|  |  |  |  |
| --- | --- | --- | --- |
|  | **PRACTICA N°** 3  **TITULO:** Operadores Lógicos con Circuitos TTL | | **SIS-420**  **LABORATORIO**  **SISTEMAS DIGITALES** |
| **GRUPO:** 2  **DIA:** martes  **HORA:** 14:00 | **NOMBRE Y APELLIDO**  **UNIV:** Gary Brayam Villca Machaca  **CARRERA:** Ingeniería De Sistemas | | **CALIFICACIÓN**  **Resumen practica**……...../10%  **Obtención de datos**…….../15%  **Procesamiento datos**……/35%  **Cuestionario**……………./10%  **Conclusiones**……………/30%  **TOTA**………/100% |
| **SUB GRUPO**  1 |
| **FECHA DE REALIZACIÓN:06**/05/2025 | | **FECHA DE ENTREGA: 13**/05/2025 | |
| **DOCENTE:** | | **AUXILIAR:** | |

1. Resumen

**Teoría**

La práctica aborda el funcionamiento de compuertas lógicas implementadas mediante circuitos integrados TTL (Transistor-Transistor Logic). Se estudian seis tipos fundamentales:

* **NAND (7400):** La salida es 0 solo cuando todas las entradas son 1.  
  *Expresión:* F = (A·B)
* **NOR (7402):** La salida es 1 solo cuando todas las entradas son 0.  
  *Expresión:* F = (A + B)
* **NOT (7404):** Inversor lógico. Cambia 0 a 1 y 1 a 0.  
  *Expresión:* F = A
* **AND (7408):** La salida es 1 solo si todas las entradas son 1.  
  *Expresión:* F = A·B
* **OR (7432):** La salida es 1 si alguna entrada es 1.  
  *Expresión:* F = A + B
* **XOR (7486):** La salida es 1 si las entradas son diferentes.  
  *Expresión:* F = A ⊕ B

Estas compuertas permiten construir circuitos lógicos que representan expresiones booleanas, utilizadas en el diseño de sistemas digitales.

**Metodología**

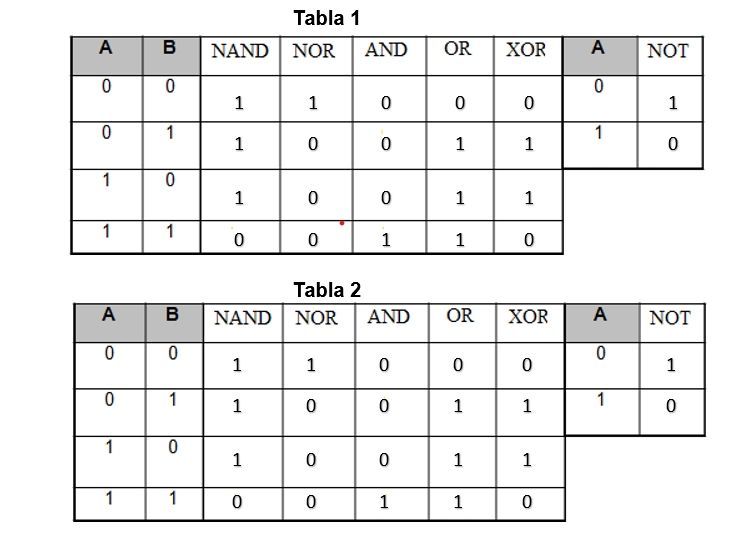
Se emplean tres entornos para el montaje y prueba de los circuitos lógicos:

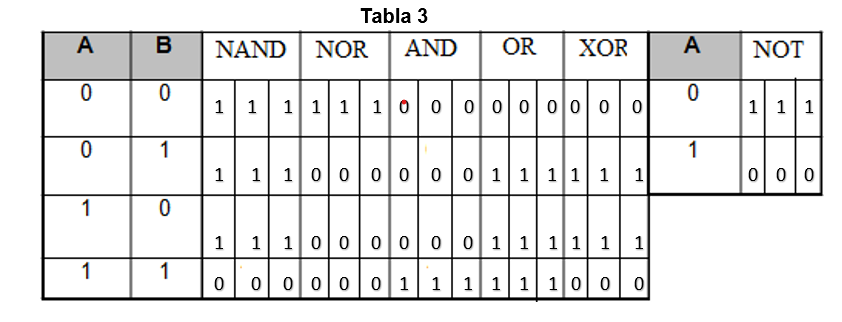
* **Placa base del equipo Simulog:** Se conectan físicamente los circuitos integrados (7400, 7402, 7404, 7408, 7432 y 7486) en una protoboard o placa simuladora, utilizando una fuente de 5V DC, cables de conexión y semáforos como indicadores de salida.
* **Simulador MULTISIM, PROTEUS o Constructor Virtual de Circuitos:** Se replican los montajes de forma virtual para validar el funcionamiento lógico de cada compuerta, completando la tabla de verdad para cada caso.
* **Tablas de verdad:** Se llenan manualmente a partir de la observación de los resultados en las salidas al variar las combinaciones de entradas.

**Conclusión**

Nos permitió comprender el funcionamiento lógico de los principales operadores digitales mediante circuitos TTL, aplicar correctamente los conceptos teóricos al interconectar físicamente y simular los circuitos, verificar experimentalmente las tablas de verdad de cada compuerta lógica y consolidar el conocimiento sobre cómo se implementan operaciones booleanas en hardware real y simuladores digitales.

