



UNIVERSIDAD DE SAN CARLOS DE GUATEMALA  
USAC  
Facultad de Ingeniería

**Laboratorio:** Organización Computacional

**Sección:** P

**Docente:** Jorge Alejandro De León Batres

## Practica #2:

### Unidad Aritmética Lógica (ALU)

| Información de alumnos              |           |               |
|-------------------------------------|-----------|---------------|
| Nombre                              | Carné     | Participación |
| Luis Antonio Catalán Pérez          | 202147534 | 100 %         |
| Hengel José Roberto Contreras Tobar | 202140315 | 100 %         |
| Geovany Josué Calderón Díaz         | 202042465 | 100 %         |
| José Alfonso Villela Villeda        | 201944260 | 100 %         |

## **INTRODUCCIÓN**

En el ámbito de la electrónica digital, los bloques MSI (Medium Scale Integration, por sus siglas en inglés) desempeñan un papel crucial en la construcción de circuitos complejos. Dentro de esta categoría, los bloques MSI tipo aritmético son especialmente importantes para realizar operaciones aritméticas básicas, como sumas, restas y multiplicaciones, de manera eficiente y confiable.

Los bloques digitales combinacionales MSI tipo aritmético están diseñados para realizar operaciones matemáticas utilizando compuertas lógicas, registros y otros componentes digitales. Estos bloques están configurados para recibir dos o más operandos y generar una salida que representa el resultado de la operación.

## DESCRIPCIÓN

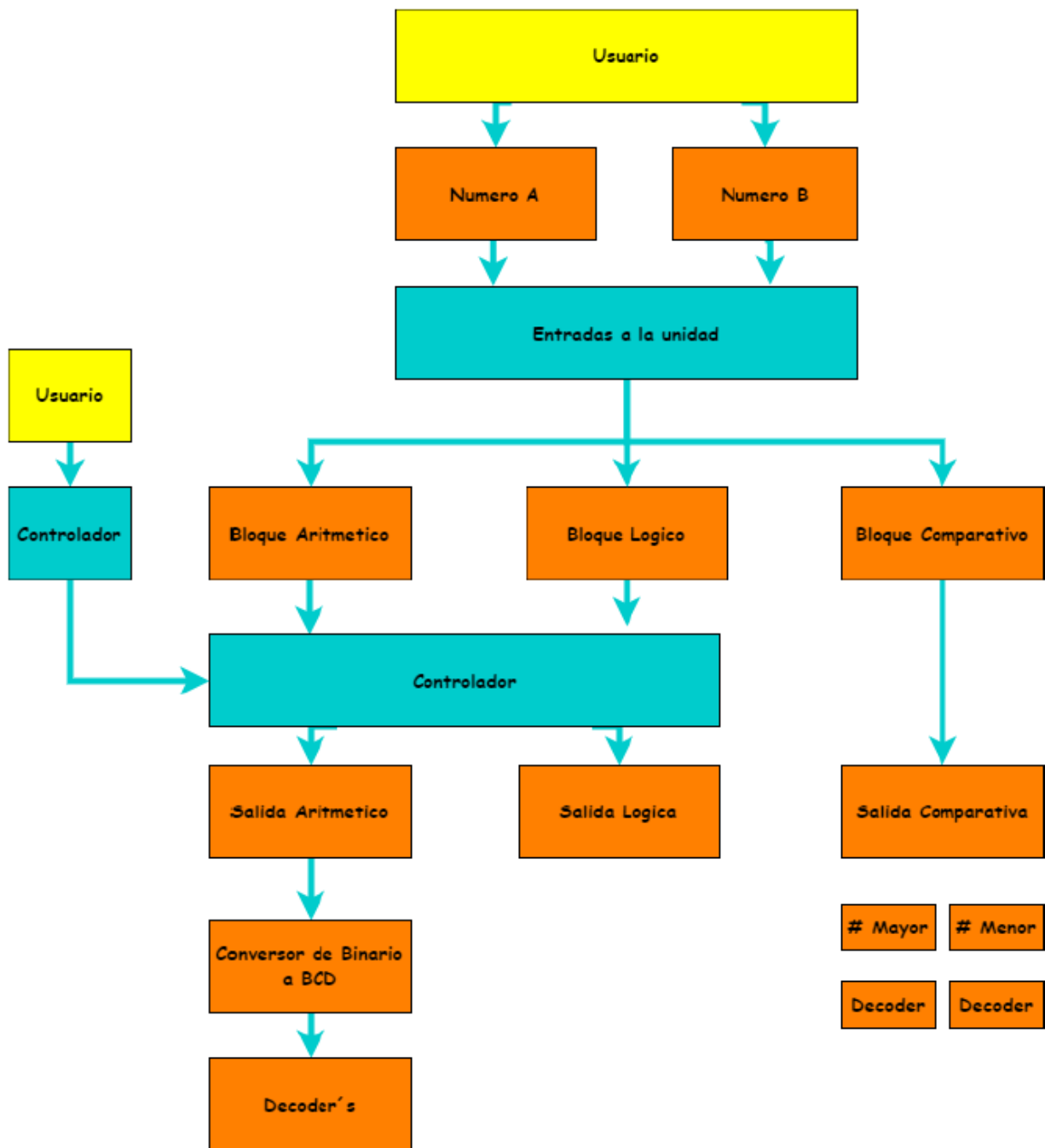
Como estudiantes del curso Organización Computacional, han sido contratados por Intel Corporation para desarrollar un prototipo de calculadora llamado "LogicCalc". Intel busca una solución óptima basada en lógica combinatorial que sea capaz de realizar cálculos aritméticos y lógicos. Para cumplir con estos requisitos, Intel ha proporcionado las especificaciones únicas para una Unidad Aritmética Lógica Básica (ALU).

