

CONESCAPANHONDURAS2025paper100.pdf

 Institute of Electrical and Electronics Engineers (IEEE)

Document Details

Submission ID

trn:oid:::14348:477731315

Submission Date

Jul 31, 2025, 10:40 PM CST

Download Date

Aug 12, 2025, 2:57 PM CST

File Name

CONESCAPANHONDURAS2025paper100.pdf

File Size

1.6 MB

6 Pages




4,098 Words

23,498 Characters

9% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Top Sources

- 9%  Internet sources
- 4%  Publications
- 0%  Submitted works (Student Papers)

Integrity Flags




0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Top Sources

- 9%  Internet sources
- 4%  Publications
- 0%  Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Internet	dvcon-proceedings.org	1%
2	Internet	www.coursehero.com	1%
3	Internet	hdl.handle.net	<1%
4	Internet	aisel.aisnet.org	<1%
5	Internet	ikee.lib.auth.gr	<1%
6	Internet	www.doctorado.us.es	<1%
7	Internet	die.ucu.edu.uy	<1%
8	Internet	app.inti.gob.ar	<1%
9	Internet	sftp.asee.org	<1%
10	Internet	www.diva-portal.se	<1%
11	Internet	spiritconsortium.org	<1%

12	Internet	www.leadhospitality.es	<1%
13	Internet	riunet.upv.es	<1%
14	Internet	www.kandh.com.tw	<1%
15	Internet	baixardoc.com	<1%
16	Internet	dspace.cvut.cz	<1%

Diseño de un ambiente de pruebas de señales mixtas para un detector de flancos

1st XXX
XXXX
XXXX, XXXX XXXX
XXXX@XXXX.XXXX

2nd XXX
XXXX
XXXX, XXXX XXXX
XXXX@XXXX.XXXX

3rd XXX
XXXX
XXXX, XXXX XXXX
XXXX@XXXX.XXXX

4rd XXX
XXXX
XXXX, XXXX XXXX
XXXX@XXXX.XXXX

Abstract—This paper presents the development of a functional verification environment for a mixed-signal edge detector using Real Number Modeling (RNM) methodology. The solution presente leverages Cadence's Xcelium simulator and Sim Vision visualization tool to validate and test the functionality of the device under test that has been adapted from conventional digital implementation to a wreal compatible model. The verification environment enables the controlled generation stimuli as square, sinusoidal and thootsaw waveform, while also is able to automate the validation of the DUT's outputs.

Index Terms—Mixed-signal verification, RNM, edge detector, Xcelium, SimVision, Verilog-AMS.

I. INTRODUCCIÓN

En la actualidad, se han desarrollado nuevas metodologías para la verificación de circuitos digitales y de señales mixtas. Entre ellas, destacan el uso de Verilog-AMS para modelado de señales analógicas y la aplicación de metodologías como UVM-AMS, que permiten la generación de estímulos y la verificación de sistemas mixtos [2].

Una alternativa a estos enfoques es el modelado de números reales (Real Number Modeling, RNM), el cual no depende de herramientas externas de simulación SPICE y reemplaza los componentes analógicos por representaciones discretas en punto flotante. Esto facilita su implementación y proporciona un mayor rendimiento en simulación, aunque puede afectar la precisión del modelo [3].

En la industria existen diversas herramientas para realizar este tipo de verificaciones, desarrolladas por empresas como Cadence y Synopsys, entre otras. Otras de las herramientas disponibles se encuentra VCS y Xcelium, la cual ofrece diversas funciones que facilitan tanto el desarrollo de circuitos digitales como su verificación [4].

Con estas herramientas es posible implementar un entorno de verificación adecuado para evaluar el funcionamiento de un detector de flancos en un circuito de señales mixtas, el cual no se posee, garantizando la correcta validación de su desempeño.

A. Objetivo

Desarrollar un entorno de verificación para un detector de flancos basado en Real Number Modeling (RNM) utilizando Xcelium de Cadence, con el propósito de simular señales mixtas de manera precisa, garantizando la validación efectiva del funcionamiento del dispositivo bajo prueba (DUT) en condiciones reales de operación.

