

Proyecto Integrador N°0

Ignacio Nahuel Chantiri 69869/1

Facultad de Ingeniería de la Universidad Nacional de La Plata

1. Introducción

El siguiente informe resume las características del sistema desarrollado, la justificación de la elección de sus partes, y un desglose de las distintas operaciones posibles. El sistema lee secuencialmente instrucciones de 8 bits almacenadas en una memoria EPROM de 4096 celdas. El principal propósito es manejar hasta 2 operandos de 4 bits y realizar dos operaciones aritméticas simples, una de suma y otra de multiplicación por dos, siguiendo una secuencia de instrucciones programable.

2. Estructura

2.1 Diagrama general

El esquema simplificado del sistema se dibuja en la Figura 1:

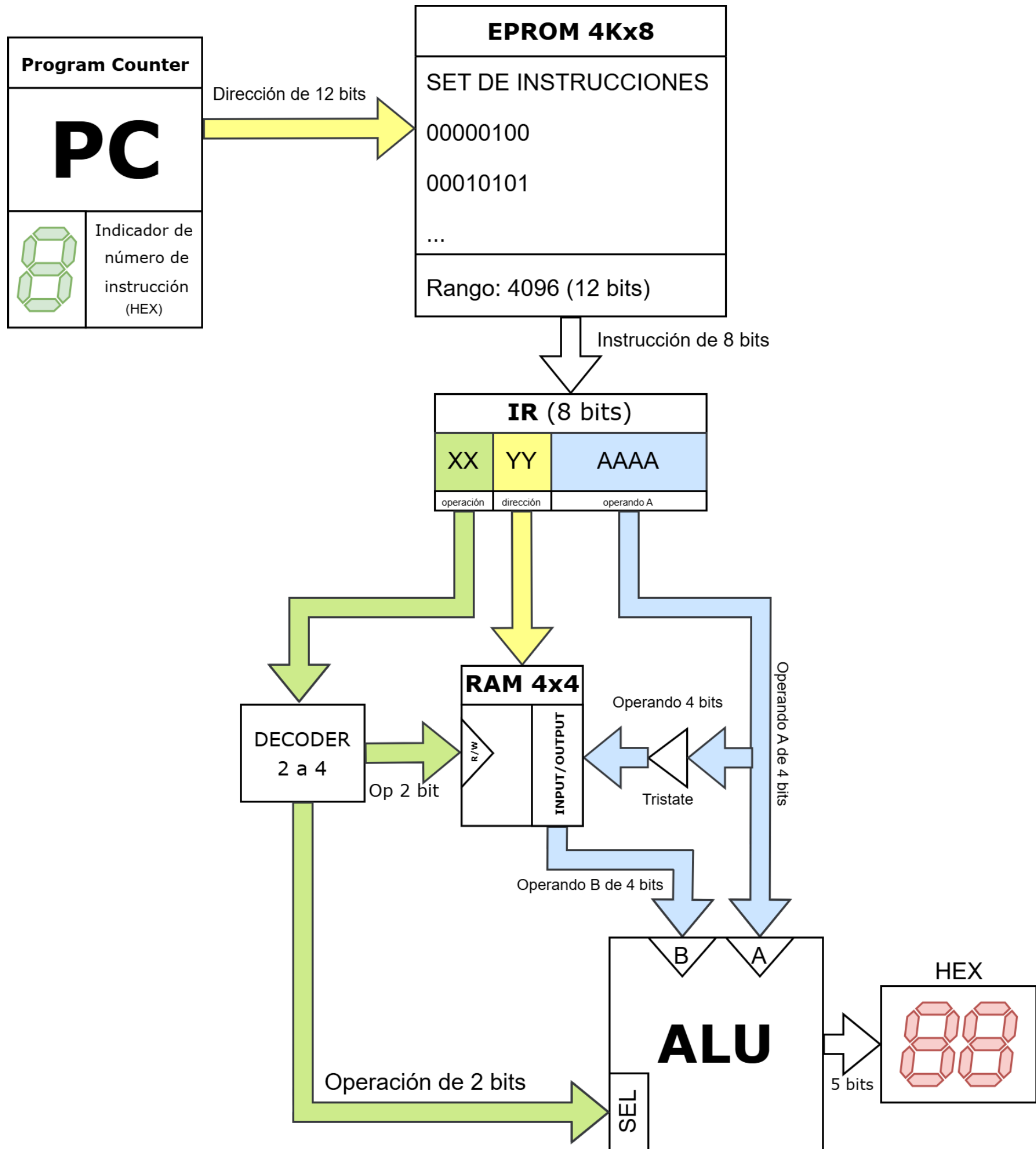


Figura 1: Diagrama general del sistema implementado.

