

Diseño de un sistema de detección de coincidencias basado en
FPGA aplicado a la caracterización de inestabilidades en
osciladores locales

por

Elias Misael Reyes Iñiguez

Presentada al Departamento de Instrumentación
en cumplimiento parcial de los requisitos para obtener el grado de

Maestro en Ciencias

en la

Universidad Autónoma de Baja California

Facultad de Ingeniería

Maestría y doctorado en ciencias e ingeniería

Autor: Elias Misael Reyes

Facultad de Ingeniería

Asesor: Dr. Daniel Avalos González

Profesor de la Facultad de Ingeniería

Mexicali, Baja California, México. Diciembre de 2025.



COMITÉ DE TESIS

DIRECTOR DE TESIS

Dr. Daniel Avalos Gonzales

Profesor

Facultad Ingeniería

REVISORES

Co Director Tesis

Dr. Ramírez Zarate José Manuel

Facultad Ingeniería

Miembro comité

Dr. Hernández Daniel Balbuena

Facultad Electrónica

Miembro comité

Dr. Galaviz Yañez Guillermo

Facultad Electrónica

Miembro comité

Dr. Perez Perz Dalila Blanca

Facultad Electrónica

Diseño de un sistema de detección de coincidencias basado en FPGA aplicado a la caracterización de inestabilidades en osciladores locales

by

Elias Misael R.

Submitted to the departamento de instrumentación
on Mayo 21, 2025 in partial fulfillment of the requirements for the degree of

MAESTRO EN CIENCIAS

RESUMEN

Local oscillators are pivotal components in communication systems, significantly influencing the overall performance of both transmitters and receivers. However, noise sources can induce instabilities, making their measurement challenging due to the complex nature of these phenomena. Although there is substantial theoretical research in this field, the application of theory under specific conditions remains necessary. This research focuses on the hardware design implementation of pulse coincidence technique, providing experimental outcomes that can be used for characterizing local oscillators in the time domain. By employing mathematical static tools and finding estimators to enhance data processing and improve the reliability of local oscillator performance metrics.

Thesis supervisor: Dr. Daniel Avalos Gonzáles

Title: Profesor Facultad de ingeniería

Agradecimientos

les dedico a mis amados padres, Manuel Reyes y Lorena Iñiguez así como a mi futura esposa, Sarah Corona, a mis hermanos y amigos, este logro, ya que ellos siempre me han apoyado con las herramientas y la confianza, así como han sentado las bases para que ahora pueda tener el tiempo y los recursos para terminar un posgrado.

A mi tutor el Dr. Daniel Avalos González por invitarme a realizar un posgrado

Índice general

<i>Índice de figuras</i>	6
<i>Índice de cuadros</i>	8
Objetivos	9
Hipótesis	10
1 Introducción	11
1.1 Discusión sobre el ruido de fase	11
1.2 Dominio del tiempo y frecuencia	13
1.3 Detección de inestabilidad	14
1.3.1 Retos de medición	16
2 Marco Teórico	17
2.1 Herramientas estadísticas	17
2.2 Ley de funciones de ruido	17
2.2.1 Avar/Mvar	18
2.2.2 Estimación de tiempos muertos de medición	19
2.3 Medición de frecuencia	19
2.3.1 Métodos digitales de medición frecuencia	20
3 Coincidencia de pulsos	27
3.1 Método coincidencia de pulsos	27
3.1.1 Consideraciones para un sistema de medición real	29
3.1.2 Topología	31
3.1.3 Estimadores	33

4	Diseño de Hardware	36
4.1	FPGA	36
4.2	Flujo de diseño	37
4.3	Transferencia de datos	39
4.3.1	Transferencia de datos	40
4.4	Implementación a nivel de registro RTL	41
4.5	Verificación	42
5	Estimador de frecuencia	44
6	Desarrollo de prototipo en FPGA	50
6.1	Implementación	50
6.1.1	Consideraciones de diseño	51
6.1.2	Hardware implementado	53
6.2	Estimador en mediciones realizadas	63
7	Resultados	66
7.1	Frecuencia instantánea	73
7.2	Estabilidad	77
8	Conclusiones	79
A	Listado de Código	82
B	Tablas de resultados	85
B.0.1	Resultados Algoritmo propuesto	85
C	Extracto de coincidencias recolectadas	90
	<i>Referencias</i>	92

Índice de figuras

1.1	Ruido de fase caracterizado en oscilador comercial.	15
1.2	Desviación de oscilador caracterizada en oscilador comercial.	16
2.1	Principio de medición de tiempo.	19
2.2	Diagrama básico de conteo convencional. [25]	20
2.3	Resolución de contador convencional.	21
2.4	Resolución contador recíproco.	23
2.5	Diagrama de método vernier.	24
3.1	Representación gráfica del método de coincidencia de pulsos	28
3.2	Diagrama bloques Coincidencia de pulsos.	31
3.3	geometría de coincidencia de pulsos	35
4.1	Bloque lógico configurable.	37
4.2	Nios II flujo de diseño.	38
4.3	Conexión de datos para escritura.	39
4.4	Protocolo de comunicación Avalon.	40
4.5	Verificación de transmisión FIFO.	43
5.1	Histograma de fases θ de cada coincidencia.	45
5.2	Selección de coincidencias sin conocer T_x	48
5.3	Diagrama de flujo algoritmo.	49
6.1	Diagrama RTL.	54
6.2	Diagrama RTL del bloque de coincidencia de pulsos.	55

6.3	Diagrama RTL de diseño implementado en Platform designer.	56
6.4	Diagrama RTL de Bloque personalizado para platform designer.	57
6.5	Diagrama RTL detallado de Bloque personalizado para platform designer. . .	58
6.6	Maquina de estados.	59
6.7	Maquina de estados.	60
6.8	Diseño implementado en platform designer.	61
6.9	Diseño implementado en platform designer.	62
6.10	Ajuste de coincidencias.	64
7.1	Comparación de errores relativos entre estimadores.	68
7.2	Comparación de errores relativos entre estimadores enfoque.	69
7.3	Resultados gráficos de medición $t_x = 16.18034\mu\text{S}$	70
7.4	Resultados gráficos de medición 195.65nS	71
7.5	Resultados gráficos de medición $31.14159\mu\text{S}$	72
7.6	Comparación de medición de frecuencia con tiempo de muestreo.	74
7.7	Comparación de medición de frecuencia con tiempo de muestreo.	75
7.8	Calculo de ADEV y MDEV para señal ideal con ruido blanco.	77
7.9	Calculo de ADEV y MDEV para señal ideal con ruido blanco.	78

Índice de cuadros

3.1	Tabla de n_x y n_o de coincidencia de pulsos.	30
6.1	Entradas y salidas del sistema digital.	54
7.1	Periodos medidos	66
7.2	Comparación de estimadores: Algoritmo Capítulo 5 vs. Estimadores	67
7.3	Conteos de $n_o(i)$ reiniciados en tiempo T_s	76
B.1	Resultado del Algoritmo	86
B.2	Frecuencia vs. Tiempo - Segundo conjunto de resultados	87
B.3	Frecuencia vs. Tiempo - Tercer conjunto de resultados	87
B.4	Frecuencia vs. Tiempo - Cuarto conjunto de resultados	88
B.5	Frecuencia vs. Tiempo - Quinto conjunto de resultados	88
B.6	Frecuencia vs. Tiempo - Sexto conjunto de resultados	89
C.1	Coincidencias y diferencias Δn_x y Δn_o	91

Objetivos

Objetivo general

Diseñar un circuito en FPGA para medir frecuencias de señales eléctricas utilizando el método de detección de coincidencias y un algoritmo que permita validar la potencial aplicación del método en la caracterización de inestabilidades en osciladores.

Objetivos específicos

- Investigar el método de detección de coincidencias y su aplicación en la medición de frecuencias de señales eléctricas.
- Definir los requisitos y especificaciones técnicas en función de la aplicación propuesta.
- Cumplir con los requisitos de software y hardware para el uso de un FPGA.
- Validar el diseño mediante pruebas y experimentación.

Metas

- Diseñar un sistema en FPGA que aplica el método de detección de coincidencias.
- Desarrollar un criterio de procesamiento de datos.
- Solicitar registro de propiedad intelectual de los productos generados.
- Demostrar el potencial de aplicación del método en la caracterización de inestabilidades en osciladores para posibles trabajos futuros.

Hipótesis

A partir de un conjunto de pares de datos $n_0(i)$ y $n_x(i)$, con $i = 1, \dots, n$, obtenidos de cada coincidencia entre dos trenes de pulsos de diferente frecuencia, es posible generar un estimador que incremente la precisión en la medición de frecuencia y que permita detectar variaciones a lo largo del tiempo, a través de un diseño en FPGA que implemente el método de coincidencias de pulsos con el potencial de ser aplicado en caracterización de inestabilidades en osciladores.

Capítulo 1

Introducción

El análisis en frecuencia es importante para la mayoría de los dispositivos de radiofrecuencia que utilizan un oscilador local, el cual se implementa en receptores superheterodinos en los receptores de radio, televisión satelital, radiotelescopios, sintetizadores de frecuencia, transeptores de radiofrecuencia, sistemas de radar, etc. Los osciladores locales se han convertido en el corazón de dichos dispositivos electrónicos, los cuales son de alta demanda en la actualidad [1].

1.1 Ruido de fase en osciladores

El ruido de fase es una perturbación aleatoria que afecta una señal eléctrica; este es uno de los factores responsables de inestabilidades en los osciladores. Este fenómeno también es conocido como jitter cuando nos referimos a nuestro análisis en el dominio del tiempo y comúnmente denominamos a esta perturbación como ruido de fase cuando nos referimos a inestabilidades en el dominio de la frecuencia. Para facilitar el análisis de este fenómeno, podemos utilizar la descripción matemática de un oscilador simple. Este componente está diseñado para tener un resultado ideal representado por la ecuación 1.2; sin embargo, conocemos que un oscilador real tiene un ruido inherente; estas inestabilidades en la señal pueden ser representadas por la siguiente ecuación:

$$f(t) = A(t) \sin(2\pi\nu(t) + \phi(t)) \tag{1.1}$$

$$f(t) = A \sin(2\pi\nu t + \phi) \quad (1.2)$$

La ecuación representa 1.3 las fluctuaciones de fase y frecuencia en el tiempo como funciones en el dominio del tiempo. En comparación con la ecuación del oscilador ideal 1.2, el valor de la fase y la frecuencia se consideran constantes. Considerar la base y la frecuencia como constantes puede reducir el tiempo de análisis cuando se desea estimar el comportamiento de un sistema, ya sea por simulación o de manera analítica; sin embargo, no es precisamente correcto para todas las aplicaciones. Este fenómeno causa errores en los dispositivos electrónicos debido a su naturaleza impredecible o aleatoria. No podemos eliminar completamente este fenómeno, el cual puede estar relacionado incluso con efectos de materiales, impurezas de fabricación y variaciones de temperatura. La Regeneración espectral y el aumento de la tasa de error de bit son errores potenciales causados por el ruido de fase, en el caso específico de las comunicaciones electrónicas, debido al ruido presente en el oscilador local [2]. También existen perturbaciones aleatorias en la amplitud de las señales eléctricas, pero el ruido de fase es de particular interés porque los niveles no son lo suficientemente altos como para mitigar y controlar en los sistemas eléctricos actuales. Además, los errores causados por este fenómeno son difíciles de mitigar debido a la naturaleza aleatoria de estas fluctuaciones.

Afortunadamente, existen diferentes métodos para corregir este fenómeno, dependiendo de las aplicaciones. Por ejemplo, en la comunicación digital se puede agregar redundancia al código o elegir códigos con distancia de Hamming para incorporar la posibilidad de corrección de errores [3].

Dependiendo de la aplicación o de la precisión requerida, es necesario realizar un análisis intensivo de este fenómeno o agregar más detalles al modelo y a las herramientas matemáticas para el análisis.

1.2 Dominio del tiempo y frecuencia

El dominio de la frecuencia, ó también denominado espectro, es una transformación matemática de funciones altamente utilizada en el análisis de señales y sistemas. Esta transformación es denominada transformada de Fourier, la cual parte del desarrollo de series de Fourier para señales periódicas. Fourier fue un matemático y político francés que en 1801 y 1811 publico sus resultados de análisis armónico [4]. La motivación de su investigación era la transferencia de calor en materiales, pero en los siguientes siglos, su teoría se extendió a otras áreas del conocimiento, y el análisis en frecuencia o espectro ha sido profundamente utilizado en comunicaciones y en el análisis de señales eléctricas. Esta descomposición espectral permite a ingenieros y científicos observar propiedades de las señales desde otro punto de vista, en otro dominio. Se denomina descomposición espectral debido a que la señal se representa matemáticamente como una combinación de señales básicas, ya que estas solo contienen una frecuencia fundamental $\sin(\omega t)$.

En el dominio del tiempo se encuentran las señales que conjuntamente utilizamos matemáticamente de una función f con una variable independiente t y una variable dependiente $f(t)$, donde cada variable dependiente solo depende de una variable independiente. Clásicamente se han analizado señales periódicas y no periódicas, de las cuales podemos observar su comportamiento (frecuencia, amplitud, periodo, etc.) a lo largo del tiempo. Una de las formas más utilizadas para caracterizar señales ha sido en el dominio del tiempo. Para osciladores en el dominio del tiempo, tenemos parámetros importantes para caracterizar, como la frecuencia, el período, el jitter y el clock skew, los cuales pueden obtenerse mediante equipos de medición y diferentes métodos en el dominio del tiempo (sin utilizar la transformada de Fourier) [5-7].

1.3 Detección de inestabilidad

El dominio de la frecuencia ha sido ampliamente utilizado por ingenieros y científicos para caracterizar osciladores. En IEEE Std 802.3-2008 [8] se define el ruido de fase 1.3.

$$\mathcal{L}(f) = \frac{S_{\varphi}(f)}{2} \quad (1.3)$$

Donde $\frac{S_{\varphi}}{2}$ es la PSD unilateral de $\varphi(t)$. Las unidades de $\mathcal{L}(f)$ son radianes cuadrados por Hertz (rad^2/Hz) y también se utilizan comúnmente (dBc/Hz). El análisis espectral tiene un papel importante en la caracterización del ruido de fase, el cual provoca inestabilidades en el dominio del tiempo. Este parámetro, en la gráfica de PSD de un oscilador local, esperamos ver, idealmente, solo una función $\delta(t)$ en la frecuencia de oscilación; sin embargo, en realidad, diferentes factores repercuten en que la señal tenga contenidos de frecuencias alrededor de la frecuencia de oscilación. Como se muestra en la Figura 1.1, la cual pertenece a la hoja de datos de un oscilador de 10 Mhz NI-10M-3500 [9], en la Figura se puede observar como la mayor cantidad de ruido se encuentra cerca de la portadora $\mathcal{L}(1hz)$. Esto es de particular interés debido a que los problemas prácticos en gran cantidad de osciladores, por ejemplo, en comunicaciones, requieren una mayor optimización del ancho de banda BW , con un menor contenido de frecuencia alrededor del oscilador portadora, y de esta manera lograr un mejor empaquetado de canales adyacentes. De esta manera, caracterizar este parámetro funge un papel importante en sistemas prácticos, en la comercialización y utilización de dispositivos electrónicos.

La detección de inestabilidades de un oscilador local también ha sido propuesta en el dominio del tiempo cuando se toman mediciones de un fenómeno o una variable aleatoria. σ^2 es la herramienta matemática que representa la desviación o distancia de un conjunto de mediciones hacia la media μ . Al utilizar esta herramienta matemática para caracterizar osciladores locales, los investigadores se percataron de que, para algunos osciladores, el resultado no convergía; esto se debía al tipo de ruido interno del oscilador, que se presenta como un proceso estocástico de un movimiento aleatorio en particular[10]. Este es uno de los

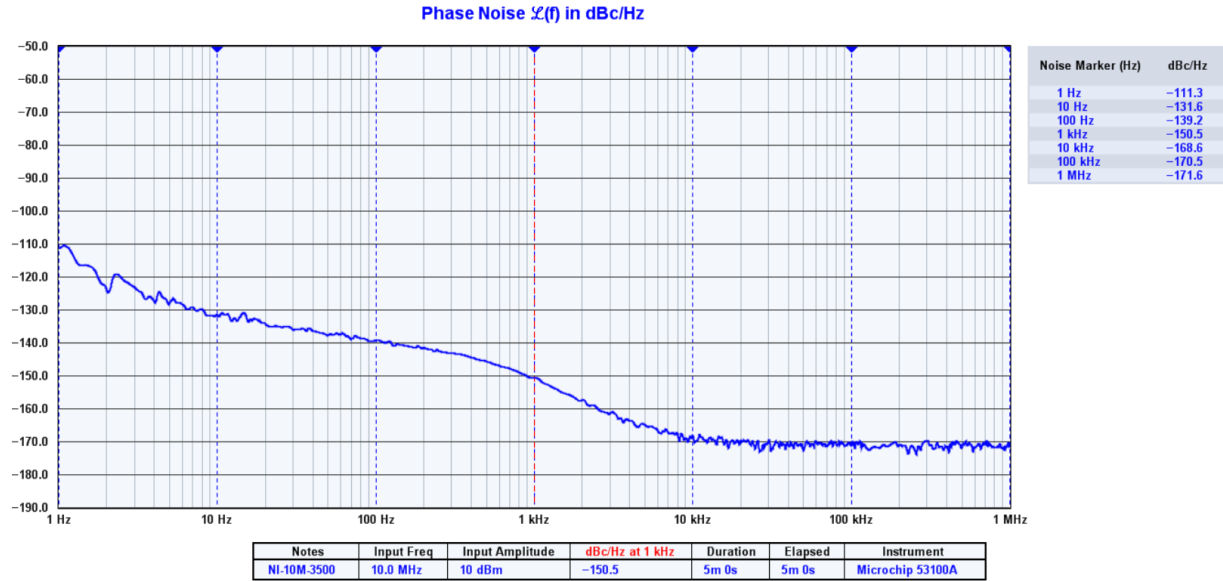


Figura 1.1: Ruido de fase caracterizado en oscilador comercial.

motivos por los cuales es común utilizar AVAR y MVAR para la caracterización de osciladores 1.4.

$$\sigma_y^2(\tau) = \frac{1}{2} \langle (\bar{y}_{n+1} - \bar{y}_n)^2 \rangle \quad (1.4)$$

Se puede observar cómo la variable independiente es τ , la cual representa diferentes tiempos de promediación. De esta gráfica se pueden obtener diferentes parámetros, por ejemplo, el tipo de ruido y la cuantización de este. Existen diferentes tipos de varianzas que son muy útiles y se pueden utilizar en diferentes casos. En la Figura 1.2 se muestra un ejemplo de una gráfica de AVAR para un oscilador comercial NI-10M-3500.

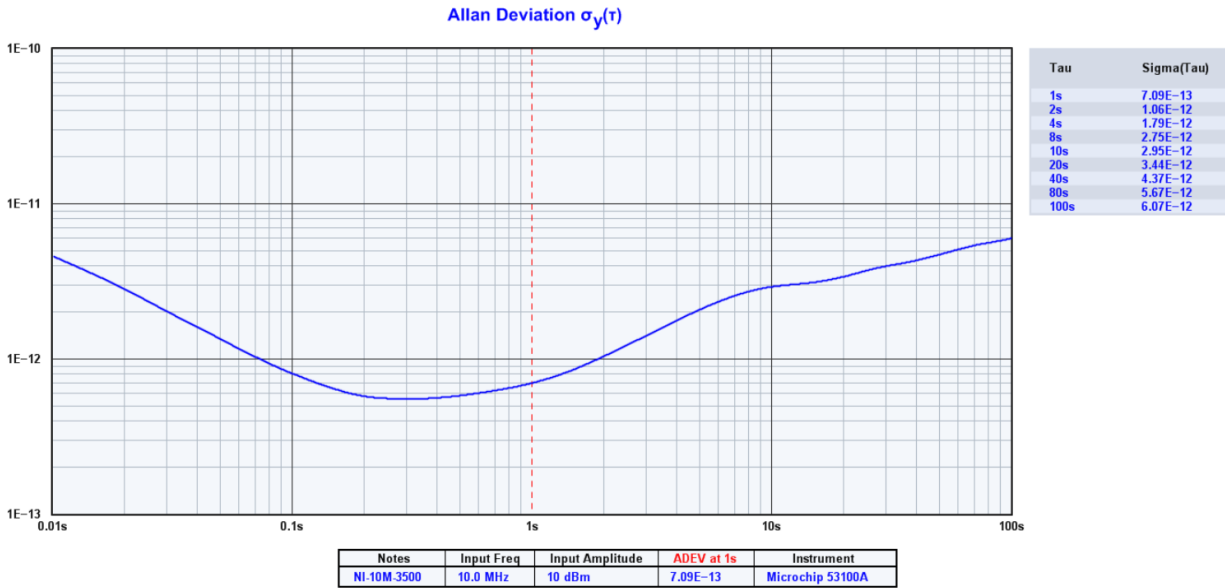


Figura 1.2: Desviación de oscilador caracterizada en oscilador comercial.

1.3.1 Descripción de los paradigmas

1. No se puede recuperar una PSD a través de AVAR. [11]
2. Con sistemas que utilizan ADC es necesario cumplir el requisito de nyquist para utilizar AVAR. [12]
3. La función de promediación $W(t)$ utilizada afecta la función de varianza. [13]
4. Una sobrepromediación puede sesgar el resultado de inestabilidad. [10]

Capítulo 2

Marco Teórico

2.1 Herramientas estadísticas

En 1966 se publicó el artículo "Statics of atomic Frequency Standards" [10], en el que se presenta un análisis de la estabilidad de osciladores atómicos, algo que en aquel tiempo se buscaba caracterizar para estimar su inestabilidad. D. Allan desarrolló un análisis para estos osciladores, resultando en que la desviación depende del tipo de ruido que afecta al sistema, la cantidad de muestras y el tiempo de medición. En este estudio se encontró que la varianza clásica diverge para tipos de ruido como el ruido blanco, desarrollando de esta manera la σ_{allan}^2 o también denominada σ^2 de dos muestras, y demostrando que converge para los tipos de ruido de interés en osciladores atómicos que se utilizan en los modelos $S_y \propto f^\alpha$ $-2 \leq \alpha \leq 2$.

También es común encontrar otras variaciones de la desviación de Allan o diferentes varianzas para la caracterización de osciladores, cada una con sus respectivas ventajas y aplicaciones.

2.2 Ley de funciones de ruido - Noise Law

Podemos estimar las fuentes de ruido a través de las leyes de potencia del ruido. Una motivación para utilizar AVAR/ADEV como un inicio es que converge para la mayoría de los tipos de ruido, mientras que la varianza clásica diverge.[14-17].

2.2.1 Avar/Mvar

Como se muestra en la ecuación 2.1, la varianza depende de los parámetros M , T y τ .

$$\sigma_y^2(M, T, \tau) = \frac{M}{M-1} \left(\frac{1}{M} \sum_{i=0}^{M-1} y_i^2 - \left[\frac{1}{M} \sum_{i=0}^{M-1} y_i \right]^2 \right) \quad (2.1)$$

Si hacemos $M = 2$ y $T = \tau$, entonces obtenemos la AVAR o también llamada varianza de dos muestras [18].

$$\sigma_y^2(\tau) = \langle \sigma_y^2(2, \tau, \tau) \rangle = \frac{1}{2} \langle (\bar{y}_{n+1} - \bar{y}_n)^2 \rangle = \frac{1}{2\tau^2} \langle (x_{n+2} - 2x_{n+1} + x_n)^2 \rangle \quad (2.2)$$

$$\sigma_y(\tau) = \sqrt{\sigma_y^2(\tau)} \quad (2.3)$$

La varianza de Allan mide la estabilidad de frecuencia de un reloj, calculando el promedio de los cuadrados de las diferencias entre lecturas sucesivas, y depende del periodo de muestreo τ . Una varianza baja indica buena estabilidad. La desviación de Allan, utilizada en gráficos y valores numéricos, refleja la estabilidad relativa y facilita la comparación entre fuentes.

Por ejemplo, una desviación de Allan de $1,3 \times 10^{-9}$ a $\tau = 1$ s indica una inestabilidad RMS de $1,3 \times 10^{-9}$, equivalente a 13 mHz para un reloj de 10 MHz. Para la estabilidad de fase, se utilizan variantes de desviación en el tiempo [19].

La varianza de Allan modificada surge para resolver la limitación de la varianza de Allan tradicional, la cual no puede distinguir entre la modulación de fase blanca y la modulación de fase parpadeante (flicker). Para superar este problema, se aplica un filtrado algorítmico que reduce el ancho de banda en un factor n . Esta modificación altera la definición y los estimadores originales, dando lugar a una nueva categoría conocida como varianza de Allan modificada. Al igual que la varianza de Allan convencional, esta medida se emplea para evaluar la estabilidad en frecuencia de una señal [20].

2.2.2 Estimación de tiempos muertos de medición

En [21], se analiza que, para un Sistema de medición basado en (Start/Stop), siempre existirán tiempos muertos en la medición. se propone utilizar el enfoque mostrado en [22], utilizando funciones de bias para corregir el sesgo generado por los tiempos muertos entre muestras.

2.3 Estimación de frecuencia

Los sistemas de medición de frecuencia que presentan su resultado de manera digital también se denominan TDC (time to digital converter) y se utilizan para la estimación de frecuencia y la conversión a digital la Figura 2.1 muestra el diagrama básico de medición de un intervalo de tiempo. Podemos definir la frecuencia de una señal periódica como el número de ciclos que ocurren en un intervalo de tiempo, comúnmente un segundo. Estimar este valor ha tomado relevancia en el campo de la electrónica debido a que muchos de sus sistemas se basan en el funcionamiento de señales periódicas; gran parte de los dispositivos electrónicos dependen de su operación mediante ciclos de reloj. Sistemas de modulación y reloj; estas son las propiedades en el dominio de la frecuencia para lograr la transmisión y recepción de información de manera inalámbrica. Esto fue un gran avance, el cual nos permitió estar altamente comunicados en la actualidad [23].

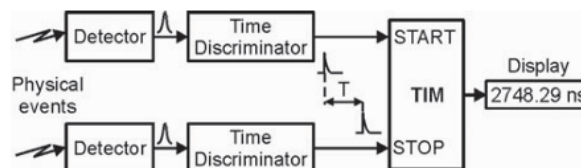


Figura 2.1: Principio de medición de tiempo.

2.3.1 Métodos digitales de medición frecuencia

La tecnología digital permitió el uso de técnicas de medición de frecuencia basadas en conteos; esta también facilita el uso de un procesador digital que puede realizar cálculos y utiliza hardware dedicado para las tareas de alto desempeño. Estas técnicas han evolucionado y se han vuelto más complejas, lo que permite incrementar la resolución, la precisión y la confiabilidad [24].

Conteo convencional

En la Figura 2.1 se muestra el diagrama básico de un contador de pulsos convencional y los componentes principales que lo caracterizan. La señal de entrada se condiciona para que sea compatible con la tecnología interna del contador; después, la frecuencia se introduce en la entrada de una compuerta AND principal.