II. METODOLOGÍA

A. Abreviaciones

A continuación, se listarán todos las abreviaturas que se utilizaran en el documento:

- AMS: Analog and Mixed-Signal (Señal Analógica y Mixta)
- DUT: Device Under Test (Dispositivo Bajo Prueba)
- RNM: Real Number Modeling (Modelado de Números Reales)
- HDL: Hardware Description Language (Lenguaje de Descripción de Hardware)
- SoC: System-on-Chip (Sistema en un Chip)
- UVM: Universal Verification Methodology (Metodología Universal de Verificación)
- DCILab: Laboratorio de Diseño de Circuitos Integrados
- EDA: Electronic Design Automation (Automatización de Diseño Electrónico)

B. Propuesta

Se realizará un espacio de trabajo para la verificación funcional del dispositivo bajo prueba basándose explícitamente en el método de verificación de Real Number Modeling (RNM) para la simulación más fidedigna de las entradas mixtas de señales analógicas cuadradas, de sierra y sinusoidal.

C. Justificación teórica

1) *Flujo de diseño*: Para los circuitos electrónicos de señales mixtas se deben cumplir ciertas especificaciones y requerimientos, por lo que es necesario determinar un flujo de diseño donde se establezcan y expliquen los bloques analógicos, digitales y las interfaces de comunicación entre bloques. En primera instancia es necesario desarrollar los bloques digitales y analógicos en un entorno de simulación para analizar de manera exhaustiva su desempeño bajo diversas condiciones operativas, luego integrar el sistema completa y realizar una verificación para detectar problemas de compatibilidad, sincronización o degradación de señales. Luego se ejecuta la implementación física del diseño en la que se incluye el layout, el enrutamiento y la verificación de reglas de manufactura, se realiza la extracción de parásitos y se simula la exposición del circuito a efectos físicos reales para su validación. El flujo de diseño culmina con la fabricación

y testeo del circuito integrado, garantizando que el producto final responde a los requisitos del cliente y del sistema [5].

2) *Verificación de circuitos integrados*: El ciclo de verificación de circuitos integrados es un proceso iterativo (Figura 1), orientado a validar que el diseño cumpla con los requerimientos definidos en la especificación funcional, por lo que es necesario generar un plan de verificación y así construir el entorno en el que se verificará. El entorno debe de estar compuesto por módulos funcionales que generen estímulos, monitoreen respuestas y evalúen el comportamiento del DUT. Posteriormente se debe depurar el código HDL y el entorno, identificando errores de lógica, sincronización o interpretación. Finalmente se realizan pruebas de regresión para validar la robustez del diseño bajo múltiples escenarios de trabajo y a su vez documentar lecciones aprendidas para retroalimentar futuros ciclos de verificación [6].

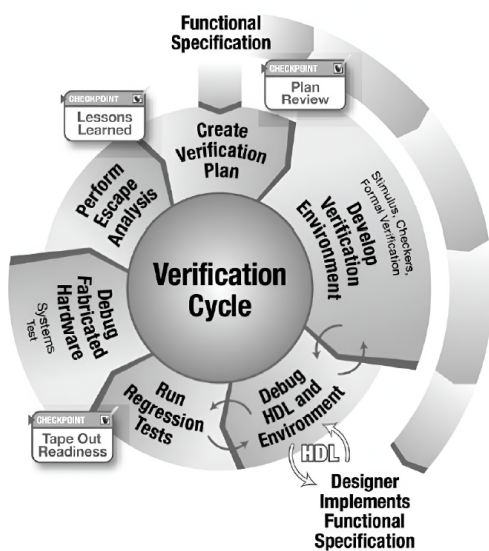


Fig. 1. Ciclo de verificación [6]

La implementación de estrategias de verificación rigurosas desde las etapas tempranas es un aspecto crítico en el desarrollo de circuitos integrados, esto debido a que el costo de corregir errores aumenta de manera exponencial conforme avanza el ciclo de vida del producto, y es necesario detectarlos en etapas tempranas del desarrollo.