En el Anexo al final de este documento se encuentra la implementación real en Proteus.

2.2 Línea temporal

El modo de trabajo del sistema se desarrolla durante dos ciclos del reloj. Siguiendo el esquema anterior, la línea temporal es la siguiente:

- Al primer flanco ascendente, el **PC** (Program Counter) se actualiza al valor 0000 0000.
- La **EPROM**, encargada de almacenar las instrucciones, recibe este valor como una dirección, y pone a su salida el contenido de la dirección 0000 0000.
- El **IR** (Registro 8 bits) toma este contenido en el siguiente flanco ascendente y lo interpreta como una instrucción con formato XX - YY - AAAA (8 bits). A partir de aquí, cada sección de la instrucción toma un camino distinto.
- La operación codificada en los dos primeros bits (XX) se decodifica en el **Decoder 2 a 4**, que indica a la **ALU** la operación aritmética que debe realizar; y a la **RAM** si debe leer, escribir o apagarse.
- La dirección del tercer y cuarto bit (YY) pasa a seleccionar una de las celdas de la **RAM**, que almacena un operando.
- Los últimos 4 bits (AAAA) son de datos, y pueden servir como operando en la **ALU**, o pueden ser guardados en la misma **RAM**, según la operación indicada.

3. Instrucciones

3.1 Formato de las instrucciones

Las instrucciones cargadas en la EPROM son de 8 bits y cumplen con el siguiente formato:

XX - YY - AAAA

Donde:

- **XX**: Operación
- **YY**: Dirección de memoria de la RAM
- **AAAA**: Operando A

3.2 Operaciones

Las posibles operaciones del sistema se indican en la siguiente tabla:

Código de Instrucción	Instrucción
11 - YY - AAAA	Leer ⁽¹⁾ de la RAM la dirección YY
00 - YY - AAAA	Escribir en la RAM el dato AAAA en YY
01 - YY - AAAA	Sumar el dato de la posición YY con AAAA
11 - YY - AAAA	Realizar la operación 2A (shift)

Cuadro 1: Tabla de operaciones

(1) El dato leído se muestra en el display del operando B.

3.3 Programa

Con el archivo .bin adjunto, se cargó en la memoria EPROM el siguiente conjunto de 14 instrucciones que el sistema recorre de manera secuencial:

Nº	OP	DIR	DATA	HEX	INSTRUCCIÓN
0	00	00	0110	06	ESCRIBE EL 0110 (6) EN LA DIRECCIÓN 00
1	00	01	0110	16	ESCRIBE EL 0110 (6) EN LA DIRECCIÓN 01
2	00	10	0001	21	ESCRIBE EL 0001 (1) EN LA DIRECCIÓN 10
3	00	11	1111	3F	ESCRIBE EL 1111 (15) EN LA DIRECCIÓN 11
4	11	00	0000	C0	LEE LO QUE HAY EN LA DIRECCIÓN 11 (15)
5	11	01	0000	D0	LEE LO QUE HAY EN LA DIRECCIÓN 10 (1)
6	11	10	0000	E0	LEE LO QUE HAY EN LA DIRECCIÓN 01 (6)
7	11	11	0000	F0	LEE LO QUE HAY EN LA DIRECCIÓN 00 (6)
8	01	00	0001	41	SUMA LO QUE ESTÁ EN LA DIRECCIÓN 00 (6) CON 0001 (1) = 7
9	01	01	0010	52	SUMA LO QUE ESTÁ EN LA DIRECCIÓN 01 (6) CON 0010 (2) = 8
A	01	10	1000	68	SUMA LO QUE ESTÁ EN LA DIRECCIÓN 10 (1) CON 1000 (8) = 9
B	01	11	0001	71	SUMA LO QUE ESTÁ EN LA DIRECCIÓN 11 (15) CON 0001 (1) = 16
C	10	00	0001	81	TOMA EL OPERANDO A (0001)(1) Y LO MULTIPLICA POR 2 = (00010) 2
D	10	00	1000	88	TOMA EL OPERANDO A (1000)(8) Y LO MULTIPLICA POR 2 = (10000) 16

Cuadro 2: Set de instrucciones

3.4 Contenido del archivo .bin

Las instrucciones entonces quedan escritas en hexadecimal y de manera secuencial del siguiente modo:

06
16
21
3F
C0
D0
E0
F0
41
52
68
71
81
88

4. Componentes

A continuación se detallan los componentes utilizados para cada bloque.

