

Práctica 8:

CONTADOR DE METALES

Prof. Rubén Galicia Mejía

Equipo:

Los Pingüinos de Madagascar

Integrantes:

- Ayala Fuentes Sunem Gizeht
- Dávila de Jesús Sandro Armando
- García García Francisco
- Guevara Badillo Areli Alejandra

Fecha de realización: *lunes, 24 de junio de 2024*

Fecha de entrega: *jueves, 27 de junio de 2024*

MATERIALES

- Sensor inductivo detector de metales LJ12A3-4-Z/BX
- Una batería de 9V
- Display de 7 segmentos de ánodo común
- 1 led
- 1 resistencia de $3.3k\Omega$
- Placa de pruebas (protoboard)
- Cables y conectores
- Placa o base para montar la FPGA

INTRODUCCIÓN

En el ámbito de la ingeniería electrónica y la automatización, los detectores de metales juegan un papel crucial en diversas aplicaciones industriales y de seguridad. En esta práctica, se ha llevado a cabo el diseño y la implementación de un detector de metales simple utilizando un sensor inductivo LJ12A3-4Z y una FPGA (Field-Programmable Gate Array).

El objetivo principal de esta práctica es comprender el funcionamiento del sensor inductivo y aprender a integrar y programar una FPGA para procesar las señales de sensores en tiempo real. A lo largo de este documento, se detallarán los componentes utilizados, el diseño del circuito, la programación de la FPGA, y los resultados obtenidos durante la implementación y prueba del detector de metales.

MARCO TEÓRICO

Se pueden utilizar varios principios de detección para distintas tareas de detección. El principio de detección más adecuado para una aplicación específica se determina a partir de varios aspectos: Estos incluyen el material del objeto que se va a detectar, el entorno de aplicación y la distancia desde la que se va a realizar la detección.

Si el objeto que se va a detectar es eléctricamente conductivo, por ejemplo, de metal, y se puede detectar de cerca, se recomienda un sensor inductivo.

Un sensor inductivo es un sensor de proximidad diseñado para detectar objetos metálicos. Podemos emplear conectar fácilmente este tipo de sensores a un autómata o procesador como Arduino.

Internamente estos sensores disponen de un generador de campo magnético y en una bobina inductora que detecta el campo generado por el propio sensor.

La presencia de un objeto metálico modifica el campo inducido. Al tener una menor resistencia magnética (reluctancia) el campo magnético se «alarga», incrementando la corriente inducida en la bobina sensora.

Este campo es detectado por la electrónica del sensor, cuya salida se activa cuando un metal es detectado. Como el sensor tiene dos estados (Ilustración 1), en ocasiones a estos dispositivos se les denomina interruptores inductivos de proximidad.

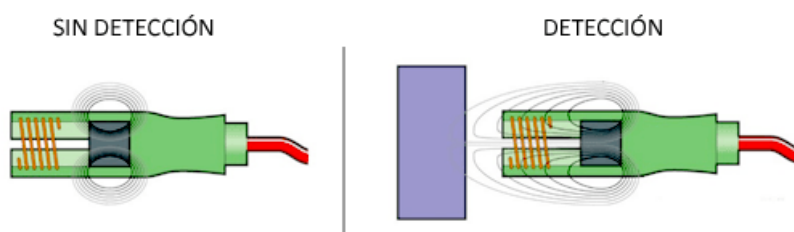


Ilustración 1. Funcionamiento del campo magnético al detectar un metal.

Los sensores de proximidad inductivos sólo pueden detectar objetos metálicos. No detectan objetos no metálicos, tales como plástico, madera, papel y cerámica. A diferencia de los sensores fotoeléctricos, esto permite que un sensor de proximidad inductivos pueda detectar un objeto de metal a través de plástico opaco.

Los sensores inductivos generalmente tienen 3 o 4 conductores codificados por colores. Normalmente incorporan una etiqueta en el propio sensor que indica el esquema de conexión.

El sensor inductivo que vamos a emplear LJ12A3-4-Z/BY. El cable azul (BU) es Gnd, el marrón (BN) es Vcc, y el negro (BK) es la salida del sensor (Ilustración 2).



Ilustración 2. Esquema de montaje.