3) *HDL (Hardware Description Language)*: Los HDL permiten representar lógica combinatorial y secuencial, temporización, estructuras jerárquicas y módulos reutilizables. Verilog y VHDL son los lenguajes más difundidos en la industria. Estos lenguajes facilitan la automatización del flujo de diseño, puesto que permiten simulaciones funcionales, pruebas de verificación, análisis de temporización y generación automática de netlist para su implementación en FPGA, lo que los convierte en pilares esenciales del diseño electrónico [7].

Se decidió utilizar Verilog como herramienta de trabajo en el bloque digital, debido a que cuenta con extensiones convenientes para desarrollo del proyecto. Verilog-AMS incorpora

capacidades adicionales para modelar sistemas analógicos y de señales mixtas, donde se integra el dominio discreto (digital), típico de Verilog y el continuo (analógico), que permite representar señales dependientes del tiempo. Lo que la convierte en una herramienta poderosa para simular convertidores A/D y D/A, VCO's, sensores analógicos, amplificadores y detectores de eventos analógicos. Verilog-AMS extiende las capacidades del lenguaje, de modo que, con la característica de los datos continuos real y wreal, se pueden representar magnitudes físicas como voltaje y corriente con resolución analógica. Además, facilita la descripción de fenómenos dinámicos mediante la formulación directa de ecuaciones diferenciales y algebraicas dependientes del tiempo, útil para representar circuitos integradores, osciladores y filtros. También admite descripciones estructurales jerárquicas como modelado conductual, promoviendo la reutilización y modularidad del código, los modelos desarrollados deben ser procesados por simuladores compatibles con co-simulación de dominios mixtos, esta capacidad de simular comportamientos digitales y respuesta analógicas de manera simultánea es esencial para validar el funcionamiento de sistemas mixtos reales [8].

SystemVerilog-AMS, una extensión no estandarizada del lenguaje SystemVerilog, facilita la verificación de sistemas de señales mixtas dentro de un único entorno de simulación, su principal ventaja es que permite integrar modelos analógicos directamente en entornos de verificación digitales, sin requerir herramientas de co-simulación, esto mediante el AMS Bridge, un componente que permite la transferencia bidireccional de señales entre los dominios digital y analógico dentro del mismo simulador, evitando errores de acople o pérdidas de información temporal. Esta extensión hereda todas las capacidades avanzadas del lenguaje base y se le añade compatibilidad con señales del tipo wreal, para representar magnitudes físicas. El entorno de programación y simulación se complementa con librerías especializadas para la declaración de disciplinas eléctricas, constantes matemáticas y físicas, así como módulos de conexión AMS que resuelven diferencias entre señales electrical, wreal y logic [9].

4) *Detector de flancos de señal mixta*: Se encarga de identificar cambios abruptos o transiciones relevantes dentro de una señal analógica continua, su función consiste en generar salidas digitales que marquen la ocurrencia de eventos de interés, como flancos ascendentes o descendentes. Mediante un esquema de evaluación continua, el sistema realiza un muestreo periódico de la señal y mantiene un registro del valor anterior, se calcula la variación entre las muestras consecutivas y se determina si ha ocurrido una transición significativa, esto se interpreta como un flanco válido, cuando este se detecta el sistema emite una señal digital breve, un pulso estrecho de un solo ciclo, que marca la detección del evento. Este pulso digitaliza la transición y la vuelve utilizable en sistemas digitales sincronizados. Para detección de flancos se pueden implementar dos estrategias. La primera consiste en comparar el valor instantáneo de la señal con otro predefinido, usualmente 0.5V. Si la señal cruza la referencia en dirección ascendente se considera un flanco de subida, si la cruza en

dirección descendente se considera un flanco de bajada. La segunda consiste en un análisis de la derivada de la señal, una variación positiva pronunciada en la derivada sugiera un flanco ascendente, una pendiente negativa un flanco descendente, esta estrategia es útil para señales ruidosas o poco definidas [10].

