

Trabajo Práctico Final

Circuitos Lógicos Programables

Manejo de encoder rotativo incremental con FPGA

Ing. Ignacio Majul
Junio de 2020



Carrera de Especialización en Sistemas Embebidos

Índice de contenido

Índice de contenido	1
Introducción	2
Objetivo	2
Herramientas utilizadas	2
Descripción del funcionamiento	3
Desarrollo	4
Código VHDL	4
Esquemático simplificado	4
Simulaciones	6
Recursos utilizados	6
Depuración de señales de hardware	7
Conclusiones	8

1. Introducción

1.1. Objetivo

El objetivo de esta memoria es mostrar el proyecto implementado para el manejo de un encoder rotativo incremental sobre FPGA. Esta memoria fue presentada como trabajo práctico final de la materia Circuitos Logicos Programables

1.2. Herramientas utilizadas

Para realizar este trabajo práctico se utilizó un encoder rotativo incremental de 360 pulsos por revolución, una placa de desarrollo Arty Z7-10, un conversor USB-serie y una Laptop. La herramienta de software para el desarrollo fue Vivado Design Suite de Xilinx, tanto para desarrollar el código en VHDL como para simular los bancos de prueba, sintetizar, implementar el diseño y depurar el hardware.

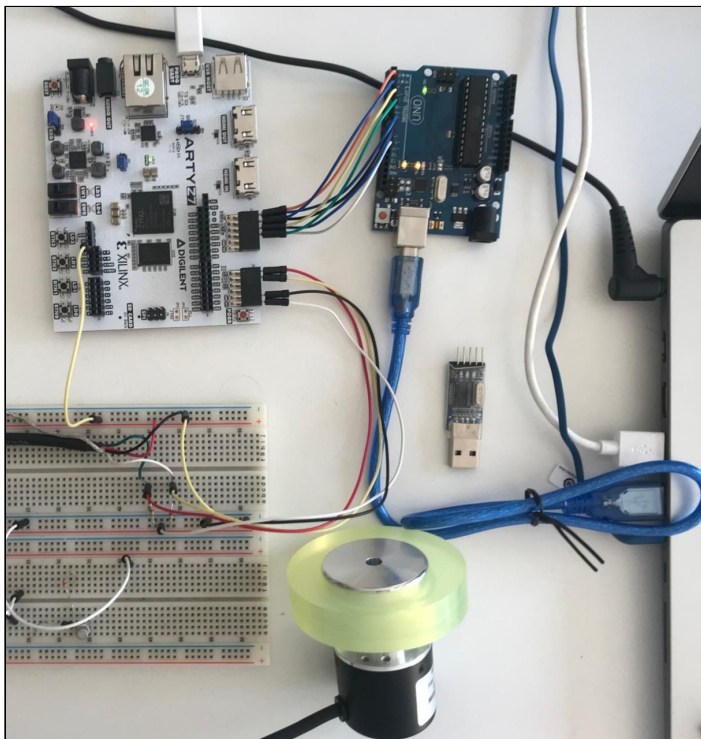
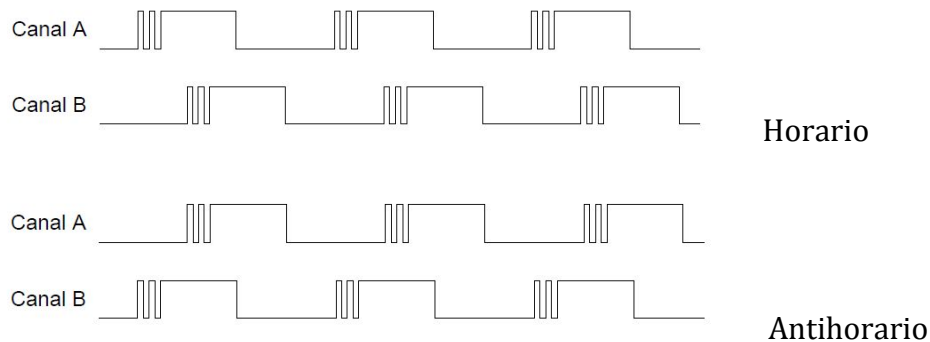