4.1 Program Counter (PC)

Es un **Contador Sincrónico de 12 bits** compuesto por flip-flops tipo 'D'. Se utilizaron 3 integrados CD4076, que contienen 4 flip-flops D cada uno. Se implementó además un bloque de lógica combinacional adecuada para conectarlos entre sí.

4.2 Memoria EPROM de instrucciones

Esta memoria de 32K (4Kx8) contiene el programa (4096 posibles instrucciones de 8 bits cada una). Se utilizó un integrado F2732.

4.3 Registro IR

El registro **IR** es de una celda de 8 bits y está conformado por dos integrados CD4076 de 4 bits cada uno. Este registro está obviamente sincronizado con el reloj, por lo que las instrucciones se actualizan con el flanco ascendente. La función del **IR** es contener la instrucción actual y dividirla en secciones para su procesamiento.

4.4 Unidad Aritmético-Lógica (ALU)

La **ALU** implementada tiene sólo dos operaciones principales:

- Una **suma de dos operandos de 4 bits (A + B)**
- Un shift de una cifra a la izquierda de los bits del operando A, o **multiplicación de A por dos (2*A)**

La elección del modo se realiza con un selector de 2 bits, OP1 y OP0, y el resultado se muestra en la salida de 5 bits.

Por un lado, la suma se realiza con un predictor de carry de 4 bits, mientras que el shift es una simple conexión entrada-salida adecuada, sin lógica combinatoria, limitando el corrimiento de bits a solo una posición. Además se incluyó un modo de alta impedancia de salida con tres tristate cuádruples 74125, que permite apagar un display de salida cuando no se está utilizando ninguna operación.

El set de operaciones posibles queda definido en la siguiente tabla:

Selección modo		Salidas activo en alto	
OP1	OP0	Operación	Descripción
0	0	Z	Salida en alta impedancia
0	1	A + B	Suma entre el operando A y el
1	0	2A	Multiplicación del operando A por dos (shift)
1	1	Z	Salida en alta impedancia

Cuadro 3: Tabla de operaciones

4.5 RAM 4X4

La memoria **RAM de 4 celdas de palabras de 4 bits** está conformada por flip-flops tipo 'D' individuales (reutilizado del ejercicio de la práctica). Tiene capacidad de lectura y escritura por el mismo bus de datos (IO) se accede a cada dirección de memoria con los selectores S1 y S0.

4.6 Decoder de operaciones

Es un pequeño **módulo decodificador de 2 a 4 bits**. Toma como entrada los primeros 2 bits de las instrucciones, y a su salida dispone dos bits hacia la **ALU** para seleccionar la operación aritmética, y otros dos hacia la **RAM** que indica lectura, escritura o apagado. La lógica que permite este decoder es la siguiente:

- Cuando se recibe la instrucción de **escribir**, pone el pin RW de la **RAM en modo escritura y en enable**, y **enciende el tristate** para el paso de la información a los pines IO de la RAM. Además selecciona el **modo alta impedancia en la ALU**.
- Cuando se recibe la instrucción de **leer**, pone el pin RW de la **RAM en modo lectura y en enable**, y **apaga el tristate** para cortar la llegada de información a los pines IO. Además selecciona el **modo alta impedancia en la ALU**.
- Cuando se recibe la instrucción de **sumar**, pone el pin RW de la **RAM en modo lectura** (para acceder al operando B) **y en enable**, y los selectores de la **ALU en modo suma**.
- Cuando se recibe la instrucción de **multiplicar A por 2, apaga la RAM con el enable** (para apagar la pantalla del operando B y solo mostrar el A), y pone los selectores de la **ALU en modo multiplicación**.

Se realizó con dos de las 4 compuertas OR del integrado 4071, un negador y las conexiones adecuadas.

4.7 Tristate de 4 bits

Se utilizó un tristate cuádruple 74125. Su función es el control de datos que llegan a los IO de la **RAM**, permitiendo en un caso conectarla a un bus para su escritura, y en el otro cortar el bus para que pueda ser leída sin interferencia.

5. Entradas y Salidas de información

5.1 Entradas

Solo se cuenta con un método de entrada:

- **BOTÓN SIGUIENTE INSTRUCCIÓN**: Es el botón que se encuentra al lado del display que indica el número de instrucción. Al presionarlo, el sistema continúa a la siguiente instrucción.