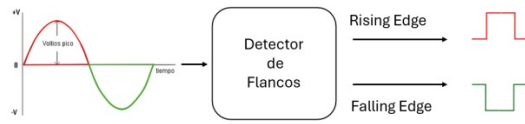


Fig. 2. Diagrama funcional de detector de flancos

5) *Circuitos de señales mixtas en la industria*: Combinar componentes analógicos y digitales dentro una misma arquitectura presenta desafíos importantes de verificación, por lo que es primordial realizarla de manera correcta para garantizar interacciones coherentes y sin errores entre ambos dominios. Uno de los principales retos en la industria es la poca experiencia de ingenieros digitales en la simulación de bloques analógicos, para abordar este problema se adoptó el modelado RNM, además herramientas como Xcelium Logic Simulator en conjunto con Spectre para co-simulación permite implementar simulaciones digitales de señales mixtas y entornos de verificación AMS completos [11].

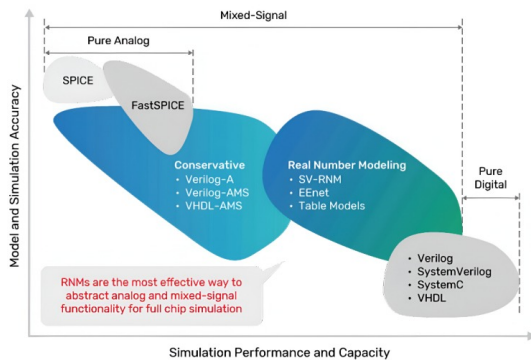


Fig. 3. Precisión del modelo y de la simulación frente al rendimiento y la capacidad en la simulación de señales [11]

Como se muestra en la Figura 3, la precisión en estas metodologías de simulación cada es más relevante y la implementación de ellas y el uso de las nuevas herramientas refleja una tendencia creciente en la industria por optimizar los flujos de verificación, reducir costos asociados a iteraciones tardías y mejorar la calidad del producto final. En consecuencia, la capacitación de los equipos de verificación en técnicas de señales mixtas y el uso de abstracciones como RNM se ha vuelto una estrategia clave para enfrentar los desafíos actuales del diseño de circuitos integrados [11].

Xcelium es una herramienta de simulación lógica que destaca frente a otros simuladores debido a su capacidad para realizar simulaciones concurrentes mediante técnicas de paralelización multicore, que reduce los tiempos de simulación

en diseños complejos a gran medida. Además, Xcelium es compatible con el RNM, lo cual permite representar señales analógicas utilizando variables real o wreal, dentro del dominio digital, reduciendo el tiempo de simulación [12].

Por su parte, Virtuoso es una herramienta que se destaca por ofrecer un entorno altamente integrado en el que se pueden realizar flujos de diseño completos desde el nivel esquemático hasta el diseño físico dentro de una misma interfaz gráfica. Esta permite la creación de esquemáticos eléctricos, simulación funcional mediante Verilog-AMS y la generación automática del layout físico, todo de manera sincronizada mediante reglas de diseño y verificación cruzada. Virtuoso ofrece capacidades avanzadas como enrutamiento asistido, verificación de reglas de diseño, extracción de parásitos y análisis de rendimiento. Otro aspecto relevante es la integración directa con herramientas de simulación como Spectre o Xcelium de manera simultánea con Virtuoso, lo que resulta útil en el desarrollo de circuitos de señal mixta. En resumen, Virtuoso es una opción robusta, flexible y altamente especializada para el diseño de circuitos integrados [13].

6) *RNM (Real Number Modeling)*: El RNM es una metodología empleada en el diseño y verificación de sistemas de señales mixtas que permite representar comportamientos analógicos mediante señales continuas, codificadas como variables del tipo real o wreal, que resultan más eficientes y simples, ya que no se requieren de sistemas de ecuaciones diferenciales ni definir parámetros específicos como las presentes en Verilog-AMS. Eliminar estas características reduce los tiempos de simulación y el consumo de recursos computacionales. La capacidad del RNM de representar los efectos funcionales de circuitos analógicos como la detección de flancos, la generación de señales continuas o el cálculo de frecuencia, permite desarrollar bancos de prueba mixtos completamente integrados, evitando la necesidad de cosimulación con herramientas externas o lenguajes analógicos puros de manera eficiente. En resumen, esta técnica es una solución práctica para enfrentar los retos de verificación en diseños modernos de señal mixta [14].