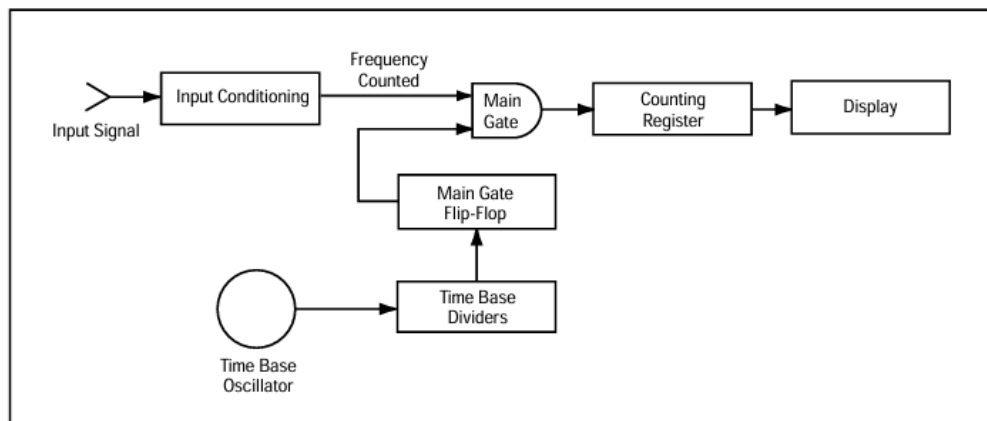


Figura 2.2: Diagrama básico de conteo convencional. [25]

Luego se muestra una señal de referencia, la cual puede ser generada por un oscilador o un patrón conocido para tener una mayor precisión; a esta señal se le aplica un divisor para generar la ventana de tiempo deseada; por lo regular, se utiliza un segundo. Como esta señal comúnmente es de corriente alterna, se hace pasar por una compuerta flip-flop para convertirla en una señal que pueda ser procesada por nuestra compuerta principal, que regularmente es una onda cuadrada. Las coincidencias de los pulsos se producen en esta compuerta principal; la salida se registra en un contador latch que se utiliza para mantener

la señal en el display hasta que cambie la frecuencia.

Como sabemos, la referencia de tiempo o de ventana es fácil de calcular; la frecuencia se calcularía con esta simple fracción $f = \frac{n}{t}$ [25]. Por ejemplo, si el tiempo de ventana para la compuerta principal es de un segundo y los pulsos contados en n son 5, tendríamos una señal de $5Hz$. Las desventajas de este método son que es necesario aumentar la ventana de tiempo de medición para obtener una mejor resolución; la incertidumbre es de ± 1 conteo en la entrada, por lo cual, a bajas y medias frecuencias, no se obtiene una buena precisión. La Figura 2.3 muestra la cantidad de números de resolución en función de la ventana de medición.

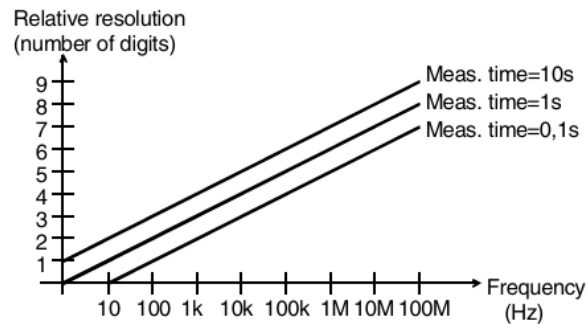


Figura 2.3: Resolución de contador convencional.

Los siguientes errores se presentan en los contadores convencionales al realizar el conteo y en la base de tiempo.

- Error de ± 1 conteo

Dado que la señal de entrada no se encuentra sincronizada con el patrón de referencia, puede ocurrir que el tiempo de medición en compuerta, cuando esté activo para obtener los conteos, se pierda un conteo debido al desfase en las señales, como se muestra en el caso número 1 de la Figura 2.3. Este error de conteo es crítico a bajas frecuencias debido a que se tienen menos conteos en el intervalo de tiempo deseado, y perder un conteo puede representar un gran error. Por eso, esta técnica de conteo es más confiable a altas frecuencias, siendo limitada por la frecuencia de corte de la tecnología digital que se está utilizando.

- Error oscilador

Cualquier error que se presente en el oscilador del contador de frecuencia resultará en un error que se acumulara en nuestra medición. este error es causado por la desviación que pueda llegar a existir entre el valor nominal del oscilador y la frecuencia que genere. existen diversas fuentes que causan un efecto negativo en nuestro patrón [26].

- Error en la medición de frecuencia

El error total en la medición se encuentra en la suma del error de conteo más el error en el oscilador y se muestra en la ecuación $\frac{\Delta f}{f} = \frac{\pm 1}{f_i}$ En la ecuación, podemos observar que conforme aumenta la frecuencia de entrada, el error relativo se reduce a \pm un conteo; el error se expresa en partes por millón.

Conteo recíproco

El contador recíproco es también conocido como un método de medición de periodo; consiste en la medición de N ciclos de periodo de entrada en un tiempo MT (Measurement time), donde M es la cantidad de conteos de pulsos de reloj y T es el tiempo de un ciclo de reloj. de esta manera, a diferencia del contador convencional, el tiempo MT está sincronizado con la señal de entrada, aboliendo de esta forma el error de ± 1 conteo de ciclos de entrada del contador convencional. Ahora, el error de medición se encuentra en ± 1 ancho del pulso de reloj.

La ecuación 2.4 muestra la relación entre los conteos registrados y la frecuencia de entrada. a diferencia del contador convencional, este cuenta con dos contadores y un microcontrolador para realizar las operaciones matemáticas [21].

$$\text{frequency} = \frac{\text{Conteo de ciclos de entrada}}{(\text{Conteo de referencia}) \times t_c} = \frac{N}{MT} \quad (2.4)$$

La resolución es mayor aún en bajas frecuencias, a diferencia del contador convencional. la Resolución relativa se muestra en la Figura 6, donde podemos observar que esta depende del tiempo de medición y se puede obtener una mejor resolución a bajas frecuencias. el rendimiento de este tipo de contadores se puede lograr aumentando la frecuencia de referencia.

Los tipos de errores que se presentan en este tipo de contadores están relacionados con el inicio y el fin de la medición en la estimación del periodo; además, existen errores de medición relacionados con incertidumbres, los cuales afectan a cualquier dispositivo real.

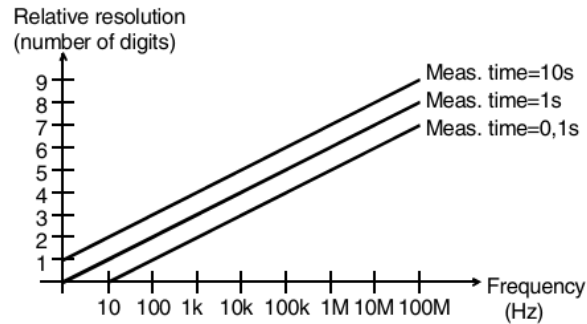


Figura 2.4: Resolución contador recíproco.

- Error de disparo

Este error es causado por el ruido en la señal de entrada y en los canales del contador. Este error puede ocasionar que la compuerta principal encargada de realizar las coincidencias tarde o se accione demasiado pronto. Este es un error aleatorio que ocasiona que la compuerta se accione en el momento incorrecto.

- Error en la medición de periodo

El contador recíproco es un método de medición de periodo; posteriormente, se puede convertir a frecuencia según se requiera. Como se observa en la tabla 1, los errores en la medición se encuentran en el de ± 1 conteo, como en el contador convencional, ya que son inherentes a los métodos de conteo. También se le puede añadir el error del tiempo base o oscilador, al estar usando este como referencia de tiempo; la precisión o imprecisión de este se verá reflejada en el resultado final.

Asumiendo que el error en el oscilador y el de disparo son despreciables, podríamos lograr esto utilizando un sistema de referencia más preciso, como un oscilador atómico, y reduciendo el ruido de entrada. Solo tendríamos el error de conteo o cuantización, mostrado en la ecuación (3) para un contador convencional y en la ecuación (5) para un contador convencional. Si utilizamos un oscilador de 10MHz y un tiempo de medición de un segundo.

Método Vernier

El esquema funcional que ilustra la medición de tiempo mediante el método de Vernier se presenta en la 2.5. Al inicio del proceso, el flip-flop START se configura en estado de activación (*set*). Simultáneamente, tanto el contador principal CTR1 como el contador Vernier CTR2 se inicializan en cero.

En el instante t_A , el flip-flop START cambia a estado *set*, lo que habilita la compuerta CC y permite que el contador principal comience a registrar los pulsos generados por el oscilador principal. De manera análoga, en el momento t_B , el contador Vernier inicia la contabilización de los pulsos provenientes del oscilador Vernier.

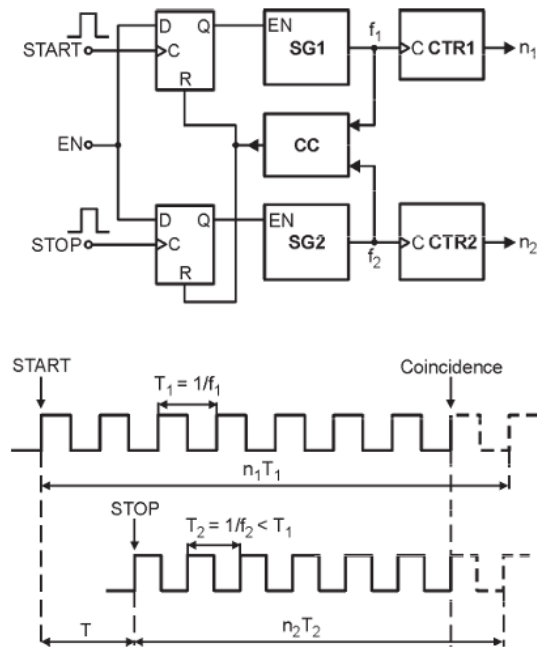


Figura 2.5: Diagrama de método vernier.

Cuando ocurre una coincidencia entre los pulsos de ambos osciladores, el detector de coincidencias emite una señal de reinicio (*reset*) a ambos flip-flops. Esta acción provoca que ambos contadores detengan su operación.

Si se consideran los periodos T_1 y T_2 correspondientes al oscilador principal y al oscilador Vernier, respectivamente, junto con los conteos n_1 y n_2 de cada contador, el intervalo de tiempo T entre los eventos t_A y t_B puede determinarse mediante la siguiente expresión [27]:

$$T = n_1 T_1 - n_2 T_2 \quad (2.5)$$

Una limitación de este método es que el tiempo de medición excede la duración del evento que se desea registrar [28].

Conteo recíproco por interpolación

Una limitación de los contadores convencionales y recíprocos se puede encontrar en la resolución del dispositivo; si se busca solucionar este problema, se enfoca en que se puede utilizar la interpolación, la cual busca medir fracciones de frecuencia, debido a que los conteos solo pueden ser de períodos completos. La interpolación se puede lograr de manera digital y analógica.

En la interpolación analógica, comúnmente se utiliza un capacitor, el cual se carga durante el tiempo fraccional de un pulso que se desea interpolar; de esta manera, el tiempo transcurrido se puede medir y sumar al del conteo de los ciclos completos. Esta es una de las maneras más básicas de implementar la interpolación; existen diversas maneras y enfoques para esta problemática de manera digital y analógica [29].

Regresión lineal

Una de las técnicas más recientes, en comparación con las anteriores, es el uso de regresión lineal para la estimación de frecuencia; la ventaja de esta implementación es que no sería necesario un disparador de medición de inicio y final, es decir, mediciones continuas. Esto se logra mediante estampas y estadísticas, por medio de las cuales se puede estimar la frecuencia. Conociendo el número de conteos y el tiempo en el que se secuencia desde un punto de referencia, se puede implementar una regresión lineal. Este método ha demostrado buenos resultados y ha realizado mediciones continuas. La desventaja es que necesita más recursos de procesamiento, por lo que su implementación en tiempo real puede ser más complicada [2]. Algunos contadores de frecuencia profesionales ya implementan esta función como una característica especial de medición, como el CNT-90 [30], lo que proporciona versatilidad al tener más inferencia en la forma en que se realiza el proceso de medición de frecuencia. Esto es importante para aplicaciones como la caracterización de osciladores, donde el método de promediado y la obtención de datos juegan un papel crucial para utilizar herramientas estadísticas como AVAR.

Capítulo 3

Coincidencia de pulsos

La medición de la frecuencia desempeña un papel importante en la instrumentación y en la medición de magnitudes físicas, así como en aplicaciones en instrumentación, aeroespacial, automotriz, médico, etc. Esto se debe a que la frecuencia es un parámetro fundamental en la ciencia y la ingeniería para describir los fenómenos oscilatorios que se encuentran en diferentes ámbitos, lo que resalta la importancia de la medición de la frecuencia. Este capítulo presenta una revisión del método de coincidencia de pulsos, sus aplicaciones, limitaciones y diferentes enfoques a lo largo del tiempo, así como sus desafíos e implicaciones prácticas.

3.1 Método coincidencia de pulsos

El método de coincidencia de pulsos fue presentado en 1975 [31] es una técnica para estimar una frecuencia desconocida f_x . Esta señal tiene un período T_x , y la frecuencia desconocida f_x se estima indirectamente a partir del período de su señal. Esto es posible mediante la adición de una señal de referencia S_o con período $T_o = \frac{1}{f_o}$. Este método se basa en contar el número de períodos de dos señales, lo cual puede llevarse a cabo en el flanco de subida o de bajada. En este punto, estas señales se consideran ondas cuadradas. Definimos estos conteos como n_x y n_o .

Es interesante la relación entre estos conteos: si realizamos la multiplicación $T_x \cdot n_x$ y la comparamos con $T_o \cdot n_o$, notamos que estas magnitudes son aproximadamente iguales en el punto de coincidencia de los dos pulsos, como se muestra en la Figura 3.1.

En el instante de coincidencia de pulsos i , se cumple:

$$T_x \cdot n_x(i) = T_o \cdot n_o(i) + e(i)$$

donde $e(i)$ representa el error entre cada aproximación.

Despejando T_x :

$$T_x = \frac{n_o(i)}{n_x(i)} \cdot T_o + \frac{e(i)}{n_x(i)} \quad (3.1)$$

De la ecuación se puede observar que, a medida que $n_x(i)$ tiende a infinito, el error $e(i)$ converge a 0 de esta forma $T_X = \frac{n_o(i)}{n_x(i)} \cdot T_o$. sin embargo, esto requiere incrementar el tiempo de medición M_t a un tiempo infinito de coincidencias, lo cual no es realizable en la practica, por esta razón, se han implementado diferentes estimadores, procesando datos obtenidos o utilizando relaciones matemáticas de Teoría de números. Comprendiendo el fenómeno, se ha logrado reducir el tiempo de medición y disminuir la probabilidad de error. La forma de lograr esto no es trivial, ya que el posicionamiento de las mejores coincidencias no siempre se encuentra al final de la medición, lo cual convierte este método de medición en un tema atractivo de análisis e investigación.

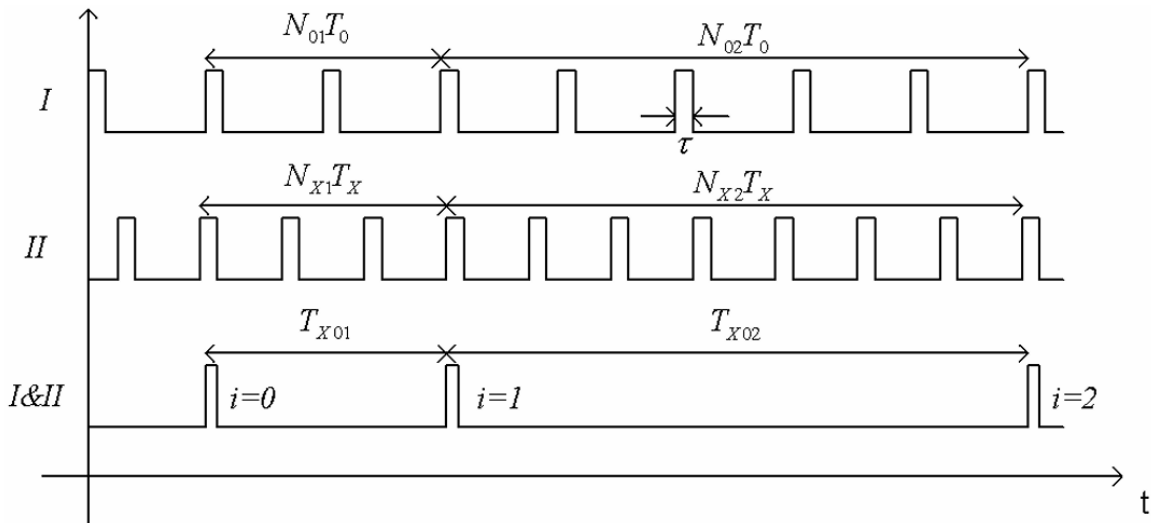


Figura 3.1: Representación gráfica del método de coincidencia de pulsos

En la tabla 3.1 se observa un ejemplo de las coincidencias de pulsos registradas por

este método; los números que se obtienen de cada conteo y las diferencias entre ellos se presentan en la última y penúltima columna. Estos números enteros contienen información acerca de ambas señales S_x y S_o . Se espera que una de ellas esté muy bien caracterizada y sea estable; por lo regular S_o podemos realizar estimaciones sobre el periodo y la frecuencia de S_x simplemente con estos números. Todos estos tienen una relación entre sí, y dentro de todos ellos es necesario realizar una selección sobre cuáles se utilizarán o como se utilizarán para generar un resultado. El tiempo de procesamiento de los números que se escojan, así como cómo fueron obtenidos por el hardware y el ancho de pulso, son factores que contribuyen a un resultado de mayor o menor calidad. En este caso, se muestran las primeras 20 coincidencias, pero en un segundo se pueden tener miles de datos con señales en el orden de los Mhz esto indica que, si se requiere un sistema en tiempo real, sea cual sea el umbral de tiempo para cada aplicación en específico, se necesita procesar los datos de manera eficiente.

3.1.1 Consideraciones para un sistema de medición real

- Ambas señales ocurren en tiempo continuo y se necesita una señal de inicio y de fin de medición para iniciar y detener el tiempo de medición. Un enfoque de modo continuo también se puede aplicar [32] o se puede utilizar regresión lineal [2]. La diferencia de tiempo entre el tiempo de inicio y final se define como tiempo de promediación τ y M_t como el tiempo total de medición
- El ancho de pulso τ_x y frecuencia f_x son desconocidos, así como la señal que se desea medir o caracterizar. Si deseamos conocer sus características sin influir en ellas.
- La señal de referencia S_o debe ser lo más estable posible, con ancho de pulso τ_o conocido se pretende utilizar valores de osciladores comerciales (i.e., 10 Mhz).

Cuando diseñamos un sistema de medición, podemos estar interesados en los siguientes parámetros técnicos:

- **Rango de Medición (MR, por sus siglas en inglés):** Se refiere al tiempo durante el cual el sistema puede medir un fenómeno de manera continua sin detenerse. Existen dos categorías de convertidores tiempo-digital (TDC): de rango corto y de rango

Índice (i)	$n_x(i)$	$n_o(i)$	Δn_{xi}	Δn_{xo}
0	0	0	0	0
1	1	146	1	146
2	2	308	1	162
3	3	470	1	162
4	6	955	3	485
5	7	1117	1	162
6	8	1279	1	162
7	11	1764	3	485
8	12	1926	1	162
9	13	2088	1	162
10	16	2573	3	485
11	17	2735	1	162
12	18	2897	1	162
13	19	3059	1	162
14	21	3382	2	323
15	22	3544	1	162
16	23	3706	1	162
17	24	3868	1	162
18	26	4191	2	323

Tabla 3.1: Tabla de n_x y n_o de coincidencia de pulsos.

amplio. El método de coincidencia de pulsos pertenece a la segunda categoría, ya que los contadores pueden funcionar indefinidamente, con la única limitación de un desbordamiento (*overflow*). Sin embargo, este problema puede resolverse utilizando el bit de acarreo como una señal para almacenar el último valor contado y compensar la suma.

- **Incertidumbre estándar de medición o error aleatorio** (s): Representa la variación estadística en las mediciones debido al ruido inherente del sistema o a fluctuaciones.
- **Tiempo muerto** (T_d): Se refiere al tiempo entre el final de una medición y el inicio de la siguiente [33].

3.1.2 Topología

El sistema de medición de frecuencia está compuesto por elementos simples: dos contadores de período, un detector de coincidencia y un registro para almacenar el par de números enteros $n_x(i)$, $n_o(i)$. La acción de estos elementos puede llevarse a cabo mediante componentes digitales, ya que el enfoque digital de la coincidencia de pulsos es el más común para su implementación. Este trabajo se centra en los enfoques digitales de la coincidencia de pulsos.

La Figura 2 muestra un diagrama de implementación digital. Los contadores binarios actúan como contadores de período, la puerta AND funciona como detector de coincidencia y el elemento de memoria es un registro digital. En este modelo, puede ser útil agregar señales de inicio y detención. Podemos seleccionar estos parámetros, pero es conveniente realizar un análisis profundo para elegir adecuadamente la coincidencia de inicio y detención [34], con el fin de mejorar la precisión del sistema.

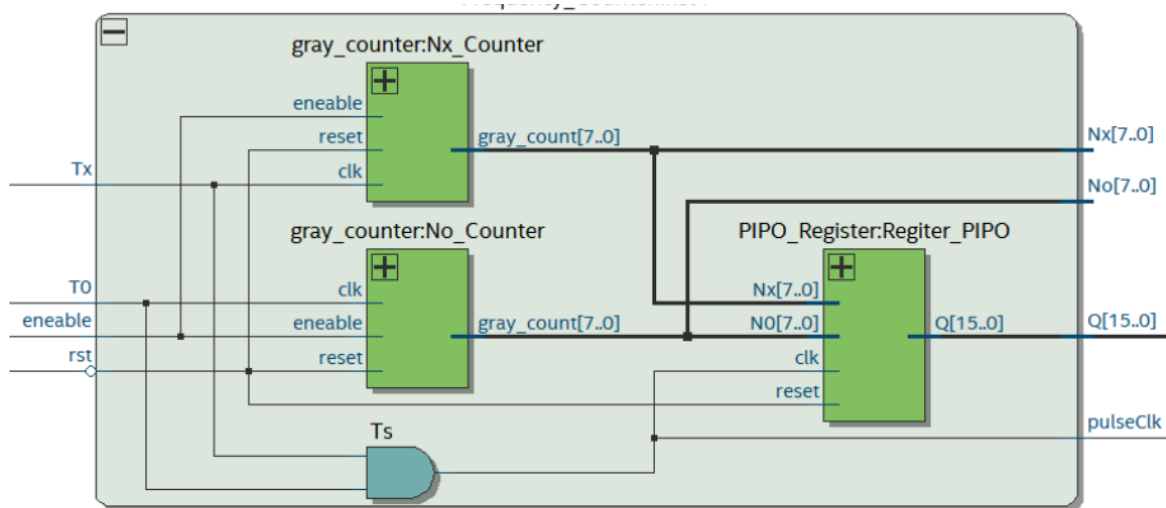


Figura 3.2: Diagrama bloques Coincidencia de pulsos.

Error de medición

El error de medición del método está altamente relacionado con el ancho de pulso de ambas señales. Debido a que en cada coincidencia lo que se registra son los flancos de subida o bajada de cada pulso cuadrado, el remanente de tiempo es considerado un error $e(i)$ mostrado en la ecuación 3.1. Ese error se define con un máximo de 2τ si ambos anchos de pulso son iguales [35]. Una coincidencia perfecta o ideal sería donde ambas señales tienen ambos flancos de subida o bajada exactamente al mismo tiempo. Esto, en sí, puede parecer el mejor caso, pero puede ocasionar que el detector de coincidencia no se active debido al ancho de pulso tan pequeño generado y que, por esta razón, no se capture el par de datos $n_x n_o$. Por eso, cada problema en la coincidencia de pulsos no suele ser trivial, y existen diferentes enfoques para resolver cada uno de ellos desde diversos puntos de vista y aplicaciones.

Errores sistemáticos en la medición: La estimación de la frecuencia nunca puede representarse como el cociente exacto de dos enteros, lo que implica una incertidumbre inherente en la medición [36-38].

Coincidencias arbitrarias en la conjunción de trenes de impulsos: La posición de los impulsos en el tiempo es aleatoria, lo que genera una secuencia infinita de coincidencias y puede afectar la estabilidad de la medición [32].

Dependencia de la duración de los pulsos y el tiempo de medición: La precisión del método está limitada por la duración de los pulsos [39].

3.1.3 Estimadores por software

Existen en la literatura diversos estimadores para el método de coincidencias de pulsos. definimos que son estimadores de software debido a que el procesamiento de las coincidencias se realiza en una computadora mediante software secuencial (i.e. MATLAB) después de la captura de las coincidencias; su contraparte serían los estimadores por hardware, que realizan las operaciones o decisiones con circuitos físicos, ya sean digitales o analógicos. El uso de estos estimadores permite mejorar la medición en diferentes parámetros o proponer una condición de selección de coincidencias; cada estimador puede tener sus mejoras, así como sus compromisos, lo que los hace más efectivos en diversas aplicaciones prácticas.

Aproximaciones racionales

En el método de coincidencia de pulsos se han investigado estimadores y como estos pueden mejorar la medición sin cambiar el hardware utilizado. uno de ellos son las aproximaciones racionales [32]. se encuentra que existe una relación entre la coincidencia de pulsos y el algoritmo de elucides $\frac{T_x}{T_o} = \{a_0, a_1, a_2, \dots\}$, donde los coeficientes representan las fracciones continuadas. se explica que, por medio de dos números enteros, no se puede obtener uno racional exactamente, pero podemos usar mediantas de α ($a = f_x/f_0$), para el caso de coincidencia de pulsos, podemos utilizar $\frac{\sum_m P_n}{\sum_m Q_n}$. La nomenclatura P_n y Q_n puede ser confundida con los valores de coincidencias $n_x(i)$ $n_o(i)$ pero en aproximaciones racionales P_n y Q_n hacen relación a las mediantas que se pueden producir en el proceso de coincidencia de pulsos

Cuando se presentan dos coincidencias, se genera una fracción o medianta, y se cumple la condición establecida en la Ecuación 3.2

$$\frac{P_{n+i}}{Q_{n+i}} < \frac{P_{n+i} + P_n}{Q_{n+i} + Q_n} < \frac{P_n}{Q_n}, \quad (3.2)$$

P_n y Q_n representan conteos independientes de pulsos desconocidos y de referencia dentro del intervalo de tiempo considerado. El índice n indica el orden de la fracción, mientras que i es un número entero positivo. Específicamente, los valores P_{n+i} y Q_{n+i} aparecen tras una iteración posterior a P_n y Q_n ; el valor $n + i$ se incrementa como resultado de la medición

temporal hasta alcanzar un número m , en el cual se obtiene la mejor aproximación. Esta propiedad es ampliamente reconocida en la meteorología clásica: al aumentar el tiempo de medición, mejora la precisión del resultado [40].

Análisis Geométrico

En este acercamiento, con el método de coincidencia de pulsos, se puede obtener una reducción en el tiempo de medición utilizando la geometría de la coincidencia de las señales S_o y S_x cuando son procesadas a través de un detector de coincidencias; este algoritmo se ha propuesto en [41]. El análisis geométrico realizado permite identificar las condiciones bajo las cuales se producen coincidencias ideales y parciales, y cómo estos eventos dependen de parámetros como el ancho de pulso y el máximo común divisor de los periodos. Se demuestra que, al ajustar estos parámetros, es posible reducir significativamente el tiempo de medición sin sacrificar la precisión, lo que representa una mejora sustancial respecto a los métodos existentes.