Figura 1: Banco de pruebas

1.3. Descripción del funcionamiento

El encoder LPD3806-360BM genera una señal cuadrada de DC = 50%, desfasadas 90° eléctricos por cada uno de los dos canales de salida. Cada señal genera 360 pulsos por cada revolución completa del encoder. La posición relativa del encoder, respecto a su punto inicial, se determina contando la cantidad de pulsos generados en ese intervalo. Para determinar el sentido de giro (horario o antihorario) se debe evaluar cuál de las dos señales está adelantada respecto a la otra:



Para obtener una lectura confiable es preciso filtrar las señales de entrada de manera de eliminar cualquier posible rebote que pudiera aparecer y que pudiera traducirse en una interpretación errónea de su comportamiento.

Las señales de entrada del sistema son:

- Clock: 50 MHz (escalar)
- Canales A y B del encoder (escalar)
- Pulsador de reset (escalar)

Las señales de salida del sistema son:

- Sentido de giro (escalar)
- Posición del encoder (vector de 8 bits)

La posición relativa del encoder se representa a partir de un vector de 8 bits interpretado como un entero signado y representa el avance del encoder en centímetros. Para eso se aplica la siguiente relación:

1 revolución : 1440 pulsos (360 flancos ascendentes y 360 flancos descendentes por canal)

Perímetro de la rueda del encoder : 22.5 cm.

1 centímetro de avance : $1440 / 22.5 = 64$ pulsos / cm

Vel max : 5000 rpm => 120.000 pulsos por segundo

El vector que representa se divide por 64 para obtener el avance de la rueda en cm.

Algoritmo de debounce:

Para filtrar los rebotes en los flancos de las señales de entrada se implementó un módulo que se activa cuando transcurren 10 us desde que la señal de entrada está estable. El módulo es un contador de N bits que rebalsa cuando N llega a $10\text{us} \times 50\text{MHz} = 500$.

2. Desarrollo

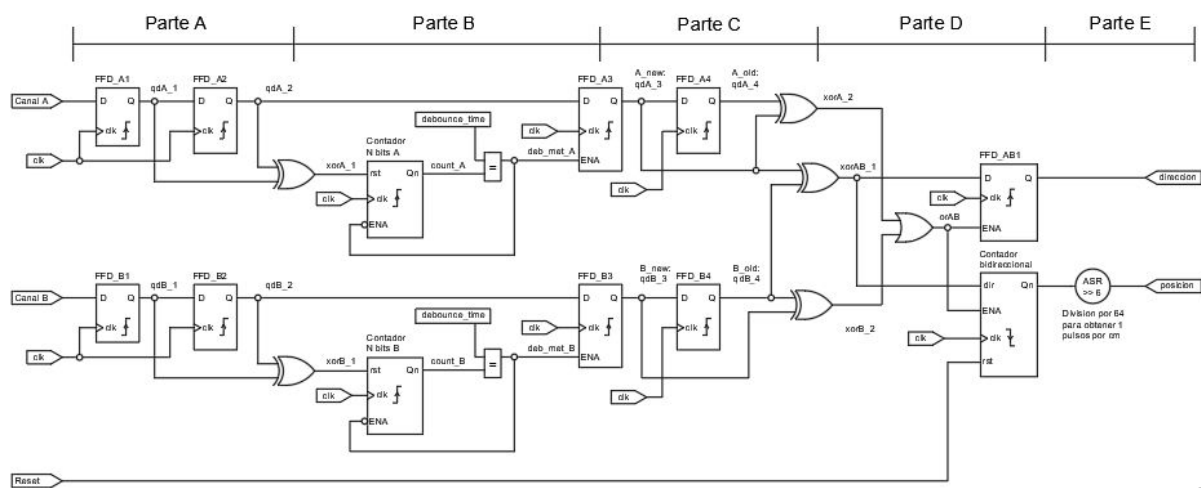
2.1. Código VHDL

Los bloques utilizados para el proyecto fueron:

- Flip Flop D. Función: sincronización y reducción de metaestabilidad
- Flip Flop D con entrada de habilitación. Función: determinación del sentido de giro
- Registros. Función: Lógica de los contadores
- Contador de bits con habilitación. Función: lógica de antirrebote.
- Contador ascendente y descendente con habilitación. Función: determinación de la posición del encoder.
- UART full duplex. (implementada, finalmente no instanciada)
- Bloque ILA para depuración de las señales de ambos canales de entrada, reset externo, sentido de giro y conteo de pulsos.

