

# Instituto Tecnológico de Costa Rica

Fundamentos de Arquitectura de Computadores

Título del documento:

Bitácora del proyecto: Lógica Combinatoria

Estudiante:

Byron Mata Fuentes      2021430403

Grupo: 2

Profesor: Luis Alberto Chavarría Zamora

Fecha de entrega: 4 de abril de 2024

I Semestre 2024

## 22/03/2024 - Creación del repositorio y de la documentación

### Descripción

Se completó la creación del repositorio Git conforme a la solicitud para actualizar la bitácora del proyecto. El primer commit incluye la creación del archivo README.md, marcando así el inicio de la rama main.

Además, se han generado los documentos entregables del proyecto utilizando LaTeX. Posteriormente, se ha comenzado a redactar el presente documento. Por otro lado, el documento del "artículo científico" ha sido organizado en secciones según lo especificado en las instrucciones de la evaluación.

### Resultados

- Se creó el repositorio: [https://github.com/ByrnMta/Proyecto-Logica\\_Combinatoria.git](https://github.com/ByrnMta/Proyecto-Logica_Combinatoria.git)
- Se creó el documento de la bitácora del proyecto.
- Se creó el documento del artículo científico del proyecto.

### Próximas Tareas

- Crear un proyecto en el software Tinkercard para la simulación del diseño.
- Realizar e implementar el diseño del circuito combinatorio para el decodificador.

## 23/03/2024 - Inicio del modulo del decodificador y simulación

### Descripción

Se creó un proyecto en el software Tinkercard con el objetivo de simular cada uno de los módulos que conforman el diseño del sistema. Inicialmente, se incorporaron algunos componentes del módulo del circuito combinatorio, centrándose específicamente en el decodificador. Para ello se determinó el rango de activación, y se procedió a su obtención teórica.

El rango de activación sería de 2 a 3 y de 6 a 7 en su representación decimal. Usando la tabla de verdad mostrada en la figura 1, se obtuvieron los valores en su representación de código Gray.

Number	Gray code		
0	0	0	0
1	0	0	1
2	0	1	1
3	0	1	0
4	1	1	0
5	1	1	1
6	1	0	1
7	1	0	0

Figura 1: Código Gray de 3 bits. Fuente [1].

Seguidamente se determinó la salida y se definieron los minterminos para la ecuación booleana que describe el funcionamiento del decodificador, tal y como se presenta en la tabla 1.

Tabla 1: Tabla de valores para el diseño del circuito combinatorio del decodificador.

Decimal	A	B	C	Y	Nombre min
0	0	0	0	0	$m_0$
1	0	0	1	0	$m_1$
2	0	1	0	1	$m_2$
3	0	1	1	1	$m_3$
4	1	0	0	0	$m_4$
5	1	0	1	0	$m_5$
6	1	1	0	1	$m_6$
7	1	1	1	1	$m_7$

A partir de los minterminos  $m_2$ ,  $m_3$ ,  $m_6$  y  $m_7$ , se determinó la ecuación boolana para el circuito combinatorio del decodificador.

$$F(A, B, C) = \sum m_2, m_3, m_6, m_7 \quad (1)$$

$$Y = \bar{A} \cdot B \cdot C + \bar{A} \cdot B \cdot \bar{C} + A \cdot \bar{B} \cdot C + A \cdot \bar{B} \cdot \bar{C} \quad (2)$$

Considerando al ecuación (2), se realizó un proceso de reducción mediante la simplificación booleana. En la figura 2 se presenta el proceso de simplificación:

▷ Simplificación :

$$\bar{A} \cdot B \cdot C + \bar{A} \cdot B \cdot \bar{C} + A \cdot \bar{B} \cdot C + A \cdot \bar{B} \cdot \bar{C}$$

$$\bar{A} \cdot B (C + \bar{C}) + A \cdot \bar{B} (\bar{C} + C) \quad \dots \text{Distributividad}$$

$$\bar{A} \cdot B + A \cdot \bar{B} \quad \dots \text{Complemento}$$

$$\therefore Y = \bar{A} \cdot B + A \cdot \bar{B}$$

Figura 2: Proceso de simplificación booleana para el circuito combinatorio del decodificador.

Finalmente, se creó la tabla 2 para la ecuación simplificada obteniendo que, el circuito requiere de una compuerta XOR.

Tabla 2: Tabla de verdad de la ecuación que describe el circuito combinatorio simplificado.

A	B	$\bar{A}$	$\bar{B}$	$\bar{A} \cdot B$	$A \cdot \bar{B}$	$\bar{A} \cdot B + A \cdot \bar{B}$
0	0	1	1	0	0	0
0	1	1	0	1	0	1
1	0	0	1	0	1	1
1	1	0	0	0	0	0

Para la simulación, se programó en “Arduino IDE” un programa que primero realiza la lectura de los pines digitales, en los cuales ingresaba una señal proveniente del Sensor Ultrasónico. Seguidamente, realiza un mapeo del rango y del valor de lectura para ajustarlo al rango deseado. Posteriormente, realiza la conversión a código Gray y finalmente lo muestra en el monitor serial para verificar que está funcionando correctamente. En las figuras 3, se muestran secciones de código del programa.

```
// Función para calcular el código Gray
int toGray(int num) {
    return num ^ (num >> 1);
}
```

Figura 3: Función para realizar la conversión del código Gray.

## Resultados

- Se creó del proyecto en Tinkercard para la simulación del sistema.
- Se obtuvo el circuito combinatorio para el modulo del decodificador del sistema.
- Se inició el programa que se cargará al Arduino UNO para la funcionalidad del sistema.
- Se verificó que el diseño para el decodificador era correcto y funciona con lo esperado.

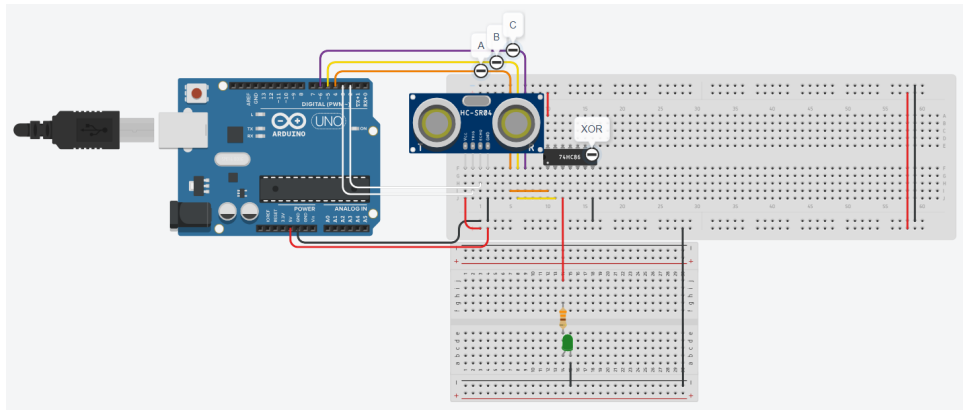


Figura 4: Diseño del modulo del decodificador incompleto.

## Próximas Tareas

- Completar el modulo de decodificador y el del desacople.

## Referencias

- [1] Sarah Harris y David Harris. Digital design and computer architecture: arm edition. Morgan Kaufmann, 2015.