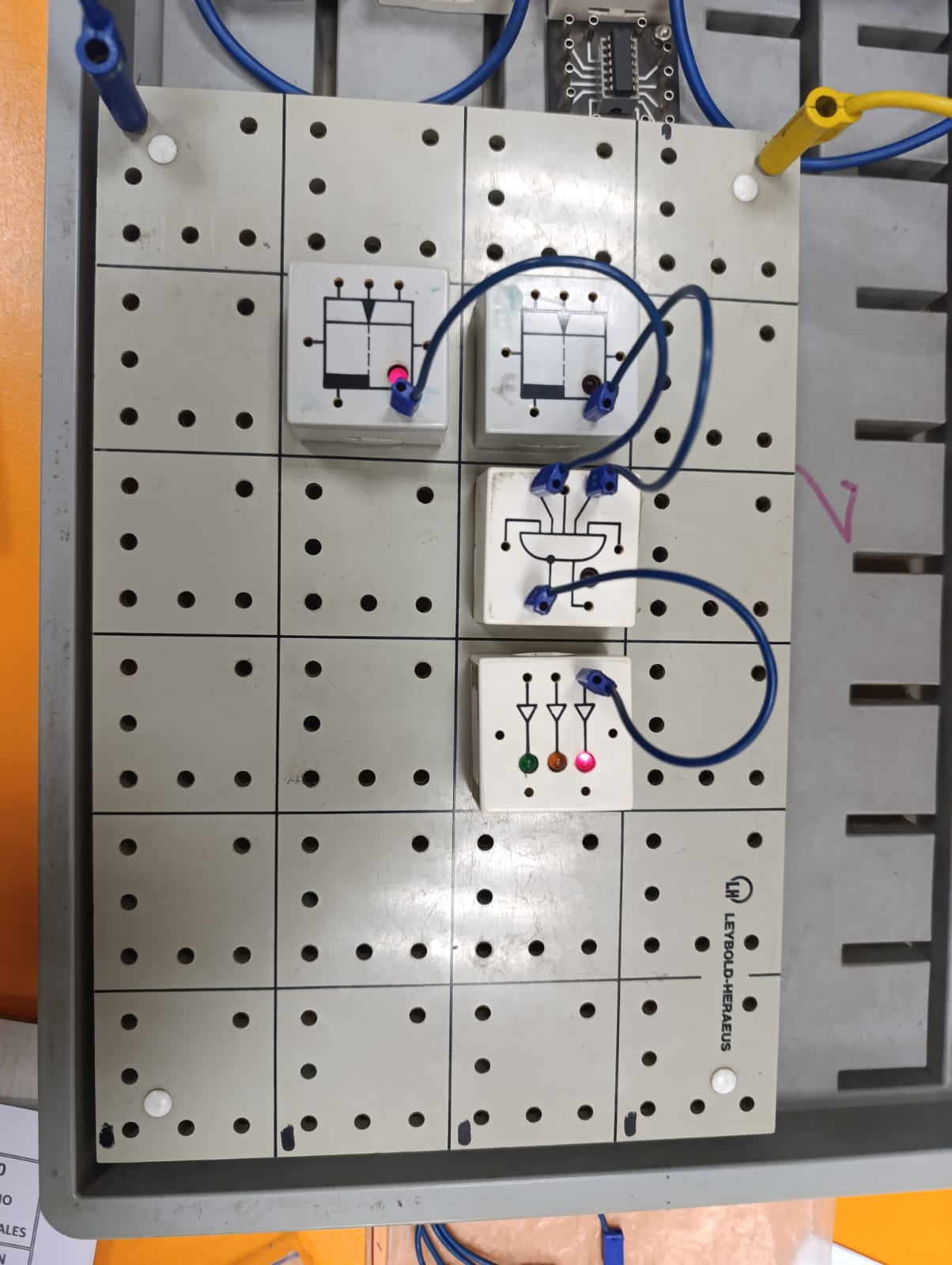
1. Obtención de datos

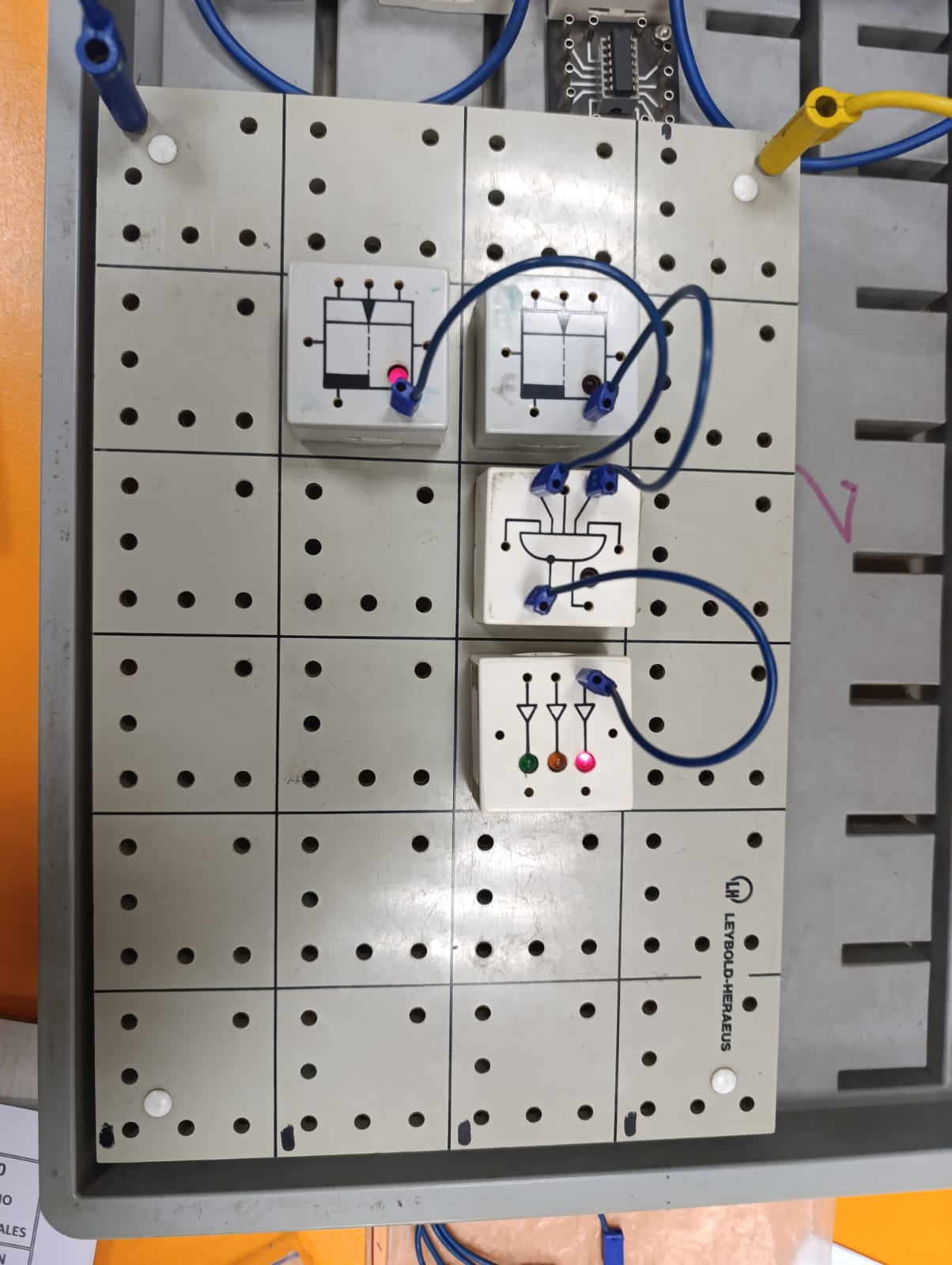