## REQUERIMIENTOS

Se desea elaborar un circuito combinatorial, el cual sea capaz de resolver operaciones Aritméticas, Lógicas y Comparativas entre 2 números binarios de 3 bits. Para lo cual se debe de implementar un controlador que será el encargado de seleccionar la operación deseada por el usuario. El controlador manejará las operaciones deseadas por medio de la definición de la siguiente tabla:

| B | A | Operación |
|---|---|-----------|
| 0 | 0 | Suma      |
| 0 | 1 | Resta     |
| 1 | 0 | AND       |
| 1 | 1 | OR        |

## DIAGRAMA DE BLOQUES ALU REQUERIDO



NOTA: la unidad comparativa no será requerida según acuerdo realizado en laboratorio, solo se trabajará sobre la unidad aritmética y la unidad lógica.

## UNIDAD ARITMÉTICA

Esta unidad estará encargada de realizar el cálculo aritmético de las entradas a la Unidad.

Por lo cual se deberá tener en cuenta la selección de la operación en el controlador:

- **Suma:** Procederá a realizar la Suma entre 2 Números binarios A y B (Entradas), teniendo en cuenta los valores de acarreo respectivamente. Para las entradas se manejarán solo números positivos. Esta operación se tendrá que implementar enteramente con compuertas lógicas, no se permite utilizar semi-sumadores o sumadores completos.
- **Resta:** Procederá a realizar la Resta entre 2 Números binarios A y B (Entradas), teniendo en cuenta los valores de préstamo respectivamente. Para las entradas se verificará si el minuendo es mayor que el sustraendo, si es así, realizar la resta y mostrar el resultado, de lo contrario, mostrar un mensaje de Error en el Display. No se permite utilizar semi-sumadores o sumadores completos.

| Validación | Resultado                               |
|------------|---|
| $A \leq B$ | Realizar resta,<br>mostrar<br>Resultado |
| $A > B$    | Mostrar EE en<br>display                |

## UNIDAD LÓGICA

Esta unidad estará encargada de realizar el cálculo lógico de las entradas a la Unidad.

Por lo cual se deberá tener en cuenta la selección de la operación en el controlador:

- **AND:** Procederá a realizar la Multiplicación Lógica entre 2 Números binarios A y B (Entradas).
- **OR:** Procederá a realizar la Suma Lógica entre 2 Números binarios A y B (Entradas).