Vcc entre 5 y 36V Al usar alimentación externa SIEMPRE poner GND común.

DESARROLLO DE LA PRÁCTICA

En el desarrollo de esta práctica, nos enfocamos en dos componentes clave: la programación del código en VHDL y el ensamblado del circuito que integra el sensor LJ12A3-4Z con la FPGA. Ambos aspectos son cruciales para la creación de un detector de metales funcional y eficiente.

Desarrollo del código en VHDL

Definición de la Entidad Counter

```
entity Counter is
    port(
        clk : in std_logic;           -- Reloj de entrada principal
        clk_out : out std_logic;      -- Salida del reloj de 1 Hz
        reset : in std_logic;         -- Señal de reinicio del contador
        count_in : in std_logic;      -- Entrada para contar eventos
        seg : OUT STD_LOGIC_VECTOR(6 DOWNTO 0); -- Salida para los segmentos del display
        digit : OUT STD_LOGIC_VECTOR(7 DOWNTO 0) -- Salida para habilitar el dígito en un
        display 7-segmentos
    );
end Counter;
```

- clk: Reloj de entrada a 50 MHz.
- clk_out: Salida del reloj de 1 Hz.
- reset: Señal de reinicio que pone el contador en estado inicial.
- count_in: Entrada que se utiliza para incrementar el contador en flanco de bajada.
- seg: Vector de 7 bits para manejar los segmentos del display 7-segmentos.
- digit: Vector de 8 bits para habilitar un dígito en un display 7-segmentos.

Arquitectura Behavioral

```
architecture Behavioral of Counter is
    signal clk_1hz : std_logic := '0';           -- Reloj interno a 1 Hz
    signal count_divider : UNSIGNED(31 downto 0) := (others => '0'); -- Divisor del reloj
    constant CLK_FREQ : integer := 50000000;    -- Frecuencia del reloj principal: 50 MHz
    constant FREQ_5HZ : integer := 5;           -- Frecuencia deseada: 5 Hz para el
    contador
```

```

    constant DIV_5HZ : UNSIGNED(31 downto 0) := to_unsigned(CLK_FREQ / (2 * FREQ_5HZ) - 1,
32); -- Valor para dividir el reloj a 1 Hz
    signal div_value : UNSIGNED(31 downto 0) := DIV_5HZ; -- Valor de división calculado
    signal count_in_last : std_logic := '1'; -- Estado anterior de `count_in` para
detectar flancos de bajada
    signal active : integer := 0; -- Variable para el estado actual del
contador

    type seg_type is array (0 to 15) of STD_LOGIC_VECTOR(6 downto 0); -- Tipo para
codificaciones de los segmentos
    constant seg_codes : seg_type := ( -- Codificaciones de 7 segmentos para
números hexadecimales
        "0000001", -- 0
        "1111001", -- 1
        "0010010", -- 2
        "0000110", -- 3
        "1001100", -- 4
        "0100100", -- 5
        "0100000", -- 6
        "0001101", -- 7
        "0000000", -- 8
        "0000100", -- 9
        "0001000", -- A
        "1100000", -- b
        "0110001", -- C
        "1000010", -- d
        "0110000", -- E
        "0111000" -- F
    );
    signal total_counter : integer := 0; -- Contador principal que va de 0 a 15

```

- clk_1hz: Reloj interno a 1 Hz generado a partir del reloj principal.
- count_divider: Contador para generar la señal clk_1hz.
- CLK_FREQ: Frecuencia del reloj principal (50 MHz).
- FREQ_5HZ: Frecuencia deseada de 5 Hz para el contador.
- DIV_5HZ: Valor calculado para dividir el reloj de 50 MHz a 1 Hz.
- count_in_last: Guarda el estado anterior de count_in para detectar el flanco de bajada.
- active: Variable de estado del contador (no utilizada en el código proporcionado, podría ser un remanente de versiones anteriores).
- seg_codes: Mapeo de números hexadecimales a códigos de segmentos del display.
- total_counter: Contador que aumenta en cada flanco de bajada de count_in, reiniciándose en 15.