III. DESCRIPCIÓN DEL PROYECTO

A. Preparación previa

1) *Configuración de la herramienta de trabajo*: Para la correcta verificación del dispositivo bajo prueba, es crucial la implementación de un entorno de trabajo capaz de simular de la forma más fiel señales analógicas. Para la aplicación en este caso, se dio uso de Xcelium de Cadence Design Systems como herramienta principal, pensando en realizar la configuración orientado a la metodología de Modelado con Números Reales (RNM).

Como primer paso, se procesó a la ejecución inicial del script "start_rnm.csh" el cual nos permite establecer los parámetros fundamentales para esta aplicación

- Configura el modo de netlisting para soportar componentes analógicos

- Habilita la optimización para sistemas de 64 bits
- Carga las variables de entorno necesarias para las herramientas Cadence

14

Posterior a esto, es posible automatizar el ambiente de trabajo permitiendo una fácil reproducibilidad, esto mediante el script "startScript24.csh" el cual nos realizara la creación y configuración del espacio de trabajo, realizara la asociación con el Process Design Kit (PDK) XH018 y también, configura de forma automática las librerías a utilizar.

2) *Adaptación del DUT:* Habiendo configurado la herramienta de trabajo, con el fin de poder implementar la herramienta de trabajo para una correcta verificación del DUT es necesario realizar unas adaptaciones al DUT con el fin de generar compatibilidad. Originalmente el dispositivo bajo prueba presenta una entrada digital y un registro intermedio llamado "a_ff", donde este depende directamente de la sincronía con un reloj "clk" el cual en cada flanco positivo se actualiza el registro intermedio mediante bloques secuenciales. Para la detección de cada flanco, se realiza una comparación de la entrada con el valor almacenado en el registro. Para la adaptación del dispositivo, se realizo primeramente la eliminación de la dependencia al reloj e introduciendo soporte para señales en tiempo continuo, estas siendo de tipo "wreal", y para lograr esto fue necesario cambiar de una entrada digital a una entrada continua "a_in".

De igual manera, para la detección de flancos se realizaron adaptaciones las cuales consistieron en la utilización de un bloque "always @ (a_in)", en el que las transiciones de flanco se detectan comparando el valor actual de la señal con su valor previo (a_prev), utilizando un umbral definido (0.5) para identificar los cruces de frontera.

Como última adaptación, las salidas "rising_edge_o" y "falling_edge_o" se generaban como pulsos sincronizados explícitamente con el dominio temporal del reloj, las cuales fueron redefinidas como señales de tipo wreal, garantizando así compatibilidad con el flujo de verificación basado en RNM y permitiendo su análisis detallado dando uso de herramientas como SimVision.

3) *Verificación post-adaptación:* Para ya dar paso al desarrollo del testbench, se realizó una verificación del funcionamiento básico del DUT con la intención de comprobar que la adaptación del DUT no presenta errores manteniendo la misma funcionalidad, respondiendo correctamente a estímulos de naturaleza continua, manteniendo consistencia a su salida bajo condiciones controladas.

El desarrollo de este testbench fue crucial para la prevención de posibles errores introducidos en la adaptación del DUT, desarrollándolo basando en RNM de manera que se de uso y se pruebe las nuevas entradas y salidas analógicas mediante señales del tipo wreal. La generación de estímulos se llevó a cabo mediante la construcción de una señal cuadrada alternante entre 0.0 V y 1.0 V, aplicada a la entrada del DUT a través de la variable de control v_value utilizando la instrucción assign_wreal.

Esta estrategia permitió simular transiciones de flanco periódicas, las cuales fueron procesadas por el DUT para

generar las correspondientes señales de salida. Adicionalmente, 3 Diseño del ambiente de verificación se incorporó un bloque always con funciones de monitoreo que registró durante el tiempo de simulación el estado de las señales de entrada y salida, incluyendo la frecuencia calculada por el DUT, lo que permitió observar de forma directa su comportamiento dinámico. Basándose en la figura 4 podemos apreciar el flujo de trabajo que se uso para establecer la verificación de las modificaciones.