## Controlador

|     |  |
|-----|--|
| S/R | Control de sumador restador (0=Resta, 1=Suma)  |
| D   | Habilita el display (0= display apagado, 1=display encendido)                                      |
| OR  | Habilita compuerta OR para A <sub>1</sub> B <sub>1</sub> (0=OR deshabilitado, 1= OR habilitado)    |
| AND | Habilita compuerta AND para A <sub>1</sub> B <sub>1</sub> (0=AND deshabilitado, 1= AND habilitado) |

| A | B | S/R | D | OR | AND |
|---|---|-----|---|----|-----|
| 0 | 0 | 0   | 1 | 0  | 0   |
| 0 | 1 | 1   | 1 | 0  | 0   |
| 1 | 0 | X   | 0 | 0  | 1   |
| 1 | 1 | X   | 0 | 1  | 0   |

|     |            |
|-----|------------|
| S/R | = A'B      |
| D   | = A'B+A'B' |
| OR  | = AB       |
| AND | = AB'      |

| S/R     |   |   |
|---------|---|---|
| A\B     | 0 | 1 |
| 0       | 0 | 1 |
| 1       | X | X |
| S/R = B |   |   |

| D      |   |   |
|--------|---|---|
| A\B    | 0 | 1 |
| 0      | 1 | 1 |
| 1      | 0 | 0 |
| D = A' |   |   |

| OR      |   |   |
|---------|---|---|
| A\B     | 0 | 1 |
| 0       | 0 | 0 |
| 1       | 1 | 0 |
| OR = AB |   |   |

| AND       |   |   |
|-----------|---|---|
| A\B       | 0 | 1 |
| 0         | 0 | 0 |
| 1         | 0 | 1 |
| AND = AB' |   |   |

## Suma

Sumador completo

| A | B | C <sub>0</sub> | S | C |
|---|---|----------------|---|---|
| 0 | 0 | 0              | 0 | 0 |
| 0 | 0 | 1              | 1 | 0 |
| 0 | 1 | 0              | 1 | 0 |
| 0 | 1 | 1              | 0 | 1 |
| 1 | 0 | 0              | 1 | 0 |
| 1 | 0 | 1              | 0 | 1 |
| 1 | 1 | 0              | 0 | 1 |
| 1 | 1 | 1              | 1 | 1 |

|   |   |
|---|---|
| S | $= A'B'C_0 + A'BC_0' + AB'C_0' + ABC_0$ |
| C | $= A'BC_0 + AB'C_0 + ABC_0' + ABC_0$    |

| S                           |    |    |    |    |
|-----------------------------|----|----|----|----|
| A\BC <sub>0</sub>           | 00 | 01 | 11 | 10 |
| 0                           | 0  | 1  | 0  | 1  |
| 1                           | 1  | 0  | 1  | 0  |
| $S = A \oplus B \oplus C_0$ |    |    |    |    |

| C                                      |    |    |    |    |
|--|----|----|----|----|
| A\BC <sub>0</sub>                      | 00 | 01 | 11 | 10 |
| 0                                      | 0  | 0  | 1  | 0  |
| 1                                      | 0  | 1  | 1  | 1  |
| $D = BC_0 + AC_0 + AB = C_0(B+A) + AB$ |    |    |    |    |

Nota: este sumador completo suma 2 números de 1 bit, pero al agregar esta misma distribución del combinacional en cascada conectando el carry C a el próximo carry C<sub>0</sub> se puede sumar 2 números de n bits.

## Resta

Para realizar la **resta** A-B, se emplea el método del **complemento a dos**. Este método permite restar números utilizando un sumador convencional. La implementación se realiza de la siguiente forma:

**Invertir B si es resta:**

- **XOR entre cada bit de B y la señal S/R:**  
Esta XOR actúa como un inversor controlado.
  - Si S/R = 1 (suma), la salida de la XOR es simplemente B (ya que  $B \oplus 0 = B$ )
  - Si S/R = 0 (resta), la salida es el complemento de B (ya que  $B \oplus 1 = B'$ ).

**Carry-in (C<sub>0</sub>) también controlado por S/R:**

- Si S/R = 0 (suma), se pone C<sub>0</sub> = 0.

