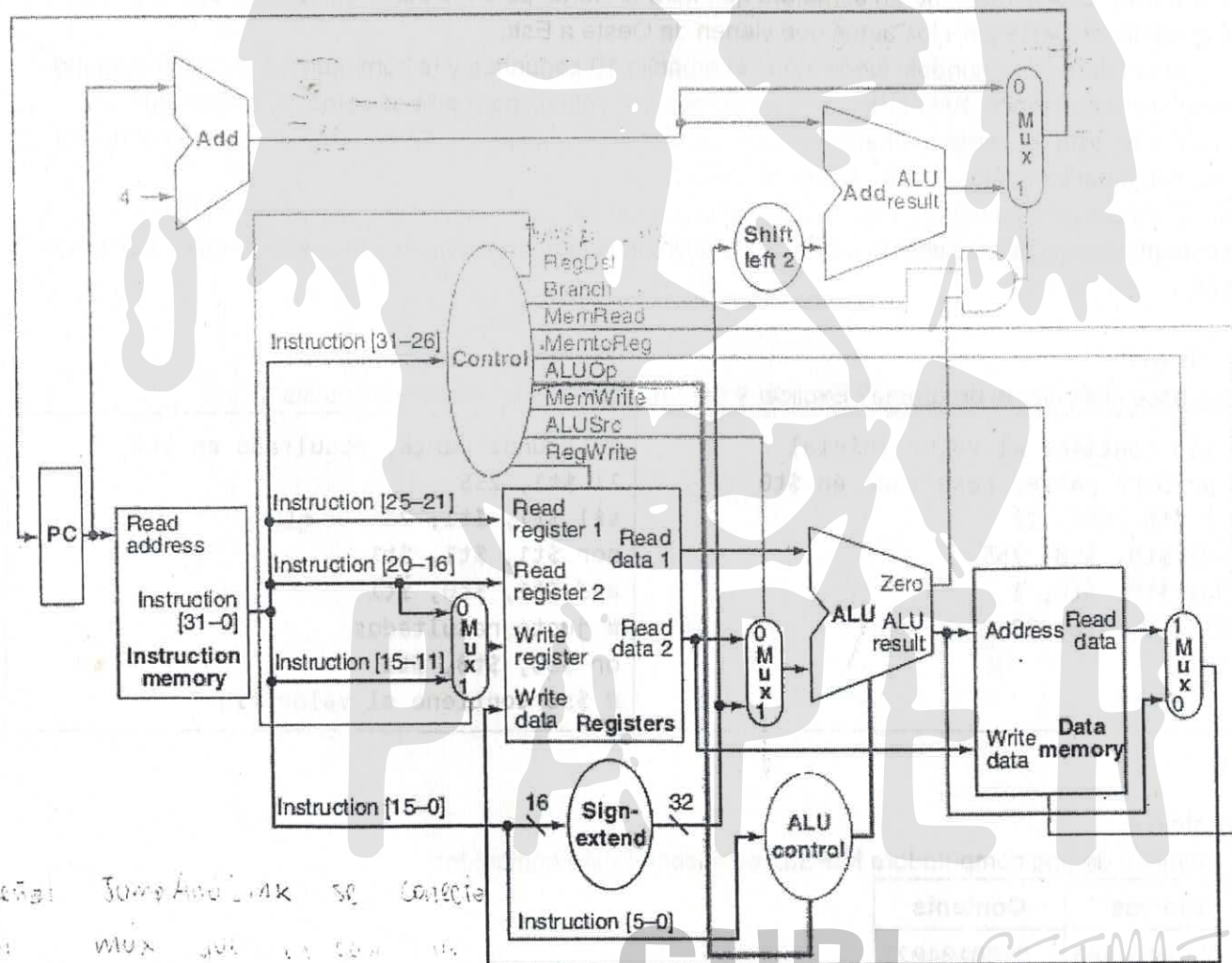
1. Explique qué sucede si la memoria está completamente llena de 0's y el PC empieza a ejecutar en 0x00000000.
2. ¿Explique cómo multiplicaría por dos un número IEEE 754 de manera rápida y sencilla?

Ejercicio 6

Modifique el diagrama de implementación de MIPS para agregar la instrucción jal. Suponga que desde el Control ya sale una señal JumpAndLink que está a 1 si y solo si el opcode es el 0x0 y la función es 0x3.

Ejercicio 7

Muestre la **implementación interna** del sumador que está en la esquina superior izquierda del diagrama de implementación de MIPS. Incluya la constante 4 que alimenta al puerto b del sumador.



La señal JumpAndLink se conecta a un mux que selecciona entre el PC actual y el PC de destino. El resultado de la ALU se conecta al puerto b del sumador.