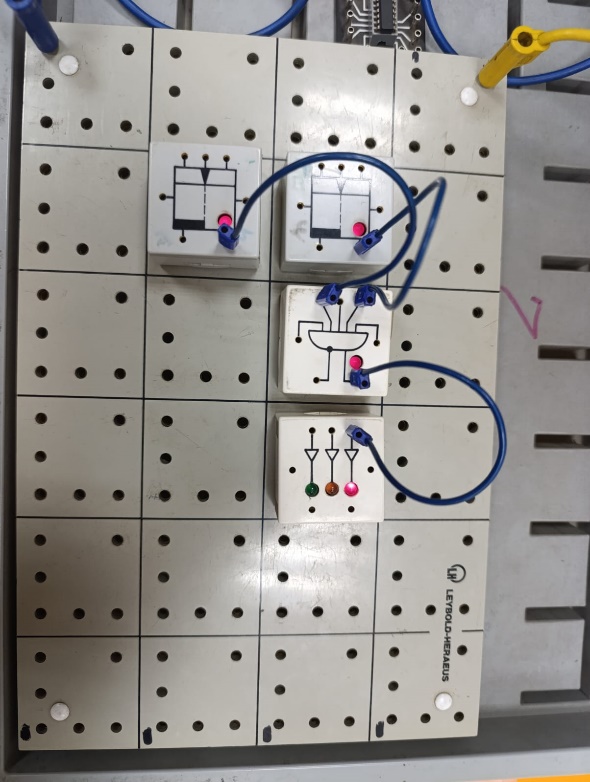
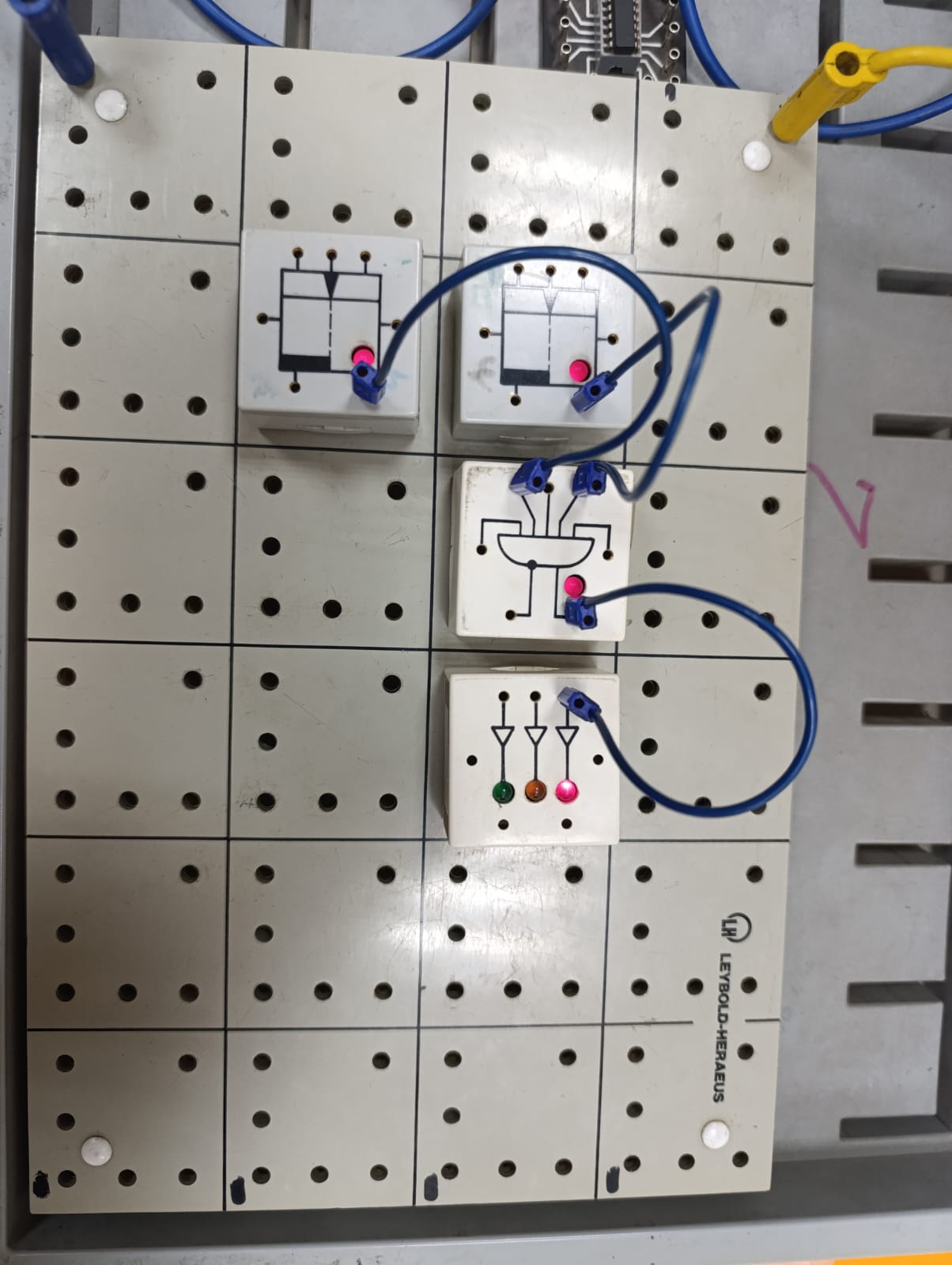