Proceso para Generar el Reloj de 1 Hz

```

process(clk, reset)
begin
    if reset = '1' then
        count_divider <= (others => '0');
        clk_1hz <= '0';
    elsif rising_edge(clk) then
        if count_divider >= div_value then
            count_divider <= (others => '0');
            clk_1hz <= not clk_1hz;
        else
            count_divider <= count_divider + 1;
        end if;
    end if;
end process;

```

A cada flanco de subida del reloj principal, el `count_divider` se incrementa hasta llegar a `div_value`. Cuando esto ocurre, se reinicia el `count_divider` y se cambia el estado del reloj `clk_1hz`, generando una señal de 1 Hz.

Proceso para Manejo del Contador Principal

```
process(clk_1hz, reset)
begin
    if reset = '1' then
        total_counter <= 0;
        count_in_last <= '1';
    elsif rising_edge(clk_1hz) then
        if count_in_last = '1' and count_in = '0' then -- Detección de flanco de bajada
            total_counter <= total_counter + 1;
            if total_counter = 15 then
                total_counter <= 0;
            end if;
        end if;
        count_in_last <= count_in; -- Actualiza el estado anterior de count_in
    end if;
end process;
```

A cada flanco de subida del reloj de 1 Hz, se verifica si hubo un flanco de bajada en `count_in`. Si es así, se incrementa `total_counter` y se reinicia a 0 cuando llega a 15.

Asignación de Salidas

```
seg <= seg_codes(total_counter);
digit <= "11111110"; -- Se asume que sólo un dígito está habilitado en el display de 7 segmentos
clk_out <= clk_1hz;
```

- `seg`: Se asigna el código de segmentos correspondiente al valor de `total_counter` usando el array `seg_codes`.
- `digit`: Habilita el primer dígito del display 7-segmentos.
- `clk_out`: Se asigna la señal `clk_1hz` como salida del reloj de 1 Hz.

Este código VHDL implementa un contador que se incrementa en cada flanco de bajada del `count_in`, con una frecuencia de conteo derivada de un reloj de 50 MHz. La señal `clk_1hz` se genera mediante la división del reloj principal y se usa para actualizar el contador principal. El valor del contador se usa para controlar un display de 7-segmentos, mostrando los dígitos del 0 al F en formato hexadecimal.

Ensamblado del circuito y Funcionamiento

Primero, conectamos el sensor inductivo LJ12A3-4-Z/BX a la protoboard. Este sensor requiere una alimentación de 6-36V, por lo que usamos la batería de 9V para alimentarlo. Conectamos el cable marrón del sensor al positivo de la batería y el cable azul al negativo (tierra). El cable negro del sensor, que es la salida, lo conectamos a un pin de entrada de la FPGA a través de un cable de conexión.

El siguiente paso fue conectar el display de 7 segmentos de ánodo común a la protoboard. Conectamos cada uno de los segmentos del display a los pines de salida de la FPGA. Usamos resistencias adecuadas para limitar la corriente a través de los segmentos del display. En este caso, utilizamos una resistencia de 3.3kΩ para el LED indicador.

Colocamos un LED en la protoboard como indicador visual adicional. Este LED se conectó en serie con la resistencia de 3.3kΩ y se controló mediante un pin de salida de la FPGA, encendiéndose cuando se detectaba un metal.

Con todo ya conectado y en su lugar como se ilustra en la Ilustración 3, comenzamos las pruebas.

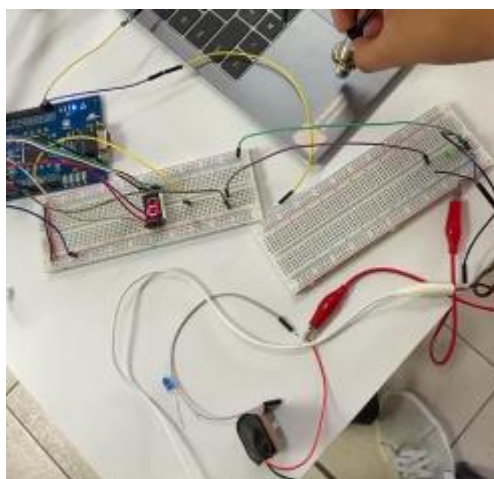


Ilustración 3. Circuito ensamblado

Realizamos varias pruebas para asegurarnos de que el circuito funcionara correctamente. Al probar el sistema, observamos cómo el display de 7 segmentos mostraba un incremento en el conteo cada vez que acercábamos un objeto metálico al sensor inductivo.