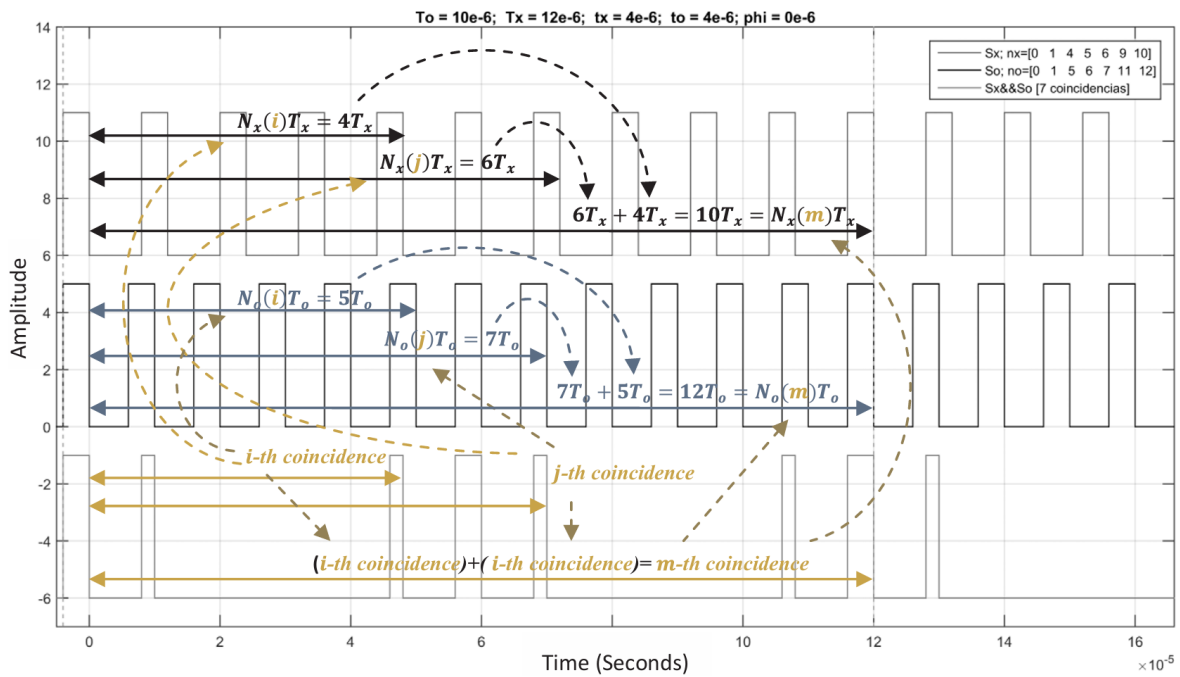


Figura 3.3: geometría de coincidencia de pulsos

Capítulo 4

Diseño de Hardware

En la actualidad, existen diferentes tecnologías en las que se puede implementar un diseño digital. En el primer acercamiento dentro del desarrollo de circuitos integrados, las compañías ofrecen sus productos de THT (through hole, traducir) de manera que los ingenieros o desarrolladores pueden implementarlos en tableros de prueba. Sin embargo, esto requiere de un gran esfuerzo y demasiada obra manual. Desde la época de los años 80 se han venido desarrollando Dispositivos integrados reprogramables con la finalidad de implementar diseños digitales de manera mas rápida y eficiente, que permitan reducir los tiempos de implementación y desarrollo. estos son llamados dispositivos lógicos programables (PLDs). Estos dispositivos iniciaron en 1960, cuando texas instruments desarrollo el circuito integrado TMS2000, el cual era una memoria reprogramable de solo lectura: PAL, PLA, GALs, CPLD y FPGA [42].

4.1 FPGA

Xilinx fue la primera empresa en implementar una tecnología de campos lógicos programables (FPGA) en 1985 [43]. esta consiste en la interconexión de un CLB (Control Unit Logic), que pueden representar un elemento lógico y/o combinacional por medio de una tabla de búsqueda (look-up table) y un flip-flop. Los diseños actuales son más complejos; en la Figura 4.1 se muestra el diagrama de un CLB de estos, los cuales están interconectados. Un programa de síntesis se encarga de optimizar el diseño con compuertas para utilizar la menor cantidad de recursos y evitar errores de diseño. En este punto, es importante conocer los lenguajes

de descripción de hardware (HDL) y los métodos de programación que son sintetizables; es decir, se refiere a que pueden ser traducidos a un nivel de registros y compuertas lógicas.

A lo largo del tiempo, diferentes empresas se han sumado a integrar sus productos de compuertas lógicas programables para diversas aplicaciones, las cuales van desde el desarrollo de pruebas y la implementación en algoritmos hasta el procesamiento de imágenes y la investigación.

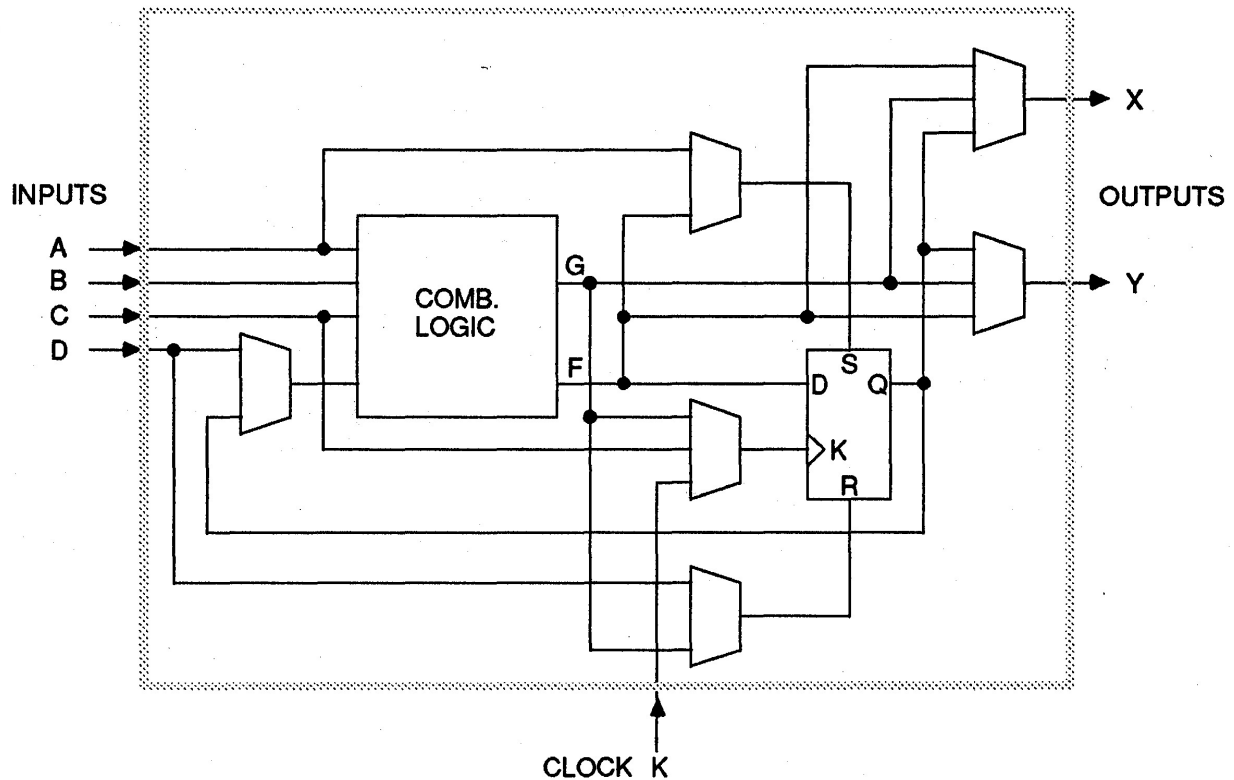


Figura 4.1: Bloque lógico configurable.

4.2 Flujo de diseño

El flujo de diseño de hardware en plataformas FPGA tiene un flujo de implementación como se muestra en 4.2. cada proveedor de plataforma de desarrollo provee un software con las herramientas necesarias para poder implementar el diseño mostrado y ofrece diferentes soluciones de diseño para implementar una cantidad diversa de aplicaciones en hardware. El proceso de diseño mostrado en la Figura 4.2 es específico para la utilización de un procesador

y una lógica de usuario personalizada. Este procesador se utiliza para la gestión de datos hacia otros periféricos, como la RAM DDR3, puertos de pruebas como JTAG, y transferencias de datos hacia una computadora de propósito general.

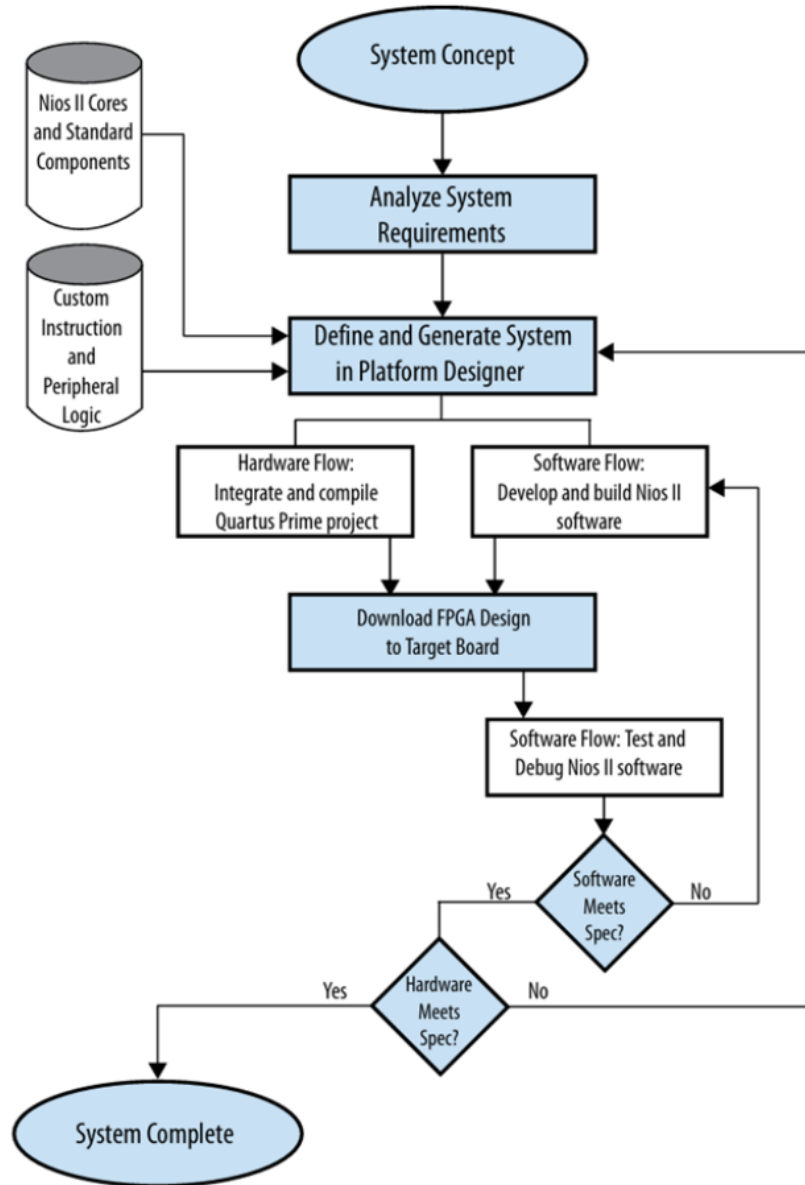


Figura 4.2: Nios II flujo de diseño.

4.3 Transferencia de datos

Es necesario un flujo de datos de alta velocidad para no sobrepasar la memoria limitada dentro del FPGA. estos bloques nos permiten almacenar información rápidamente, pero son recursos limitados. es necesario el uso de un elemento de memoria externo, como una memoria flash, EEPROM, RAM, ROM, etc., donde contener nuestros datos ya procesados o listos para procesar. en este caso, se utilizó el bus de datos Avalon, que nos permite leer y escribir información entre diferentes periféricos. la Figura 4.3 muestra el diagrama a bloques de cómo se diseñó este elemento. Por la lógica de usuario, los datos se transfieren a una memoria FIFO, y el bus de datos Avalon genera las señales para escribir los datos en otros elementos. esto se realiza por medio de direcciones; por esta razón, se menciona que es una transferencia de memoria mapeada. Esto quiere decir que accedemos a registros de memoria de otros componentes para leer o escribir en ellos y poder implementar diferentes operaciones.

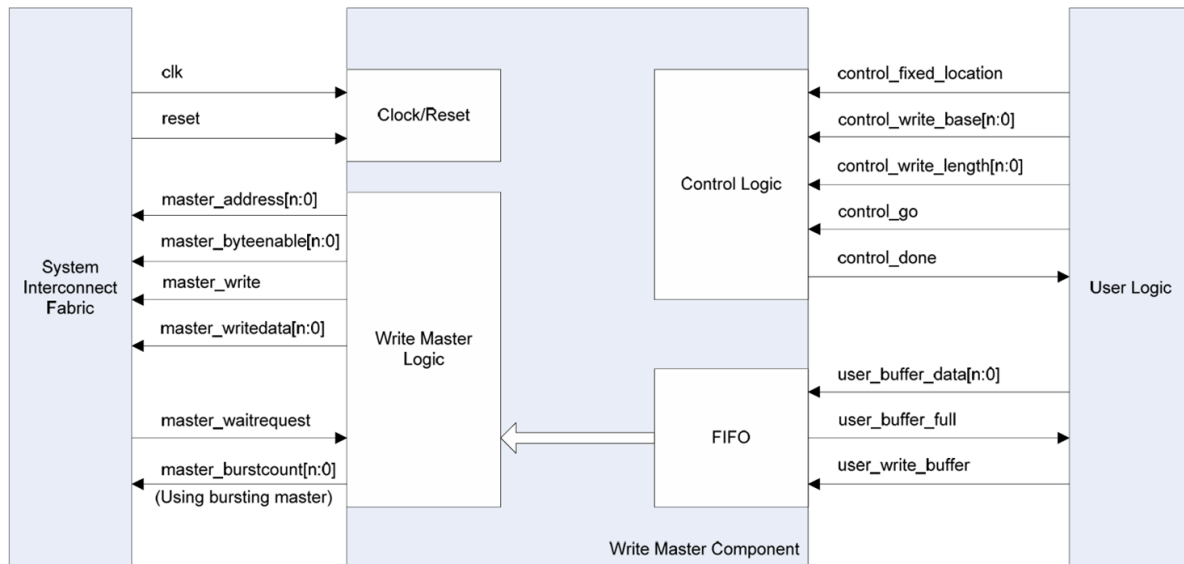


Figura 4.3: Conexión de datos para escritura.

4.3.1 Protocolo de comunicación en silicio

Los diseños digitales comúnmente se encuentran diseñados por bloques que cumplen una tarea específica. Estos bloques digitales realizan tareas tales como leer una memoria ROM, escribir en una memoria RAM, controlar interrupciones de un procesador, etc. Estos bloques tienen que comunicarse entre sí para la transferencia de información o comandos, la plataforma FPGA de Intel provee compatibilidad con el protocolo de comunicación Avalon, el cual es de código abierto y se maneja por medio de un comportamiento maestro-esclavo y registros con direcciones; por eso también se le llama de memoria mapeada (Avalon MM) [44], un ejemplo en el diagrama de tiempo de este protocolo de comunicación se muestra en la Figura 4.4.

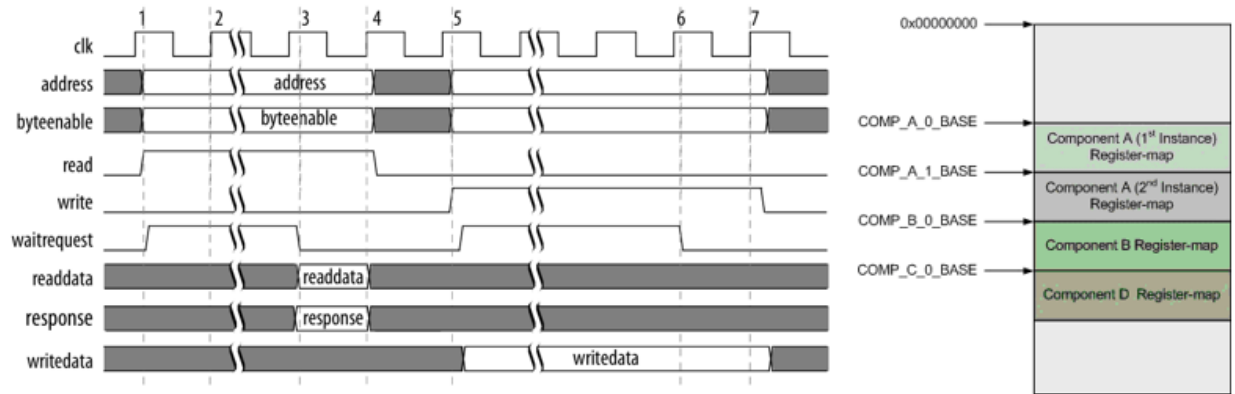


Figura 4.4: Protocolo de comunicación Avalon.

4.4 Implementación a nivel de registro RTL

Los sistemas digitales desarrollados en plataformas como FPGA se programan con lenguajes HDL (lenguaje descriptivo de Hardware) como VHDL y Verilog; estos lenguajes difieren de los lenguajes de programación debido a que permiten la concurrencia, es decir, eventos que ocurren paralelamente, ya que se describe el comportamiento de un circuito digital, con compuertas lógicas, flip-flops, registros, etc. Existen diferentes niveles de abstracción en los lenguajes HDL. Se puede utilizar el estilo de programación por descripción de Hardware, en el que se procura describir los elementos lógicos y la descripción de comportamiento, donde se programa pensando en el comportamiento del sistema que se está diseñando de esta manera, al momento de la síntesis de código, se puede observar el hardware que se generó. Otro nivel de abstracción en lenguajes es utilizar lenguajes HDL con características de programación orientada a objetos, como SystemVerilog; estos permiten crear objetos y brindan más libertad al momento de programar, lo cual los hace muy plausibles para describir comportamientos complejos o sistemas a gran escala, con la desventaja de que inferir el hardware que será sintetizado es más complejo.

Cuando los lenguajes HDL son sintetizados, se dice que tenemos la implementación a nivel de registro RTL; esto quiere decir que el diseño se creará con registros y compuertas combinatoriales, de tal forma que se pueden realizar simulaciones sobre el comportamiento del circuito si la implementación no cumple con los requerimientos, se puede modificar el código, como se observa en la [4.2](#), por eso, es necesario verificar y realizar simulaciones con código de prueba del circuito diseñado. Esto puede ser una tarea laboriosa, incluso más que el diseño del circuito propuesto.

4.5 Verificación

En el diseño de hardware, en la parte del flujo de trabajo, se encuentra la verificación, la cual puede ser incluso más intensiva que el diseño propio del circuito digital que se quiere implementar. La verificación juega un papel importante, pues se enfoca en buscar errores y crear los casos de prueba necesarios para asegurar el correcto funcionamiento del circuito digital. Para una correcta verificación del diseño digital, existen diferentes metodologías, como UVM (Universal Verification Methodology), pero no es la única forma; cada empresa puede establecer su propio flujo de verificación.

La verificación más sencilla que se puede implementar es un bloque de VHDL ó Verilog que envuelva el diseño que se quiere verificar, denominado DUT (Device Under Test), y establezca los estímulos de prueba como entradas, observe las salidas y las compare con el resultado deseado. esto se puede lograr por medio de software de simulación como ModelSim, el cual contiene el ambiente de pruebas para observar las señales, generar reportes y depurar un diseño digital.

En la Figura 4.5 se puede observar una simulación de verificación para el llenado de una memoria FIFO (first in, first out), la cual se utiliza para la transmisión de datos. La verificación consiste en un código VHDL que instancia el hardware de transmisión de datos; se envían números en secuencia o conocidos. Esta es relativamente sencilla, pero es de suma importancia no omitir este paso, dado que se pueden detectar comportamientos erráticos, comportamientos inesperados y se tiene un conocimiento de cómo debe comportarse el circuito implementado físicamente. En caso de que esto funcione incorrectamente, se logrará una corrección de errores en el menor tiempo posible. También es importante conocer los ciclos de operación, en cuántos ciclos de reloj se realiza cada operación. Esto no siempre es fácil de detectar cuando se está programando hardware o se instancian componentes de propiedad intelectual, y esa información solo se encuentra en la documentación.

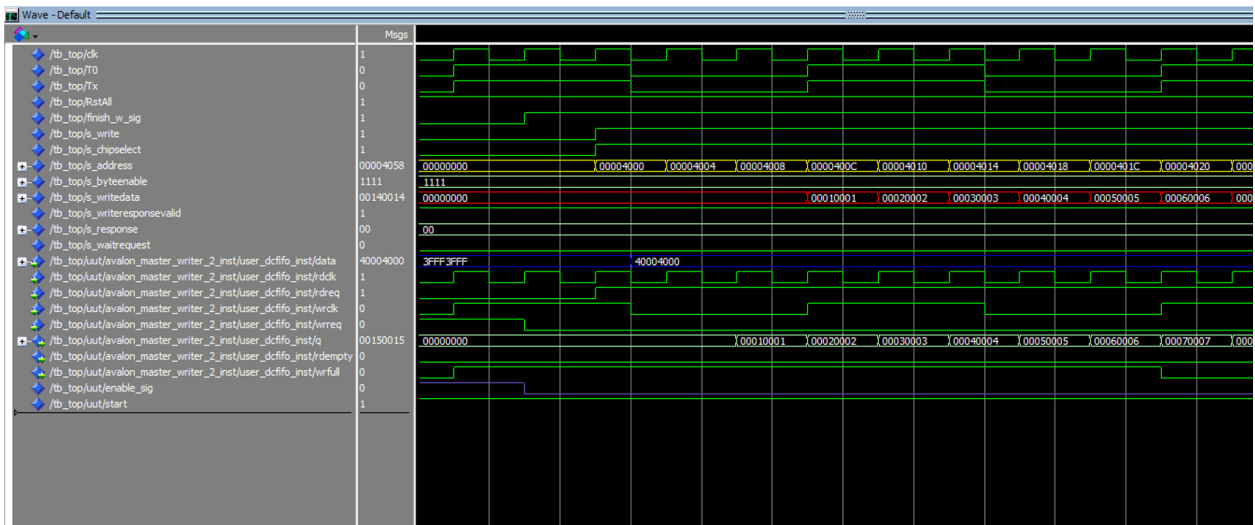


Figura 4.5: Verificación de transmisión FIFO.

Capítulo 5

Estimador de frecuencia

Cuando recolectamos los valores de los contadores almacenados, tendremos diferentes aproximaciones del valor de frecuencia desconocida, de esta manera, surge el dilema de cual valor escoger o que criterio utilizar. Un estimador básico puede ser el utilizar el promedio de los datos en un tiempo específico τ , para esta aplicación, conociendo las condiciones, tales como que no es necesario tener un resultado en tiempo real, se propone el siguiente estimador.

Tenemos dos señales S_o y S_x , donde la primera es la señal de referencia y la segunda es la señal cuya frecuencia deseamos estimar, con periodos T_o y T_x como podemos observar en la imagen, los errores de medición se producen al inicio de la medición y al final de la medición la diferencia de fase de cada medición la podemos definir:

$$\begin{aligned}\theta(1) &= T_0 n_o(1) - T_x n_x(1) \pm \theta_1 \\ \theta(2) &= T_0 n_o(2) - T_x n_x(2) \pm \theta_2 \\ &\quad \vdots \\ \theta(k) &= T_0 n_o(i) - T_x n_x(i) \pm \theta_k\end{aligned}$$

Donde θ_k es la fase de inicio de medición causada por los disparadores de inicio de medición. estas diferencias de fase en las coincidencias están relacionadas con el error máximo debido al ancho de los pulsos. Esto genera una gráfica de densidad de probabilidad, como se muestra en la 5.1. Con un histograma, tiende a comportarse como una variable aleatoria uniforme [45].

$$\theta(k) = T_0 n_o(i) - T_x n_x(i) \pm \theta_k \quad (5.1)$$

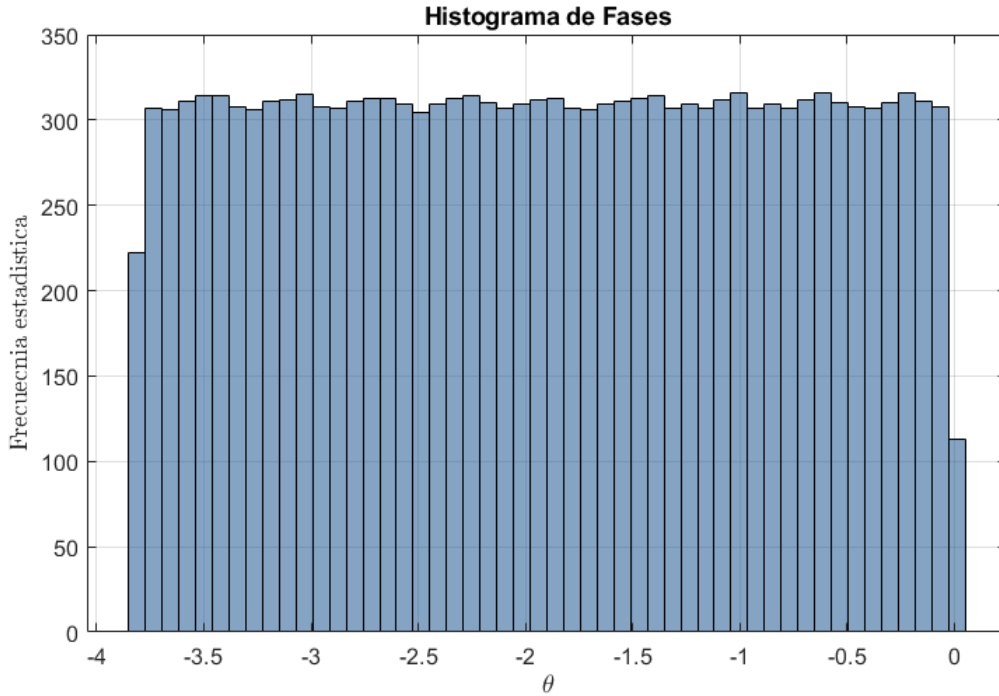


Figura 5.1: Histograma de fases θ de cada coincidencia.

La ecuación (5.1) nos permite observar la diferencia de fase de cada coincidencia. Para el propósito de esta investigación, definimos los mejores valores cuando $0 \leq \theta(k) \leq |\text{umbral}_{error}|$. Este umbral se tiene que definir dentro de valores conservadores, ya que un umbral demasiado pequeño puede que no exista ningún valor de $\theta(i)$ que los satisfaga, y no tendríamos datos de medición. En la Figura se muestra el valor de estas diferencias con respecto al número de coincidencias. Para aplicar este criterio, es necesario conocer T_x , el cual es el periodo que deseamos estimar. Sin embargo, podemos obtener un estimado con una coincidencia m seleccionada y obtener su valor utilizando la ecuación (5). De esta manera, obtenemos (5.2),

donde no contemplamos el valor de θ_i , ya que este solo genera una descompensación en el eje y y solo buscaremos el conjunto de valores relativos mas cercanos a 0.

$$\theta(k) = T_0 n_o(i) - \left[\frac{n_o(m)}{n_x(m)} \cdot T_o + \frac{e(m)}{n_x(m)} \right] n_x(i) \quad (5.2)$$

De la siguiente manera, se propone un algoritmo para estimar el periodo desconocido T_x en un tiempo de promediado τ , utilizando la ecuación de análisis (5.2).

1. Definir Coincidencia inicial en el centro de un paquete de coincidencias.
 1. Calcular diferencias $\Delta n_x(i)$ y $\Delta n_o(i)$.
 2. Detectar indices i con incrementos de mayor valor, comparados con el resto de los incrementos.
 3. Seleccionar coincidencia de inicio entre el primer y segundo incremento positivo i_1 i_2 de $\Delta n_x(i)$ y $\Delta n_o(i)$ obteniendo el centro de un paquete con $i_c = \left\lceil \frac{i_2 - i_1}{2} \right\rceil$ y compararlos, si difieren existe un error de conteo.
2. Ajustar $n_x(i)$ y $n_o(i)$ en base al índice de coincidencia inicial seleccionado.
3. Escoger una coincidencia final m con la menor probabilidad de error posible dentro del rango de tiempo promedio τ , $n_x(m)$ y $n_o(m)$.
4. Sustituir en (5.2) los valores previamente seleccionados y obtener $\theta(i)$.
5. Definir un umbral de error para seleccionar el conjunto de coincidencias de estimación.
6. Obtener el conjunto de coincidencias que cumpla el siguiente criterio $|\theta_k| \leq \text{umbral}$.
7. Con el nuevo conjunto de coincidencias podemos obtener el promedio de estas y este sería el período T_x estimado en el tiempo τ .

En la Figura (5.2) se muestra el desarrollo del algoritmo con datos simulados para una señal de referencia $T_o = 1\mu s$, una señal desconocida $T_x = 1,61803398874989\mu s$ y un umbral = 5×10^{-10} se puede observar que el conjunto de coincidencias que se encuentra mas cercano a 0

en la tercera gráfica corresponde a las coincidencias que están mas cercanas a la convergente del método de coincidencia de pulsos, las cuales están encerradas en círculos rojos. este conjunto de datos puede ser utilizado para obtener un promedio con menor desviación σ^2 . Este algoritmo también se comporta como un discriminador, de tal manera que las coincidencias o datos atípicos que no corresponden a una secuencia de la convergente son filtrados al momento de seleccionar el conjunto de coincidencias mas cercanas a 0 de la diferencia de fases, ya que estas se encontraran fuera del umbral de selección.

