



TECNOLÓGICO NACIONAL DE MÉXICO  
INSTITUTO TECNOLÓGICO DE TLAXIACO

---

**PRACTICA 1 CIRCUITOS ARITMETICOS Y LOGICOS**

---

**CARRERA:**

Ingeniería en Sistemas Computacionales

**ASIGNATURA:**

Arquitectura de computadoras

**SEMESTRE:**

5

**PRESENTA:**

**NOMBRE DE LOS INTEGRANTES DEL EQUIPO:**

Roldan Uriel Arcadio Ávila 22620100

Ariadna Monserrat López Aparicio 22620052

Tlaxiaco, Oax., 02 de sep de 2024.





## OBJETIVO:

El alumno implementará las operaciones de suma, resta, multiplicación y comparación de 1 bit, basadas en circuitos integrados la familia TTL y/o tecnología MSI, para validar y comprobar su funcionamiento.

## Materiales

- Laptop
- Software de simulación de circuitos digitales (LiveWire, Logisim, Proteus, Multisim, etc.)

## Desarrollo

### 4.1 Circuito Sumador

#### 4.1.1 Implementación

#### Componentes necesarios:

- 2 compuertas AND
- 2 compuertas XOR
- 1 compuerta OR

#### Pasos:

##### 1. Configurar entradas:

- Coloca tres entradas digitales para A, B y Cin (Carry In).

##### 2. Sumar bits:

- Conecta A y B a una compuerta XOR para obtener la primera suma parcial (S1).
- Conecta S1 y Ci a otra compuerta XOR para obtener la suma final S.

##### 3. Calcular el acarreo:

- Conecta A y B a una compuerta AND para obtener el primer acarreo parcial (C1).
- Conecta S1 y Ci a otra compuerta AND para obtener el segundo acarreo parcial (C2).
- Conecta C1 y C2 a una compuerta OR para obtener el acarreo de salida Cout.

#### 4.1.2 Tabla de Verdad

| A | B | Ci | Cout | Suma |
|---|---|----|------|------|
| 0 | 0 | 0  | 0    | 0    |
| 0 | 0 | 1  | 0    | 1    |
| 0 | 1 | 0  | 0    | 1    |
| 0 | 1 | 1  | 1    | 0    |
| 1 | 0 | 0  | 0    | 1    |
| 1 | 0 | 1  | 1    | 0    |
| 1 | 1 | 0  | 1    | 0    |
| 1 | 1 | 1  | 1    | 1    |