- Si  $S/R = 1$  (resta), se pone  $C0 = 1$ , lo que equivale a sumar el +1 del complemento a dos.

### Sumador completo:

Se conecta en cascada por cada bit. Recibe:

- A
- B (modificado con XOR según S/R)
- $C0 = S/R$

$$A - B = A + (B' + 1)$$

### XOR final en el Carry Out:

Una XOR en el carry final para ajustar el acarreo dependiendo del modo de operación.

### Resultado negativo

Se comparan las entradas de A y B con un LS85, la salida de  $A < B$  combinado con la entrada del controlador S/R para manejar la salida directa al display.

### Mostrar números decimales en 2 display

- Usa un **sumador 74LS83** para mostrar resultados de suma o resta.
- Muestra el **dígito de las unidades** (0–9) en el **primer display**.
- Muestra un **1 o 0** en el **segundo display** dependiendo si el resultado total es **mayor o igual a 10 o menor que 10**.
- (ejemplo): A la operación se le suma **+6** para que en caso de resultado negativo en complemento a dos (como  $1110 = -2$ ), al sumarle 6 muestre 4 en el display.



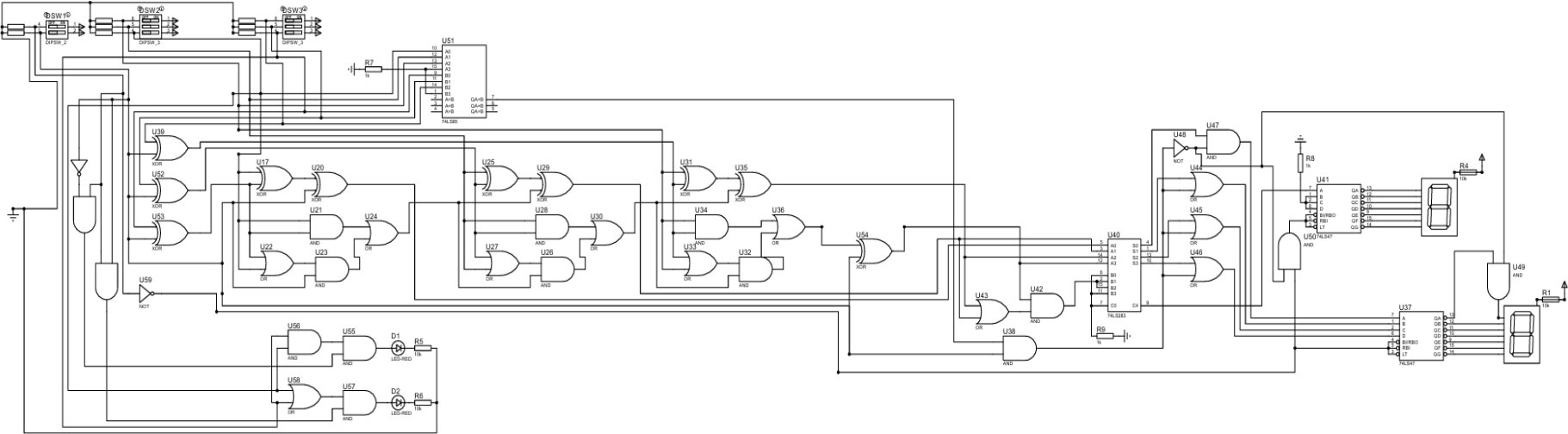
## OR y AND

| Controlador |   |                |                |    |     |
|-------------|---|----------------|----------------|----|-----|
| A           | B   | A <sub>1</sub> | B <sub>1</sub> | OR | AND |
| 1           | 0   | 0              | 0              | 0  | 0   |
| 1           | 0   | 0              | 1              | 1  | 0   |
| 1           | 0   | 1              | 0              | 1  | 0   |
| 1           | 0   | 1              | 1              | 1  | 0   |
| 1           | 1   | 0              | 0              | 0  | 0   |
| 1           | 1   | 0              | 1              | 0  | 0   |
| 1           | 1   | 1              | 0              | 0  | 0   |
| 1           | 1   | 1              | 1              | 0  | 1   |
| OR          | $= AB' B_1 A_1' + AB' B_1' A_1 + AB' B_1 A_1$ |                |                |    |     |
| AND         | $= AB B_1 A_1$                                |                |                |    |     |