5.2 Salidas

La información se accede mediante displays y leds:

- **DISPLAY 7 SEGMENTOS OPERANDO A**: Muestra el dato de 4 bits del operando A en formato hexadecimal.
- **DISPLAY 7 SEGMENTOS OPERANDO B**: Muestra el dato de 4 bits del operando B en formato hexadecimal. La operación de lectura muestra el dato leído a través de este mismo display.
- **DOBLE DISPLAY 7 SEGMENTOS RESULTADO**: Solo se enciende cuando la ALU se utiliza alguna de las operaciones aritméticas. Muestra el resultado de 5 bits en formato hexadecimal.
- **DISPLAY NÚMERO DE LÍNEA DE INSTRUCCION**: Muestra la instrucción actual, en formato hexadecimal.
- **LEDS INDICADOR DE OPERACIÓN**: El led encendido indica la operación actual. R = READ, W = WRITE, + = SUMA, X = MULTIPLICACIÓN.

6. Anexo: Implementación en Proteus

A continuación se muestra el diagrama final implementado en Proteus junto con el tablero de control.

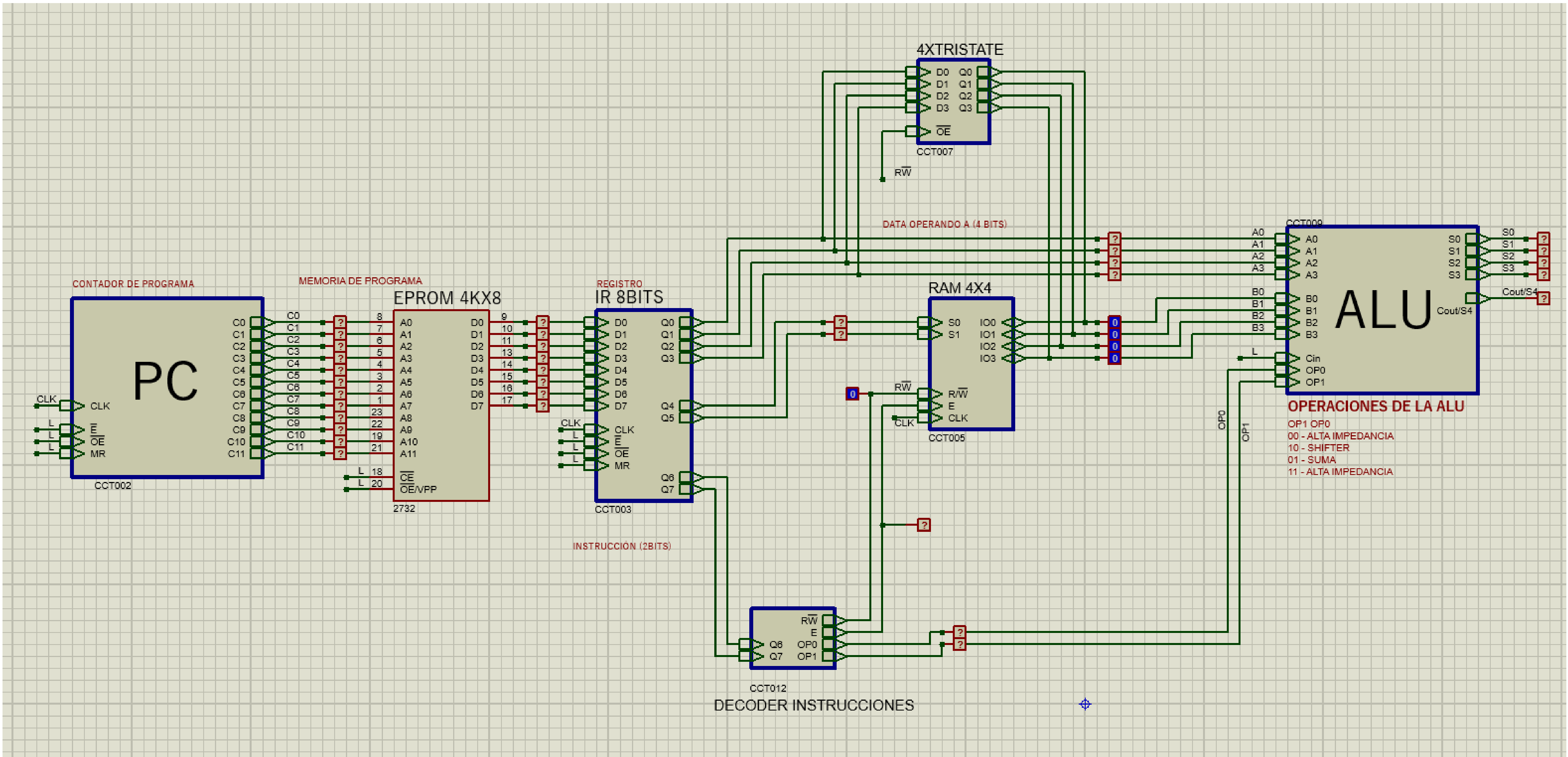


Figura 2: Implementación del sistema en Proteus

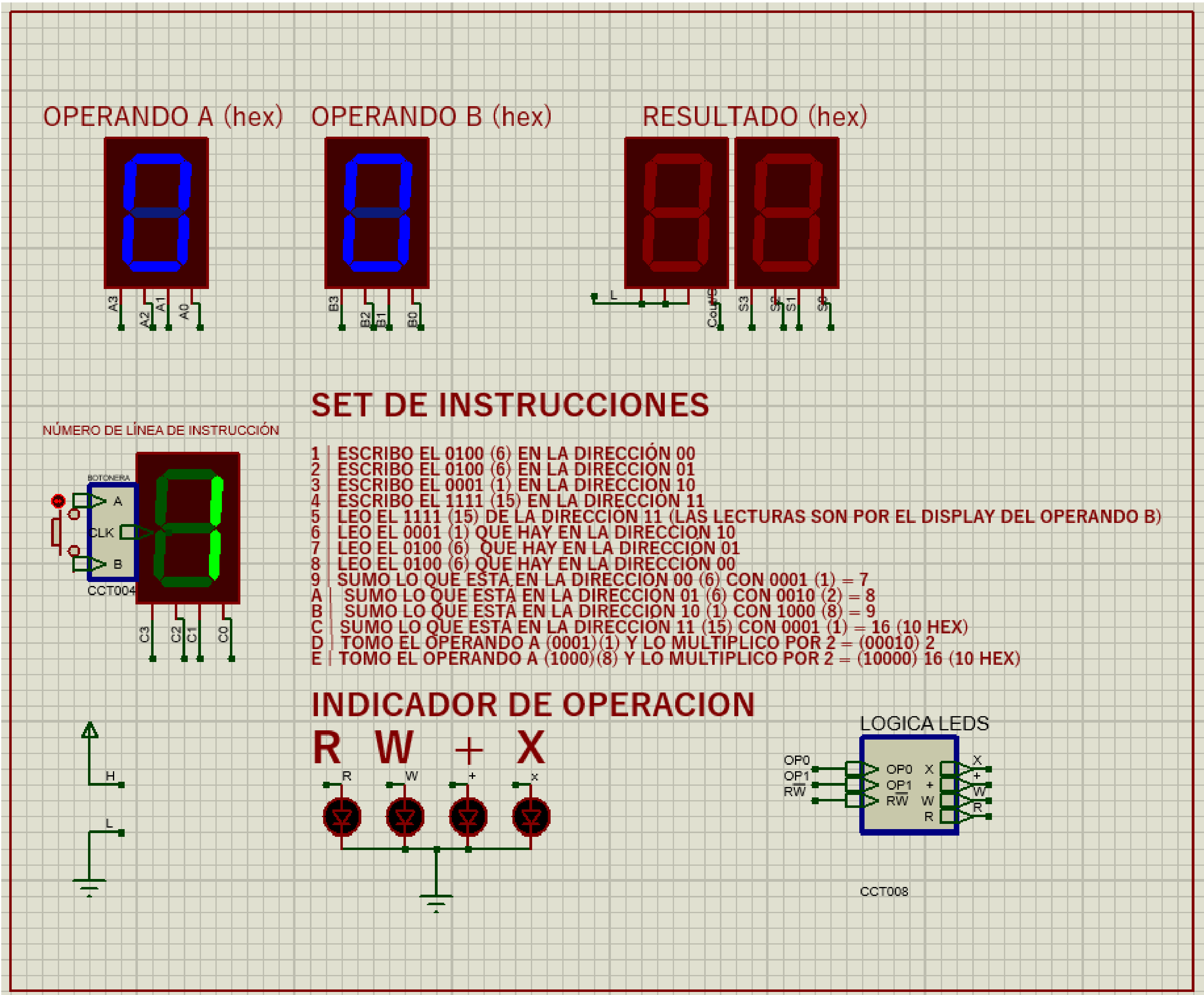


Figura 3: Tablero de control