#### 4.1.3 Simulación

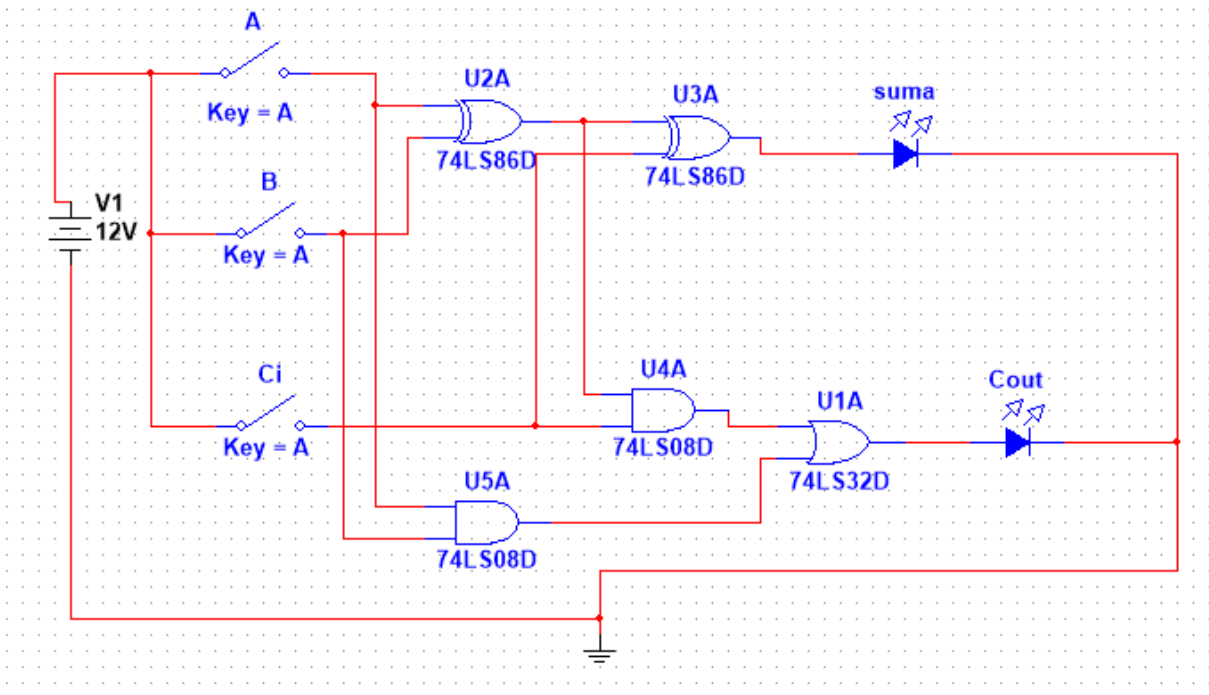
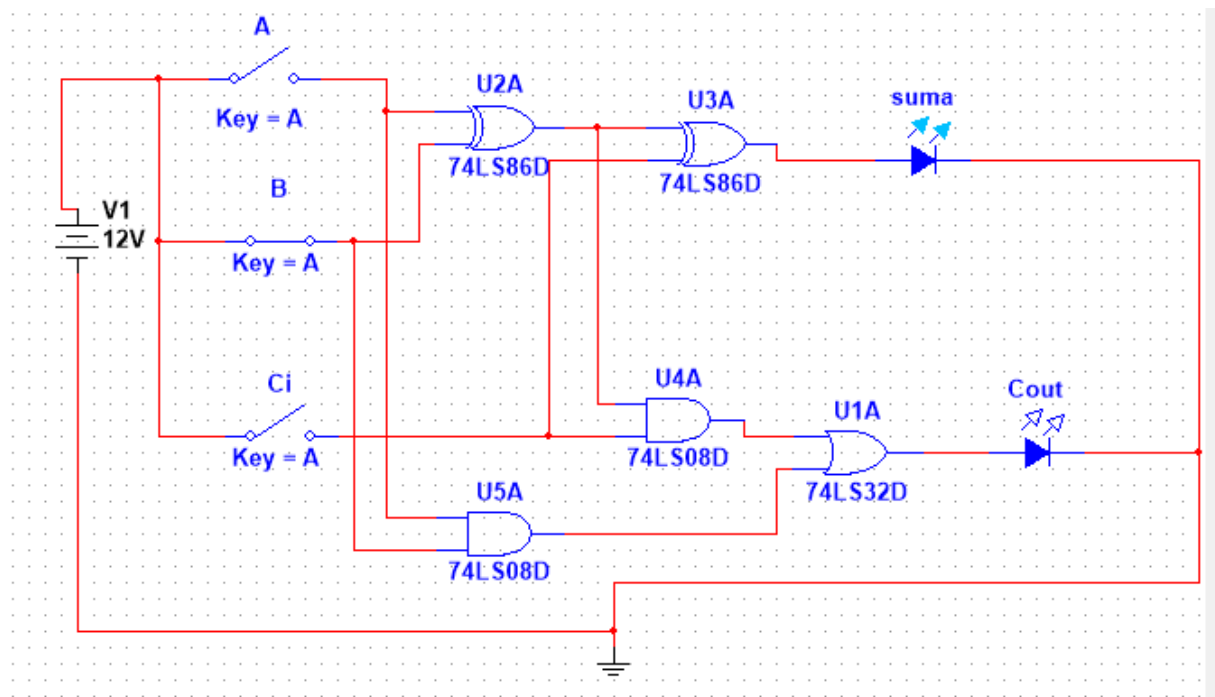
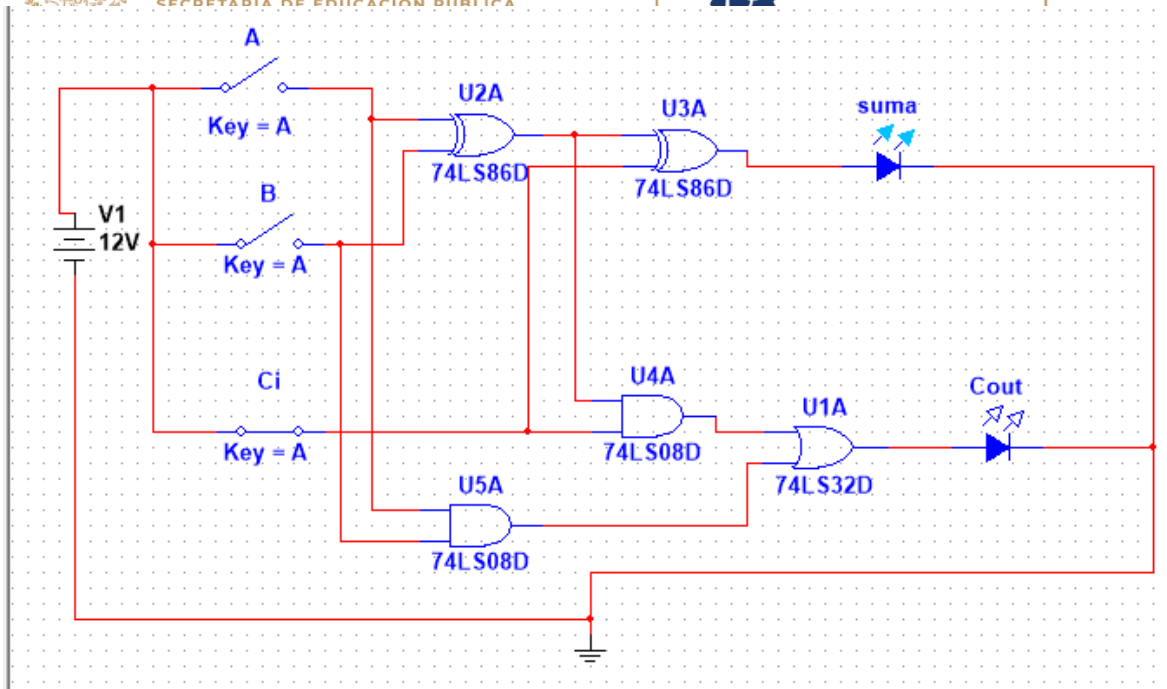
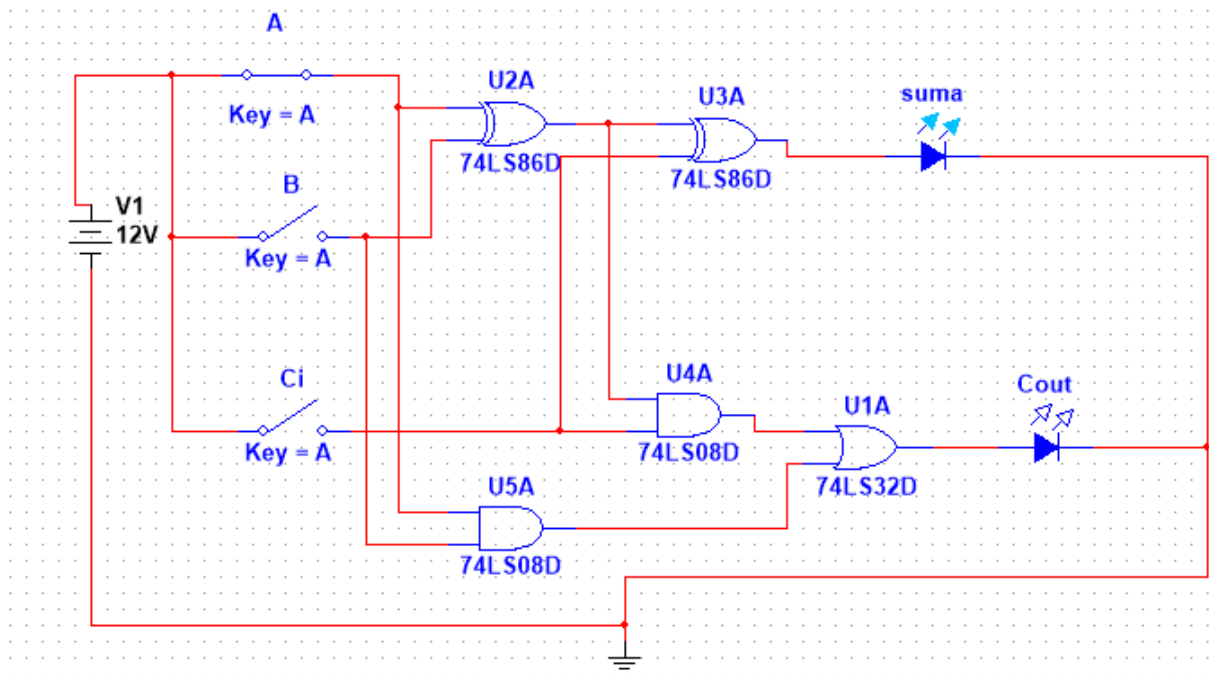