| OR                                     |    |    |    |    |
|--|----|----|----|----|
| A B \ B <sub>1</sub><br>A <sub>1</sub> | 00 | 01 | 11 | 10 |
| 00                                     |    |    |    |    |
| 01                                     |    |    |    |    |
| 11                                     |    |    |    |    |
| 10                                     |    | 1  | 1  | 1  |
| OR = $AB' B_1 + AB' A_1$               |    |    |    |    |

| AND                                    |    |    |    |    |
|--|----|----|----|----|
| A B \ B <sub>1</sub><br>A <sub>1</sub> | 00 | 01 | 11 | 10 |
| 00                                     |    |    |    |    |
| 01                                     |    |    |    |    |
| 11                                     |    |    |    |    |
| 10                                     |    |    |    |    |
| AND = $AB B_1 A_1$                     |    |    |    |    |

Diagrama



## **Equipo utilizado**

Para el desarrollo de la práctica numero 2:

Herramientas utilizadas

- Pinza pelacables
- Pinza de corte
- Pinzas para electrónica
- Multímetro
- Cautín para soldadura
- Mano de apoyo para soldadura
- Tijeras
- Pistola para fundir silicón
- Esponja de acero para limpiar cautín
- Marcador

## **Presupuesto**

### **Gastos**

Cable para protoboard: 2 metros (Q 2.0 c/u mts)

Protoboard: 3 protoboard (Q 70.0)

Jumpers: 2 kit (Q32.00 kit)

Resistencias de 330Ω: 12 resistencias (Q1 c/u)

Resistencias de 1KΩ: 3 resistencias (Q1 c/u)

IC TTL 74LS08 (AND): 4 ands (Q8 c/u)

IC TTL 74LS04 (NOT): 1 not (Q8 c/u)

IC TTL 74LS32 (OR): 3 or (Q8 c/u)

IC TTL 74LS86 (XOR): 3 xor (Q8 c/u)

IC TTL 74LS85 (Comparador de 4 bits): 1 comparador (Q10 c/u)

IC TTL 74LS83 (Sumador de 4 bit): 1 sumador (Q 20 c/u)

IC TTL 74LS47 (Decodificador BCD): 2 decoder (Q12 c/u)

LED: 2 (Q 1.00)

Display de 7 segmentos ánodo común: 2 display (Q7.00)

Fuente de Voltaje de 5V (Cargador de celular)

Deep Switch de 3 interruptores: 2 (Q3 c/u)

Deep Switch de 2 interruptores: 1 (Q3.25 c/u)

Estaño: 1mtr (Q 3.00 c/metro)

Recipiente para encapsulado: Q28.00

Presupuesto: Q 479.25

Total, de presupuesto en material para la practica 2, sin herramienta es de: Q479.25

### **Aporte individual de cada integrante**

Luis Catalán

Aporto Q 159,75 en compra de material para la práctica, más el aporte de herramienta ya adquirida previamente para la realización de la práctica no. 2

José Villela

Aporto Q 159,75 en compra de material para la práctica, más el aporte de herramienta ya adquirida previamente para la realización de la práctica no. 2

Geovany Calderón

Aporto Q 159,75 en compra de material para la práctica, más el aporte de herramienta ya adquirida previamente para la realización de la práctica no. 2

Hengel Contreras

Aporto Q 159,75 en compra de material para la práctica, más el aporte de herramienta ya adquirida previamente para la realización de la práctica no. 2

## **CONCLUSION**

En la practica el desarrollo de la unidad de procesamiento aritmético lógico también denominado ALU, se comprendió el funcionamiento de componente como los decodificadores para el uso de componentes como el display de 7 segmentos al recibir números binarios y mostrarlos en decimal, así mismo el funcionamiento de comparadores para comprender si los números cumplían, para poder operarse según la lógica de operaciones como la suma y la resta, también se comprendido el uso de componentes como el sumador para el manejo de los números en binario a operar logrando diseñar una unidad ALU con un numero eficiente de componentes como compuertas lógicas y los antes mencionados.