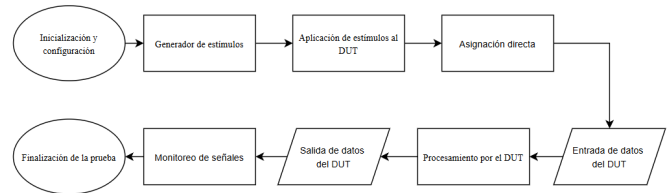


Fig. 4. Dispositivo seleccionado

B. Propuesta final del testbench

El proceso de verificación funcional del DUT, como se menciono previamente, requiere de la configuración precisa del entorno de simulación, para el cual se dio uso de Xcelium como herramienta principal, que mediante la metodología de RNM se busca obtener el mejor resultado de la simulación. Para el entorno de verificación final se implemento un testbench completo en un único archivo Verilog-AMS debido a las limitantes de las herramientas de trabajo la cual no permite la segmentación en módulos. Esta unificación integra 5 módulos funcionales principales los cuales trabajan conjuntamente para la validación del comportamiento del detector de flancos, el generador de estímulos, el driver de aplicación, un monitor, el verificador automático y el scoreboard para la consolidación de los resultados. Se puede apreciar en la figura 5 el flujo de trabajo de este testbench

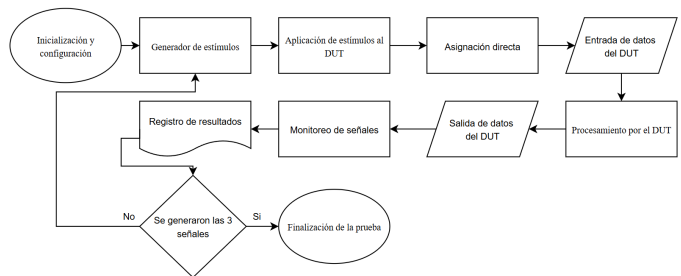


Fig. 5. Dispositivo seleccionado

Basándose específicamente en el monitor, este realiza un muestreo constante de las salidas cada 10 ns, capturando los eventos en cada rising edge y falling edge, al igual que el valor de frecuencia. El verificador compara los valores esperados en tiempo real, aplicando una tolerancia de ± 1 ciclo de muestreo (10 ns) para la detección de flancos y manteniendo un porcentaje de error menor del 1% para el calculo de la

8

frecuencia. Todo esta verificación de los datos se registran en un archivo CSV.

Para la implementación específica del detector, este sigue un flujo secuencial, donde comienza con la inicialización de parámetros. Este usa construcciones de Verilog-AMS como randcase para la selección aleatoria del tipo de onda y estas poseen un valor de frecuencia especificado, para que posteriormente se proceda a la síntesis de las formas de onda que se realiza mediante bucles temporales donde se actualiza el valor de la señal de entrada en base a su función matemática.

El testbench posee la capacidad de autoevaluación mediante el checker el cual detecta inmediatamente discrepancias entre los valores esperados en base al scoreboar y el valor medido en la salida del DUT, donde se generan mensajes de error específicos cuando se superan los umbrales de tolerancia.

El testbench final brinda una solución completa y autónoma permitiendo una verificación del detector de flancos, permitiendo generar de forma fiel los distintos estímulos, realizando un monitoreo preciso de la respuesta del dispositivo y, validación automática de resultados. El diseño modular pero integrado en un único archivo simplifica su ejecución y mantenimiento. De igual manera el uso de wreal permite una representación eficiente de las señales mixtas sin necesidad de simulaciones con herramientas analógicas.

IV. ANÁLISIS DE RESULTADOS

Para la sección de resultados se debe configurar de manera adecuada el entorno de verificación junto con la implementación del dispositivo bajo prueba. Para ello se crea una librería con la implementación adecuada del DUT dentro de las herramientas de diseño de Cadence. Es importante tomar en cuenta las herramientas necesarias para poder simular adecuadamente las señales mixtas.