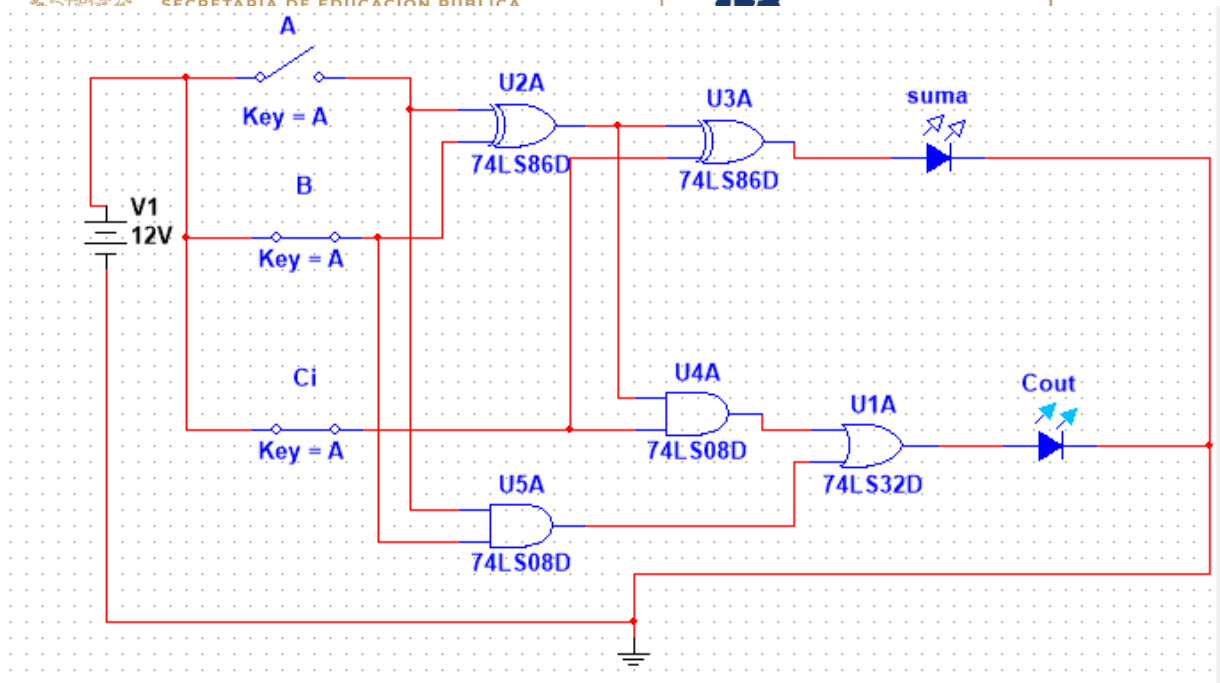
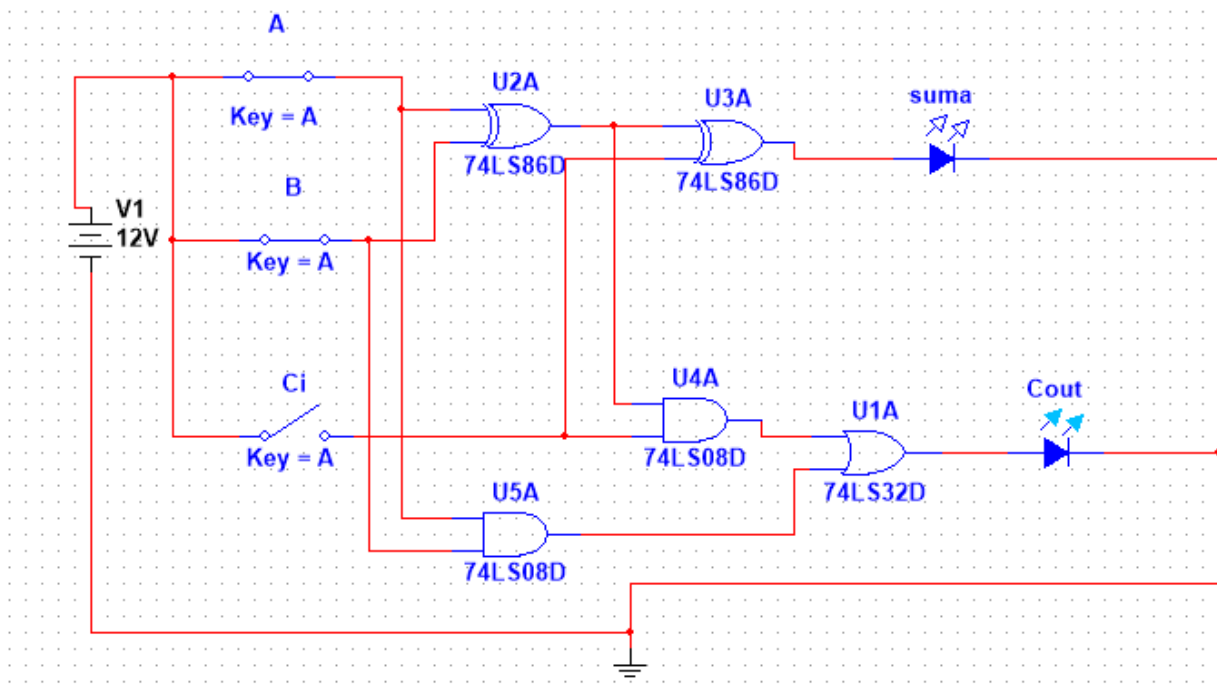
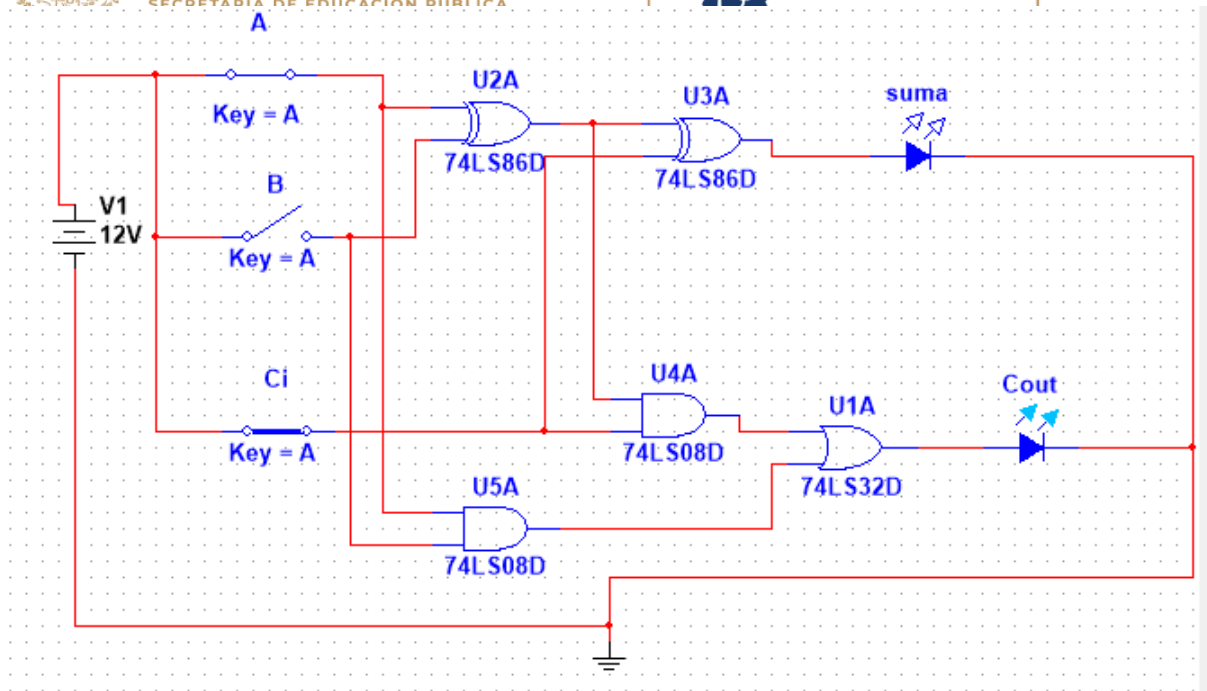
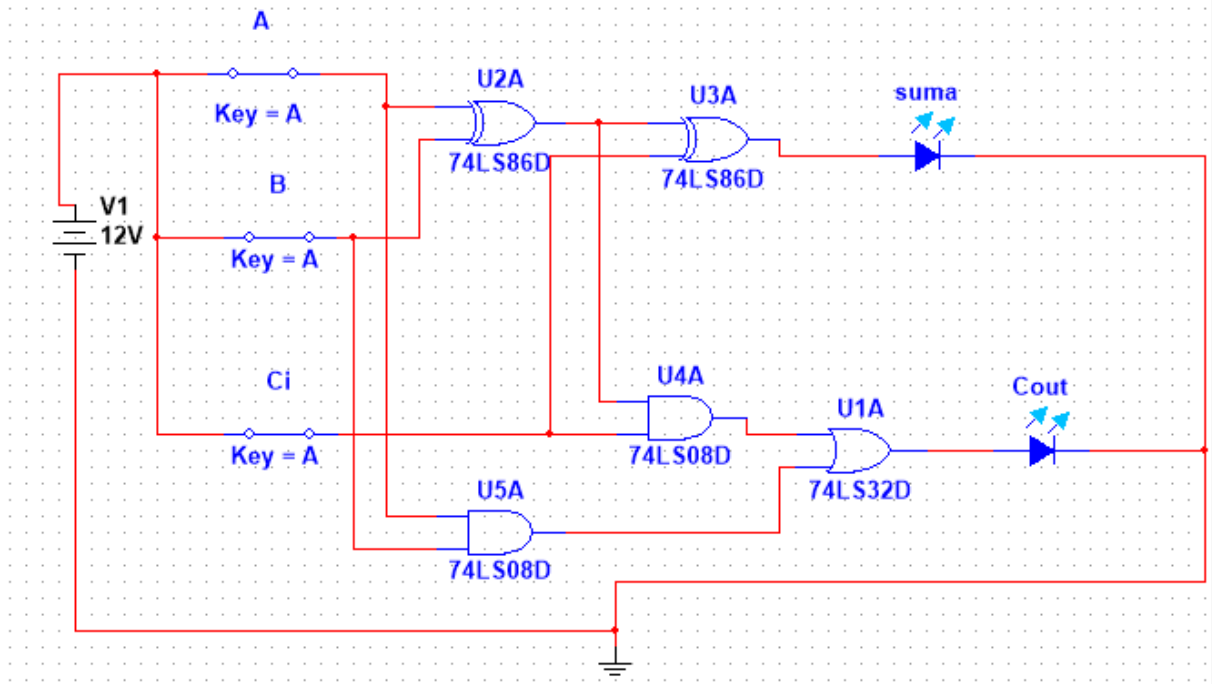