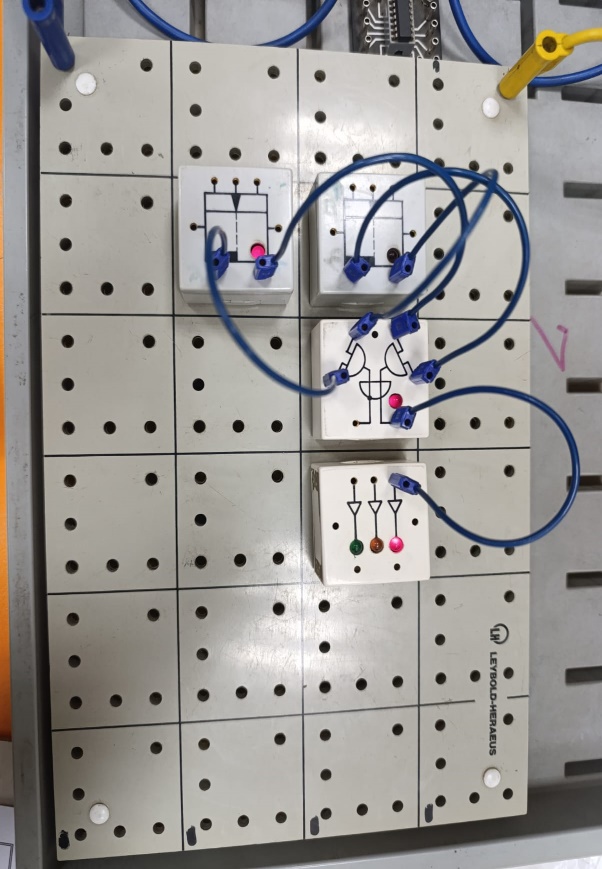
1. Procedimiento de datos

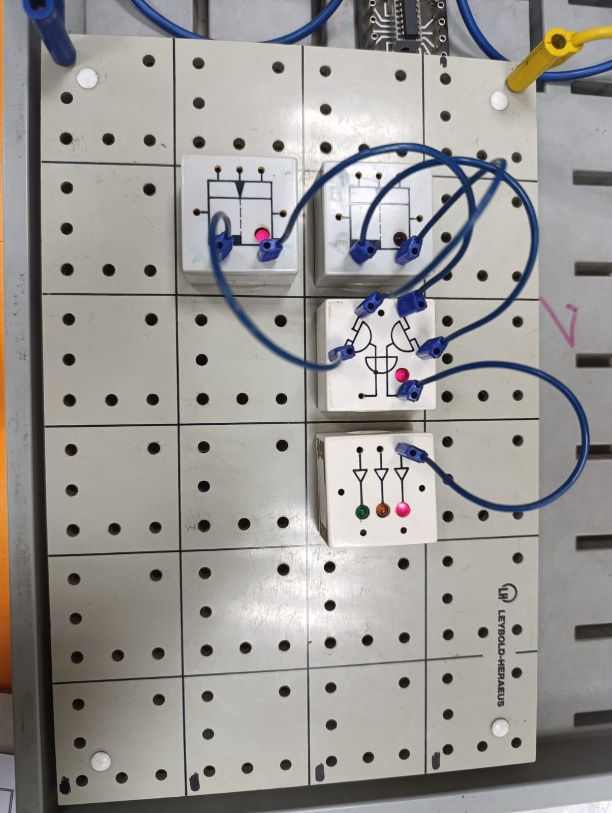
**TABLA 1:**

**NAND NOR**

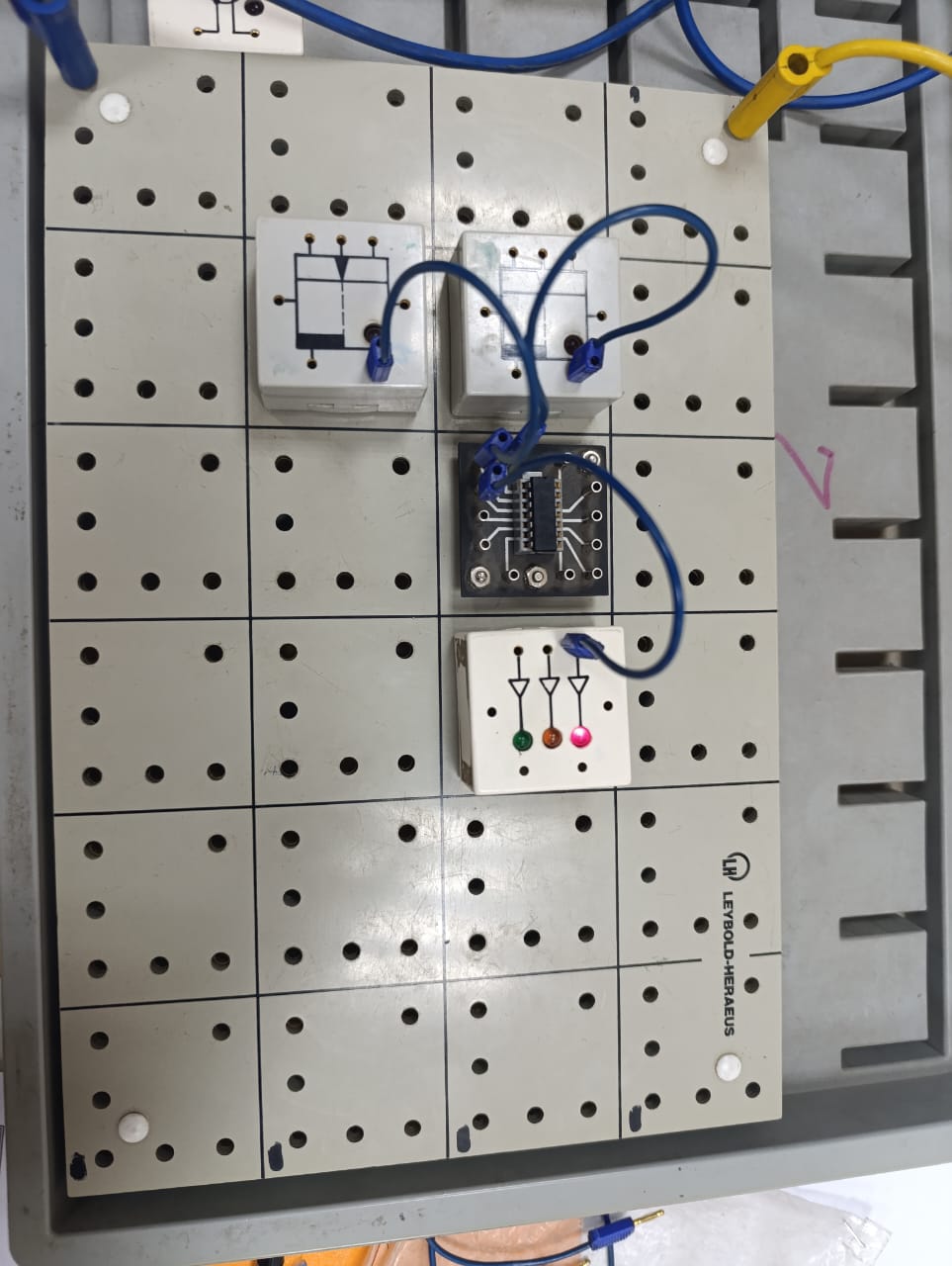
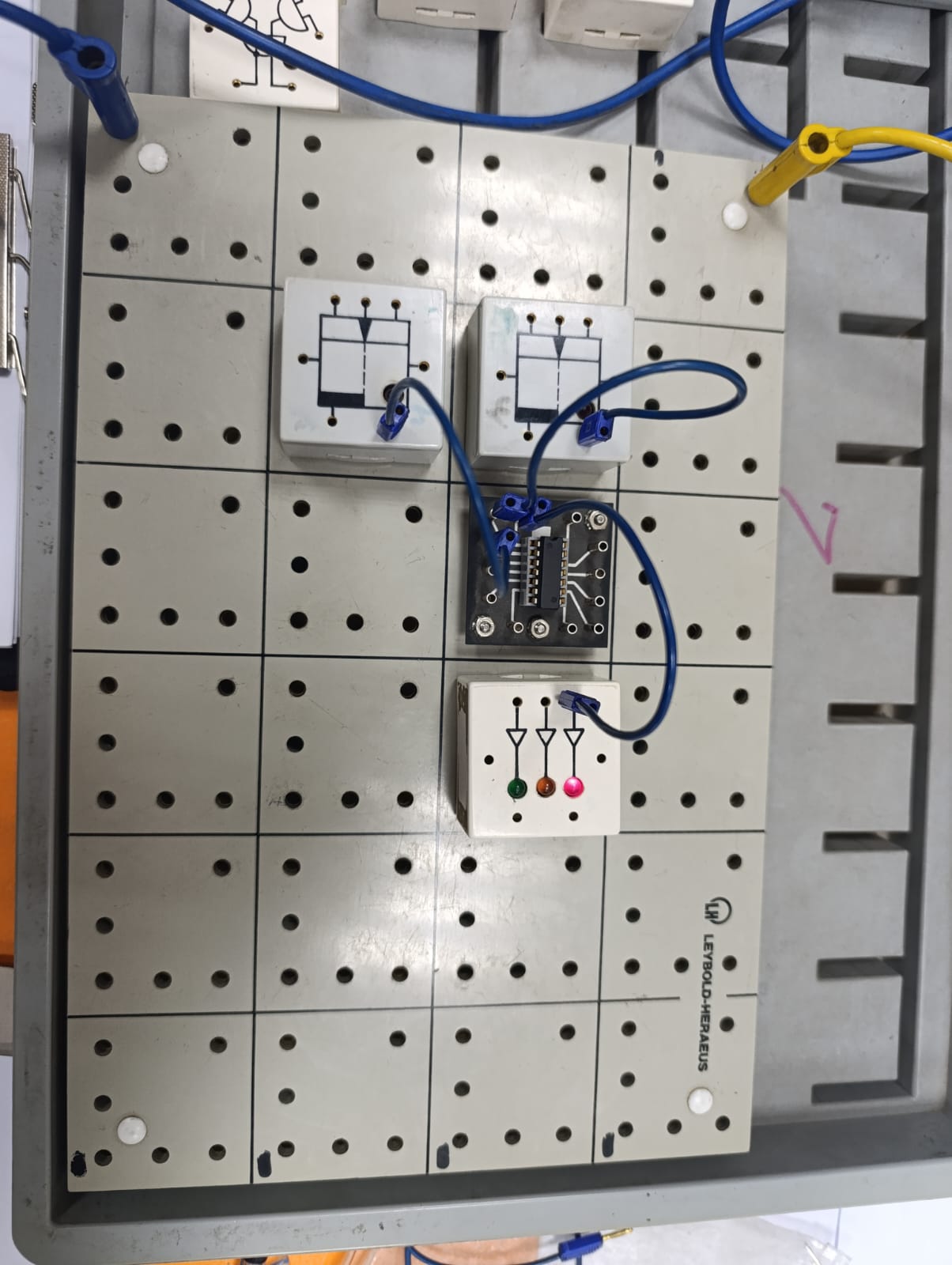