2.2. Esquemático simplificado

A continuación se muestra un diagrama esquemático simplificado con las partes más importantes del sistema.



Parte A: Se implementan dos Flip Flop para sincronizar las señales de entrada y reducir la metaestabilidad. La compuerta XOR se utiliza para detectar cambios en la señal de entrada.

Parte B: Contador de N bits para eliminar el rebote ascendente y descendente de las señales. El parámetro debounce se calcula como [Tiempo de estabilización] x [Frecuencia del clock]. El tamaño de los registros internos es variable para optimizar el uso de memoria y se calcula como $\text{Log}_2(N)$.

Parte C: Se implementan dos Flip Flop para captura de cambios en las señales filtradas.

Parte D: Determinación del sentido de giro y la posición del encoder: el sentido de giro se determina haciendo una XOR entre las señales de ambos canales, desfasadas por un ciclo de clock.

set_origin_n	Previous Inputs		New Inputs		Results	
	a_prev	b_prev	a_new	b_new	direction	position
1	0	0	1	0	1	Increment
1	1	0	1	1	1	Increment
1	1	1	0	1	1	Increment
1	0	1	0	0	1	Increment
1	0	0	0	1	0	Decrement
1	0	1	1	1	0	Decrement
1	1	1	1	0	0	Decrement
1	1	0	0	0	0	Decrement
0	X	X	X	X	No change	0

Parte E: Normalización: se divide la señal de salida por 64 mediante un shift aritmético a derecha de 6 posiciones para obtener el avance relativo de la rueda en centímetros.

2.3. Simulaciones

Se utilizaron las herramientas gráficas de Vivado para generar los bancos de prueba que permitieron testear tanto el módulo principal como los sub-bloques. Las simulaciones permitieron verificar el correcto comportamiento a nivel funcional de los bloques descritos en el esquemático. Se simularon condiciones límites de operación mecánica del encoder y máximo rebote de las señales.

Los principales resultados de las simulaciones se incluyen en el Anexo I.

2.4. Recursos utilizados

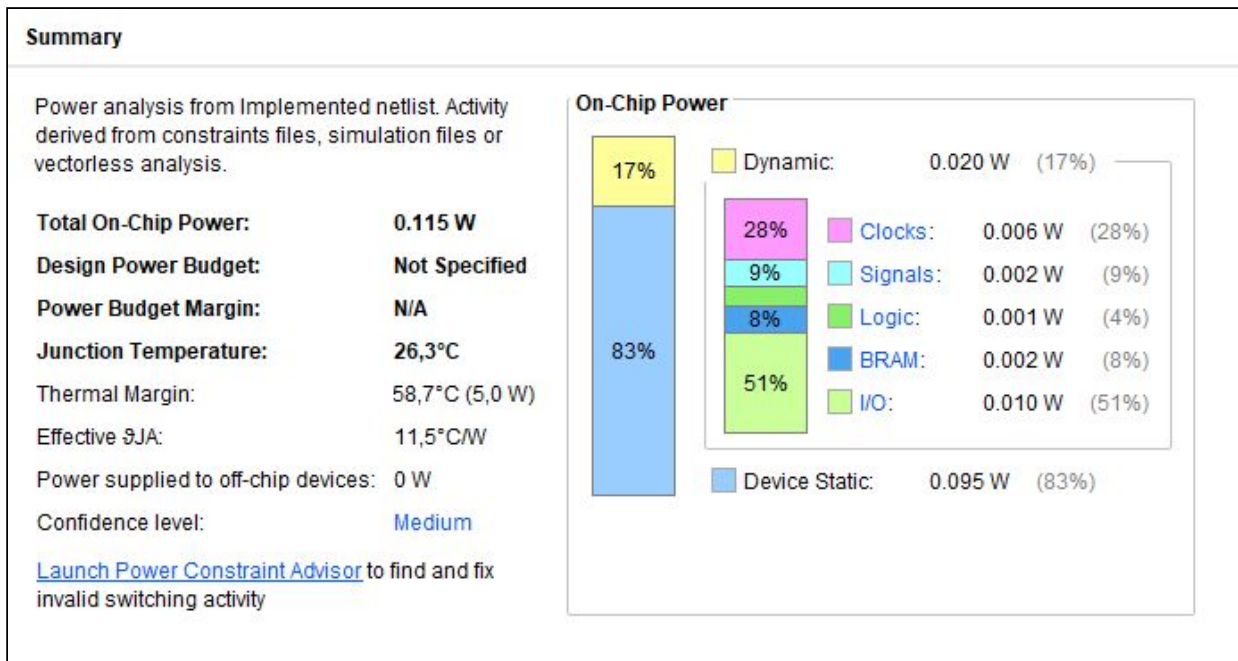
A continuación se muestran las gráficas principales que indican los recursos utilizados de la FPGA.