Ilustración 1 Simulación sumador









## 4.2 Circuito Restador

### 4.2.1 Implementación

#### Componentes necesarios:

- 2 compuertas XOR
- 3 compuertas AND
- 1 compuerta OR
- 1 inversor (NOT)

#### Pasos:

##### 1. Configurar entradas:

- Coloca tres entradas digitales para A, B y Cin.

##### 2. Calcular la diferencia:

- Conecta A y B a una compuerta XOR para obtener la primera diferencia parcial (D1).
- Conecta D1 y Cin a otra compuerta XOR para obtener la diferencia final S.

##### 3. Calcular el préstamo:

- Conecta A a un inversor (NOT) para obtener  $\sim A$ .



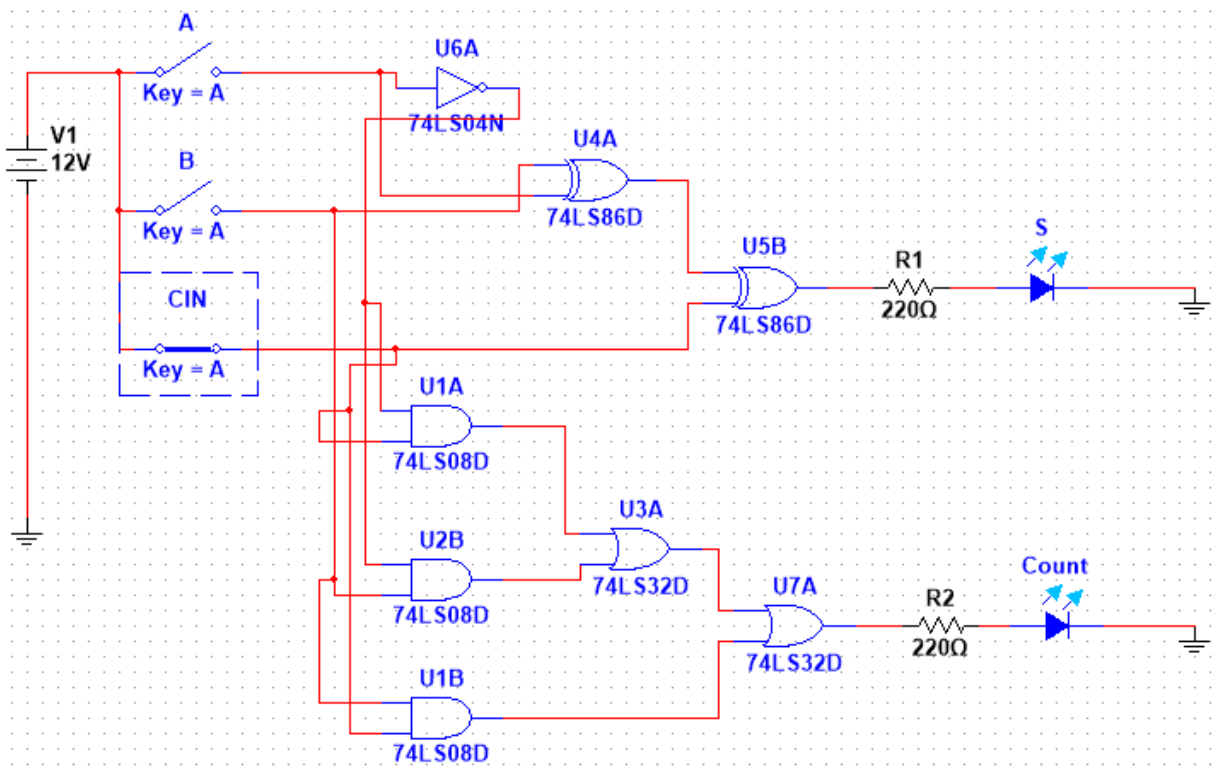
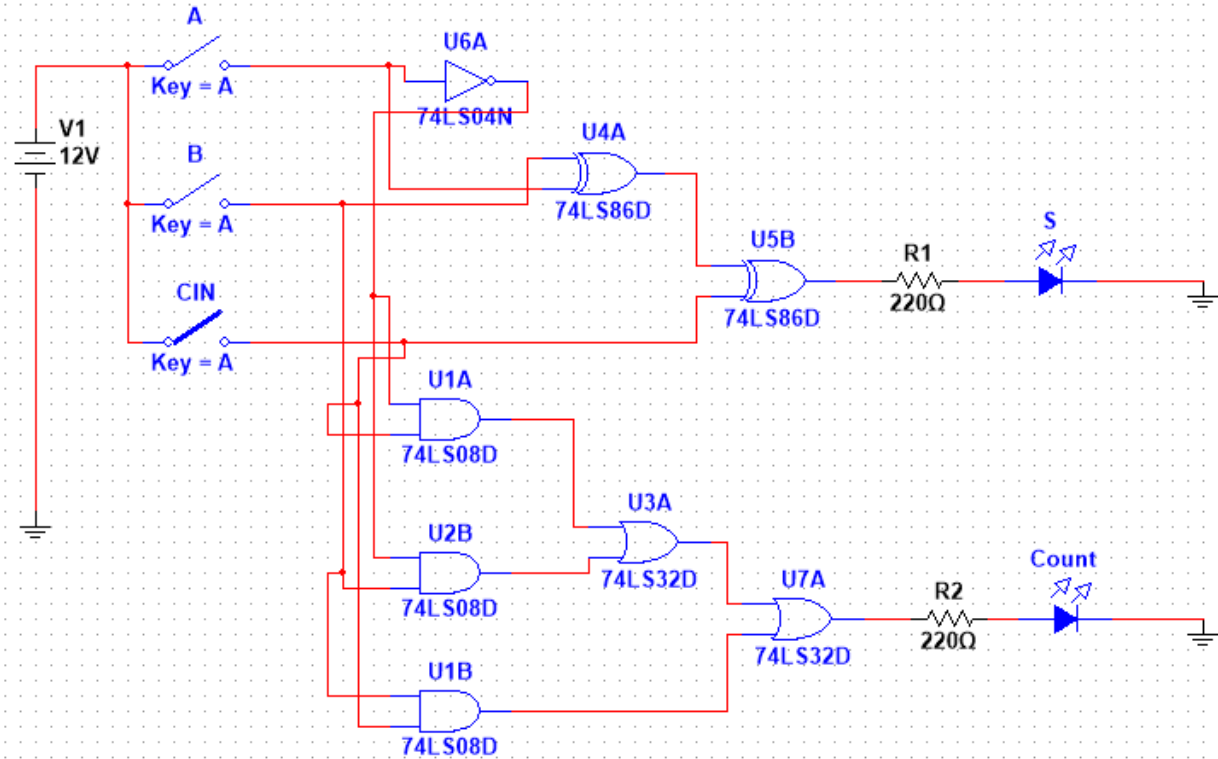
- Conecta  $\sim A$  y B a una compuerta AND para obtener el primer préstamo parcial (B1).
- Conecta D1 y Bin a otra compuerta AND para obtener el segundo préstamo parcial (B2).
- Conecta B1 y B2 a una compuerta OR para obtener el préstamo de salida Cout.