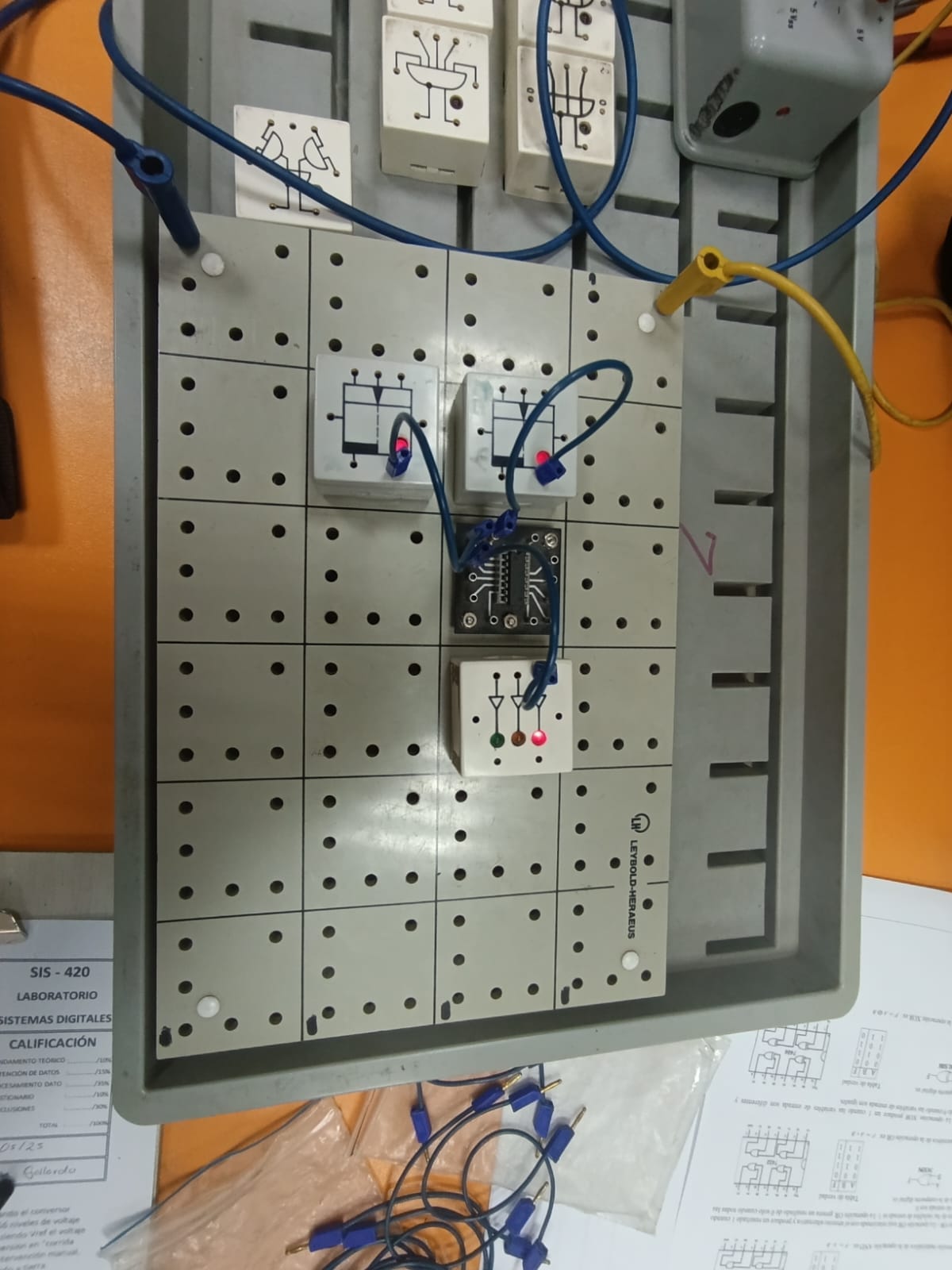
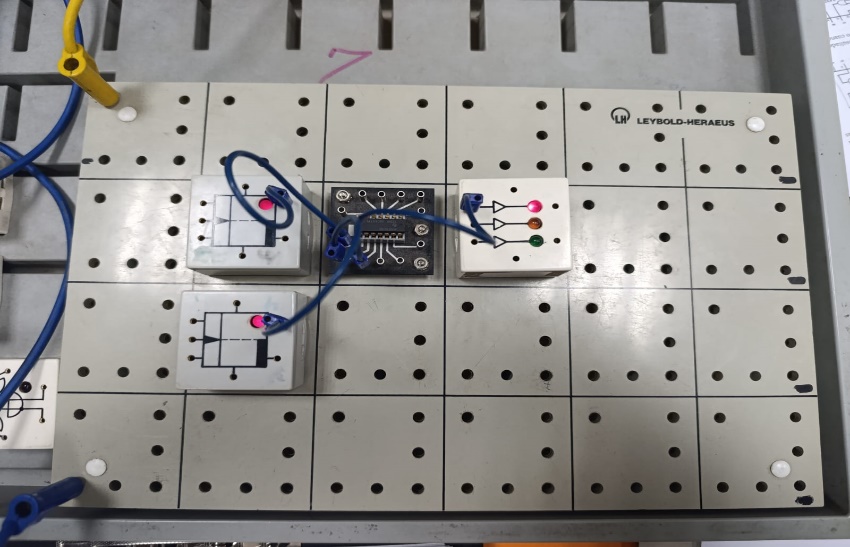
 **AND** **OR**