Recursos luego de la síntesis y luego de la implementación: se observa un incremento en la ocupación de Block RAM y LUTs por la instanciación del IP core ILA.

Utilization				Post-Synthesis	Post-Implementation
				Graph	Table
Resource	Estimation	Available	Utilization %		
LUT	68	17600	0.39		
FF	43	35200	0.12		
IO	13	100	13.00		
BUFG	1	32	3.13		

Utilization				Post-Synthesis	Post-Implementation
				Graph	Table
Resource	Utilization	Available	Utilization %		
LUT	1494	17600	8.49		
LUTRAM	120	6000	2.00		
FF	2153	35200	6.12		
BRAM	38	60	63.33		
IO	13	100	13.00		
BUFG	2	32	6.25		

Reporte de energía del sistema:



2.5. Depuración de señales de hardware

Para verificar el correcto funcionamiento del sistema, se conectó el encoder a los pines de entrada asignados en los archivos de restricciones y se conectaron puntas de prueba del bloque ILA en señales de bajo nivel.

Las señales de entrada se generaron a 400 rpm debido a la dificultad operativa de alcanzar velocidades altas en el banco de pruebas montado. Para tener un espectro amplio de análisis se aumentó el número de muestras tomado por el bloque hasta 65.536 lo cual aumenta la cantidad de recursos implementada por la herramienta.

Las señales depuradas fueron:

- Canal de entrada A : 1 bit
- Canal de entrada B: 1 bit
- Pulsador de reset : 1 bit
- Contador de pulsos : 16 bits
- Sentido de giro : 1 bit

Los principales resultados se incluyen en el Anexo II.