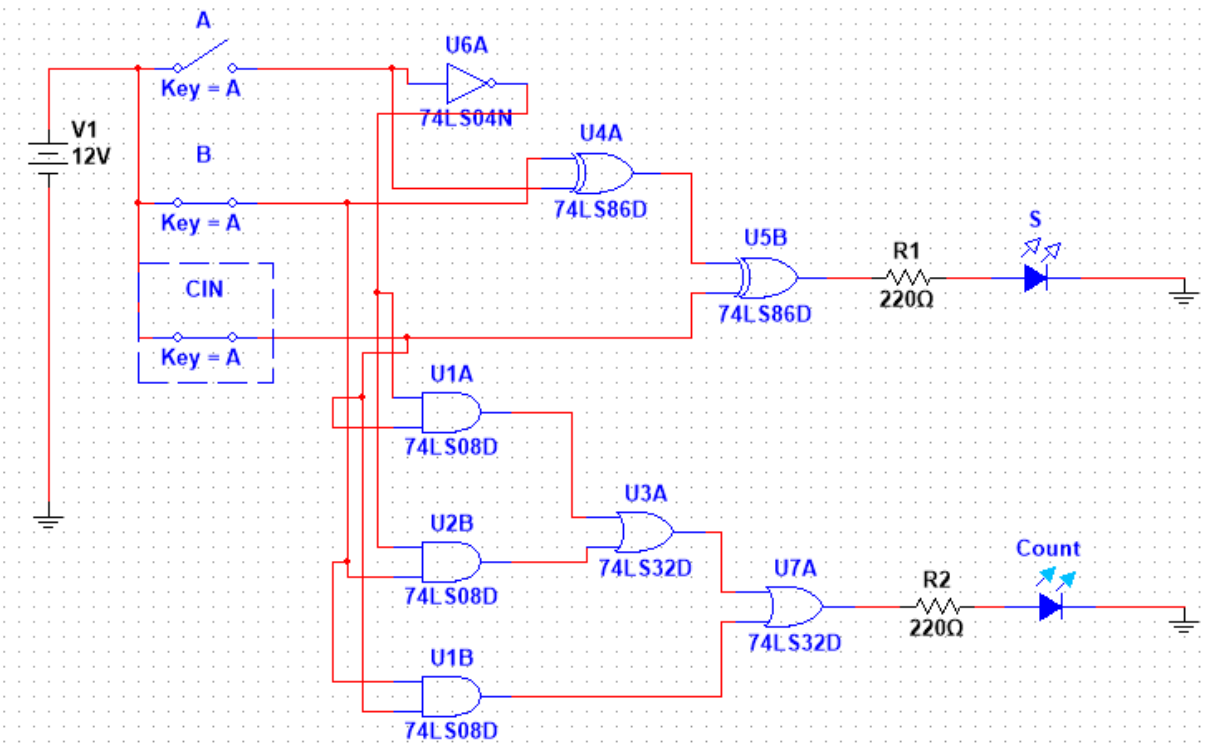
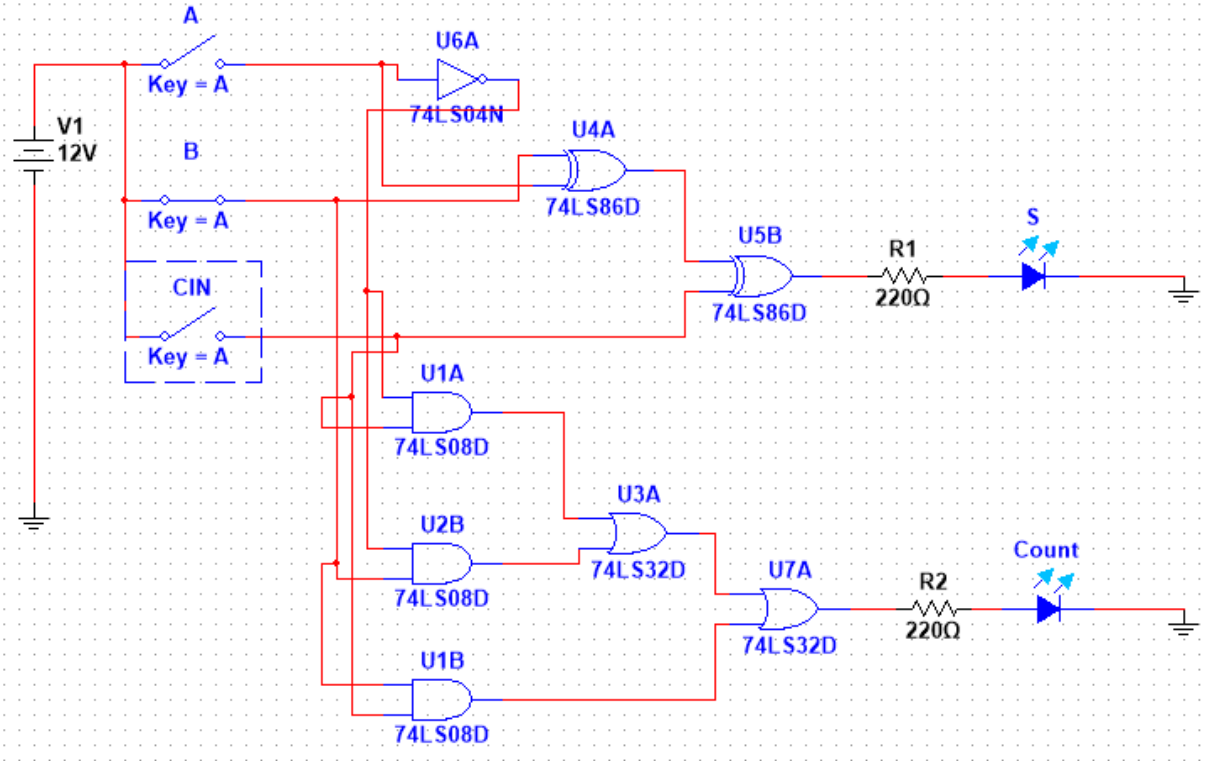
#### 4.2.2 Tabla de Verdad