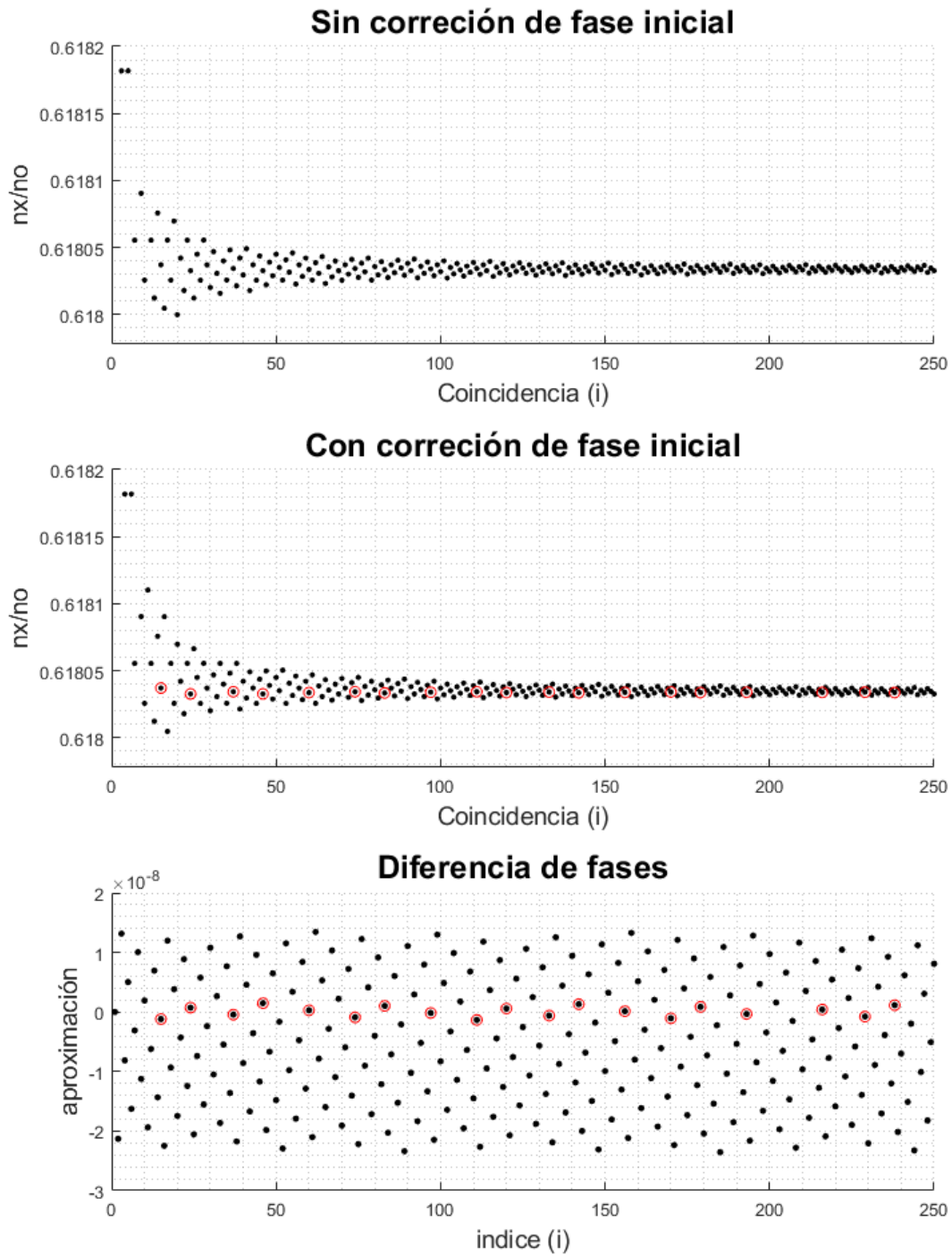


Figura 5.2: Selección de coincidencias sin conocer T_x .

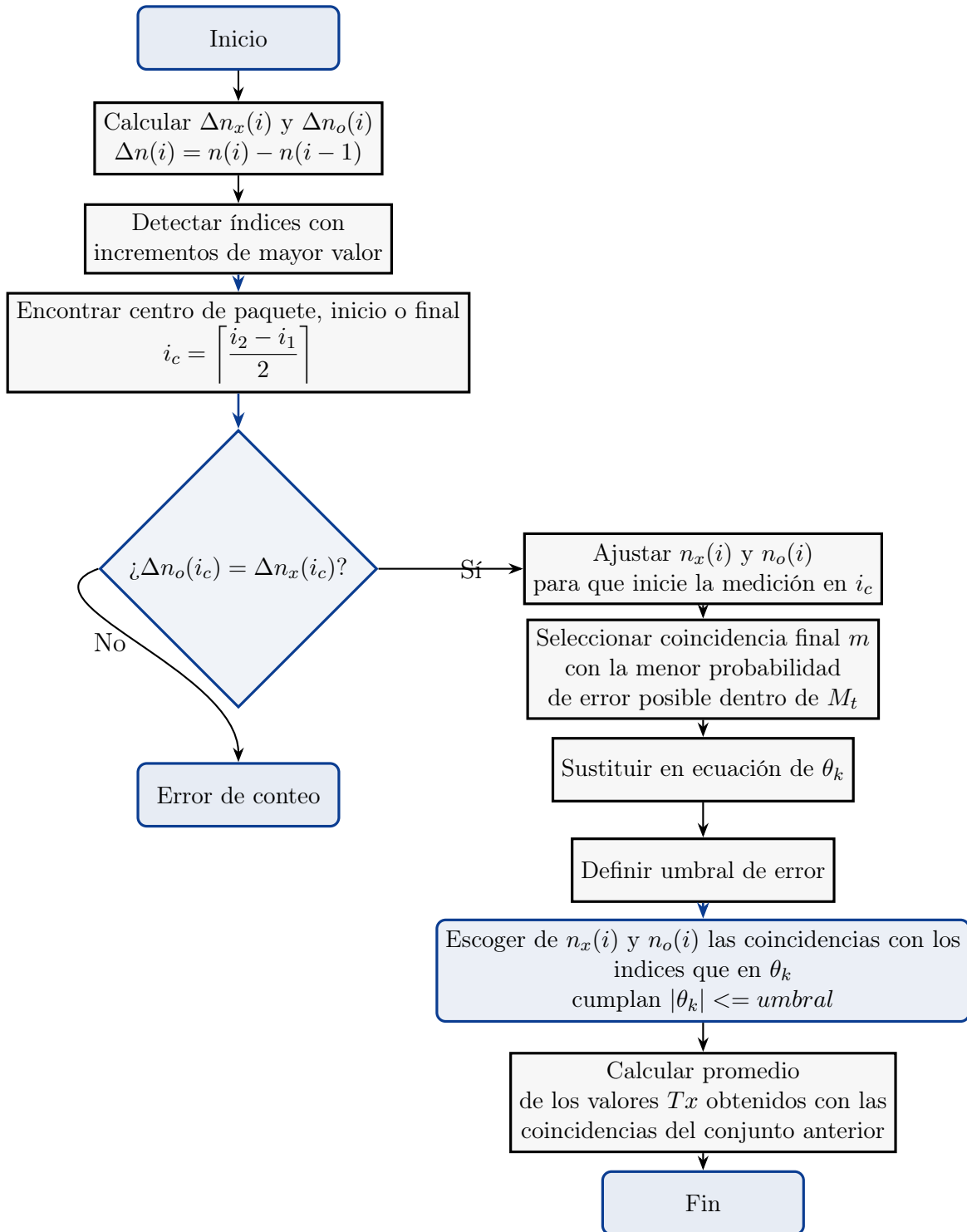


Figura 5.3: Diagrama de flujo algoritmo.

Capítulo 6

Desarrollo de prototipo en FPGA

En este capítulo se presenta el diseño del hardware, las limitaciones encontradas, así como los requerimientos necesarios para el funcionamiento e implementación del procesamiento de datos propuesto en esta investigación.

6.1 Implementación

El diseño propuesto se implementó utilizando la tarjeta de desarrollo Altera MAX 10 el diseño, escrito en código VHDL, implementa el método de coincidencia de pulsos utilizando el sintetizador RTL de Quartus la ventaja de utilizar un sistema re-programable como los FPGA es que el hardware es escalable y mantenible, permitiendo realizar modificaciones y ajustes futuros. Dentro de las desventajas se encuentra que, al tener bloques configurables para realizar cada operación lógica, como se observa en la Figura 4.1, se requieren más transistores que su equivalente no re-programable; por lo tanto, el desempeño puede verse afectado. Sin embargo, las frecuencias requeridas para esta investigación son estables en esta plataforma y cumplen con los propósitos de la investigación.

Con los datos obtenidos del diseño en FPGA, se procesaron las coincidencias mediante MATLAB para calcular los resultados utilizando el estimador mencionado en el capítulo 5. Para este desarrollo se utilizan los temas de los capítulos anteriores, y en este capítulo se mostrarán las consideraciones de diseño, así como la implementación del algoritmo de coincidencia de pulsos en código VHDL y el estimador de las mediciones realizadas.

6.1.1 Consideraciones de diseño

Las consideraciones del diseño se basaron en la experiencia utilizando el FPGA y son las siguientes:

- En el sistema digital se encuentran dos dominios de frecuencias: T_s (coincidencia de pulsos) y Clk (reloj del sistema, 10 MHz, 25 MHz, 50 MHz, hasta 100 MHz).
- El comportamiento del método de coincidencia de pulsos tiende a ser más similar al de un sistema digital asíncrono.
- Es necesario tener un mínimo de 10 Mx32 bits de memoria o un protocolo de transferencia de información rápido para obtener 1 s de medición estable con una señal de referencia de 10 MHz.
- Es de vital importancia el tipo de flujo de datos, ya que de esto dependerá la velocidad máxima de transferencia y el tamaño de la memoria.
- La memoria cache de alta velocidad es limitada, e.g., 12 kbits en FPGA Max 10.

Los puntos anteriores acentúan los retos que se enfrentan al realizar un prototipo funcional implementado en una plataforma de diseño de Hardware para el método de coincidencia de pulsos utilizando lenguajes descriptivos de Hardware, por ejemplo VHDL, así como las limitaciones que se encuentran en el Hardware. Estas variarán dependiendo de la plataforma física que el usuario desee utilizar.

Que en el sistema digital se encuentren dos dominios de frecuencia se debe a que las señales de entrada S_x y S_o generan el reloj para capturar los datos por medio de la compuerta AND, a través de la cual se detectan las coincidencias. $S_x \&\& S_o$ como la compuerta AND es controlada por frecuencias diferentes, podemos decir que el reloj de lectura trabaja en dos dominios de frecuencia de manera asíncrona. La tecnología digital asíncrona, por lo general, no se estudia y no se encuentra en todos los libros de diseño digital; por ejemplo, en [46], este libro solo se enfoca en circuitos digitales síncronos y explica cómo los circuitos digitales asíncronos pueden presentar un mejor rendimiento, sin embargo, el proceso de diseño puede ser complicado y poco escalable, como lo es para los circuitos digitales con máquinas de estados, mapas de Karnaugh, etc. Por eso, se considera que el comportamiento del sistema de coincidencia de pulsos tiende a ser asíncrono en el momento de capturar las coincidencias.

Por ello, es necesario implementar estrategias para que el circuito funcione. Al respecto de la memoria necesaria para el diseño, no solo hay una solución para este problema; dependerá de diferentes factores en este caso, se toman en cuenta la velocidad de escritura $Wclks$ y la velocidad de lectura $Rclks$ para nosotros $Wclks = S_x \ \&\& \ S_o$, dado que es la señal que da la pauta para guardar un par de coincidencias n_x, n_o . cada palabra se compone de 32 bits: 16 bits para n_x y los 16 bits restantes para n_o concatenados, si se tiene que $Wclks = S_x \ \&\& \ S_o = 10Mhz$, entonces la memoria necesaria será 10 Mx32 bits/s.

6.1.2 Hardware implementado

En esta sección se detallará como está implementado el diseño, en base a qué tecnología y plataforma, así como la descripción RTL. Se muestran los diferentes bloques lógicos, así como las entradas y salidas que generan el funcionamiento deseado.

Diagrama RTL

A continuación, se muestra el código sintetizado en Register Transfer Level, como se menciona en el **capítulo 4.4** la implementación digital para la estimación de frecuencia tiene el siguiente diagrama RTL, mostrado en la Figura 6.1, este consta de componentes de propiedad intelectual, como un procesador Nios II y memoria RAM proporcionada por el fabricante, para aumentar el desempeño del diseño en la tarjeta de desarrollo, debido a que estos están optimizados para utilizar implícitamente recursos del FPGA, como bloques destinados a memoria, PLL y diseños optimizados para la tarjeta de desarrollo, estos recursos son explícitamente declarados por las herramientas de diseño, como **Quartus Prime y Platform Designer**. Para este trabajo, se decidió utilizar estos bloques de propiedad intelectual reservados (IP) en partes de diseño que no están sujetas a investigación. se espera el máximo desempeño de estas estructuras, tanto como transmisión de datos hacia una computadora como en la recepción de comandos. De esta forma, se pueden enfocar los recursos de tiempo hacia los bloques más importantes para el desarrollo de este trabajo y esperar el mejor desempeño de los periféricos que lo rodean, aproximándonos así más a entornos de aplicaciones reales.

En la Figura 6.3 se pueden observar las entradas y salidas del circuito digital. A la izquierda se muestran las entradas y, a la derecha, las salidas. En la tabla 6.1 se muestra la descripción de los pines de entrada y salida

En el diagrama 6.2 se pueden observar cuatro bloques principales. el bloque *Coin* es una compuerta Latch, la cual se encarga de presentar la señal de inicio de coincidencia cuando la compuerta AND se activa y la entrada *Start/Read* esta en alto; entonces, ambas señales de coincidencia se encuentran en alto; por eso, T_x y T_o entran primero a unas compuertas AND. la salida de *Coin* es usada en el bloque *Write Control Logic*. Este bloque se encarga de, por

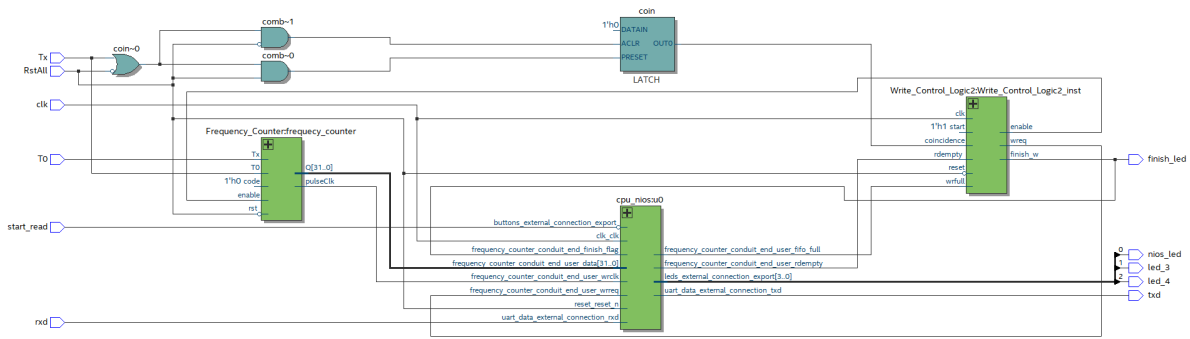


Figura 6.1: Diagrama RTL.

Pin	Tipo	Descripción
Entradas generales		
clk	Entrada	Señal de reloj principal para sincronización del sistema.
RstAll	Entrada	Reinicia todos los módulos del sistema a su estado inicial.
start_read	Entrada	Activa el inicio del conteo de frecuencia.
rx	Entrada	Lectura serial UART
Contador de frecuencia		
To	Entrada	Señal para coincidencia de pulsos S_x
Tx	Entrada	Señal para coincidencia de pulsos S_o
Salidas del sistema		
led[3:0]	Salida	Indicadores visuales del estado del sistema o señales de control.
tx	Salida	salida serial de información hacia computadora o dispositivo capaz de interpretar UART

Tabla 6.1: Entradas y salidas del sistema digital.

medio de una máquina de estados, gestionar la escritura en memoria RAM y, mediante una memoria FIFO, necesita estrictamente que dos condiciones no se cumplan; no se escribe en la memoria cuando esta está llena y no se lee cuando la memoria está vacía. ambas condiciones

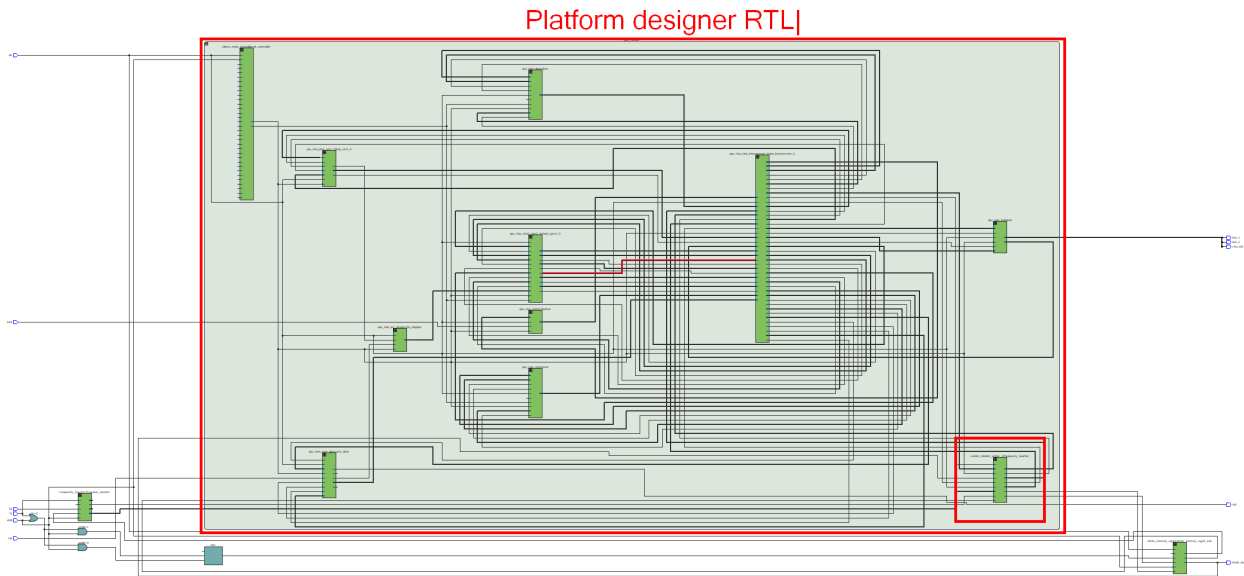


Figura 6.3: Diagrama RTL de diseño implementado en Platform designer.

En la Figura 6.4 se puede observar el diseño RTL, pero se muestra con más detalle el bloque instanciado en Platform Designer, el cual contiene otro bloque importante dentro, resaltado en rojo, que es el diseño personalizado para manejo de datos entre el circuito de coincidencia de pulsos de la Figura 6.2, la memoria RAM y el procesador NIOS II, que lee los datos de esta memoria y los transfiere por UART a una computadora. En la Figura 6.4 Se muestran con más detalles las señales de entrada y salida; cada una de ellas tuvo que ser programada para comportarse de acuerdo con el protocolo de transferencia de datos Avalon especificado en el **capítulo 4.3.1**.

Este bloque es de suma importancia, dado que nos permite tener un flujo de datos y escribirlos de manera rápida, resolviendo un problema común que se presenta al trabajar en plataformas con FPGA, cuando uno tiene sus datos o modelo, se requiere observar esos datos sin perderlos, ya sea en una memoria o computadora. existen diversas soluciones; algunas incluyen utilizar software con licencias de costo, como MATLAB HDL Coder o LabVIEW MyRIO, en este caso, se optó por diseñar el hardware que implemente las funciones necesarias, como este bloque custom, y utilizar las herramientas de libre uso en Quartus Prime y Platform Designer, dado que nos permiten observar el hardware sintetizado, así como especificarlo mediante código.

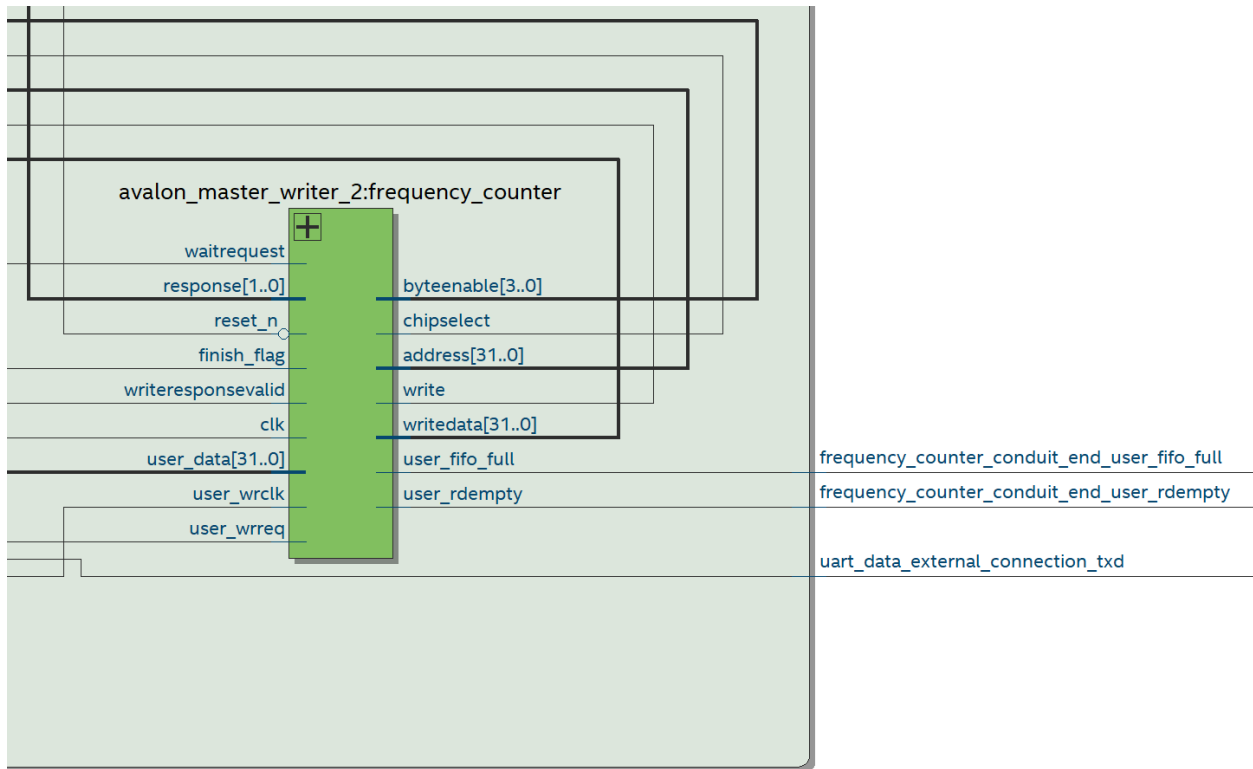


Figura 6.4: Diagrama RTL de Bloque personalizado para platform designer.

El bloque de la Figura 6.5 muestra el diagrama RTL del bloque personalizado, se puede observar que se compone de multiplexores, registros y compuertas básicas. Las señales de salida se mantienen a través de los registros. En rojo se muestra resaltada la memoria FIFO explicada en el **capítulo 4.3**. este es un componente clave debido a que se manejan frecuencias diferentes en la escritura de datos y la transferencia de protocolos, como por ejemplo el protocolo UART, de esta manera, los datos se van almacenando a medida que las señales S_x y S_o generan coincidencias, y se pueden leer a la velocidad de un protocolo de comunicación sencillo como el UART a 115200baud/s .

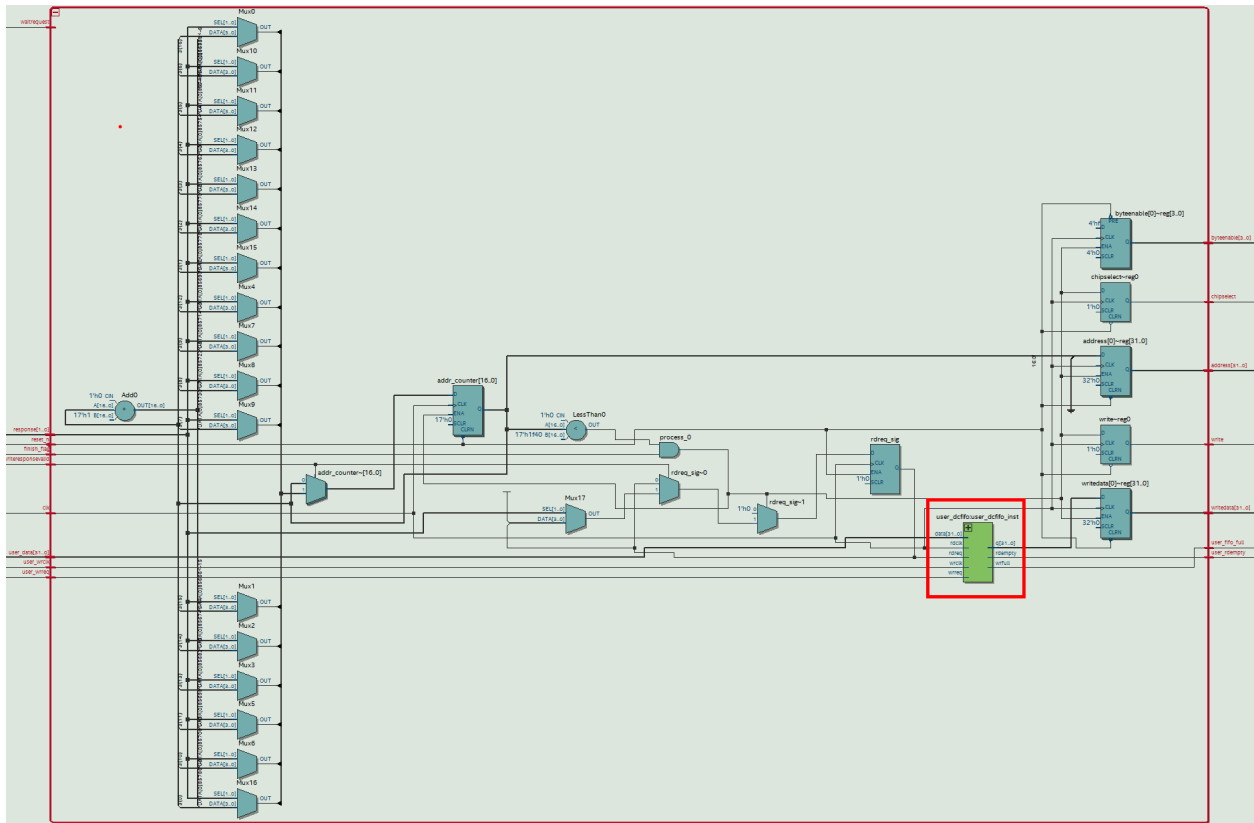


Figura 6.5: Diagrama RTL detallado de Bloque personalizado para platform designer.

Maquinas de estado finito

Las máquinas de estado finito son esenciales en el diseño digital; nos permiten generar lógica secuencial; y se utilizan para automatizar acciones, específicamente para el inicio y el final de la medición. A continuación, se presenta la máquina de estados utilizada para controlar la escritura de datos:

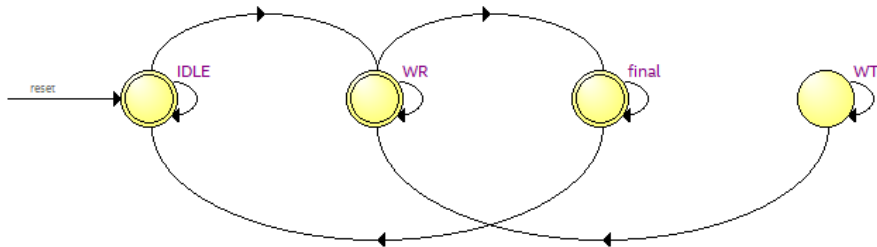


Figura 6.6: Máquina de estados.

