

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
- 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.
- Iniciar el submódulo del decodificador.

23/03/2024 - Inicio del submódulo 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	1	1	m_2
3	0	1	0	1	m_3
4	1	1	0	0	m_4
5	1	1	1	0	m_5
6	1	0	1	1	m_6
7	1	0	0	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:

Handwritten simplification process:

$$\begin{aligned} &\bar{A} \cdot B \cdot C + \bar{A} \cdot B \cdot \bar{C} + A \cdot \bar{B} \cdot C + A \cdot \bar{B} \cdot \bar{C} \\ &\bar{A} \cdot B (C + \bar{C}) + A \cdot \bar{B} (\bar{C} + C) \quad \dots \text{Distribuidad} \\ &\bar{A} \cdot B + A \cdot \bar{B} \quad \dots \text{Complemento} \\ &\therefore Y = \bar{A} \cdot B + A \cdot \bar{B} \end{aligned}$$

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 submódulo 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.

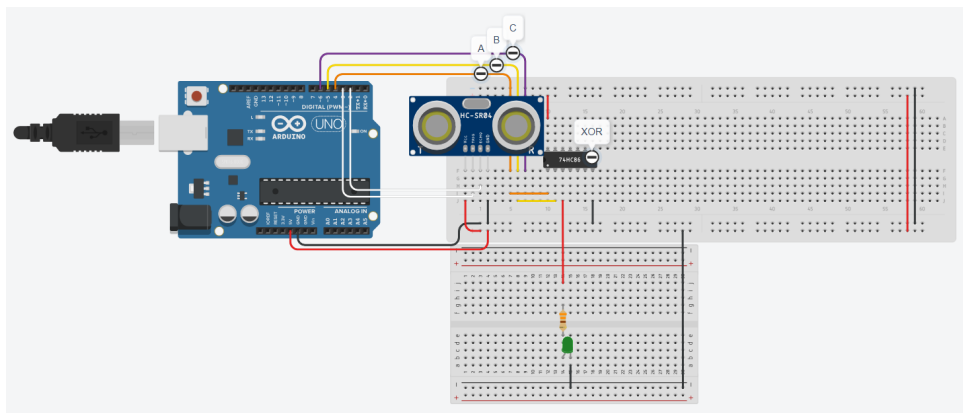


Figura 4: Diseño del submódulo del decodificador incompleto.

Próximas Tareas

- Completar el submódulo de decodificador y el del desacople.

24/03/2024 - Submódulo del decodificador y desacople completado

Descripción

Dado el avance del día de ayer, se trabajó en completar la etapa de desacople del diseño del sistema. Esta corresponde precisamente al submódulo del decodificador, así como a las etapas del accionador y del desacople. Dado que ya se contaba con los primeros procesos del decodificador y el accionador, se investigó únicamente cómo hacer que un transistor funcione como un switch (un interruptor) para el arranque de un motor DC, siguiendo las recomendaciones del profesor. El resultado de esta investigación se muestra en el diagrama de la siguiente figura:

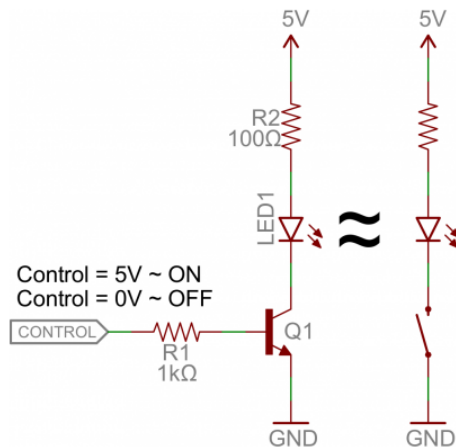


Figura 5: Diagrama para la aplicación de un transistor como switch. Fuente [2].

A partir del diagrama anterior, se consideró el valor de la resistencia de base, además del uso de un transistor BJT NPN. Únicamente se remplazaría la carga del colector para realizar la conexión directa con el motor dc. Todo lo anterior fue trasladado al proyecto de Tinkercard para su simulación.

Resultados

- Se comprobó el funcionamiento adecuado del transistor como un switch para encender un motor DC en el rango dado al decodificador.
- Se completo en su totalidad el submódulo del decodificador y de la etapa del desacople del sistema.

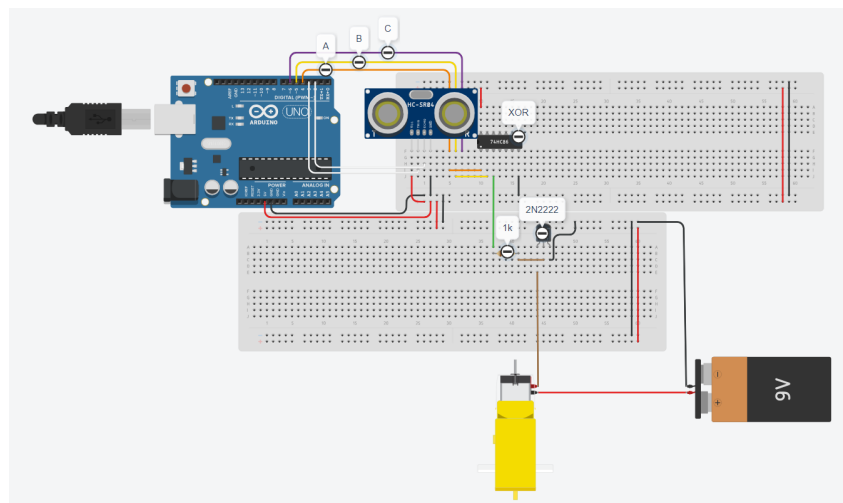


Figura 6: Diseño del submódulo del decodificador completo.