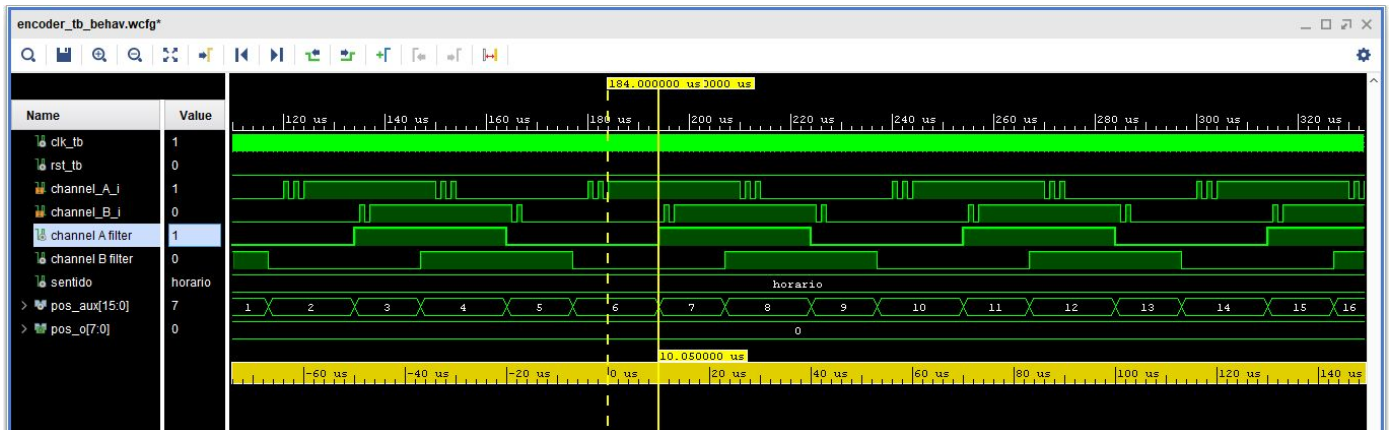
3. Conclusiones

La implementación de la solución para manejo de encoders rápidos y de alta resolución resultó ser muy eficiente con el uso de lógica programable ya que logra una total independencia en el procesamiento de las señales de entrada y logra un eficaz filtro que evita errores de interpretación. En comparación con la implementación de este mismo sistema sobre microcontroladores en lenguaje C se denota una clara mejoría en la performance por lograr el paralelismo real de procesamiento.

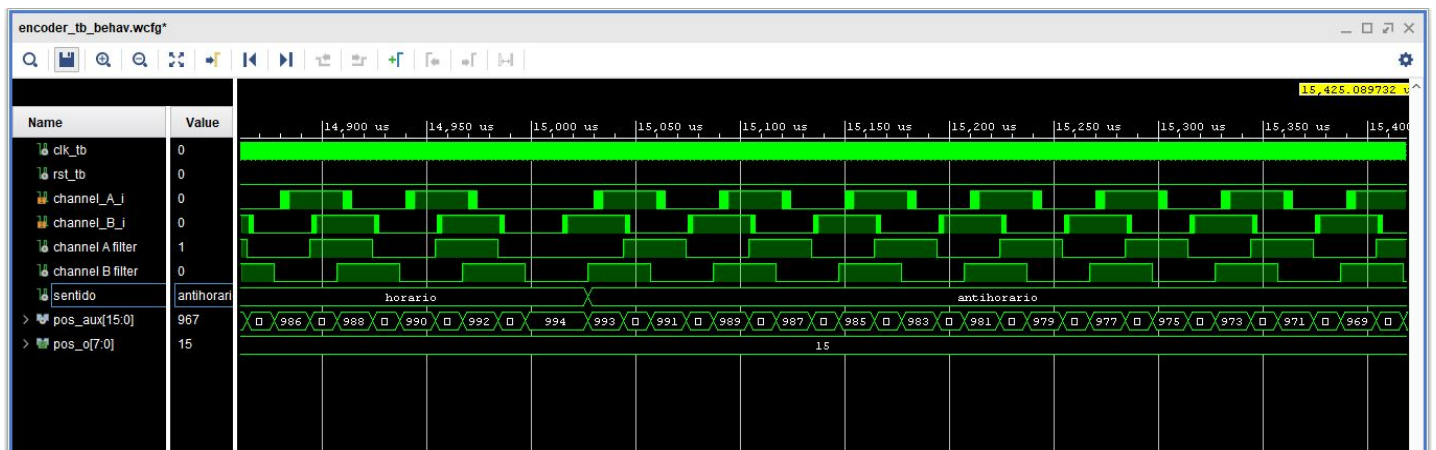
El uso de la herramienta ILA fue de gran utilidad para la visualización de la evolución temporal real de las señales de entrada y para verificar el comportamiento esperado del encoder en las condiciones de trabajo.

Anexo I - Simulaciones

Verificación del cambio en el sentido de giro:

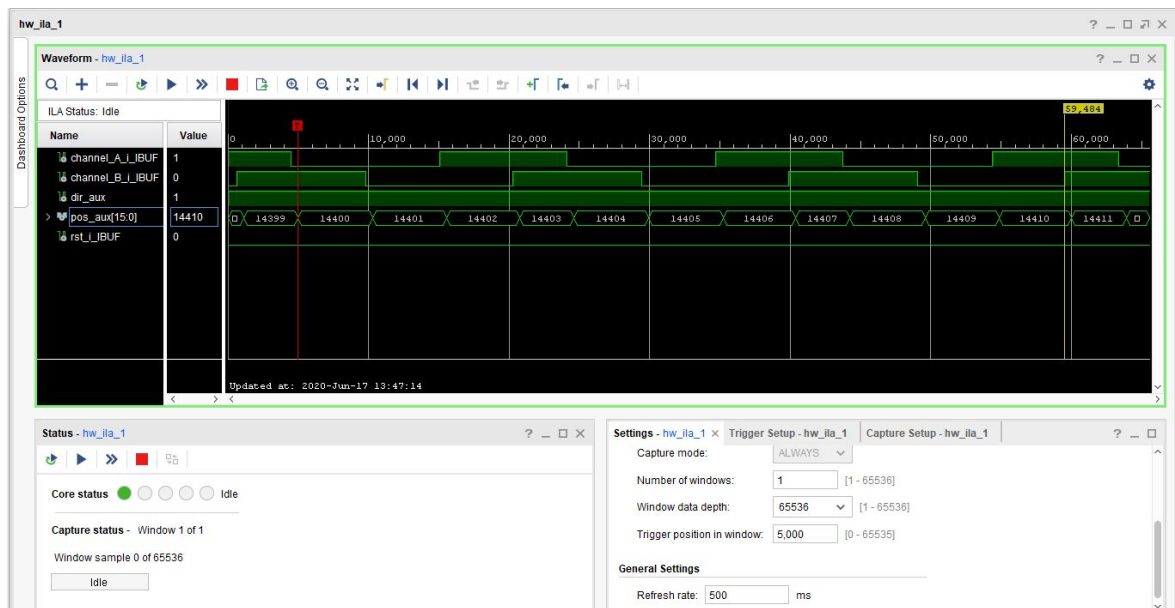


Verificación de desfase por filtrado de rebotes:

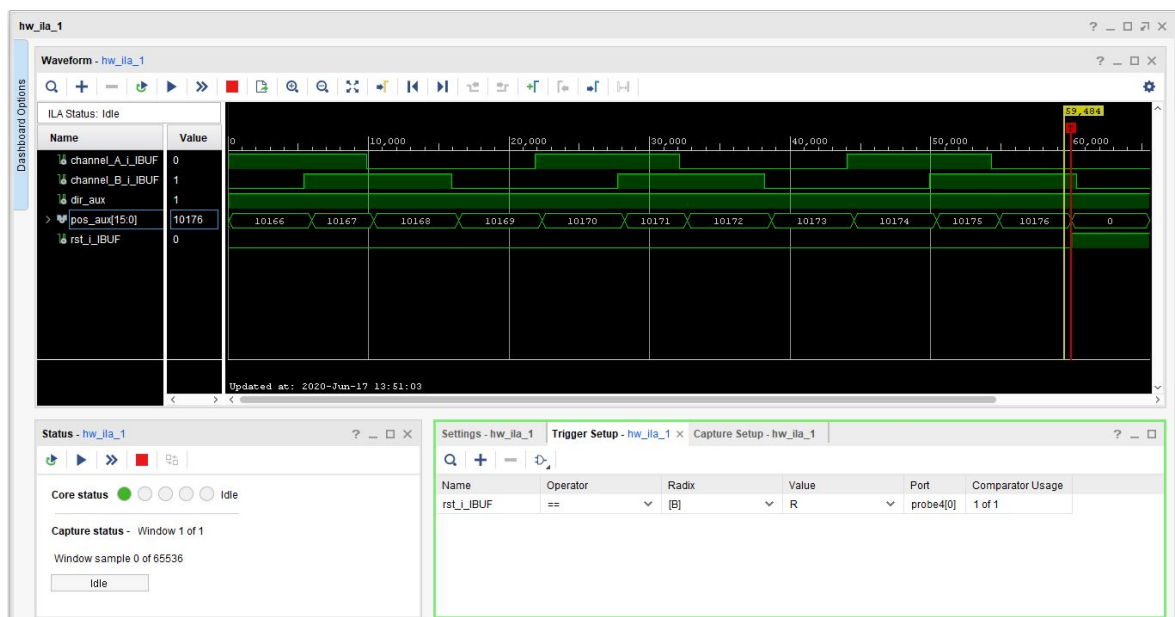


Anexo II - Depuración de Hardware

Trigger en 10 revoluciones (14400 pulsos)



Trigger en reset



Verificación de desfase