El diagrama de máquina de estados anterior 6.6 representa una máquina de estados Moore ya que los cambios de estado no dependen de la salida. Sabemos esto porque en las flechas de transición no se observan cambios en la salida, lo que quiere decir que los cambios suceden dentro de los estados y no en la transición de estos, una característica de las máquinas de estado Moore.

La tabla 6.7 muestra las transiciones de la máquina de estados. La primera columna representa el estado en el que se encuentra, la segunda columna es el estado al que debe transitar cuando se cumpla la condición de la columna número tres. Algo muy importante a resaltar es que ninguna transición se encuentra sin especificar, dado que esto puede provocar una mala síntesis de compuertas digitales [48]. Esta es la descripción de cada estado:

	Source State	Destination State	Condition
1	final	final	(!reset)
2	final	IDLE	(reset)
3	IDLE	WR	(process_0)
4	IDLE	IDLE	(!process_0)
5	WR	WR	(twrfull)
6	WR	final	(wrfull)
7	WT	WR	(rdempty)
8	WT	WT	(!rdempty)

Figura 6.7: Maquina de estados.

- IDLE: En este estado, se espera que el reset cambie para comenzar la acción de escritura; aquí se utiliza el reset como un accionador.
- WR: En este estado se permite la escritura de las coincidencias; si la memoria fifo esta llena, se va al estado WT
- Estado final cuando se terminan de escribir datos.
- WT: En este estado, se espera a que la memoria fifo libere espacio para seguir escribiendo.

La Figura 6.8 muestra el diagrama RTL de la máquina de estados; aquí se pueden observar las compuertas y registros que son parte del diseño, así como las señales de entrada y salida.

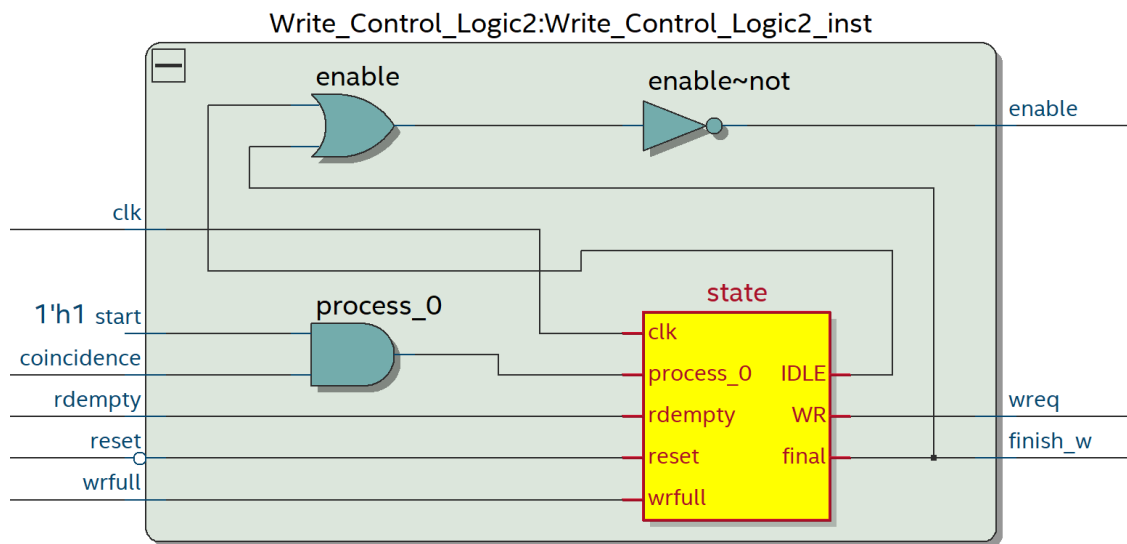


Figura 6.8: Diseño implementado en platform designer.

Platform designer

Platform designer es un módulo del programa quartus. La importancia de utilizar este módulo es que nos permite usar los recursos de la tarjeta FPGA (es decir, conectar módulos físicos como ROM, RAM, Ethernet, etc.) de una manera más eficiente en tiempo y rendimiento. Se debe tener conocimiento de qué es lo que se está diseñando, cómo funciona el platform designer para poder conectar los diferentes módulos entre sí, y saber que lo que se quiere realizar es posible.

En esta parte se busca conectar nuestro diseño con una memoria y leer esta memoria a través de un microprocesador llamado NIOS II. En la imagen 6.9 se puede observar cómo se conecta o diseña el hardware en platform designer. El cuadro en rojo muestra el bloque propio que se programó en VHDL y se adaptó para que fuera compatible con el bus de datos Avalon, y de esta manera, poder escribir la información en una memoria RAM. La línea amarilla muestra el flujo de los datos tal como puede ser leído por el microcontrolador. La salida de los datos se realiza a través de JTAG UART. El microcontrolador se encarga de leer los datos grabados en la memoria y enviarlos por UART hacia una computadora para ser procesados.

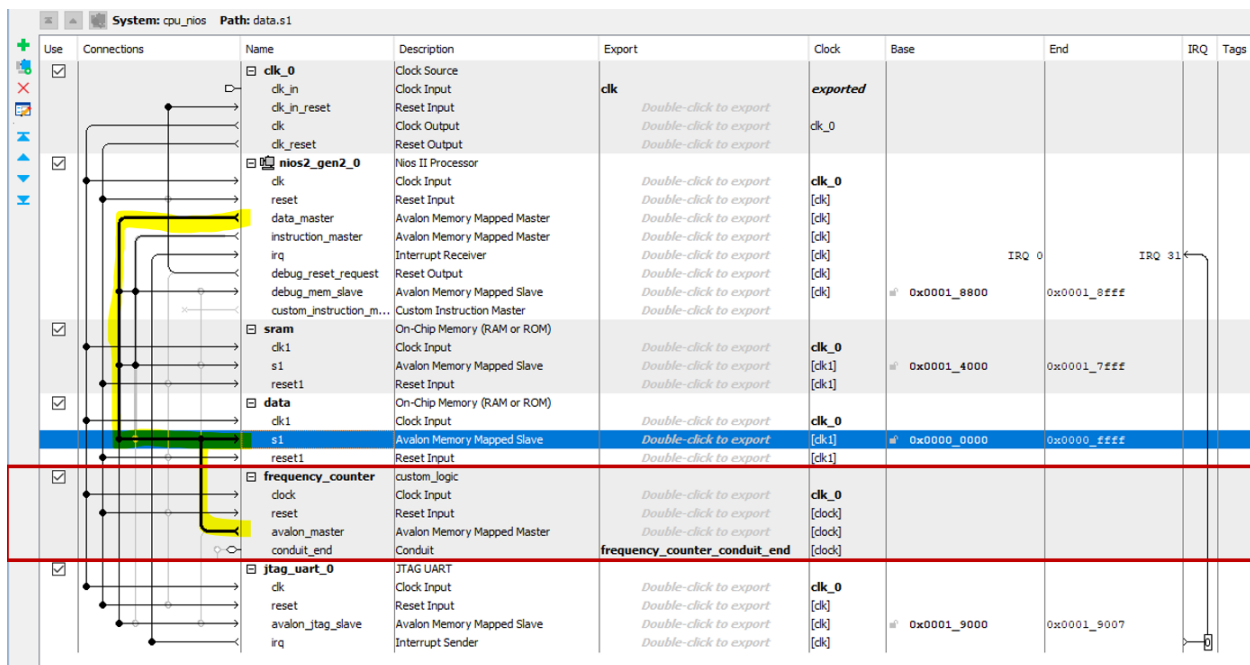


Figura 6.9: Diseño implementado en platform designer.

6.2 Estimador en mediciones realizadas

El estimador de frecuencia propuesto se utilizó en las mediciones de osciladores con el sistema de coincidencias de pulsos diseñado en FPGA. estas coincidencias seleccionadas son promediadas, lo cual nos permite estimar \bar{y}_k cuál es el valor que el usuario puede utilizar como medición. \bar{y}_k tiene sus características estadísticas, que son necesarias para saber el nivel de confianza con el que se está trabajando al tratar de estimar la frecuencia de algún dispositivo.

Se recolectaron las coincidencias del Hardware diseñado, como se muestra en la **sección 6.1.2**. el resultado es un par ordenado de números enteros $n_x, n_o \in \mathbb{Z}$, debido a que en el proceso de disparo de medición se pueden presentar errores de medición; además, comenzar conteos antes de tiempo puede ocasionar un error de acarreo, que se puede observar en la gráfica de mediciones como una curva ascendente o descendente. Para solucionar esto, se propone corregirlo ajustando n_x y n_o , con el siguiente algoritmo:

- Encontrar paquetes de medición.
- Escoger como referencia el inicio, final o coincidencia intermedia de un paquete de coincidencias n_s .
- Eliminar las coincidencias anteriores a la seleccionada y registrar esta coincidencia como la primera.
- Restar el número de coincidencia al resto de coincidencias: $n_{o1} - n_s, n_{o2} - n_s, n_{o3} - n_s, \dots$, donde n_s es el número de coincidencia seleccionado y n_{oi} son las coincidencias que deben aplicarse a n_o .
- De forma paralela restar el número de coincidencia al resto de coincidencias: $n_{x1} - n_s, n_{x2} - n_s, n_{x3} - n_s, \dots$

De esta forma, se ajusta el inicio de la medición y no se afecta la relación entre los conteos. El sacrificio es el tiempo de medición antes del par n_x, n_o seleccionados. En la Figura 6.10 se muestra el comportamiento de este ajuste; como en el primer renglón, el arreglo de coincidencias muestra el comportamiento mencionado anteriormente. Después de aplicar la corrección, se puede observar cómo cambia el patrón de coincidencias; esto ocurre

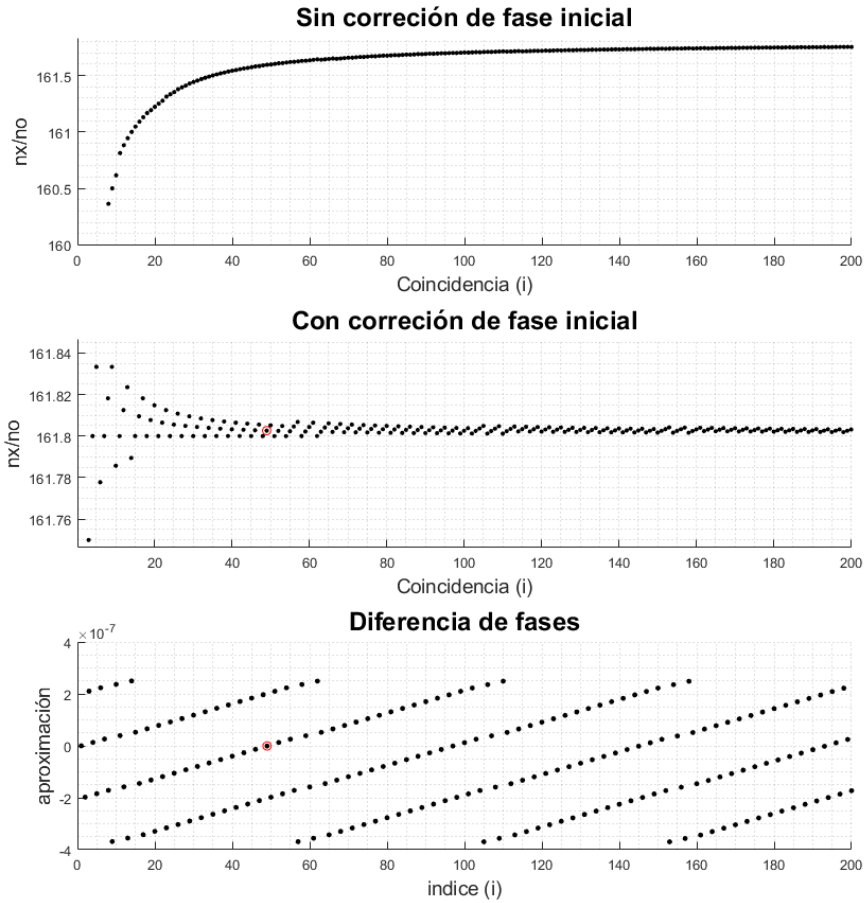


Figura 6.10: Ajuste de coincidencias.

debido a que la fase inicial se modifica al alterar artificialmente la coincidencia inicial, por lo que el error que aporta el primer par de datos varía y permite aplicar algoritmos, así como el procesamiento de estimadores, incluyendo el presentado en este trabajo de investigación.

Una vez corregido el conjunto de datos n_x , n_o , se pueden aplicar estimadores o algoritmos como los descritos en el **capítulo 3**, donde se describen estimadores por hardware o software. Se propone utilizar el estimador descrito en el capítulo 5, desarrollado en esta investigación, para la detección de inestabilidad en osciladores. Dado que obtener el resultado en tiempo real no es una limitante, se puede aprovechar esto para realizar un procesamiento después de tomar las mediciones; por lo cual, este algoritmo se clasificaría en la categoría de estimadores por

software. Este algoritmo busca encontrar las mejores aproximaciones; filtrar datos anómalos también se busca reducir el conjunto de coincidencias a las mejores, aportando a la hipótesis que esto resultará en un buen valor \bar{y}_k .

Capítulo 7

Resultados

En este capítulo se presentarán los resultados encontrados con las mediciones realizadas y comparaciones con diferentes tratamientos de los datos. Estos datos fueron recolectados por medio del hardware diseñado y se aplicaron diferentes estimadores. De la misma forma, aquí se presentan los descubrimientos sobre el desarrollo en plataformas de hardware, así como recomendaciones para que se puedan utilizar de referencia en futuras aplicaciones.

Frecuencia de referencia 10Mhz

Periodo
16.18034 μS
20.87229 μS
27.18281 μS
31.14159 μS
195.65 ns

Tabla 7.1: Periodos medidos

Los resultados obtenidos mediante el algoritmo mostrado en el capítulo 5, utilizando el filtrado inicial descrito previamente, se presentan a continuación para las frecuencias de la tabla 7.1. El umbral en el algoritmo utilizado para todas las mediciones es 5×10^{-10} ; este número influye en la cantidad de coincidencias seleccionadas; por lo tanto, la cantidad de datos varía en las diferentes mediciones. El conjunto de datos restante nos representa una tendencia en los datos sobre la medición; a este conjunto le aplicamos el promedio, el cual será

el resultado de este algoritmo. Estos resultados se pueden observar en la primera columna de la tabla 7.2. La segunda columna representa la varianza del conjunto de datos obtenidos por este algoritmo antes de aplicarles el promedio. Las últimas cuatro columnas son estimadores con los que comparamos los resultados.

Los estimadores con los que se compara el algoritmo son los siguientes: el promedio de todos los datos medidos, la mediana de los datos y el último dato de medición o de los últimos datos. Esto se debe a que en el método de coincidencia de pulsos se muestra que, conforme pasa el tiempo, la medición tiende a tener un menor error[34], y este ha sido utilizado en trabajos de investigación como un buen estimador de este método [49].

La varianza de los datos del algoritmo y del estimado del promedio se muestra para observar qué tan dispersos se encuentran los valores entre sí; de esta manera, podremos obtener conclusiones acerca de la calidad de los datos o del filtrado del procesamiento de los datos. El promedio y la mediana se eligen como estimadores, ya que estos proporcionan un punto de referencia hacia el cual converge el valor del periodo T_x . Hemos de recordar que este es el valor de medición que se dará como resultado en el tiempo de medición total M_t .

Periodo	Algoritmo Capítulo 5		Datos Normalizados		Estimadores	
	Periodo Prom.	Varianza	Periodo Prom.	Varianza	Mediana	Último Dato
16.18034	161.8026262	$1,30 \times 10^{-26}$	103.1829656	6047.5448	161.7979204	161.8016308
31.14159	311.4144542	$4,23 \times 10^{-15}$	195.7020264	22623.2811	311.3534552	311.4019621
195.65	1.95651387	$8,77 \times 10^{-12}$	1.43580153	0.74667	1.95626939	1.95630551
27.18281	271.8268122	$7,57 \times 10^{-15}$	171.2700081	171.2700081	271.8163364	17219.5274
20.87229	208.7219302	$1,11 \times 10^{-14}$	132.1676107	10105.4274	208.6658342	208.6658342

Tabla 7.2: Comparación de estimadores: Algoritmo Capítulo 5 vs. Estimadores

En la Figura 7.1 se puede observar el error relativo utilizando la ecuación 7.1. El valor verdadero utilizado es el del periodo establecido en el generador de frecuencias; estos valores se encuentran en la primera columna de la tabla 7.2. El error relativo se muestra como un porcentaje para poder tener una referencia sobre el error entre 0 y 100 por ciento.

$$\varepsilon_{\text{rel}}(\%) = \frac{|x_{\text{medido}} - x_{\text{verdadero}}|}{|x_{\text{verdadero}}|} \times 100\% \quad (7.1)$$

El error relativo se muestra en dos figuras para poder apreciar los errores que se encuentran en una escala muy pequeña; por lo tanto, en la Figura 7.2 Se realiza un enfoque sobre los resultados. La barra de color azul representa el error relativo calculado por el algoritmo del capítulo 5 y con el procesamiento mencionado anteriormente.

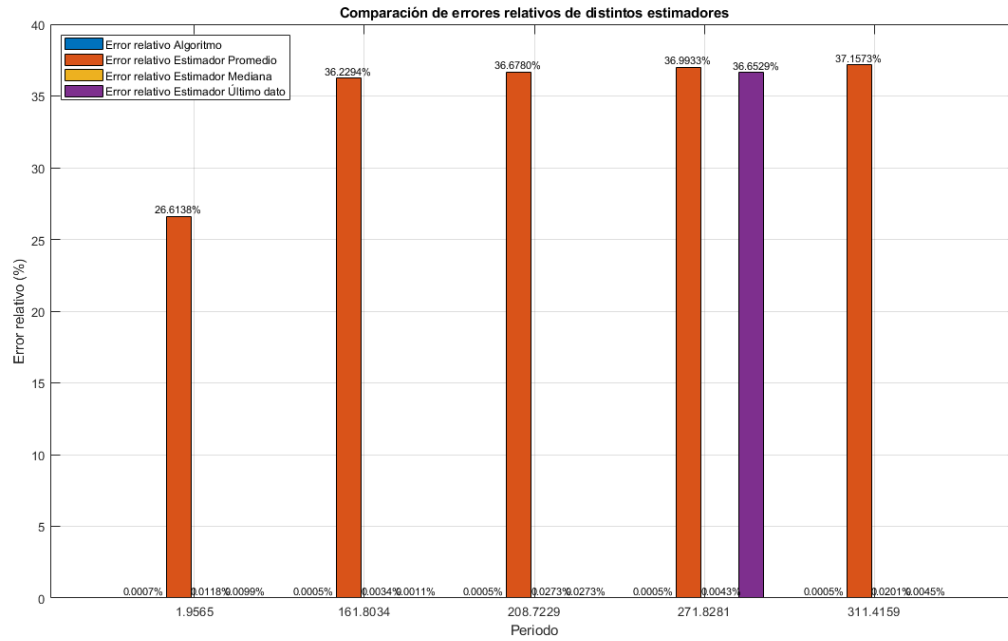


Figura 7.1: Comparación de errores relativos entre estimadores.

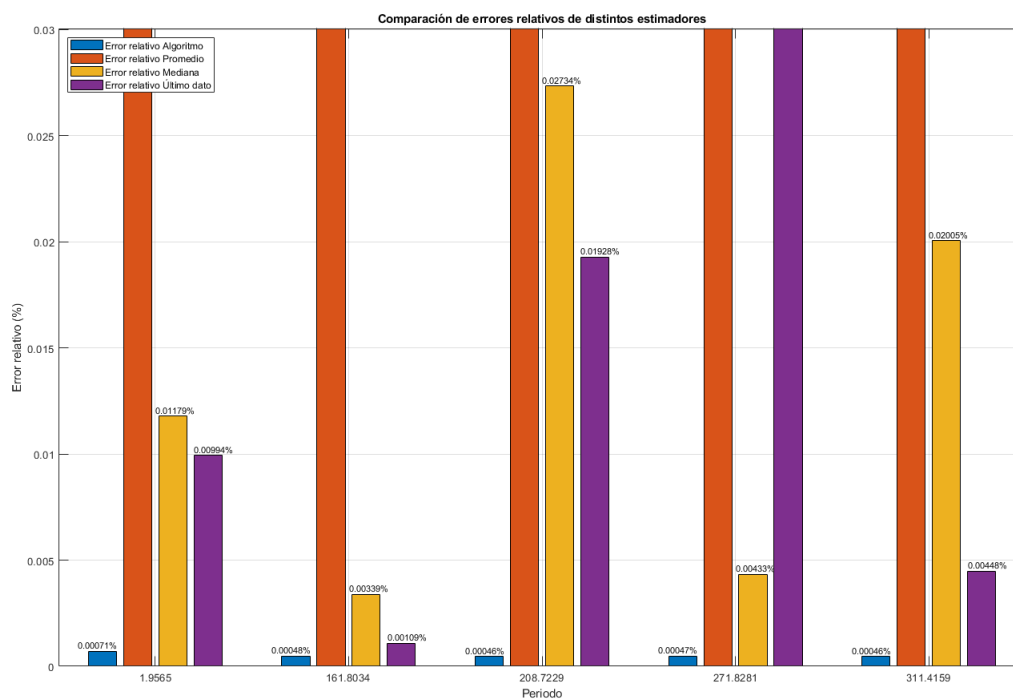


Figura 7.2: Comparación de errores relativos entre estimadores enfoque.

La Figura 7.3 nos muestra gráficas de los resultados para el periodo de medición $tx = 16,18034\mu S$ en el primer plano se presenta el resultado del calculo con las coincidencias seleccionadas; en el segundo, se muestra un promedio móvil; y en el tercero, de arriba hacia abajo, un histograma de estos datos. es interesante observar que la mayoría se agrupan cerca del periodo de referencia, pero se observan algunos que se encuentran alejados de este grupo. La Figura 7.4 muestra un comportamiento similar para la medición del periodo $Tx = 195,65nS$. estos datos alejados en el histograma pertenecen a las primeras coincidencias detectadas. Aunque se tiene un buen conjunto de coincidencias, el valor de estimación sigue mejorando conforme continua el tiempo. La Figura 7.5 muestra un histograma mas consistente con datos mas agrupados.

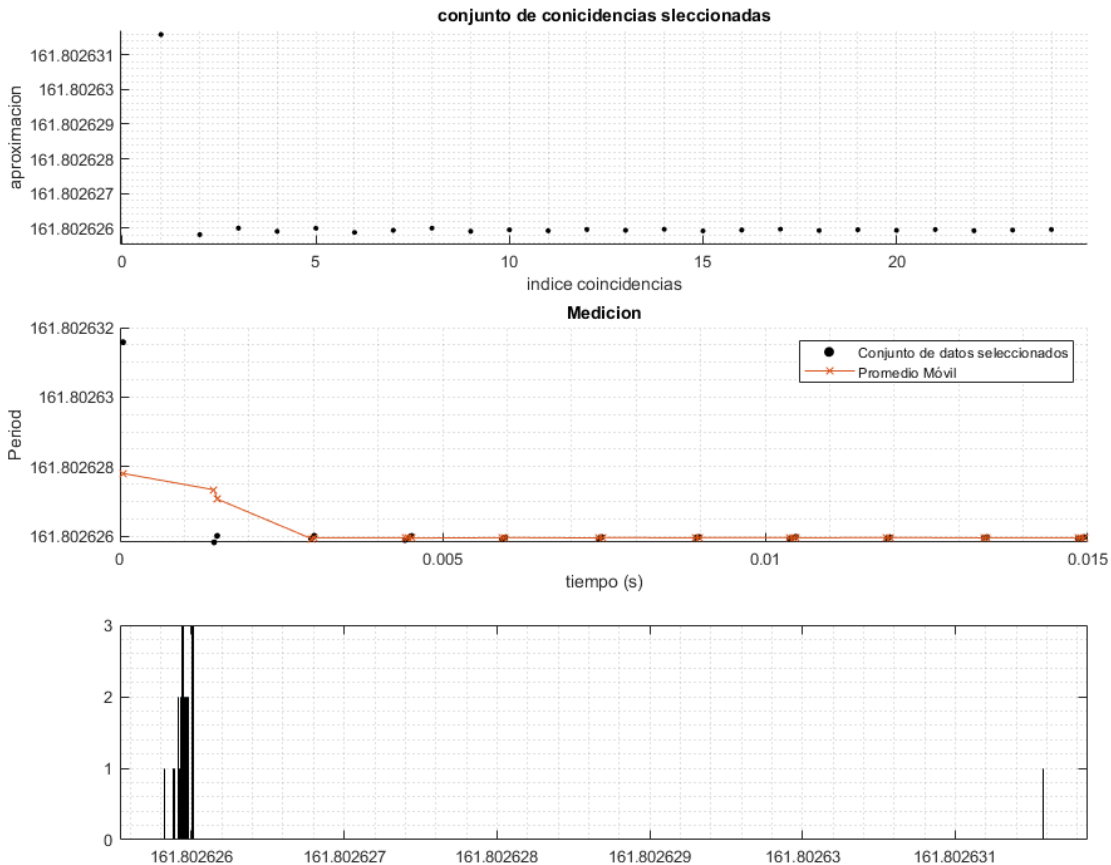


Figura 7.3: Resultados gráficos de medición $tx = 16.18034\mu S$.

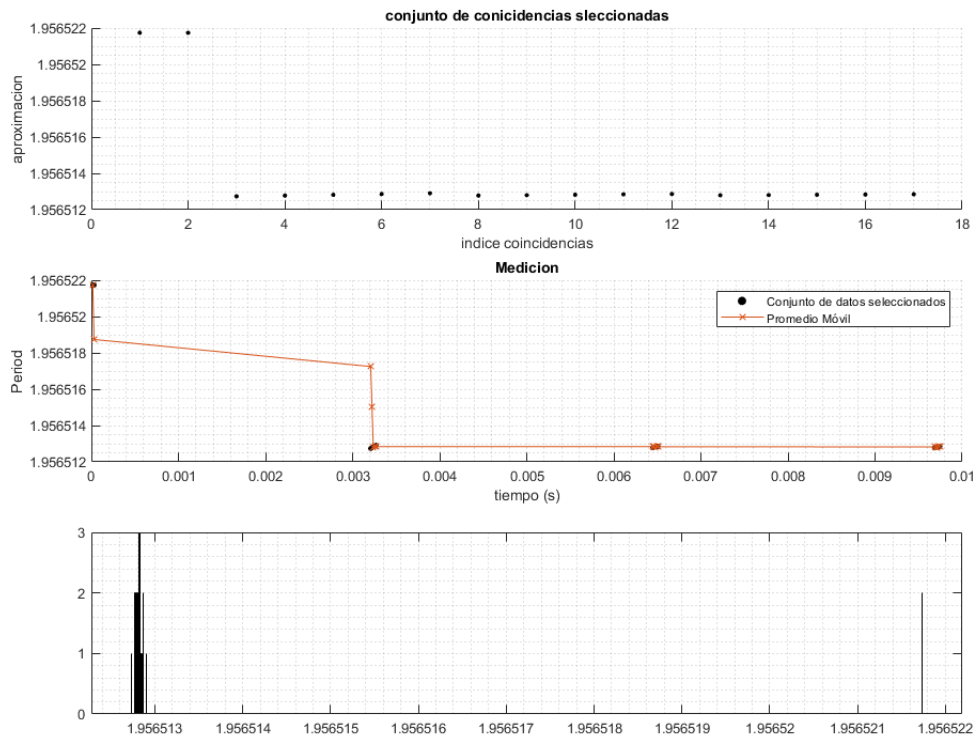


Figura 7.4: Resultados gráficos de medición 195.65nS.