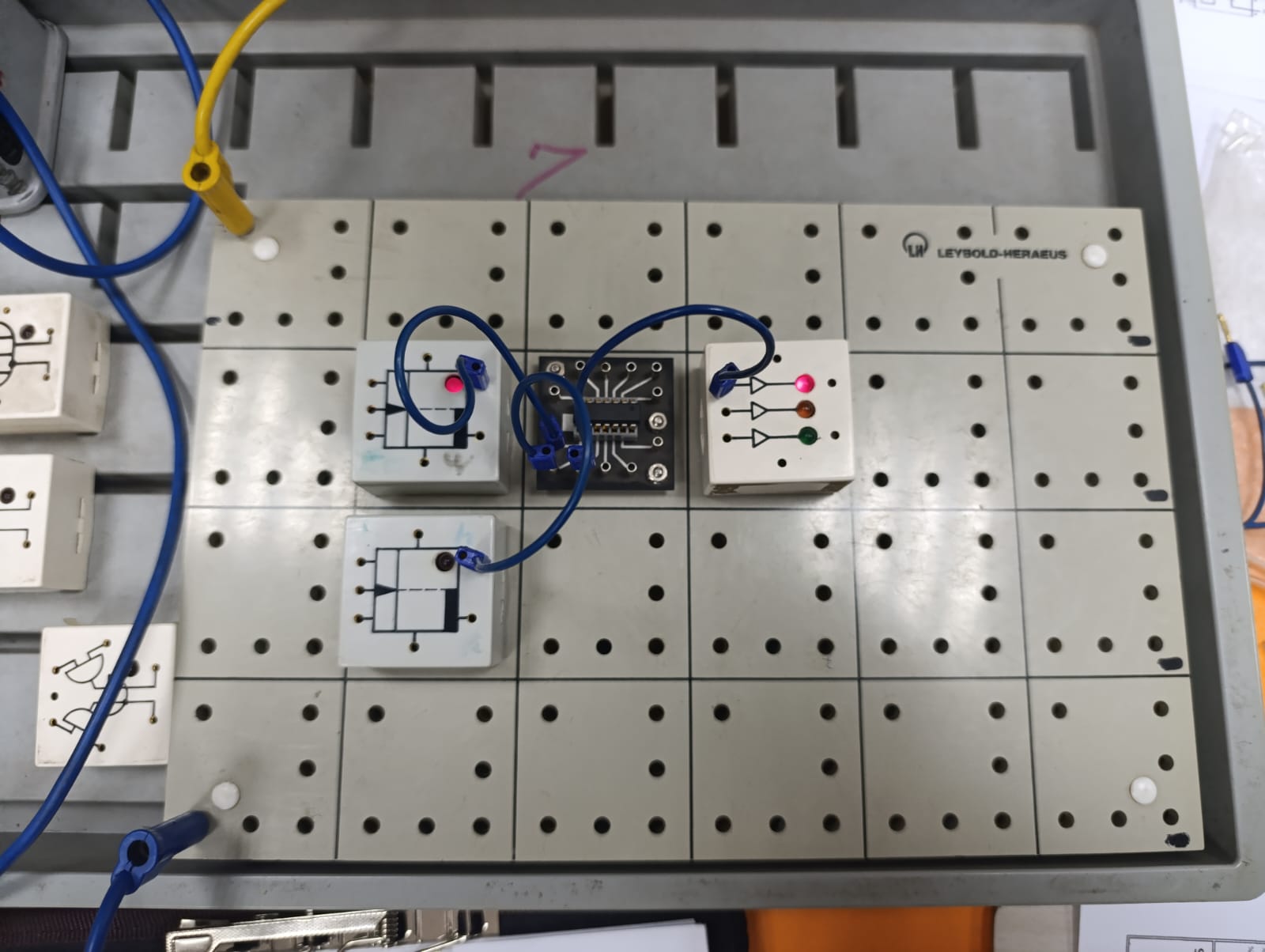
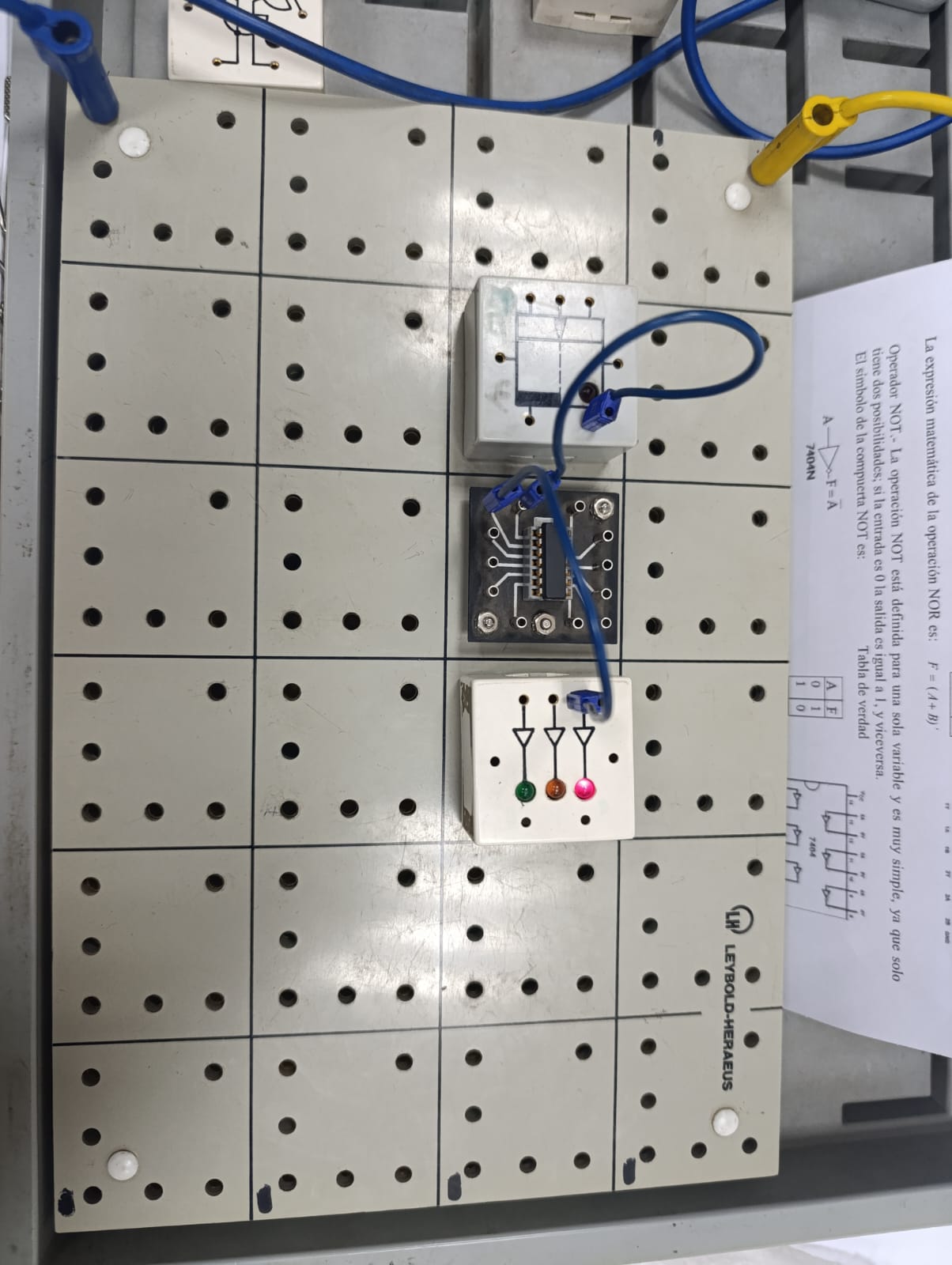
**X-OR NOT**