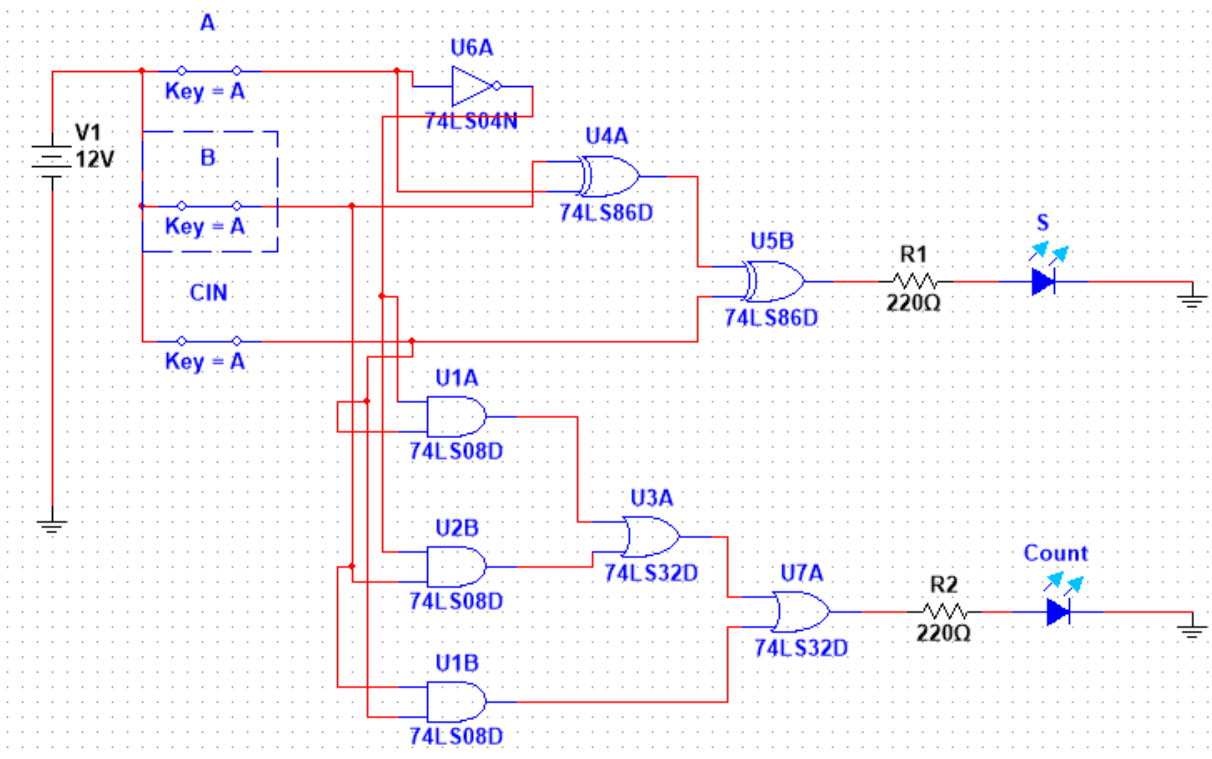
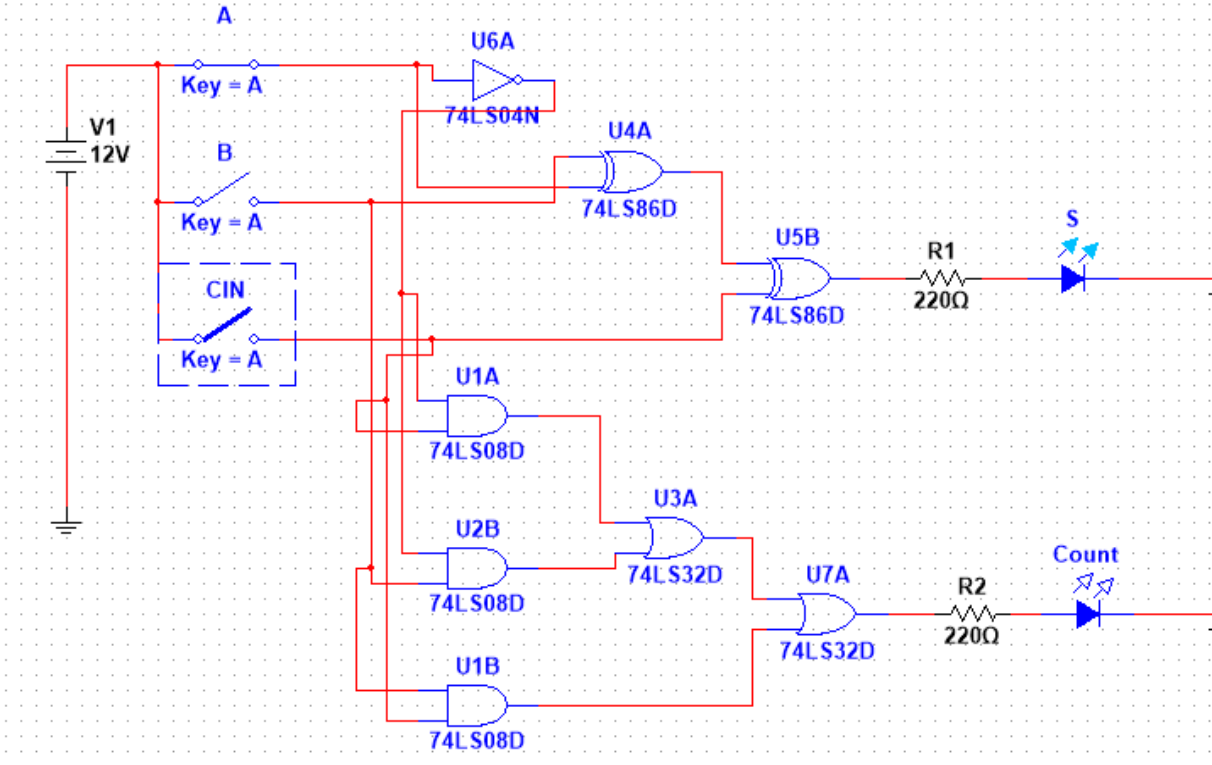
| A | B | Cin | S | Cout |
|---|---|-----|---|------|
| 0 | 0 | 0   | 0 | 0    |
| 0 | 0 | 1   | 1 | 1    |
| 0 | 1 | 0   | 1 | 1    |
| 0 | 1 | 1   | 0 | 1    |
| 1 | 0 | 0   | 1 | 0    |
| 1 | 0 | 1   | 0 | 0    |
| 1 | 1 | 0   | 0 | 0    |
| 1 | 1 | 1   | 1 | 1    |

#### 4.2.3 Simulación









### 4.3 Circuito Comparador

#### 4.3.1 implementación



### Componentes necesarios:

- 2 compuerta AND
- 1 compuerta NOR
- 2 inversor (NOT)

### Pasos:

#### 1. Configurar entradas:

- Coloca dos entradas digitales para A y B.