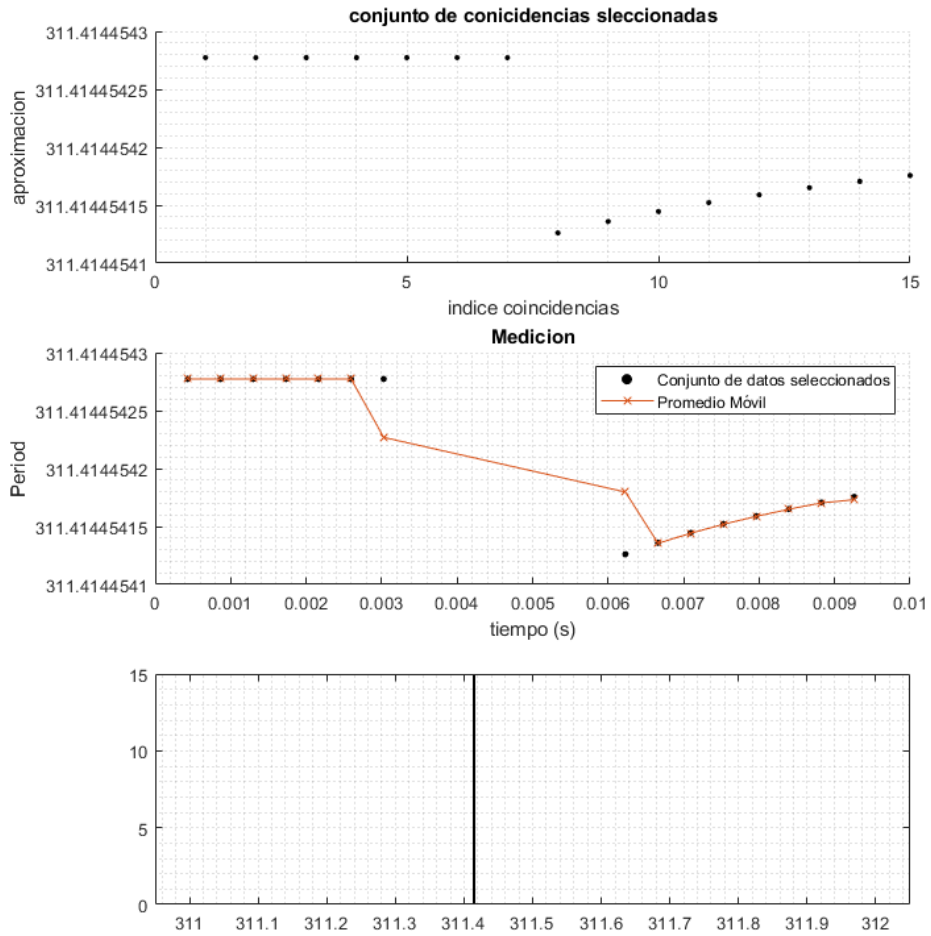


Figura 7.5: Resultados gráficos de medición 31.14159uS.

7.1 Frecuencia instantánea

La frecuencia instantánea es lo que se busca principalmente en un equipo de medición de frecuencia; en estos métodos, se busca definir la frecuencia que se obtuvo en un momento determinado. El resultado anterior muestra como se estima el valor de frecuencia en todo el tiempo de medición M_t , pero otra perspectiva es realizar mediciones de frecuencia en tiempos de muestreo igualmente espaciados, obteniendo como resultado un valor de frecuencia para cada tiempo de muestreo. de esta forma, estimamos mediciones continuas por medio del método de coincidencia de pulsos y podremos, por ejemplo, tratar de estimar la estabilidad.

Para realizar esto, podemos tomar los conteos de las coincidencias medidas $n_x(i)$ y $n_o(i)$ y reiniciar los contadores en el tiempo de muestreo, apoyándonos en la ecuación $t_i = n_o(i) * T_o$ una vez reiniciados los contadores, tendremos un subconjunto de coincidencias entre los periodos de muestreo, cada subconjunto debe ser procesado mediante un estimador, se presenta la comparación de utilizar el promedio como estimador, así como el algoritmo presentado en el **capítulo 5** , al resultado en frecuencia de cada subconjunto; lo denominaremos \bar{y}_k . El valor del estimador promedio de frecuencia es $T_s, 2T_s, 3T_s$. La tabla muestra un extracto de los resultados de estos valores obtenidos para el contador $n_o(i)$, con un $T_s = 5ms$ y utilizando $T_o = 10Mhz, T_x = 16,18034uS$, de esta medición se obtuvieron 77 columnas que representan los datos dentro de ese tiempo de muestreo; cada columna tiene aproximadamente 200 renglones esto se debe a que en cada intervalo T_s no se encontraron exactamente la misma cantidad de coincidencias. esta tabla se repite para $n_x(i)$ y cada coincidencia tiene su equivalente en tiempo para estimar el periodo con la ecuación [3.1](#).

En la Figura [7.6](#) se muestra la comparación de los resultados en tiempo, utilizando simplemente el promedio para estimar \bar{y}_k y utilizando el estimador propuesto en esta investigación. se puede observar como los valores de frecuencia son menos dispersos. Se encontraron tres frecuencias atípicas en ambas mediciones; en el mismo tiempo T_s estos se observan en ambas gráficas como tres puntos demasiado alejados del valor esperado, lo cual pudo ocurrir debido a un mal conteo justo en ese tiempo de medición se decide dejar los datos para mantener la

continuidad de la medición y como parte del experimento de un sistema de medición en la practica. Cabe observar que aquí se esta presentando la frecuencia, no el periodo, como en los resultados anteriores.

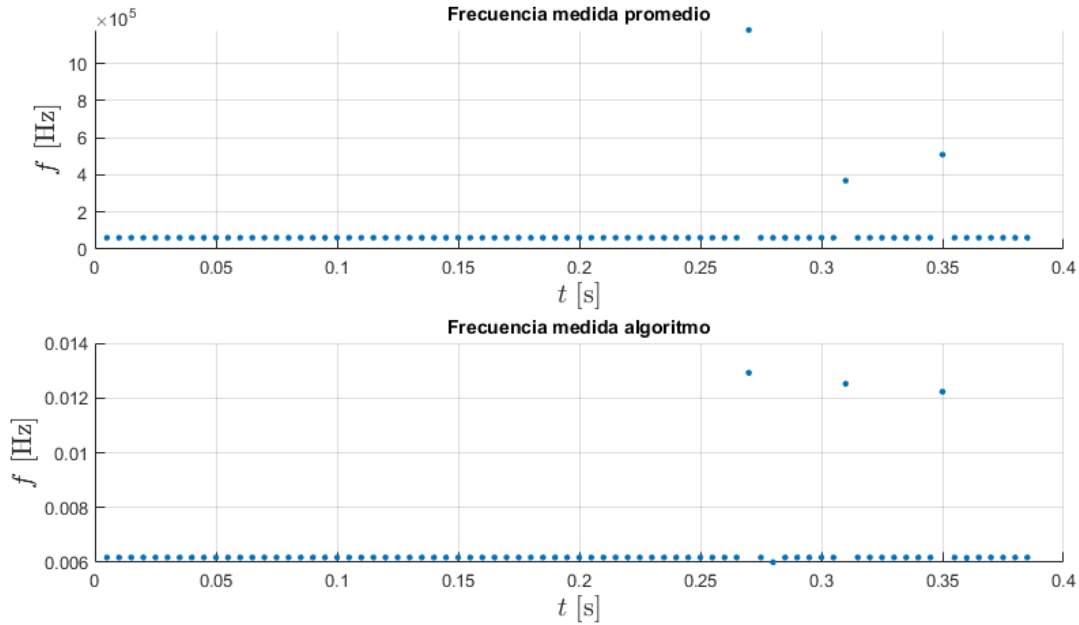


Figura 7.6: Comparación de medición de frecuencia con tiempo de muestreo.

La Figura 7.7 muestra los mismos resultados, pero con un mayor acercamiento a los datos. Los datos atípicos quedan fuera de la imagen, pero ahora podemos observar mejor como están dispersas las mediciones. la escala en el eje y de la segunda medición del algoritmo es menor, 1×10^{-3} ; por lo tanto, aunque en la gráfica se pueda ver una cierta variación en los datos, esta es mucho menor.

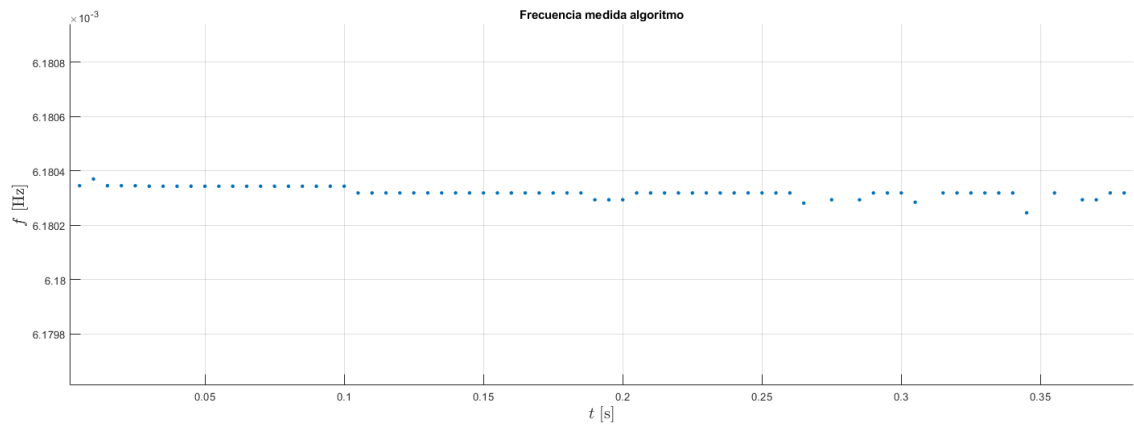
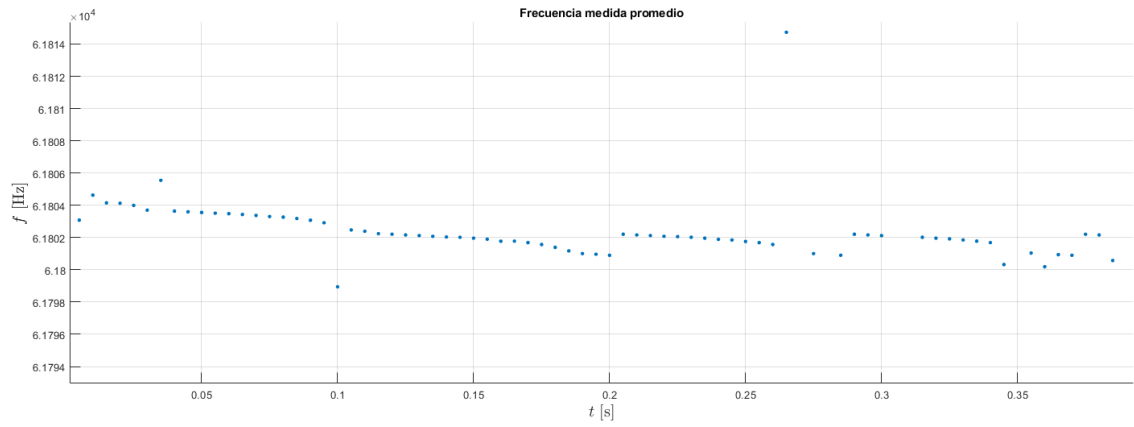


Figura 7.7: Comparación de medición de frecuencia con tiempo de muestreo.

T_s	$2T_s$	$4T_s$	$4T_s$	$5T_s$	$6T_s$	$7T_s$	$8T_s$	$9T_s$	$10T_s$
0	0	0	0	0	0	0	0	0	0
162	485	162	162	162	162	162	162	162	162
647	647	485	485	485	647	647	647	647	647
809	809	647	647	647	809	809	809	809	809
971	971	809	809	809	971	971	971	971	971
1456	1294	971	971	971	1456	1456	1456	1456	1456
1618	1456	1294	1294	1456	1618	1618	1618	1618	1618
1780	1618	1456	1456	1618	1780	1780	1780	1780	1780
1942	1780	1618	1618	1780	2265	2265	2265	2265	2265
2265	2103	1780	1780	2265	2427	2427	2427	2427	2427
2427	2265	2265	2265	2427	2589	2589	2589	2589	2589
2589	2427	2427	2427	2589	3074	3074	3074	3074	3074
2751	2589	2589	2589	3074	3236	3236	3236	3236	3236
3074	3074	3074	3074	3236	3398	3398	3398	3398	3398
3236	3236	3236	3236	3398	3883	3883	3883	3883	3883
3398	3398	3398	3398	3883	4045	4045	4045	4045	4045
3560	3883	3883	3883	4045	4207	4207	4207	4207	4207
4045	4045	4045	4045	4207	4692	4692	4692	4692	4692
4207	4207	4207	4207	4692	4854	4854	4854	4854	4854
4369	4692	4692	4692	4854	5016	5016	5016	5016	5016
4854	4854	4854	4854	5016	5501	5501	5501	5501	5501
5016	5016	5016	5016	5501	5663	5663	5663	5663	5663
5178	5501	5501	5501	5663	5825	5825	5825	5825	5825
5663	5663	5663	5663	5825	6310	6310	6310	6310	6310
5825	5825	5825	5825	6310	6472	6472	6472	6472	6472
5987	6310	6310	6310	6472	6634	6634	6634	6634	6634
6472	6472	6472	6472	6634	7119	7119	7119	7119	7119
6634	6634	6634	6634	7119	7281	7281	7281	7281	7281
6796	7119	7119	7119	7281	7443	7443	7443	7443	7443
7281	7281	7281	7281	7443	7928	7928	7928	7928	7605
7443	7443	7443	7443	7928	8090	8090	8090	8090	7928
7605	7928	7928	7928	8090	8252	8252	8252	8252	8090
8090	8090	8090	8090	8252	8737	8737	8737	8414	8252
8252	8252	8252	8252	8737	8899	8899	8899	8737	8414
8414	8737	8737	8737	8899	9061	9061	9061	8899	8737

Tabla 7.3: Conteos de $n_o(i)$ reiniciados en tiempo T_s

7.2 Estabilidad

Para finalizar, se presentan a continuación los resultados de la estimación de estabilidad con los datos recolectados del sistema diseñado en FPGA. Con base en la teoría presentada en el **capítulo 1**, podemos utilizar los datos de frecuencia con un tiempo de muestreo determinado para realizar posteriormente cálculos estadísticos; en este caso, la desviación de Allan AVAR.

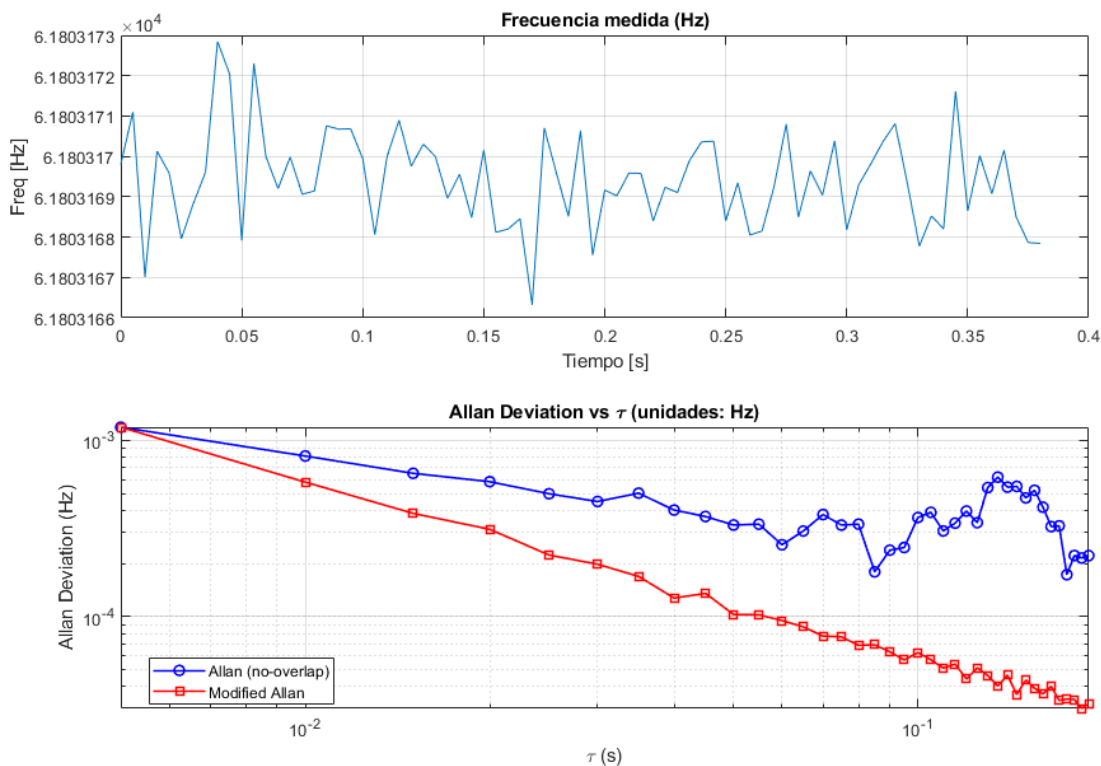


Figura 7.8: Cálculo de ADEV y MDEV para señal ideal con ruido blanco.

En la Figura 7.8 se pueden observar los resultados de la simulación de la estimación de AVAR Y MDEV la simulación tiene las características de la señal medida en FPGA, $T_s = 5ms$, $T = 16,18034\mu S$, y se le aplica ruido blanco con una desviación típica de 1×10^{-3} esto se debe a que, si se intenta calcular AVAR sin ninguna desviación, una señal que se mantuviera perfecta en el tiempo no tendría desviación.

La Figura 7.9 muestra el calculo de AVAR y MDEV, como en la simulación, pero con los datos medidos utilizando la frecuencia instantánea explicado anteriormente reiniciando los contadores en un tiempo de muestreo especifico, se puede observar que, para ambas estimaciones, la desviación sigue un patron similar. Sin embargo, para los resultados con el algoritmo, la magnitud de la estabilidad es mucho menor por los grados de magnitud que se muestran alrededor de 10×10^{-3} , algo que también se puede destacar es que el comportamiento es similar al de la simulación.

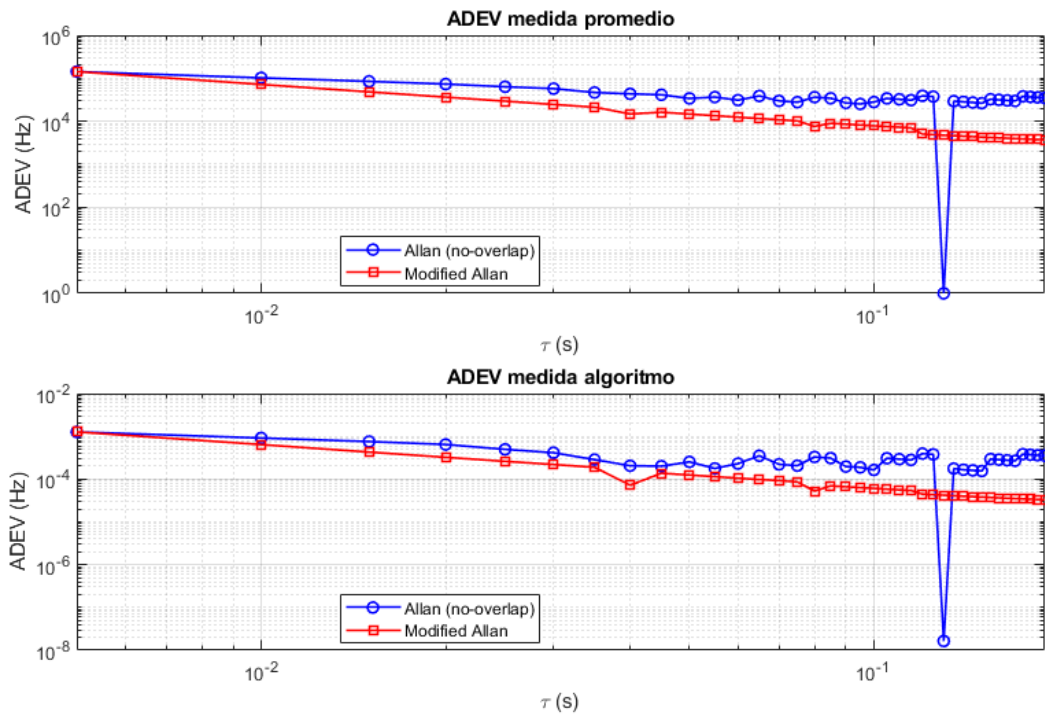


Figura 7.9: Calculo de ADEV y MDEV para señal ideal con ruido blanco.

Capítulo 8

Conclusiones

En este trabajo de investigación se presenta un diseño digital para la implementación de coincidencia de pulsos por medio de Circuitos lógicos programables; en este caso, FPGA, se presenta el proceso de diseño, así como las barreras y limitaciones que se encuentran al querer almacenar y transmitir los pares de coincidencias capturados, se explica como es el proceso de diseño en VHDL y la utilización de las plataformas (i.e., Platform Designer, Quartus) necesarias para crear un Hardware que realice la estimación de frecuencia en un procesamiento posterior. se encontró que la utilización de protocolos de comunicación internos permite transmitir la información de manera mas práctica, como lo son los protocolos basados en memoria mapeada, en este caso el protocolo Avalon MM, teniendo como limitación su velocidad de transferencia.

Como objetivo, esta investigación se enfoco en realizar y diseñar un Hardware digital capaz de almacenar los conjuntos de coincidencias. en el capitulo 6.1 Se muestran las consideraciones de diseño. Como resultado de la experimentación, se observa que no es un problema trivial, debido a que, conforme se trabaje con mayores frecuencias, los recursos y la selección de procesamiento de los datos deben ser tomados con mayor cuidado, se observo como fundamental el uso de memorias FIFO para la rapida transferencia de las coincidencias, así como protocolos de comunicación mas rápidos. Otros recursos que deben ser tomados en cuenta en la memoria caché, o memoria que se encuentra disponible de rapido acceso. Estas memorias suelen estar dentro del circuito integrado, por lo cual es mas rápido interactuar con ellas, pero suelen ser

limitadas en comparación con una memoria externa fuera del circuito integrado. El manejo de códigos de descripción de Hardware tiene una curva de aprendizaje mas lenta debido a que se debe tener en cuenta que el código representa circuitos digitales, con las limitaciones que esto conlleva. Programar todos los módulos necesarios puede ser una tarea compleja que requiera bastante tiempo de implementación. Por ello, se encuentran IP (propiedad intelectual por sus siglas en ingles) de libre acceso que pueden ser utilizadas e incluso resultar en un mejor desempeño que el código escrito por el usuario, ya que estas hacen llamadas a configuraciones del hardware creado por el fabricante. Una buena combinación entre el código de descripción de hardware proporcionado por el usuario y la implementación correcta de IP es una receta viable para lograr un buen diseño.

Otro objetivo y producto de esta investigación es plantear un estimador del conjunto de coincidencias recolectadas. esto se debe a que, cuando se generan las coincidencias, estas forman diferentes patrones y no necesariamente la ultima coincidencia tiene la mejor estimación de la frecuencia. esto crea la necesidad de utilizar un estimador que otorgue un solo valor para todo el conjunto en un tiempo designado en este trabajo se presenta un algoritmo como estimador, con el cual se busca aproximar las coincidencias con menor desviación de fase y se compara con estimadores promedio, la mediana del conjunto de datos y la ultima coincidencia. Se realizaron diferentes mediciones y, como resultado, se obtiene que, con el algoritmo, la desviación disminuye y los datos muestran un menor error en comparación con los estimadores anteriores. Este se utiliza como un estimador por software.

Al final, se plantean las bases para la estimación de estabilidad de osciladores mediante el método de coincidencia de pulsos la importancia de utilizar estimadores demostró ser clave; de esta manera, se puede mejorar el proceso de medición. También es importante conocer el proceso de promediación para saber si el calculo de AVAR es confiable[12]; por eso, una ventaja de conocer el proceso de medición es saber cómo se recolectan y procesan los datos para obtener y_n de esta forma, se pueden estudiar los efectos que pueden afectar el resultado con mayor profundidad en futuras investigaciones. se encontró una relación entre los datos obtenidos y las simulaciones; sin embargo, es necesario profundizar en el estimador y realizar

mas mediciones con diferentes tiempos de muestreo y comparaciones con patrones de mayor estabilidad.

Se demuestra que el proceso y el hardware diseñados muestran una mejora en el proceso de coincidencia de pulsos. en los resultados, se presenta una comparativa de como se reduce el error, comparando los resultados obtenidos con el valor de frecuencia de referencia. estos valores tratan de asemejar números irracionales dentro de las posibilidades de los instrumentos electrónicos de generarlos, para que, de esta manera, el método de coincidencia no se vea beneficiado con valores que son múltiplos entre si, los cuales dan buenas estimaciones en las primeras coincidencias.

Tambien se observa como la fase inicial de las mediciones afecta el error inicial de medición; por esto, se propuso un algoritmo para corregir la fase inicial y, de este modo, lograr un procesamiento posterior. Al final de esta investigación se muestran los resultados utilizando los productos elaborados durante la investigación. se presenta como es el proceso de diseño de un sistema digital, asi como un estimador que puede ser utilizado con los datos obtenidos. se observa que hay una mejora en los resultados y como pueden utilizarse como base para la estimación de estabilidad, se presenta como la base para futuras investigaciones.