Próximas Tareas

- Iniciar el submódulo del encodificador.

25/03/2024 - Inicio del submódulo del encodificador

Descripción

Se realizó una breve investigación para determinar en qué consiste el código binario con exceso 3. A partir de ello, se elaboró una tabla considerando la recomendación del profesor. Esta indicaba que se

hiciera un enfoque especialmente en los primeros 3 bits del código binario en representación de exceso 3, que van del menos significativo hasta el más significativo, para lograr que fuera circular. El resultado de esto se muestra a continuación:

Tabla 3: Tabla de los código Gray y Exceso 3.

Decimal	Gray	Exceso 3
0	000	011
1	001	100
2	011	101
3	010	110
4	110	111
5	111	000
6	101	001
7	100	010

Considerando la tabla anterior, se asignó una etiqueta de salida para cada bit del código con exceso 3. Esto implica que, según la tabla 3, se designó Y1 para la columna del bit más significativo, Y2 para la columna del bit intermedio y Y3 para la columna del bit menos significativo.

Tabla 4: Tabla de los código Gray y Exceso 3 con etiquetas por columna de bits.

Decimal	Gray	Y1	Y2	Y3
0	000	0	1	1
1	001	1	0	0
2	011	1	0	1
3	010	1	1	0
4	110	1	1	1
5	111	0	0	0
6	101	0	0	1
7	100	0	1	0

A partir de la tabla 4, se determinaron las ecuaciones booleanas que describen la conversión entre códigos. Este proceso se llevó a cabo mediante mapas K para cada etiqueta (columna) del código exceso 3. A continuación, se presenta el proceso realizado, determinando que sería necesario el uso de tres compuertas adicionales aparte del XOR, con el cual ya se contaba. Estas compuertas serían la NOT, AND y OR.

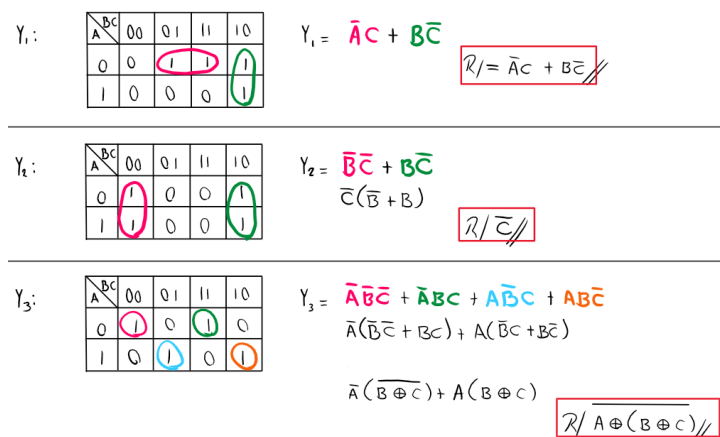


Figura 7: Proceso de simplificación mediante mapas K para el código binario con exceso 3.

Tras el proceso anterior, se llevó a cabo la implementación del mismo en la simulación, realizando las conexiones necesarias. Se adicionaron dos etapas más: por un lado, la del visualizador con LED's para mostrar la salida del código en exceso 3, y por otro lado, la etapa de retroalimentación.

Resultados

- Se determinó el circuito combinatorio para el submódulo del encodificador.
- Se completó el submódulo del encodificador.
- Se terminó la etapa del circuito combinatorio con los submódulos del decodificador y encodificador.

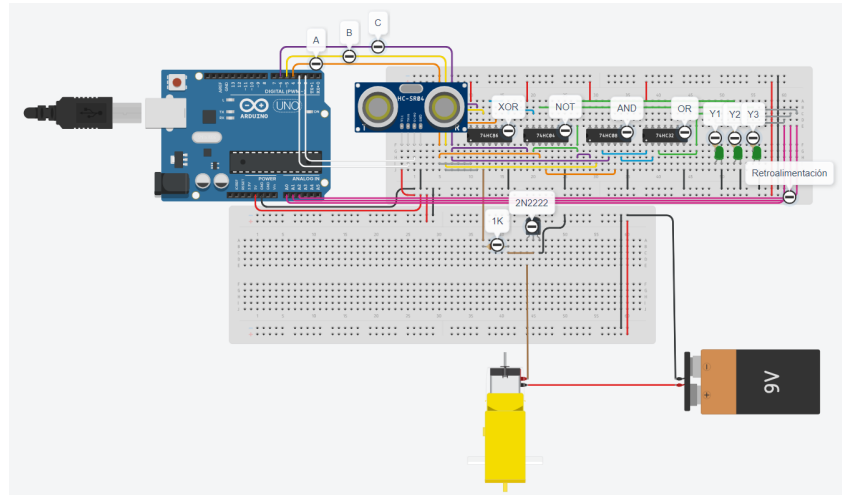


Figura 8: Diseño de los submódulos del decodificador y encodificador completos.

Próximas Tareas

- Implementar la etapa del visualizador de 7 segmentos de la señal proveniente de la retroalimentación.

Referencias

- [1] Sarah Harris y David Harris. Digital design and computer architecture: arm edition. Morgan Kaufmann, 2015.
- [2] SparkFun, "Transistors: Applications I: Switches", SparkFun, <https://learn.sparkfun.com/tutorials/transistors/applications-i-switches> (Consultado: 24 mzo., 2024).