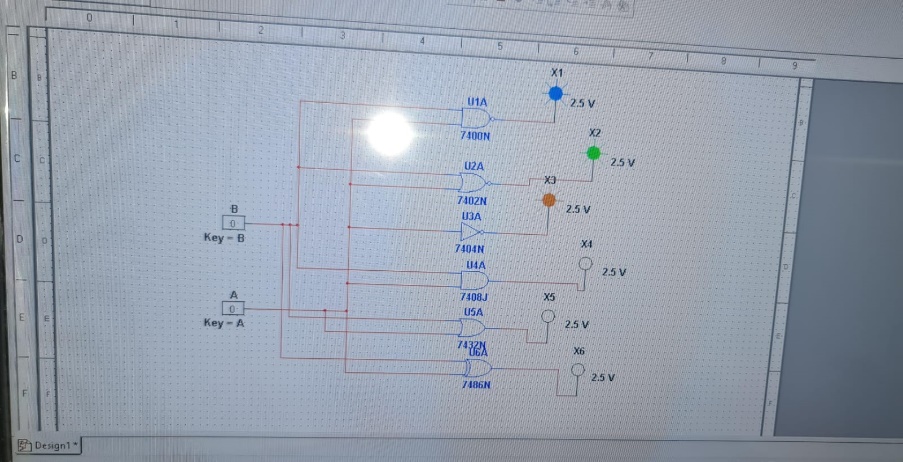
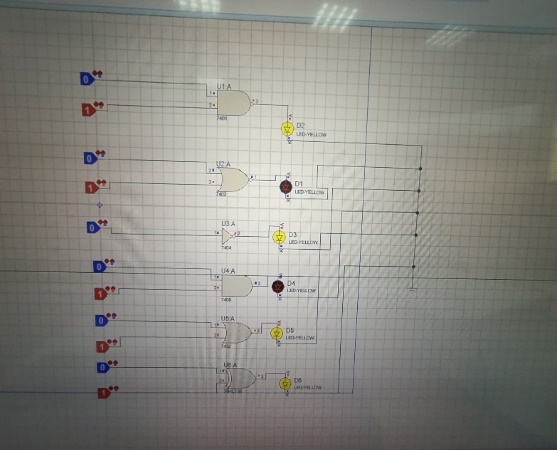
**TABLA 2:**

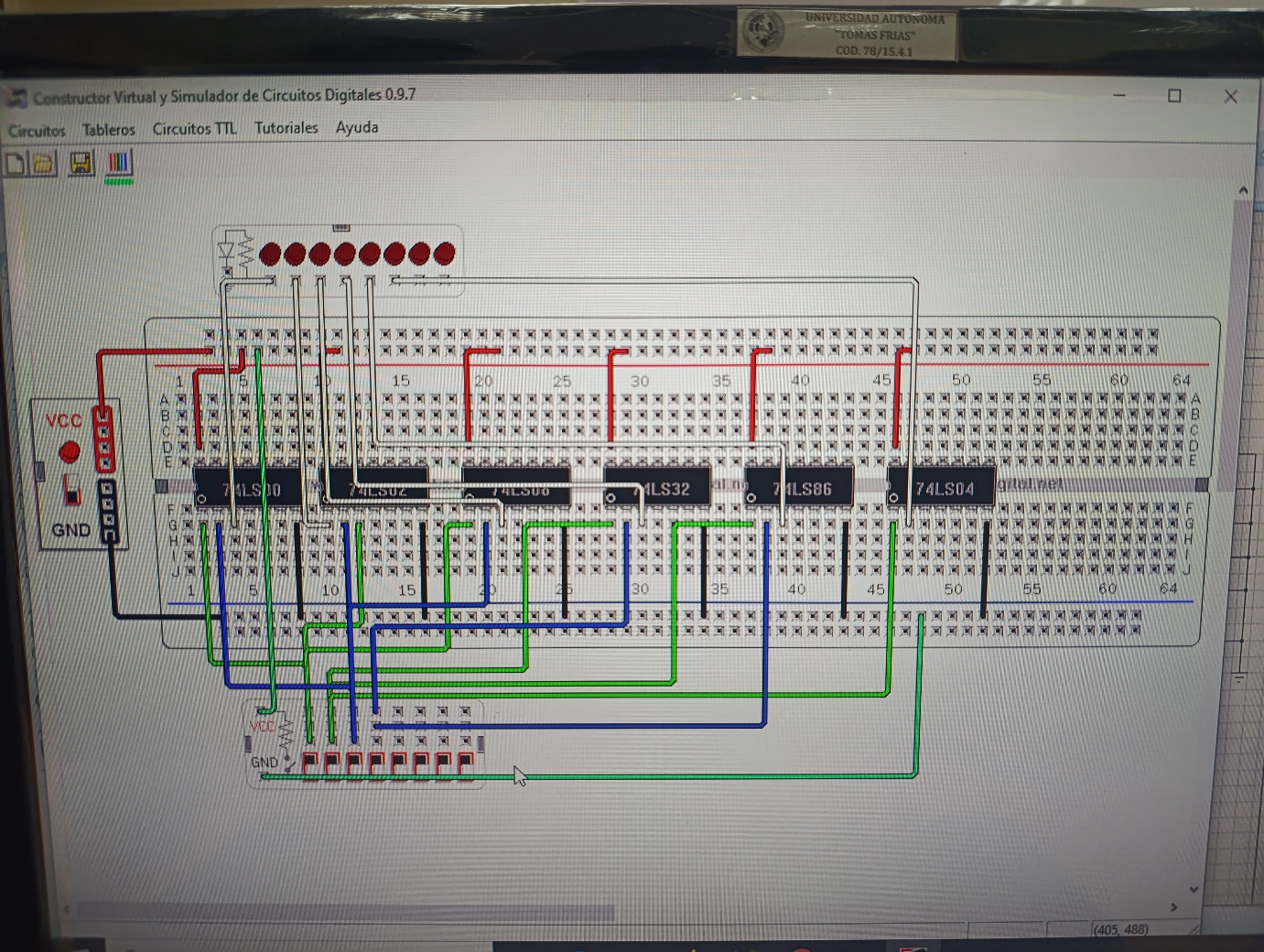
 **NAND NOR**

 **AND OR**

 **X-OR NOT**

**TABLA 3:**

 **MULTISIM PROTEUS**

CONSTRUCTOR VIRTUAL DE CIRCUITOS

1. Cuestionario

**¿Cuál es la forma de uso de un protoboard y de una placa base del equipo Simulog?**

El **protoboard** es una herramienta que permite el montaje rápido de circuitos electrónicos sin necesidad de soldaduras. Se compone de una matriz de orificios conectados internamente que permiten insertar componentes como resistencias, cables y circuitos integrados (C.I.). Para utilizarlo correctamente, se deben identificar las filas comunes para alimentación (Vcc y GND) y las columnas conectadas internamente donde se insertan los pines de los C.I.

La **placa base del equipo Simulog**, por otro lado, es un dispositivo educativo diseñado para trabajar con módulos lógicos ya incorporados (como compuertas) y facilitar el armado de circuitos digitales. Estas placas incluyen fuentes de alimentación, indicadores visuales como LEDs o semáforos, y conectores que permiten realizar las conexiones mediante cables. Es ideal para ensayos prácticos sin necesidad de utilizar componentes discretos.

**Demostrar de la tabla de verdad que 𝐹 = 𝐴⨁𝐵 = 𝐴̅𝐵 + 𝐴𝐵̅**

La compuerta **XOR (⊕)** tiene una salida de 1 únicamente cuando las entradas son distintas. La tabla de verdad es la siguiente:

| **A** | **B** | **F = A⊕B** |
| --- | --- | --- |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

**Demostrar de la tabla de verdad para un operador AND, utilizando maxtérminos, que**

**𝐹 = 𝐴 ∙ 𝐵 = (𝐴 + 𝐵)(𝐴 + 𝐵̅)(𝐴̅ + 𝐵)**

La operación AND (A·B) da como resultado 1 solo cuando A=1 y B=1. En términos de maxtérminos, se representan las combinaciones donde la salida es 0. La tabla de verdad es:

| **A** | **B** | **F = A·B** |
| --- | --- | --- |
| 0 | 0 | 0 → (A + B) |
| 0 | 1 | 0 → (A + B̅) |
| 1 | 0 | 0 → (A̅ + B) |
| 1 | 1 | 1 → No se incluye |

Multiplicando estos maxtérminos:

F = (A + B)(A + B̅)(A̅ + B)

Este producto es equivalente a la función lógica AND, ya que anula todas las combinaciones excepto cuando A y B son ambos 1.