Apéndice A

Listado de Código

Código en Matlab para calcular ADEV, MDEV con los datos obtenidos del sistema diseñado.

```
1
2
3 % Parámetros del oscilador y muestreo
4 T_osc = 16.1804e-6;           % periodo del oscilador (s)
5 f_osc = 1./T_osc;           % frecuencia portadora (Hz)
6 Ts = 0.005;                 % periodo de muestreo (s)
7 T_total = 0.386465600000000; % duración total (s)
8
9 % Vector tiempo y número de muestras
10 N = round(T_total / Ts);
11 t = (0:N-1)' * Ts;
12
13 % Simulación de la frecuencia medida (Hz)
14 rng(0);
15 sigma_freq_Hz = 1e-3;       % desviación típica en Hz (ajustable)
16
17 freq_Hz = f_osc + sigma_freq_Hz * randn(N,1);
18
19 freq_Hz = Yn(1:end -1);
20 freq_Hz = y(1:end -1);
21
22 % Tiempo de muestreo para allan (tau0)
23 tau0 = Ts;
24
25 % Calcular ADEV y Modified ADEV sobre series en Hz
26 [taus_Hz, adev_Hz, mdev_Hz] = allan_deviation(freq_Hz, tau0);
27
28 % Graficar solo las desviaciones (Hz)
29
30 % figure;
31 % subplot(2,1,1);
32 % plot((1:length(freq_Hz)).*Ts, freq_Hz);
33 % xlabel('Tiempo [s]');
34 % ylabel('Freq [Hz]');
35 % title('Frecuencia medida (Hz)');
36 % grid on;
37
38 figure;
```

```

39 subplot(2,1,1);
40 scatter((1:length(freq_Hz)).*Ts, freq_Hz, 12, 'filled', 'MarkerFaceColor',[0 0.4470 0.7410]);
41 xlabel('Tiempo [s]');
42 ylabel('Freq [Hz]');
43 title('Frecuencia medida (Hz)');
44 grid on;
45
46 subplot(2,1,2);
47 loglog(taus_Hz, adev_Hz, 'b-o', taus_Hz, mdev_Hz, 'r-s', 'LineWidth', 1.2);
48 xlabel('\tau (s)');
49 ylabel('Allan Deviation (Hz)');
50 legend('Allan (no-overlap)', 'Modified Allan', 'Location', 'best');
51 title('Allan Deviation vs \tau (unidades: Hz)');
52 grid on;
53
54 % Mostrar algunos valores resumen
55 disp(' \tau (s) , ADEV_Hz , MODADEV_Hz ');
56 for k = 1:min(length(taus_Hz), 10)
57     fprintf('%8.6e , %8.6e , %8.6e\n', taus_Hz(k), adev_Hz(k), mdev_Hz(k));
58 end
59
60 %% Función: calcula ADEV y Modified ADEV
61 function [taus, adev, mdev] = allan_deviation(x, tau0, max_m)
62     % x      : vector de muestras (Hz)
63     % tau0   : tiempo de muestreo en segundos
64     % max_m  : máximo m a evaluar (opcional)
65     if nargin < 3
66         max_m = floor(length(x)/2);
67     end
68
69     N = length(x);
70     taus = [];
71     avar = [];
72     mvar = [];
73
74     y = x(:); % asegurar columna
75
76     for m = 1:max_m
77         M = floor(N / m);
78         if M < 2
79             break
80         end
81
82         % --- AVAR no-overlap (promedios en bloques de tamaño m) ---
83         avg_x = zeros(M,1);
84         for i = 1:M
85             idx1 = (i-1)*m + 1;
86             idx2 = i*m;
87             avg_x(i) = mean(y(idx1:idx2));
88         end
89         dif = avg_x(2:end) - avg_x(1:end-1);
90         avar_m = 0.5 * mean(dif.^2);
91
92         % --- Modified Allan variance (overlapping) ---
93         h = ones(m,1)/m;
94         y_avg = filter(h,1,y); % longitud N
95         y_avg = y_avg(m:end); % recortar transitorio
96         dif2 = diff(y_avg);
97         K2 = length(dif2);
98         if K2 > 0
99             mvar_m = (1 / (2 * K2)) * sum(dif2.^2);

```

```
100     else
101         mvar_m = NaN;
102     end
103
104     avar = [avar; avar_m];
105     mvar = [mvar; mvar_m];
106     taus = [taus; m * tau0];
107 end
108
109 adev = sqrt(avar);
110 mdev = sqrt(mvar);
111 end
```

Apéndice B

Tablas de resultados

B.0.1 Resultados Algoritmo propuesto

Tabla B.1: Resultado del Algoritmo

Frecuencia (Hz)	Segundos
161.802631578947	4.90E-05
161.802625820569	1.46E-03
161.802626005930	1.51E-03
161.802625914765	2.97E-03
161.802626005930	3.01E-03
161.802625883711	4.42E-03
161.802625945483	4.47E-03
161.802626005930	4.52E-03
161.802625914765	5.92E-03
161.802625960717	5.97E-03
161.802625933236	7.42E-03
161.802625969818	7.47E-03
161.802625945483	8.93E-03
161.802625975869	8.97E-03
161.802625927971	1.04E-02
161.802625954198	1.04E-02
161.802625980184	1.05E-02
161.802625937835	1.19E-02
161.802625960717	1.19E-02
161.802625945483	1.34E-02
161.802625965777	1.34E-02
161.802625933236	1.48E-02
161.802625951587	1.49E-02
161.802625969818	1.49E-02

Tabla B.2: Frecuencia vs. Tiempo Segundo conjunto de resultados

Frecuencia (Hz)	Segundos
311.4144543	4.330E-04
311.414454277286	8.670E-04
311.414454277286	1.300E-03
311.414454277286	1.732E-03
311.414454277286	2.164E-03
311.414454277286	2.595E-03
311.414454277286	3.027E-03
311.414454126089	6.228E-03
311.414454135914	6.661E-03
311.414454144541	7.093E-03
311.414454152175	7.526E-03
311.414454158979	7.960E-03
311.414454165081	8.393E-03
311.414454170585	8.827E-03
311.414454175574	9.258E-03

Tabla B.3: Frecuencia vs. Tiempo Tercer conjunto de resultados

Frecuencia (Hz)	Segundos
1.956521739130430	1.600E-05
1.956521739130430	3.100E-05
1.956512735556010	3.208E-03
1.956512778235770	3.223E-03
1.956512820512820	3.238E-03
1.956512862392810	3.253E-03
1.956512903881320	3.268E-03
1.956512778235770	6.446E-03
1.956512799424280	6.461E-03
1.956512820512820	6.476E-03
1.956512841502090	6.491E-03
1.956512862392810	6.506E-03
1.956512792372590	9.684E-03
1.956512806464860	9.699E-03
1.956512820512820	9.714E-03
1.956512834516650	9.729E-03
1.956512848476580	9.744E-03

Tabla B.4: Frecuencia vs. Tiempo Cuarto conjunto de resultados

Frecuencia (Hz)	Segundos
271.826812059012	9.950E-04
271.826812428078	1.109E-03
271.826812059012	1.988E-03
271.826812253563	2.102E-03
271.826812191103	3.094E-03
271.826812159002	4.087E-03
271.826812253563	4.201E-03
271.826812139453	5.080E-03
271.826812216362	5.194E-03
271.826812191103	6.188E-03
271.826812172832	7.181E-03
271.826812227074	7.295E-03
271.826812159002	8.174E-03
271.826812206933	8.288E-03
271.826812191103	9.281E-03

Tabla B.5: Frecuencia vs. Tiempo Quinto conjunto de resultados

Frecuencia (Hz)	Segundos
208.721929824561	7.270E-04
208.721930243669	1.336E-03
208.721930095886	2.063E-03
208.721930243669	2.671E-03
208.721930153961	3.400E-03
208.721930243669	4.009E-03
208.721930179269	4.735E-03
208.721930132024	5.463E-03
208.721930193439	6.073E-03
208.721930153961	6.801E-03
208.721930202498	7.410E-03
208.721930168693	8.139E-03
208.721930208789	8.746E-03
208.721930179269	9.473E-03

Tabla B.6: Frecuencia vs. Tiempo Sexto conjunto de resultados

Frecuencia (Hz)	Segundos
216.394299287410	2.710E-04
216.394299287410	5.380E-04
216.394299287410	8.070E-04
216.394299287410	1.074E-03
216.394299287410	1.342E-03
216.394299287410	1.610E-03
216.394299287410	1.878E-03
216.394299287410	2.147E-03
216.394299287410	2.415E-03
216.394299287410	2.685E-03
216.394299287410	2.952E-03
216.394299287410	3.221E-03
216.394299287410	3.489E-03
216.394299287410	3.757E-03
216.394299287410	4.026E-03
216.394299287410	4.294E-03
216.394299287410	4.563E-03
216.394299287410	4.830E-03
216.394299287410	5.099E-03
216.394299287410	5.367E-03
216.394299287410	5.635E-03
216.394299287410	5.904E-03
216.394299287410	6.172E-03
216.394299287410	6.441E-03
216.394299287410	6.708E-03
216.3942993	6.977E-03
216.3942993	7.245E-03
216.3942993	7.513E-03
216.3942993	7.782E-03
216.3942993	8.049E-03
216.3942993	8.318E-03
216.3942993	8.585E-03
216.3942993	8.853E-03
216.3942993	9.122E-03
216.3942993	9.390E-03
216.3942993	9.660E-03

Apéndice C

Extracto de coincidencias recolectadas

nx(i)	no(i)	Δn_x	Δn_o	nx_i_2	nxo_2	Δn_x	Δn_o
0	0	0	0	0	464	0	0
1	146	1	146	3	949	3	485
2	308	1	162	4	1111	1	162
3	470	1	162	5	1273	1	162
6	955	3	485	8	1758	3	485
7	1117	1	162	9	1920	1	162
8	1279	1	162	10	2082	1	162
11	1764	3	485	13	2567	3	485
12	1926	1	162	14	2729	1	162
13	2088	1	162	15	2891	1	162
16	2573	3	485	16	3053	1	162
17	2735	1	162	18	3376	2	323
18	2897	1	162	19	3538	1	162
19	3059	1	162	20	3700	1	162
21	3382	2	323	21	3862	1	162
22	3544	1	162	23	4185	2	323
23	3706	1	162	24	4347	1	162
24	3868	1	162	25	4509	1	162
26	4191	2	323	26	4671	1	162
27	4353	1	162	29	5156	3	485
28	4515	1	162	30	5318	1	162
29	4677	1	162	31	5480	1	162
32	5162	3	485	34	5965	3	485
33	5324	1	162	35	6127	1	162
34	5486	1	162	36	6289	1	162
37	5971	3	485	39	6774	3	485
38	6133	1	162	40	6936	1	162
39	6295	1	162	41	7098	1	162
42	6780	3	485	44	7583	3	485

Tabla C.1: Coincidencias y diferencias Δn_x y Δn_o

Referencias

- [1] E. Dahlman, S. Parkvall y J. Sköld. «Chapter 26 - RF Technologies at mm-Wave Frequencies». En: *5G NR (Second Edition)*. Ed. por E. Dahlman, S. Parkvall y J. Sköld. Second Edition. Academic Press, 2021, págs. 527-550. ISBN: 978-0-12-822320-8. DOI: <https://doi.org/10.1016/B978-0-12-822320-8.00026-X>. URL: <https://www.sciencedirect.com/science/article/pii/B978012822320800026X>.
- [2] E. Rubiola, M. Lenczner, P.-Y. Bourgeois y F. Vernotte. «The Ω Counter, a Frequency Counter Based on the Linear Regression». En: *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control* 63.7 (2016), págs. 961-969. DOI: [10.1109/TUFFC.2016.2570604](https://doi.org/10.1109/TUFFC.2016.2570604).
- [3] I. N. John, P. W. Kamaku, D. K. Macharia y N. M. Mutua. «Error Detection and Correction Using Hamming and Cyclic Codes in a Communication Channel». En: *Pure and Applied Mathematics Journal* 5.6 (2017), págs. 220-231. DOI: [10.11648/j.pamj.20160506.17](https://doi.org/10.11648/j.pamj.20160506.17). URL: <https://doi.org/10.11648/j.pamj.20160506.17>.
- [4] J. Fourier. *Théorie analytique de la chaleur*. Basado en trabajos presentados en 1807 y 1811 ante la Academia de Ciencias de París. Paris: Chez Firmin Didot, père et fils, 1822. URL: <https://www3.nd.edu/~powers/ame.20231/fourier1878.pdf>.
- [5] D. D. Kim, J. Kim, C. Cho y D. Lim. «A GHz-digital clock jitters in time and frequency». En: (2007). Conference paper, A-SSCC 2007, IBM Research. URL: <https://research.ibm.com/publications/a-ghz-digital-clock-jitters-in-time-and-frequency>.
- [6] A. A. Wojciechowski, K. Marcinek y W. A. Pleskacz. «Relative Jitter Measurement Methodology and Comparison of Clocking Resources Jitter in Artix 7 FPGA». En:

- Electronics* 12.20 (2023), pág. 4297. DOI: [10.3390/electronics12204297](https://doi.org/10.3390/electronics12204297). URL: <https://doi.org/10.3390/electronics12204297>.
- [7] D. Singh, K. A. Jenkins et al. «On-Chip Circuit for Measuring Period Jitter and Skew of Clock Distribution Networks for CICC 2007». En: (2007). Conference paper, IBM Research. URL: <https://research.ibm.com/publications/on-chip-circuit-for-measuring-period-jitter-and-skew-of-clock-distribution-networks>.
- [8] «IEEE Standard for Jitter and Phase Noise». En: *IEEE Std 2414-2020* (2021), págs. 1-42. DOI: [10.1109/IEEESTD.2021.9364950](https://doi.org/10.1109/IEEESTD.2021.9364950).
- [9] Taitien. *NI-10M-3500 Series, Double Oven Controlled Crystal Oscillator*. Accessed: July 21, 2025. 2024.
- [10] D. W. Allan y J. Levine. «A Historical Perspective on the Development of the Allan Variances and Their Strengths and Weaknesses». En: *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control* 63.4 (2016), págs. 513-519. DOI: [10.1109/TUFFC.2016.2524687](https://doi.org/10.1109/TUFFC.2016.2524687).
- [11] F. De Marchi, M. K. Plumaris, E. A. Burt y L. Iess. «An Algorithm to Estimate the Power Spectral Density from Allan Deviation». En: *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control* 71.4 (2024). Cited by: 0, págs. 506-515. DOI: [10.1109/TUFFC.2024.3372395](https://doi.org/10.1109/TUFFC.2024.3372395). URL: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85187997509&doi=10.1109%2fTUFFC.2024.3372395&partnerID=40&md5=b0a7f5ed9dcc209c2e6896d32fe35617>.
- [12] S. Stein. «The allan variance - Challenges and opportunities». En: *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control* 57.3 (2010). Cited by: 24, págs. 540-547. DOI: [10.1109/TUFFC.2010.1445](https://doi.org/10.1109/TUFFC.2010.1445). URL: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-77949421375&doi=10.1109%2fTUFFC.2010.1445&partnerID=40&md5=57d1bdb91240a1f21703544f3011e1ab>.
- [13] E. Rubiola. «On the measurement of frequency and of its sample variance with high-resolution counters». En: *Review of Scientific Instruments* 76.5 (mayo de 2005). ISSN: 1089-7623. DOI: [10.1063/1.1898203](https://doi.org/10.1063/1.1898203). URL: <http://dx.doi.org/10.1063/1.1898203>.

- [14] J. A. Barnes et al. *Characterization of Frequency Stability*. Inf. téc. NBS Technical Note 394. National Bureau of Standards, 1970. URL: <https://tf.boulder.nist.gov/general/pdf/264.pdf>.
- [15] D. Howe, R. Beard, C. Greenhall, F. Vernotte, W. Riley y T. Peppler. «Enhancements to GPS operations and clock evaluations using a “total” Hadamard deviation». En: *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control* 52 (sep. de 2005), págs. 1253-1261. DOI: [10.1109/TUFFC.2005.1515160](https://doi.org/10.1109/TUFFC.2005.1515160).
- [16] Z. Chunlei, Z. Qi e Y. Shuhua. «Power law noise identification using the LAG 1 autocorrelation by overlapping samples». En: *2011 10th International Conference on Electronic Measurement & Instruments (ICEMI)*. Vol. 2. 2011, págs. 110-113. DOI: [10.1109/ICEMI.2011.6035615](https://doi.org/10.1109/ICEMI.2011.6035615).
- [17] E. Rubiola y F. Vernotte. «The Companion of the Enrico’s Chart for Phase Noise and Two-Sample Variances». En: *IEEE Transactions on Microwave Theory and Techniques* 71.7 (2023), págs. 2996-3025. DOI: [10.1109/TMTT.2023.3287841](https://doi.org/10.1109/TMTT.2023.3287841).
- [18] W. J. Riley y D. A. Howe. *Handbook of Frequency Stability Analysis*. Special Publication (NIST SP 1065). Accessed June 9, 2024. Gaithersburg, MD: National Institute of Standards y Technology (NIST), 2008. URL: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=50505.
- [19] E. Rubiola. «On the measurement of frequency and of its sample variance with high-resolution counters». En: *Proceedings of the Joint IEEE International Frequency Control Symposium and Precise Time and Time Interval Systems and Applications Meeting*. Vancouver, Canada, 2005, págs. 46-49. URL: <https://arxiv.org/abs/physics/0411227>.
- [20] D. A. Howe y J. A. All an. *A modified “Allan variance” with increased oscillator characterization ability*. Inf. téc. TN-394. Gaithersburg, MD: National Institute of Standards y Technology (NIST), 1981. URL: <https://tf.nist.gov/general/pdf/560.pdf>.
- [21] S. Johansson. «New frequency counting principle improves resolution». En: *Proceedings of the 2005 IEEE International Frequency Control Symposium and Exposition, 2005*. 2005, pág. 8. DOI: [10.1109/FREQ.2005.1574007](https://doi.org/10.1109/FREQ.2005.1574007).

- [22] J. A. Barnes y D. W. Allan. *Variances Based on Data with Dead Time Between the Measurements*. Inf. téc. Technical Note 1318. Accessed: 2025-10-22. National Institute of Standards y Technology (NIST), 1990. URL: <https://tf.nist.gov/general/pdf/878.pdf>.
- [23] D. D. Chowdhury. *NextGen Network Synchronization*. Springer Nature Switzerland, 2021. ISBN: 978-3-030-71179-5. DOI: [10.1007/978-3-030-71179-5](https://doi.org/10.1007/978-3-030-71179-5). URL: <https://link.springer.com/book/10.1007/978-3-030-71179-5>.
- [24] P. Banerjee y D. Matsakis. «Time and Frequency Measurements». En: *An Introduction to Modern Timekeeping and Time Transfer*. Springer Series in Measurement Science and Technology. Springer Nature Switzerland, 2023, págs. 109-125. DOI: [10.1007/978-3-031-30780-5_5](https://doi.org/10.1007/978-3-031-30780-5_5). URL: https://link.springer.com/chapter/10.1007/978-3-031-30780-5_5.
- [25] HP. *AN200 Fundamentals of Electronic Counters*. Visited September 25, 2025. 2000. URL: http://elektron.pol.lublin.pl/elekp/ap_notes/hp_an200.pdf.
- [26] T. Margesan. «How to Reduce Digitizing Oscilloscope Delta-Time Errors». En: *EE Times* (jul. de 2003). Publicado 12 Jul 2003. URL: [https://www.eetimes.com/how-to-reduce-digitizing-oscilloscope-delta-time-errors/#:~:text=There%20are%20six%20primary%20sources,and%20interpolation%20\(aliasing\)%20error%20..](https://www.eetimes.com/how-to-reduce-digitizing-oscilloscope-delta-time-errors/#:~:text=There%20are%20six%20primary%20sources,and%20interpolation%20(aliasing)%20error%20..)
- [27] T. S. Rathore. *Digital Measurement Techniques*. Pangbourne, UK: Alpha Science International Ltd., 2003. ISBN: 9781842650431.
- [28] D. A. González. «Método teórico de medición de cambio de frecuencia por coincidencia de pulsos». Accedido en octubre de 2025. Tesis de Doctorado. Mexicali, Baja California: Universidad Autónoma de Baja California, Facultad de Ingeniería, 2019. URL: <https://repositorioinstitucional.uabc.mx/entities/publication/ac1aeb82-97b5-4f56-952d-809612696cea/full>.
- [29] P. Boven. *Attachment from Time-Nuts mailing list*. Online PDF. Accessed: 2025-10-22. 2007. URL: <https://www.febo.com/pipermail/time-nuts/attachments/20071201/e7833af5/attachment.pdf>.

- [30] Pendulum Instruments. *CNT-90 Timer/Counter/Analyzer Programmer's Handbook*. Tenth Edition. Accessed October 2025. Pendulum Instruments / Altaria Services. Mayo de 2017. URL: <https://pendulum-instruments.com/wp-content/uploads/2022/05/CNT-90ph-2.pdf>.
- [31] V. Tyrsa. «Error reduction in conversion of analog quantities to digitized time intervals». En: *Measurement Techniques* 18 (1975). Traducido de Izmeritel'naya Tekhnika, No. 3, pp. 26–27, marzo de 1975, págs. 357-360. DOI: [10.1007/BF00818658](https://doi.org/10.1007/BF00818658).
- [32] D. Hernández-Balbuena y V. Tyrsa. «Método para la medición de frecuencia usando aproximaciones racionales». En: *Revista Mexicana de Física* 52.4 (2006). Cited by: 0, págs. 359-364. URL: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-33750029884&partnerID=40&md5=c03642c67d78458f9557726a38a8d459>.
- [33] A. Author. «Title of the TDC review». En: *Journal Name* 1.1 (2024), págs. 1-10.
- [34] D. Avalos-Gonzalez, D. Hernandez-Balbuena, V. Tyrsa, V. Kartashov, M. Kolendovska, S. Sheiko, O. Sergiyenko, V. Melnyk y F. N. Murrieta-Rico. «Application of Fast Frequency Shift Measurement Method for INS in Navigation of Drones». En: *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*. 2018, págs. 3159-3164. DOI: [10.1109/IECON.2018.8591377](https://doi.org/10.1109/IECON.2018.8591377).
- [35] D. A. González, P. L. A. R. Méndez, D. H. Balbuena, J. M. R. Zarate, O. Sergiyenko y F. N. M. Rico. «Medición de frecuencia por coincidencia de pulso y aproximaciones racionales con aplicación en micro balanzas basadas en FDS de zeolita recubierta». En: *XXXI Congreso de Instrumentación, SOMI 2016*. ISSN 2395-8499. Sociedad Mexicana de Instrumentación (SOMI). Ciudad de Guatemala, Guatemala, oct. de 2016, págs. 1-6.
- [36] B. Frost. *Measuring Performance: Using the New Metrics to Deploy Strategy and Improve Performance*. Second revised edition. Measurement International, 2000, pág. 96. ISBN: 0970247117.
- [37] M. Grabe. *Measurement Uncertainties in Science and Technology*. First edition. Springer-Verlag GmbH, 2006, pág. 270. ISBN: 3540209441.

- [38] N. M. Downie. *Fundamentals of Measurement: Techniques and Practices*. 2nd edition. New York: Oxford University Press, 1967, pág. 482. ISBN: 0195008715.
- [39] F. N. Murrieta-Rico, O. Y. Sergiyenko, V. Petranovskii, D. Hernandez-Balbuena, L. Lindner, V. Tyrsa, U. J. Tamayo-Perez y J. I. Nieto-Hipolito. «Optimization of pulse width for frequency measurement by the method of rational approximations principle». En: *Measurement* 125 (2018), págs. 463-470. ISSN: 0263-2241. DOI: <https://doi.org/10.1016/j.measurement.2018.05.008>. URL: <https://www.sciencedirect.com/science/article/pii/S0263224118303877>.
- [40] P. Arora y A. S. Gupta. «Time and Frequency Metrology». En: *Handbook of Metrology and Applications*. Ed. por D. K. Aswal, S. Yadav, T. Takatsuji, P. Rachakonda y H. Kumar. Springer, Singapore, 2023, págs. 403-407. ISBN: 978-981-99-2073-0. DOI: [10.1007/978-981-99-2074-7_19](https://doi.org/10.1007/978-981-99-2074-7_19). URL: https://link.springer.com/rwe/10.1007/978-981-99-2074-7_19.
- [41] D. Avalos-Gonzalez, O. Sergiyenko, D. Hernandez-Balbuena, V. Tyrsa, V. Kartashov, M. Rivas-Lopez, J. Rodriguez-Quiñonez, W. Flores-Fuentes y F. N. Murrieta-Rico. «Constraints definition and application optimization based on geometric analysis of the frequency measurement method by pulse coincidence». En: *Measurement* 126 (2018), págs. 184-193. ISSN: 0263-2241. DOI: [10.1016/j.measurement.2018.05.025](https://doi.org/10.1016/j.measurement.2018.05.025). URL: <https://www.sciencedirect.com/science/article/pii/S0263224118304068>.
- [42] S. Leibson. «How the FPGA Came To Be, Part 4». En: *EEJournal* (dic. de 2021). EPLDs and CPLDs Followed PALs, and Were the Next Evolutionary Step for PLDs. URL: <https://www.eejournal.com/article/how-the-fpga-came-to-be-part-4/>.
- [43] J. Schneider e I. Smalley. *¿Qué es una matriz de puertas programables en campo (FPGA)?* Consultado el 4 de agosto de 2025. 2024. URL: <https://www.ibm.com/es-es/think/topics/field-programmable-gate-arrays>.
- [44] I. Corporation. *Avalon® Interface Specifications*. Document ID: 683091, Version 20.1. Ene. de 2022. URL: <https://www.intel.com/content/www/us/en/docs/programmable/683091/20-1/introduction-to-the-interface-specifications.html>.

- [45] O. Sergiyenko, D. Hernández Balbuena, V. Tyrsa, P. L. A. Rosas Méndez, M. R. Lopez, W. Hernandez, M. Podrygalo y A. Gurko. «Analysis of jitter influence in fast frequency measurements». En: *Measurement* 44.7 (2011), págs. 1229-1242. ISSN: 0263-2241. DOI: <https://doi.org/10.1016/j.measurement.2011.04.001>. URL: <https://www.sciencedirect.com/science/article/pii/S0263224111001254>.
- [46] M. Mano y M. Ciletti. *Digital Design*. Pearson College Division, 2006. ISBN: 9780132348485. URL: <https://books.google.com.do/books?id=AC6tPAAACAAJ>.
- [47] R. Alugonda y K. Koilada. «Area-Efficient Gray Code Counter Design Using Hybrid Flip-Flop». En: *International Research Journal on Advanced Engineering Hub (IRJAEH)* 2.11 (nov. de 2024), págs. 2581-2585. ISSN: 2584-2137. DOI: [10.47392/IRJAEH.2024.0355](https://doi.org/10.47392/IRJAEH.2024.0355). URL: <https://pdfs.semanticscholar.org/fe58/e64b19b2ff1d19f1eff20b9e9ac72595c329.pdf>.
- [48] N. Malpica. *Síntesis de circuitos secuenciales síncronos: Máquinas de estados finitos*. Accedido: 2025-10-22. 2014.
- [49] F. Murrieta Rico, J. García Valenzuela, A. López Martínez y F. López Martínez. «Frequency Measurement by Principle of Rational Approximation for Aerospace Frequency Domain Mechanical Parameter Sensors». En: *Sensors & Transducers Journal* 165.2 (2014), págs. 155-161. URL: https://www.researchgate.net/publication/269980543_Frequency_Measurement_by_Principle_of_Rational_Approximation_for_Aerospace_Frequency_Domain_Mechanical_Parameter_Sensors.

Precision and Versatility in Color Sensing: A Comprehensive Characterization of the TCS3200 Using Time Domain Analysis

Misael Reyes I.
Facultad de Ingeniería
Universidad Autónoma de Baja California
Mexicali, México
mreyes77@uabc.edu.mx

Daniel Avalos-Gonzalez
Facultad de Ingeniería
Universidad Autónoma de Baja California
Mexicali, México
davalos@uabc.edu.mx

Abstract— This study introduces an innovative approach to characterizing the TCS3200 color sensor through pulse coincidence techniques within embedded measurement systems, contrasting with benchmarks from professional-grade equipment. The investigation unveils the sensor's versatility in a broad range of applications, from environmental monitoring to health diagnostics, by leveraging frequency domain sensors for superior noise immunity and precision. Despite widespread utilization, a notable gap in methodological documentation, especially in frequency measurement, often leads to nonlinear responses at elevated frequencies. Through the development of a prototype allowing precise adjustments in color and positioning, alternative measurement methodologies are explored, aiming to enhance sensor characterization. Results, demonstrating a minimum relative error of 0.48 percent and an average error rate of 5.41 percent, underscore the method's efficacy, while also identifying opportunities for future improvements in accuracy, precision, and real-time data analysis. This research not only expands the TCS3200 sensor's applicability across various engineering disciplines but also lays the groundwork for further investigation into automatic measurement techniques and application-specific research, striving to surpass existing limitations and unlock novel possibilities in sensor technology.

Keywords—TCS3200, pulse coincidence, FDS sensor, color sensor, measurement, frequency

I. INTRODUCTION

The development and application of Frequency Domain Sensors (FDS) have evolved significantly since their inception in the early 20th century. Initially, Frequency Domain Sensors (FDS) did not garner significant attention, as the frequency measurement methods available in 1930 (direct counting and reciprocal counting) lacked the accuracy achieved in contemporary times with the advent of frequency counters. It was this advancement that led to FDS receiving increased focus [1], [2]. FDS sensors, as an informative parameter (frequency, period, duty cycle, time interval, phase shift, etc.), exhibit several advantageous features compared to voltage output sensors parameters (voltage and current). Notably, they offer high noise immunity, a wide dynamic range, and high accuracy of frequency standards [3], [4], [5], [6]. Frequency signals are

not confined by the value of voltage source; for example, a dynamic range of 100 dB can be readily achieved [7]. Moreover, FDS sensors exhibit high precision, as frequency references tend to be more stable than voltage references, such as a crystal quartz oscillator [8]. Additionally, there is no need for an analog-to-digital converter when using pulse coincidence, which serves as an ideal integrator that can operate indefinitely. Furthermore, multiplexers for frequency sensors and transducers are very simple and do not contribute any significant errors [6]. Furthermore, it is crucial to address the limitations and drawbacks associated with FDS. Understanding these limitations underscores the benefits of frequency domain analysis and the utility of frequency signals [6].