El LED se encendía de manera consistente con la detección del metal, proporcionando una confirmación visual de que el sistema estaba funcionando correctamente.

Video del funcionamiento:

https://drive.google.com/file/d/1HqI9b_M8oXPIPd4KtKmbBDACHBZLg5jM/view?usp=sharing

CONCLUSIONES

CONCLUSIONES 1 (Ayala Fuentes Sunem Gizeth)

El desarrollo de este proyecto nos permitió profundizar en el funcionamiento de los sensores inductivos y su integración con una FPGA para crear un sistema de detección de metales eficiente. A lo largo de la práctica, aprendimos a programar en VHDL y a diseñar un circuito funcional, lo cual fortaleció nuestras habilidades en diseño y programación de sistemas digitales. Observar cómo el sistema respondía de manera confiable a la presencia de objetos metálicos, incrementando el conteo en el display de 7 segmentos, fue especialmente gratificante. Este proyecto no solo me ayudó a consolidar mis conocimientos teóricos, sino que también me brindó una valiosa experiencia práctica que será muy útil en mi futura carrera como ingeniera electrónica.

CONCLUSIONES 2 (Dávila de Jesús Sandro Armando)

La práctica representó una excelente oportunidad para aplicar conceptos teóricos en un entorno práctico, permitiéndome entender mejor la integración de sensores inductivos con FPGAs. Trabajar en la programación en VHDL y en el ensamblaje del circuito me hizo consciente de la importancia de la precisión y la atención al detalle en el diseño de sistemas electrónicos. Además, las pruebas exhaustivas que realizamos subrayaron la necesidad de verificar cada componente para asegurar que el sistema funcionara de manera fiable. La experiencia adquirida durante esta práctica me será invaluable en el futuro, especialmente en proyectos que requieran la integración de hardware y software.

CONCLUSIONES 3 (García García Francisco)

A través de esta práctica, pude entender la relevancia y las aplicaciones industriales de los sensores inductivos. La implementación del detector de metales me permitió familiarizarme con el hardware y software necesarios para este tipo de proyectos. La correcta configuración y calibración de los

componentes fueron cruciales para obtener resultados precisos, y esto se reflejó en nuestras pruebas, donde el sistema detectó metales de manera consistente. Esta experiencia práctica complementó mi formación teórica, proporcionando una comprensión más profunda de cómo los principios de detección se aplican en situaciones reales. Además, trabajar en equipo y enfrentar desafíos técnicos me preparó mejor para futuras colaboraciones en proyectos de ingeniería.

CONCLUSIONES 4 (Guevara Badillo Areli Alejandra)

Trabajar en este proyecto fue una experiencia enriquecedora que combinó teoría y práctica de manera efectiva. La construcción del circuito y la programación de la FPGA fueron desafíos que me permitieron aplicar y consolidar conocimientos adquiridos en clases anteriores. Las pruebas realizadas mostraron que nuestro sistema podía detectar objetos metálicos de manera precisa, encendiendo el LED y aumentando el conteo en el display de 7 segmentos. Este éxito me hizo apreciar la importancia de los sensores inductivos en la industria, especialmente en la detección de objetos metálicos. La experiencia adquirida no solo reforzó mis habilidades técnicas, sino que también mejoró mi capacidad para resolver problemas y trabajar en equipo, lo cual es esencial en el campo de la ingeniería.

BIBLIOGRAFÍAS

- Marky. (n.d.) SENSOR INDUCTIVO: DETECTOR DE METALES CON ARDUINO. Recuperado de <https://www.murkyrobot.com/guias/sensores/sensor-inductivo>
- Pepperl+Fuchs. (2021, October 11). Sensores inductivos | Método de detección y función. Recuperado de <https://www.pepperl-fuchs.com/mexico/es/39731.htm>
- Características de los sensores de proximidad inductivos | Fundamentos del sensor: Guía de sensores para fábricas clasificados por principios | KEYENCE. (n.d.). Recuperado de <https://www.keyence.com.mx/ss/products/sensor/sensorbasics/proximity/feature/>