**Demostrar de la tabla de verdad para un operador OR, utilizando mintérminos, que**

**𝐹 = 𝐴 + 𝐵 = 𝐴̅𝐵 + 𝐴𝐵̅ + 𝐴𝐵**

La operación OR (A + B) tiene una salida de 1 en todos los casos excepto cuando A=0 y B=0. Los mintérminos representan las combinaciones donde la salida es 1. Veamos la tabla:

| **A** | **B** | **F = A + B** |
| --- | --- | --- |
| 0 | 0 | 0 → No se incluye |
| 0 | 1 | 1 → A̅·B |
| 1 | 0 | 1 → A·B̅ |
| 1 | 1 | 1 → A·B |

Sumando estos mintérminos:

F = A̅·B + A·B̅ + A·B

Esta suma lógica cubre todos los casos en que al menos una entrada es 1, lo que corresponde al comportamiento de la compuerta OR.

**Indique su tabla de verdad, símbolo y función de una compuerta lógica XNOR.**

La compuerta XNOR (equivalencia lógica) es la inversa de la XOR. Su salida es 1 cuando ambas entradas son iguales, ya sean 0 o 1.

**Tabla de verdad:**

| **A** | **B** | **F = A ⊙ B** |
| --- | --- | --- |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

**Símbolo lógico:** Un símbolo similar al de la XOR pero con un círculo adicional al final, indicando la negación.

**Función lógica:**  
F = (A ⊕ B)′ = A·B + A̅·B̅

1. Conclusiones

A través del desarrollo del laboratorio logramos comprender de forma teórica y experimental el funcionamiento de los principales operadores lógicos implementados mediante circuitos integrados TTL. Se evidenció cómo las compuertas NAND, NOR, NOT, AND, OR y XOR permiten representar operaciones booleanas fundamentales en el diseño de sistemas digitales.

También la implementación de los circuitos en la placa base del equipo Simulog, así como en plataformas de simulación como MULTISIM o PROTEUS, permitió observar en la práctica cómo se comportan las compuertas ante diversas combinaciones de entrada, validando sus respectivas tablas de verdad. Este enfoque reforzó la comprensión de conceptos clave como el álgebra booleana, las expresiones lógicas, la equivalencia funcional entre diferentes formas de representar una misma operación y la importancia del cableado correcto en circuitos físicos.

Por ultimo la práctica fomentó el desarrollo de habilidades prácticas como la correcta identificación de pines en los C.I., el uso adecuado del protoboard y la capacidad de traducir una lógica teórica a un diseño concreto y funcional.

**Recomendaciones**

1. **Revisar bien las hojas técnicas de los C.I.** antes de conectarlos, especialmente para identificar correctamente la alimentación (Vcc), tierra (GND) y pines de entrada/salida, evitando daños o errores de conexión.
2. **Verificar cuidadosamente las conexiones** en la protoboard o placa Simulog antes de energizar el circuito, ya que un mal contacto puede generar resultados erróneos o incluso dañar los componentes.
3. **Utilizar simuladores digitales** antes del montaje físico como herramienta de verificación previa, ya que permiten identificar fallos lógicos sin poner en riesgo componentes reales.
4. **Registrar sistemáticamente los resultados** en tablas de verdad al realizar pruebas, para facilitar el análisis y comparación con la teoría.
5. **Repetir los ejercicios con diferentes combinaciones de entradas** para afianzar el entendimiento del comportamiento de cada compuerta lógica y observar patrones de funcionamiento.
6. **Complementar la práctica con ejercicios de simplificación lógica**, de modo que se vincule el análisis teórico de expresiones booleanas con su implementación mediante compuertas.

**Cuestionario**

**1. ¿Cómo se utiliza un protoboard y una placa base del equipo Simulog?**

El **protoboard** es una herramienta fundamental para el montaje rápido de circuitos electrónicos sin necesidad de realizar soldaduras. Está compuesto por una matriz de orificios interconectados internamente que permiten insertar diversos componentes electrónicos, como resistencias, cables y circuitos integrados (C.I.). Para un uso adecuado, es esencial identificar correctamente las filas destinadas a la alimentación (Vcc y GND), así como las columnas conectadas internamente para la inserción de los pines de los C.I.

En cambio, la **placa base del equipo Simulog** es un dispositivo de carácter educativo que incorpora módulos lógicos preinstalados (como compuertas lógicas), lo cual facilita la construcción de circuitos digitales. Esta placa cuenta con fuentes de alimentación, indicadores visuales como LEDs o semáforos, y conectores que permiten realizar conexiones con cables de manera sencilla. Es ideal para prácticas de laboratorio, ya que evita el uso de componentes discretos.

**2. Demuestre con la tabla de verdad que 𝐹 = 𝐴 ⊕ 𝐵 = 𝐴̅𝐵 + 𝐴𝐵̅**

La compuerta XOR (⊕) genera una salida igual a 1 únicamente cuando sus entradas son diferentes. Su tabla de verdad es:

| **A** | **B** | **F = A ⊕ B** |
| --- | --- | --- |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

A partir de esta tabla se verifica que la salida es 1 en los casos en que A y B son distintos, lo cual corresponde a la expresión booleana:  
**F = A̅·B + A·B̅**

**3. Demuestre, utilizando maxtérminos, que 𝐹 = 𝐴 · 𝐵 = (𝐴 + 𝐵)(𝐴 + 𝐵̅)(𝐴̅ + 𝐵)**

La compuerta AND (A·B) solo produce una salida igual a 1 cuando ambas entradas son 1. Su tabla de verdad es:

| **A** | **B** | **F = A·B** |
| --- | --- | --- |
| 0 | 0 | 0 → (A + B) |
| 0 | 1 | 0 → (A + B̅) |
| 1 | 0 | 0 → (A̅ + B) |
| 1 | 1 | 1 → No se incluye |

Multiplicando los maxtérminos de los casos donde la salida es 0:

**F = (A + B)(A + B̅)(A̅ + B)**

Esta expresión representa la función lógica AND, pues solo permite una salida de 1 cuando A y B son ambos 1.

**4. Demuestre, utilizando mintérminos, que 𝐹 = 𝐴 + 𝐵 = 𝐴̅𝐵 + 𝐴𝐵̅ + 𝐴𝐵**

La compuerta OR (A + B) produce una salida igual a 1 cuando al menos una de sus entradas es 1. La tabla de verdad es:

| **A** | **B** | **F = A + B** |
| --- | --- | --- |
| 0 | 0 | 0 → No se incluye |
| 0 | 1 | 1 → A̅·B |
| 1 | 0 | 1 → A·B̅ |
| 1 | 1 | 1 → A·B |

Sumando los mintérminos:

**F = A̅·B + A·B̅ + A·B**

Esta es la forma equivalente de la función OR en términos de mintérminos.

**5. ¿Cuál es la tabla de verdad, símbolo y función de la compuerta lógica XNOR?**

La compuerta **XNOR** (o equivalencia lógica) es la inversa de la XOR. Genera una salida igual a 1 únicamente cuando ambas entradas son iguales.

**Tabla de verdad:**

| A | B | F = A ⊙ B |