#### 2. Comparar bits:

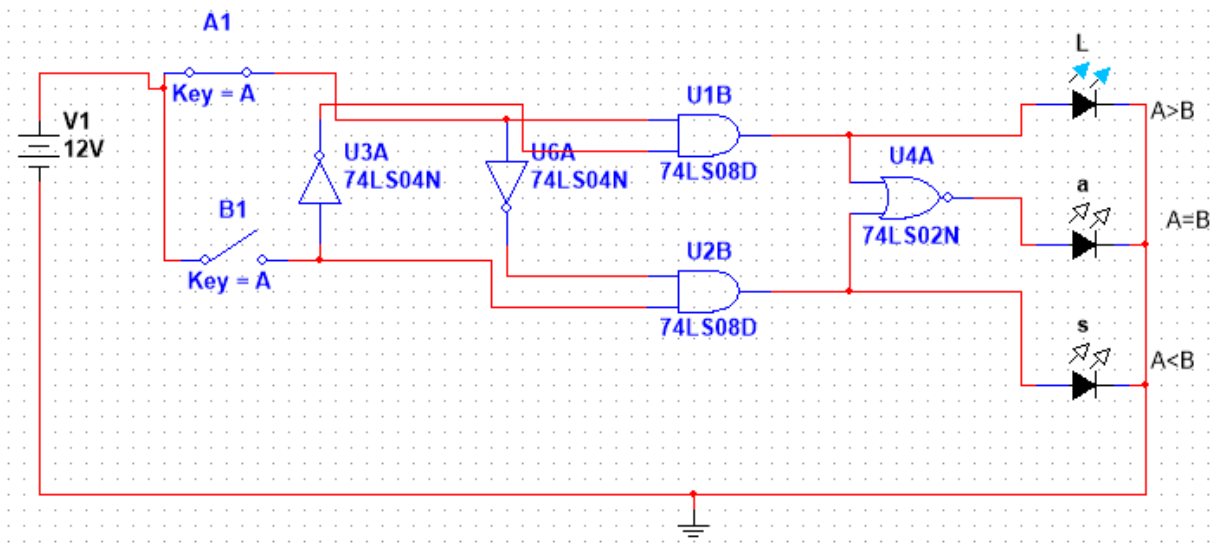
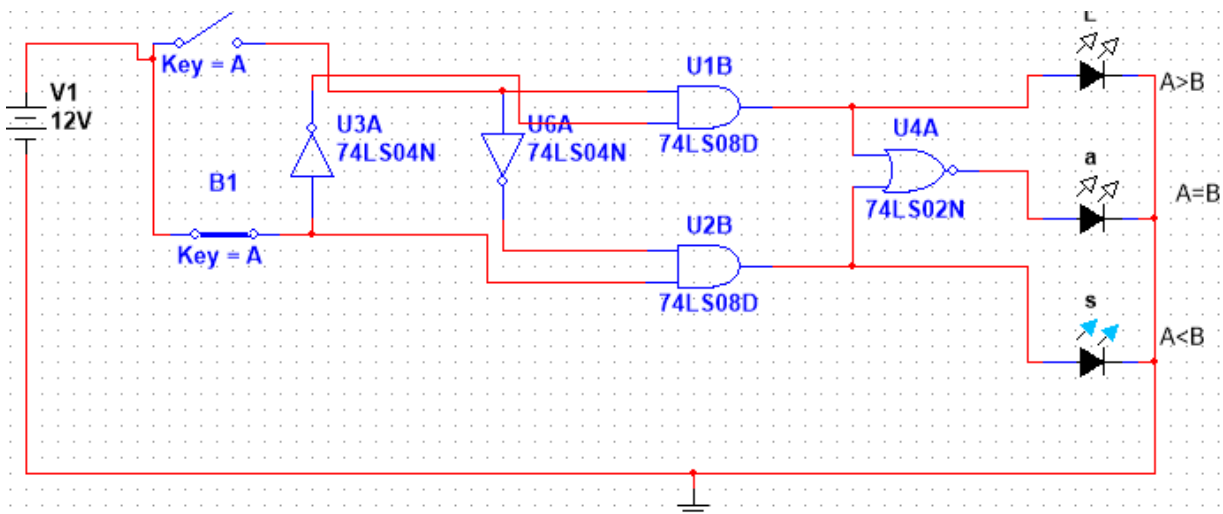
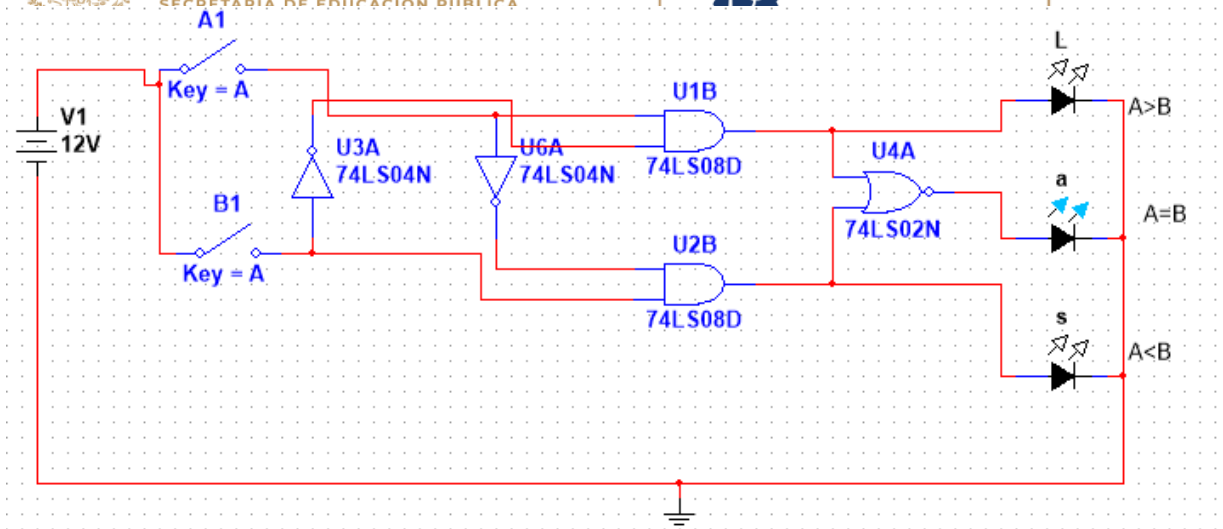
- Para  $A > B$ : Conecta A a una compuerta AND y B a un inversor (NOT). Conecta la salida del inversor y A a la compuerta AND para obtener  $A > B$ .
- Para  $A = B$ : Conecta A y B a una compuerta XOR y pasa la salida a un inversor (NOT) para obtener  $A = B$ .
- Para  $A < B$ : Conecta B a una compuerta AND y A a un inversor (NOT). Conecta la salida del inversor y B a la compuerta AND para obtener  $A < B$ .

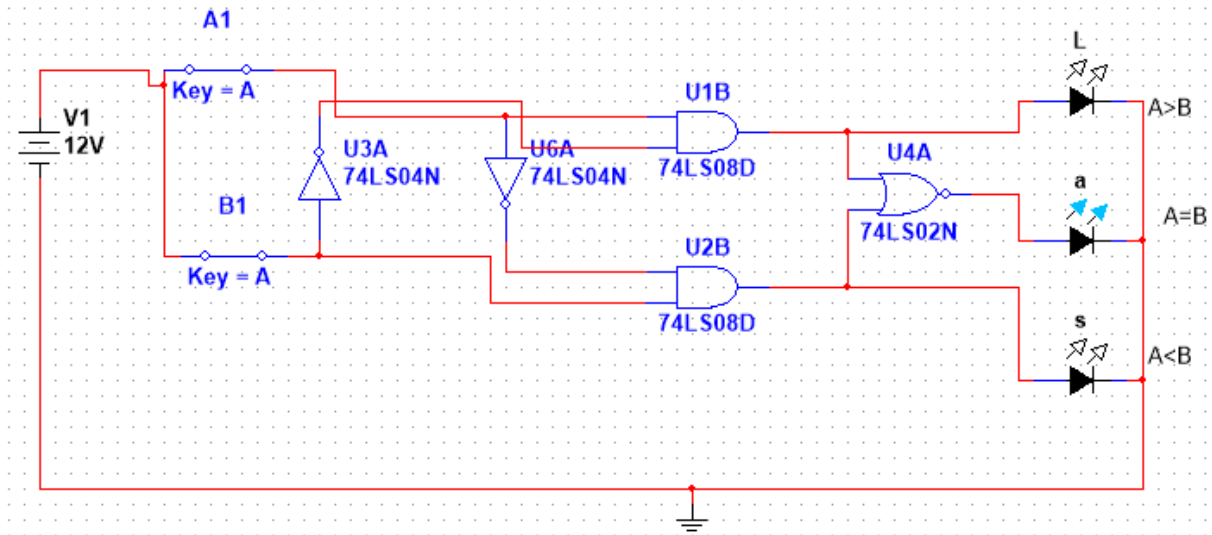
#### 3. Configurar salidas: Coloca leds para mostrar los valores de $A > B$ , $A = B$ , y $A < B$ .