TCS3200 is a color FDS that has been used in various fields and applications to measure the color of objects. The pulse coincidence method and frequency counters can be used to obtain his frequency output. Frequency counters, which are typically specialized laboratory equipment for measuring frequency, are also known as time-to-digital converters (TDC) and frequency-to-digital converters [9]. In pulse coincidence method, the pulses of an unknow signal and a reference oscillator are counted and recorded when these signals produce a coincidence [10].

II. COLOR SENSOR AND THEIR APPLICATIONS

Color is determined by the wavelengths of light an object absorbs and reflects, based on its molecular structure, and is detectable by the human eye within the 400–700 nm range [11]. Color sensors detect and measure light intensity across different wavelengths, finding use in various fields like water quality, artificial intelligence, and agriculture [12], [13], [14].

Studies on the TCS3200 sensor highlight its versatility, refer to Table I. For example, it has been used to detect color changes in porphyrins, showing a nonlinear response due to measurement noise, sensitivity, LED intensity, and firmware [15]. It has also been employed in developing a pulse oximetry sensor that measures hemoglobin wavelength for accurate

blood SpO2 determination [16], and in detecting glucose concentration by correlating color coordinates with wavelengths [17].

The TCS3200 serves as a 1D and 2D scanner for characterizing electron beam profiles, with color changes in glass plates indicating radiation dose [18]. It detects cyanide concentrations linearly within 0-20 μM [19]. Another application involves blood analysis via colorimetry, focusing on the blue-to-red absorption ratio to determine erythrocyte and hemoglobin levels. The findings are based on the red-to-blue absorption ratio, with measurement times ranging from 0.5 to 2.5 seconds [20]. Additionally, the sensor has been applied in the detection of rhodamine B, which has been shown to affect the cerebellum tissue and brainstem in *Rattus norvegicus* with increasing doses [21], [22]. The measurement of calcium levels in blood using the TCS3200 required a biochip for the separation of blood plasma. The outcomes facilitated the classification into three categories: hypo, normal, and hypercalcemia, indicating low, normal, and high calcium levels, respectively [23].

Various applications using this sensor exist across engineering disciplines, though many studies do not detail their frequency measurement methods, often relying on microcontrollers or Arduino with the pulseIn() function. This might explain the observed nonlinear responses at high frequencies [15]. This paper explores alternative measurement methods for characterizing the sensor in embedded systems.

III. TCS3200 BASIC MEASUREMENT ALGORITHM

The output of the TCS3200 is a square wave with a 50% duty cycle [24]. The frequency of this square wave is proportional to the wavelength of the light received, and it is necessary to select the filters to obtain the RGB properties. A common method for this selection is illustrated in Fig. 1.

IV. PULSE COINCIDENCE

The pulse coincidence technique involves simultaneously counting the periods of two periodic signals. The initial reference point is established when the two signals first coincide in time since the process began. The values counted in subsequent coincidences, which indicate when both signals align, reflect the ratio of the signal frequencies with an associated error or tolerance. This principle is applied to approximate the frequency of an unknown signal.

One of these signals has a frequency f_o and period T_o . This serves as a reference signal (clock) and assist in estimating the period T_x of an unknown signal and his frequency value f_x . The relationship between f_o and f_x is determined by comparing the time intervals $N_o(i)T_o$ and $N_x(i)T_x$, where $N_o(i)$ and $N_x(i)$ denote the number of complete cycles at the i -th coincidence of T_o and T_x , respectively [28]. The unknown period T_x can be calculated with (1), and the unknown frequency can also be calculated with $f_x = 1/T_x$; it's important to note that the accuracy of this method is subject to an inherent error (T_e), which is dependent on the pulse width (t) of the signals involved: $-2t < T_e < 2t$ [29].

$$T_x = N_o(i)T_o / N_x(i) \quad (1)$$

The relationship facilitates the determination of the unknown signal's frequency, as illustrated in Fig. 2; where the T_x & T_o row represents coincidence pulses.

The state-of-the-art highlights systems designed to measure the frequency of sensor parameters, such as maximum error, measurement time, hardware, and others [30-34]. These parameters are crucial due to the need to process information, which depends on the FDS application. Table II provides a comparison of different research studies.

TABLE I. TCS3200 APPLICATIONS IN DIFFERENT FIELDS

Ref.	Application	Frequency measurement system (Hardware)	Sensor Model	Characteristics	Year
[15]	Detect changes in porphyrin reflectance characteristics upon infection with targets such as UAV borne chemical sensing	Parallax system	TCS3200-DB	1D and 2D scanner developed	2014
[16]	Measure the SpO2 of the blood	Arduino Uno	TCS3200	Applied math model	2016
[17]	Measure the concentration of glucose	A/D Converter		Uses CIE color diagram	2016
[18]	1D and 2D scanner for electron beams	Not specified	TCS3200	Minimum cost based on this method	2018
[19]	Determination of the cyanide concentration	Arduino Uno	TCS3200	Uses CIE color diagram	2020
[20]	Blood analysis Using colorimetry	Microcontroller ATmega328P	TCS3200	Lowest error of 3.3%	2020
[21]	Determine Rhodamine B level in Syrup	Arduino Uno	TCS3200	Accuracy 81.8% (at 1mg/L)	2022
[23]	Measure the proportion of calcium in the blood	Arduino Uno	TCS3200-DB	Biochip developed	2022
[25]	Cocoa Ripeness Level Sorting	Arduino Uno	TCS3200	Sorting accuracy of 70%	2023
[26]	Sense the color gradation from dissolved heavy metal	Microcontroller	TCS3200	Average accuracy 96%	2023
[27]	IoT system to measure Real Time spirulina Growth	Node MCU ESP266	TCS3200	TCS3200 method presented measure up to 1.7 while a UV-Vis measure 3.0 and above of absorbance value	2024

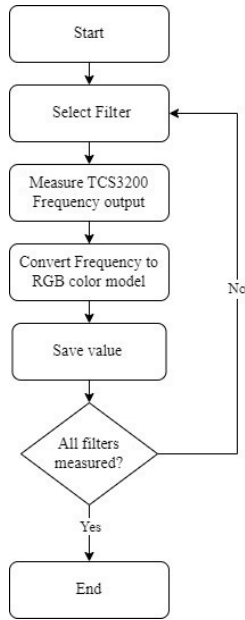


Fig. 1. RGB Filter Selection Process for Light Wavelength Detection.

V. TCS3200 SENSOR CONFIGURATION AND POSITIONING SYSTEM

To carry out the measurements, it was necessary to implement a prototype capable of alternating between different colors and maintaining a fixed position for the sensor. This requirement arises from the observation that frequency values change rapidly with varying positions of the objects. Consequently, a 3D model was developed, as shown in Fig. 3.

A TFT screen was utilized and mounted on the 3D-printed structure to facilitate color changes. This model collocates the distance of the sensor at 9 millimeters distance to the TFT screen. Additionally, a PCB was required, housing the components and connections for the screen and sensor configuration adjustments. The corresponding schematic is presented in Fig. 4.

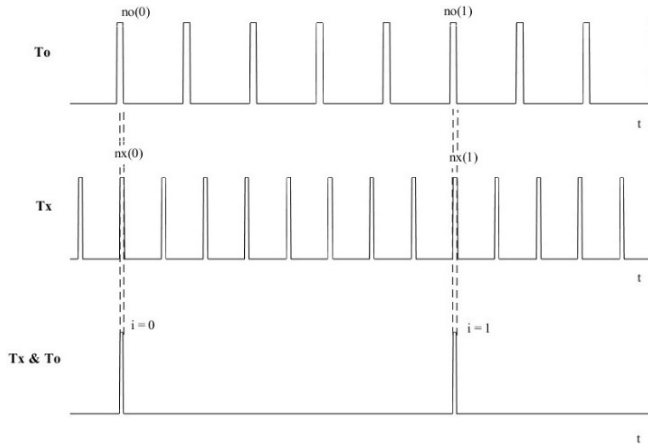


Fig. 2. Pulse coincidence representation.

TABLE II. RELATED WORK OF FDS FREQUENCY MEASUREMENT

Ref.	Application	Max Error	Measured time	Hardware	Method	Characteristics & Notes	Year
[30]	Quartz crystal balance frequency output measurement	0.1hz @ 100Mhz	1s	FPGA Spartan 3E	Count the pulses in one second gate interval	Calibrated using 10MHz standard rubidium oscillator	2016
[31]	Shift detection for INS in navigation drones	Not reported	1s	Simulation	Pulse Coincidence	Rational approximations	2018
[32]	Frequency Measurement for soil Moisture Content	0.005%*	Not reported	Not reported	Direct counting and Reciprocal	Switch between two methods	2023
[33]	Measure QCM sensor array	0.33hz@10Mhz	1s	FPGA XC6SLX9	Reciprocal counter	Multi-Channel	2023
[34]	Multidimensional Seismometric System	1hz@600 MHz	1ms	FPGA Kintex 7	Linear Regression	Multi-Channel, No Crossovers	2023

*. Percent of error

In the schematic, the connections to the microcontroller and the frequency output pinout are located on the left, whereas the connections to the TFT screen are on the right. RV1 adjusts the TFT screen's brightness, while SW3A manages the LEDs embedded in the sensor. The schematic is depicted in the PCB design shown in Fig. 5.

The PCB is designed with a two-layer copper layout to enhance stability against electrical noise. Additionally, nearly all spaces between ground connections have been filled to further mitigate interference. Fig. 6 illustrates the assembled prototype and details the connections among all components.

The pulse counter circuit depicted in Fig. 7 enables the acquisition of a pair of numbers corresponding to each approximation of the unknown frequency. This acquisition is facilitated through the pulse coincidence method, which is employed for the characterization of the target using a color sensor (TCS3200). To accomplish this, the output of the color sensor is connected to a pulse counter circuit. This setup is then synchronized with a reference oscillator signal generated by a waveform generator Tektronix AFG3102.

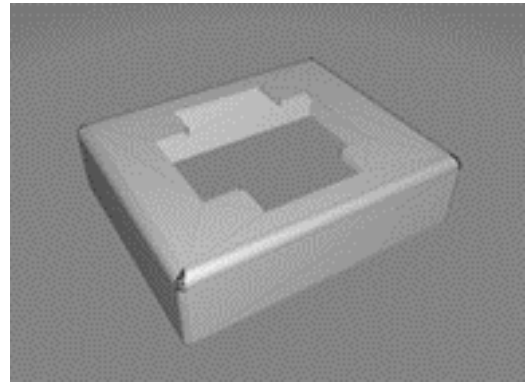


Fig. 3. 3D-Printed Mount for Positioning a TCS3200 Sensor in Front of a TFT Screen.

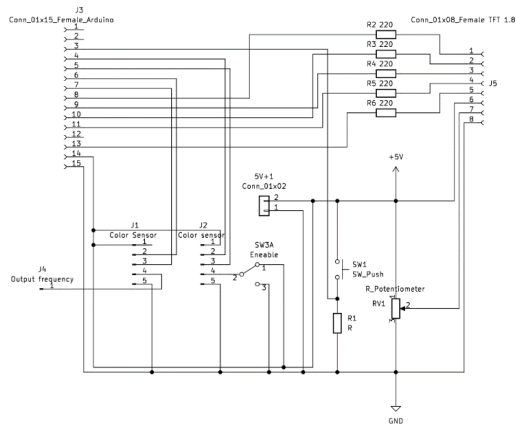


Fig. 4. Schematic of Screen Control and Sensor Configuration.

VI. EXPERIMENT

This study presents an implementation based on an embedded measurement system that utilizes pulse coincidence techniques. It also involves a comparison of these measurements against those obtained from professional-grade equipment. The Agilent 53131A frequency counter served as the standard device, offering a resolution of 10 digits, a gate time of 1 s, and a bandwidth of 225 MHz [35]. The color sensor facilitates the selection among various frequency scales, enhancing the sensor's resolution and optimizing the detection of light variations [24]. For the purposes of this experiment, these values were set to a constant 20%. Additionally, the color sensor enables switching among four filter options: red, blue, green, and no filter, with measurements conducted using each filter.

In this study, we present examples of RGB color values that were implemented, including (255, 255, 55) for a variant of yellow, (0, 0, 0) for black, (255, 0, 0) for red, (0, 255, 0) for green, and (0, 0, 255) for blue. These exemplify the broader range of colors assessed to evaluate the performance of a color sensor. Displayed on a TFT screen, these selected colors were part of the experimental setup designed to face directly the sensor.

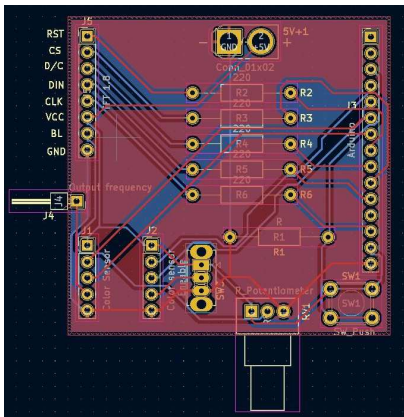


Fig. 5. PCB Layout Showcasing Microcontroller Connections, TFT Screen, and LED Control.

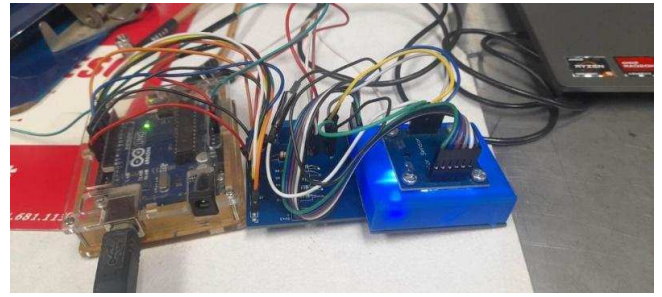


Fig. 6. Integrated Prototype Displaying Screen Control Interface and Sensor Configuration Settings.

A key indicator of measurement quality is the delta difference between measurements, including the minimum (3) and maximum (4) delta values; this is illustrated for some RGB colors. Table III shows information for RGB values (255,255,255); Table IV shows information for RGB values (0,255,0); Table V shows information for RGB values (255,0,0); Table VI shows information for RGB values (0,0,0); and Table VII shows information for RGB values (0,0,255).

VII. RESULTS

The minimum relative error (2) observed in measurements taken with the experimental pulse counter was 0.48%, and the maximum error reached 54.35%. The average error rate (5) was calculated at 5.41%, although 80% of measurements exhibited error rates lower than 3.7%, refer to Table VII. These findings suggest that implementing certain post-processing techniques could enhance the accuracy and precision of the measurements [29].

Minimum standard deviation calculated with the square root of (6) was $\sigma = 2.21$ Hz.

$$Relative\ error = \frac{|f_x - \frac{n_x}{n_0} f_0|}{f_x} \quad (2)$$

$$\Delta f_{min} = \min|f_i - f_{i-1}| \quad (3)$$

$$\Delta f_{max} = \max|f_i - f_{i-1}| \quad (4)$$

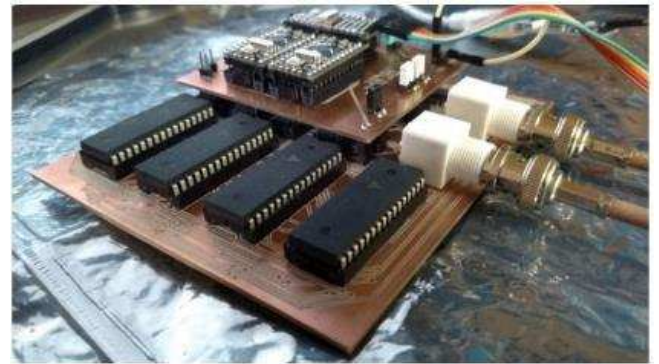


Fig. 7. Pulse counter circuit for frequency measurement [21].

$$\bar{x} = \sum \frac{f_i}{n}, \quad (5)$$

where:

f_i is the i -th frequency value measured.
 n is the number of measurements.

$$\sigma^2 = \frac{\sum(X_i - X)^2}{N}, \quad (6)$$

where:

X_i is the i -th frequency value measured.
 X is the population mean.
 N is the number of measurements.

TABLE III. DEVIATION OF MEASUREMENTS RGB (255,255,255)

RGB (255,255,255)		
Hz	Circuit	Standard
Δ min	0.3070892	0.004467
Δ max	1333.33333	69.88253

TABLE IV. DEVIATION OF MEASUREMENTS RGB (0,255,0)

RGB (0,255,0)		
Hz	Circuit	Standard
Δ min	.0003	6.50E-06
Δ max	100	6.660197

TABLE V. DEVIATION OF MEASUREMENTS RGB (255,0,0)

RGB (255,0,0)		
Hz	Circuit	Standard
Δ min	0.03730856	8.00E-07
Δ max	1111.1111	9.650533

TABLE VI. DEVIATION OF MEASUREMENTS RGB (0,0,0)

RGB (0,0,0)		
Hz	Circuit	Standard
Δ min	0.00165518	6.00E-06
Δ max	580.612245	4.497138

TABLE VII. DEVIATION OF MEASUREMENTS RGB (0,0,255)

RGB (0,0,255)		
Hz	Circuit	Standard
Δ min	0.01311673	8.00E-05
Δ max	333.333333	1.108745

TABLE VIII. RELATIVE ERROR

RGB	Red filter	Green filter	Blue filter	No filter
255,255,255	1.26%	0.48%	3.63%	3.57%
0,0255	1.63%	11.35%	2.54%	4.10%
0,255,0	2.06%	1.56%	2.10%	0.50%
255,0,0	2.66%	1.79%	1.20%	0.78%
0,0,0	0.85%	8.30%	3.47%	54.35%

VIII. CONCLUSION AND FUTURE WORK

Certain limitations were observed in the application of the coincidence method with the existing circuitry. The calculation of pulse widths was necessary to achieve coincidences when measuring specific frequencies, attributable to the pulse widths. Additionally, the measurement duration was constrained to one second, limiting the number of samples obtainable from a single measurement. Future research aims to enhance these aspects to optimize the utilization of the pulse coincidence method. This includes the ability to produce results in real-time and employ post-processing techniques that facilitate automatic measurements. Such improvements are anticipated to enhance both the accuracy and precision of the method, thereby enabling quicker analysis of results.

IX. DISCUSSION

To ensure accurate measurements, we designed a 3D-printed model for the TCS3200 sensor, addressing its high sensitivity to distance variations. Consequently, the sensor was maintained at a constant distance from the RGB screen. It is recommended to perform calibration procedures adjusting this distance while ensuring external light does not penetrate the system, except for that emitted by the sensor's integrated LEDs.

It was observed that the stability of the obtained frequencies exhibits notable variation that must be considered when measuring a sample with the TCS3200 sensor, thus continuing to limit its applications in scenarios demanding high color measurement accuracy.

The implemented frequency measurement method was conducted without applying any of the previously demonstrated post-processing criteria that improve accuracy [9], [28], [29], [31]. Future investigations will incorporate these criteria in application-specific research.

According to [6], it is possible to multiplex the sensor signals, enabling the use of the TCS3200 sensor in applications necessitating multiple sensors across extended distances, such as multiple production lines, and to be able to monitor the sensors together.

REFERENCES

- [1] S. Johansson and in Applied Physics, "New frequency counting principle improves resolution.", in Proc. Freq. Control Symp., Vancouver, BC, Canada, Aug. 29–31, 2005, pp. 628–635.
- [2] S. Y. Yurish, "High-end sensors: Challenges for the future," in *Journal of Physics: Conference Series*, Institute of Physics Publishing, Nov. 2019. doi: 10.1088/1742-6596/1379/1/012003.
- [3] J. Fraden, *Handbook of Modern Sensors*. New York, Ny Springer New York, 2010.
- [4] G. C. M. Meijer, "Smart Sensor Systems," *Smart Sensor Systems*, pp. 1–385, Dec. 2008, doi: 10.1002/9780470866931.
- [5] N. Vladimirovich. Kirianaki, *Data acquisition and signal processing for smart sensors*. J. Wiley, 2002.
- [6] S. Y. Yurish, "Sensors and transducers: frequency output versus voltage output Sensors" *Sensors & Transducer Mag.* Vol. 49, pp. 302–305.
- [7] X. Wang, W. Wong, and R. Hornsey, "A high dynamic range CMOS image sensor with inpixel light-to-frequency conversion," *IEEE Trans Electron Devices*, vol. 53, no. 12, pp. 2988–2992, Dec. 2006, doi: 10.1109/TED.2006.885642.
- [8] P. Salzenstein, A. Kuna, and F. Lefebvre, "Evaluation of the accuracy of the method for measuring state-of-the-art ultra-high stability quartz crystal oscillators," *2013 Joint European Frequency and Time Forum and International Frequency Control Symposium, EFTF/IFC 2013*, pp. 157–159, 2013, doi: 10.1109/EFTF-IFC.2013.6702072.
- [9] E. Rubiola, M. Lenczner, P. Y. Bourgeois, and F. Vernotte, "The Ω Counter, a frequency counter based on the linear regression," *IEEE Trans Ultrason Ferroelectr Freq Control*, vol. 63, no. 7, pp. 961–969, Jul. 2016, doi: 10.1109/TUFFC.2016.2570604.
- [10] D. Hernández Balbuena, O. Sergiyenko, V. Tyrsa, L. Burtseva, and M. R. López, "Signal frequency measurement by rational approximations," *Measurement*, vol. 42, no. 1, pp. 136–144, Jan. 2009, doi: 10.1016/J.MEASUREMENT.2008.04.009.
- [11] Todd Bishop and Glenn Lee, "TAOS Colometry Tutorial," TAOS, Austria, Premstaetten, Rep. 20, 2006.
- [12] V. Karthikeyan, Y. P. Visu, and E. Raja, "Integrated intelligent system for water quality monitoring and theft detection," *Water Pract Technol*, vol. 18, no. 12, pp. 3035–3047, 2023, doi: 10.2166/wpt.2023.200.
- [13] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis and Machine Vision*. Springer, 2013.
- [14] G. Gordon, *Interior lighting for designers*. Hoboken, N.J.: Wiley, 2013.
- [15] B. J. Johnson *et al.*, "Miniaturized reflectance devices for chemical sensing," *Meas Sci Technol*, vol. 25, no. 9, Sep. 2014, doi: 10.1088/0957-0233/25/9/095101.
- [16] Z. J. V Cohen, S. Haxha, and A. Aggoun, "Pulse oximetry optical sensor using oxygen-bound haemoglobin," *Opt Express*, vol. 24, no. 9, p. 10115, May 2016, doi: 10.1364/oe.24.010115.
- [17] J. S. Kim *et al.*, "A study on detection of glucose concentration using changes in color coordinates," *Bioengineered*, vol. 8, no. 1, pp. 99–104, Jan. 2017, doi: 10.1080/21655979.2016.1227629.
- [18] M. Askarbioki, M. B. Zarandi, S. Khakshourmia, S. P. Shirmardi, and M. Sharifian, "Design and fabrication of 1D and 2D optical scanner for electron beams using color centre generation," *Journal of Instrumentation*, vol. 13, no. 12, Dec. 2018, doi: 10.1088/1748-0221/13/12/T12003.
- [19] H. Singh, G. Singh, D. K. Mahajan, N. Kaur, and N. Singh, "A low-cost device for rapid 'color to concentration' quantification of cyanide in real samples using paper-based sensing chip," *Sens Actuators B Chem*, vol. 322, Nov. 2020, doi: 10.1016/j.snb.2020.128622.
- [20] L. A. Szolga and F. Mudure, "Blood (rBC/HGB) analysis using colorimetry," in *2020 8th E-Health and Bioengineering Conference, EHB 2020*, Institute of Electrical and Electronics Engineers Inc., Oct. 2020. doi: 10.1109/EHB50910.2020.9280195.
- [21] M. S. Surbakti *et al.*, "Development of Arduino Uno-Based TCS3200 Color Sensor and Its Application on the Determination of Rhodamine B Level in Syrup," *Indonesian Journal of Chemistry*, vol. 22, no. 3, pp. 630–640, 2022, doi: 10.22146/ijc.69214.
- [22] D. R. Sulistina and S. Martini, "The effect of rhodamine b on the cerebellum and brainstem tissue of rattus norvegicus," *J Public Health Res*, vol. 9, no. 2, pp. 101–104, 2020, doi: 10.4081/jphr.2020.1812.
- [23] Z. H. Al-Sawaff, Z. M. Rashid, M. F. R. Al-Okby, and F. Kandemirli, "A microcontroller based microfluidic biochip for calcium percentage detection in blood," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 26, no. 1, pp. 135–142, Apr. 2022, doi: 10.11591/ijeecs.v26.i1.pp135-142.
- [24] Taos, "Programmable color light-to frequency converter," TCS3200, TCS3210 datasheet, July. 2009.
- [25] B. Siregar, R. Pradaning, and A. Hizriadi, "Cocoa Ripeness Level Sorting System Using Integrated Computer Vision Technology On Conveyor Belt," in *ICEEIE 2023 - International Conference on Electrical, Electronics and Information Engineering*, Institute of Electrical and Electronics Engineers Inc., 2023. doi: 10.1109/ICEEIE59078.2023.10334634.
- [26] Yulkifli, K. S. Dewantara, M. B. Sari, Ramli, F. Razi, and W. Handayani, "A Low-cost Digital Spectrophotometer for Heavy Metal Absorbance Measurements," *Instruments and Experimental Techniques*, vol. 66, no. 1, pp. 147–155, Feb. 2023, doi: 10.1134/S0020441223010256/METRICS.
- [27] H. R. Lim *et al.*, "Evaluation of Real-Time Monitoring on the Growth of Spirulina Microalgae: Internet of Things and Microalgae Technologies," *IEEE Internet Things J*, vol. 11, no. 2, pp. 3274–3281, Jan. 2024, doi: 10.1109/JIOT.2023.3296525.
- [28] D. Hernandez-Balbuena, O. Sergiyenko, V. Tyrsa, L. Burtseva, and D. Hernández-Balbuena, "Method for fast and accurate frequency measurement," 2009. [Online]. Available: <https://www.researchgate.net/publication/228811071>
- [29] D. Avalos-Gonzalez *et al.*, "Constraints definition and application optimization based on geometric analysis of the frequency measurement method by pulse coincidence," *Measurement (Lond)*, vol. 126, pp. 184–193, Oct. 2018, doi: 10.1016/j.measurement.2018.05.025.
- [30] M. A. Syahbana, D. J. H. D. Santjojo, and S. P. Sakti, "High-resolution multiple channel frequency counter using spartan-3E FPGA," *Proceeding - 2016 International Seminar on Sensors, Instrumentation, Measurement and Metrology, ISSIMM 2016*, pp. 111–114, Jan. 2017, doi: 10.1109/ISSIMM.2016.7803734.
- [31] D. Avalos-Gonzalez, D. Hernandez-Balbuena, V. Tyrsa, V. Kartashov, M. Kolendovska, S. Sheiko, O. Sergiyenko, V. Melnyk, and F. N. Murrieta-Rico, "Application of fast frequency shift measurement method for ins in navigation of drones," in *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, pp. 3159–3164, 2018.
- [32] S. Takimoto, R. Shigemasa, M. Fujisono, S. Ota, and M. Futagawa, "Development of frequency measurement circuit for soil moisture content and electric conductivity measurement system with detection of soil transient response characteristics for on-site measurement," *Electronics and Communications in Japan*, vol. 106, no. 3, p. e12415, Sep. 2023, doi: 10.1002/ECJ.12415.
- [33] A. O. Triqadafi, T. N. Zafirah, H. A. Dharmawan, and S. P. Sakti, "Four-Channels High-Resolution Frequency Counter for QCM Sensor Array Using Generic FPGA XC6SLX9 Board," *Journal of Electrical and Computer Engineering*, vol. 2023, 2023, doi: 10.1155/2023/5182455.
- [34] R. Szplet, D. Sondej, P. Kwiatkowski, M. Chodnicki, J. Kowalski, and H. A. Kowalski, "Precise measurement module for laser two frequency gyroscope-based seismometer," in *2023 Joint Conference of the European Frequency and Time Forum and IEEE International Frequency Control Symposium (EFTF/IFCS)*, pp. 1–2, 2023.
- [35] Agilent, "High-performance, low-cost counters simplify and speed systems and bench frequency measurements", 53131A/132A/181A datasheet, Nov. 2006 [Revised Sep. 2006].