### 4.3.2 Tabla de verdad

| A | B | $A > B$ | $A = B$ | $A < B$ |
|---|---|---------|---------|---------|
| 0 | 0 | 0       | 1       | 0       |
| 0 | 1 | 0       | 0       | 1       |
| 1 | 0 | 1       | 0       | 0       |
| 1 | 1 | 0       | 1       | 0       |

### 4.3.3 Simulación





## 4.4 Circuito Multiplicador

### 4.4.1 Implementación

#### Componentes necesarios:

- 1 compuerta AND

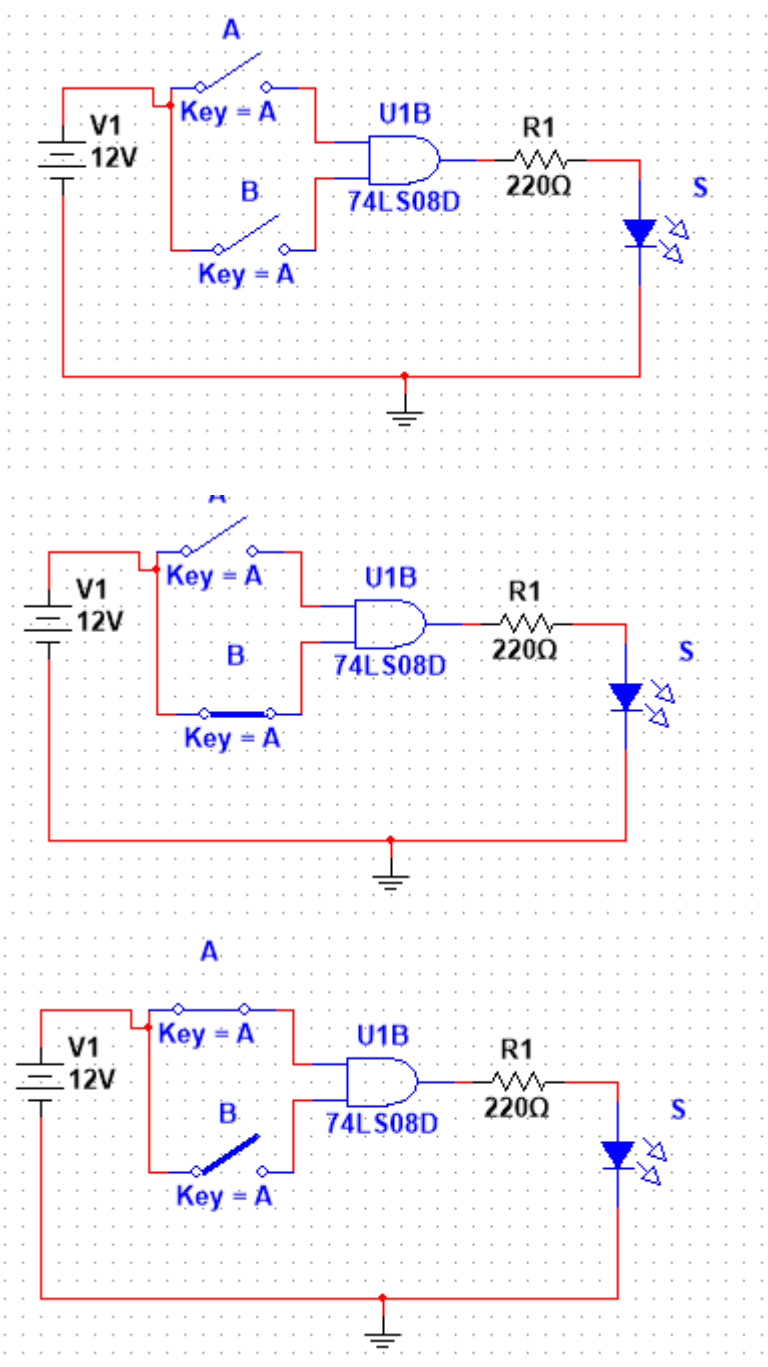
#### Pasos:

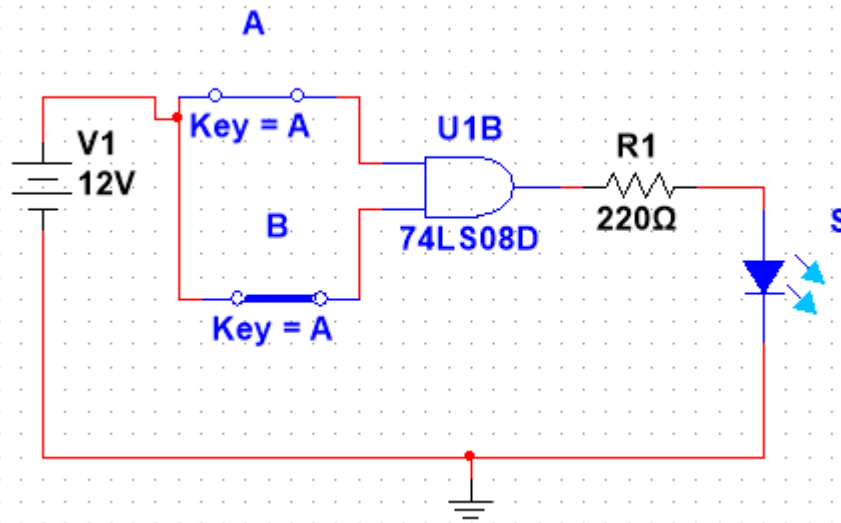
1. Configurar entradas:
  - Coloca dos entradas digitales para A y B.
2. Conecta A y B a una compuerta AND.
3. Configurar salida:
  - Coloca un led para mostrar el valor del producto S.

### 4.4.2 Tabla de verdad

| A | B | S |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

### 4.4.3 Simulación





## Conclusiones

### Roldan Uriel Arcadio Avila

La creación de circuitos de un solo bit en Multisim demuestra que es posible implementar operaciones lógicas fundamentales utilizando una combinación de compuertas básicas. A través de esta práctica, se observa la importancia de comprender la interconexión de las compuertas lógicas para lograr un diseño funcional y eficiente en el desarrollo de sistemas digitales.

### Ariadna Monserrat López Aparicio

La implementación de circuitos de un solo bit en Multisim permite realizar operaciones lógicas básicas al combinar compuertas fundamentales. Esta práctica subraya la relevancia de dominar la conexión adecuada entre compuertas para asegurar que el diseño sea tanto funcional como eficiente en su aplicación dentro de sistemas digitales.

## Referencias

<https://www.ni.com/es/support/downloads/software-products/download.multisim.html#452133>