Posterior a la configuración de parámetros se ejecuta la verificación en Xcelium. Para las pruebas se somete el diseño a tres señales diferentes: diente de sierra, senoidal y cuadrada. La primer señal inducida al DUT es la senoidal la cual muestra los siguientes resultados

```
-- PRUEBA Senoidal: 88.0 Hz --
t=0 a in=0.500 rising=1.0 falling=0.0 freq=0.00 Hz
t=100000000 a in=0.528 rising=1.0 falling=0.0 freq=0.00 Hz
t=200000000 a in=0.555 rising=1.0 falling=0.0 freq=0.00 Hz
t=300000000 a in=0.583 rising=1.0 falling=0.0 freq=0.00 Hz
t=400000000 a in=0.610 rising=1.0 falling=0.0 freq=0.00 Hz
t=500000000 a in=0.636 rising=1.0 falling=0.0 freq=0.00 Hz
t=600000000 a in=0.663 rising=1.0 falling=0.0 freq=0.00 Hz
t=700000000 a in=0.690 rising=1.0 falling=0.0 freq=0.00 Hz
t=800000000 a in=0.714 rising=1.0 falling=0.0 freq=0.00 Hz
t=900000000 a in=0.739 rising=1.0 falling=0.0 freq=0.00 Hz
t=1000000000 a in=0.763 rising=1.0 falling=0.0 freq=0.00 Hz
t=1100000000 a in=0.786 rising=1.0 falling=0.0 freq=0.00 Hz
```

Fig. 6. Resultado de la onda senoidal con una frecuencia de 88 Hz

Se observa una pequeña variación en la frecuencia calculada por el DUT en respuesta al primer estímulo es de 88.50 Hz, como se muestra en la siguiente imagen:

```
t=3400000000 a in=0.475 rising=0.0 falling=1.0 freq=87.72 Hz
t=3410000000 a in=0.503 rising=0.0 falling=1.0 freq=87.72 Hz
t=3410000000 a in=0.503 rising=1.0 falling=0.0 freq=88.50 Hz
t=3420000000 a in=0.000 rising=1.0 falling=0.0 freq=88.50 Hz
t=3420000000 a in=0.000 rising=0.0 falling=1.0 freq=88.50 Hz
```

Fig. 7. Resultado de onda senoidal con una frecuencia de 88 Hz

En la segunda prueba se aplicó una señal senoidal de 79 Hz, tal como se indica a continuación:

```
-- PRUEBA Senoidal: 79.0 Hz --
t=3920000000 a in=0.500 rising=0.0 falling=1.0 freq=88.50 Hz
t=3930000000 a in=0.525 rising=1.0 falling=0.0 freq=196.08 Hz
t=3940000000 a in=0.550 rising=1.0 falling=0.0 freq=196.08 Hz
t=3950000000 a in=0.574 rising=1.0 falling=0.0 freq=196.08 Hz
t=3960000000 a in=0.599 rising=1.0 falling=0.0 freq=196.08 Hz
t=3970000000 a in=0.623 rising=1.0 falling=0.0 freq=196.08 Hz
t=3980000000 a in=0.647 rising=1.0 falling=0.0 freq=196.08 Hz
t=3990000000 a in=0.670 rising=1.0 falling=0.0 freq=196.08 Hz
t=4000000000 a in=0.693 rising=1.0 falling=0.0 freq=196.08 Hz
t=4010000000 a in=0.716 rising=1.0 falling=0.0 freq=196.08 Hz
t=4020000000 a in=0.738 rising=1.0 falling=0.0 freq=196.08 Hz
t=4030000000 a in=0.760 rising=1.0 falling=0.0 freq=196.08 Hz
t=4040000000 a in=0.783 rising=1.0 falling=0.0 freq=196.08 Hz
```

Fig. 8. Resultado de señal senoidal con frecuencia de 79 Hz

Al realizar esta simulación se obtiene un comportamiento equivalente pues la señal enviada es de 79 Hz y el dispositivo calcula una frecuencia de 79 Hz.

Posteriormente se aplica al DUT una señal cuadrada de 167 H, sin embargo, dada la naturaleza de las señales cuadradas, abruptas y bien definidas, permite una detección exacta de los flancos, por consiguiente un cálculo de la frecuencia altamente preciso.

```
-- PRUEBA Cuadrada: 167.0 Hz --
t=82300000000 a in=1.000 rising=0.0 falling=1.0 freq=79.37 Hz
t=82300000000 a in=1.000 rising=1.0 falling=0.0 freq=196.08 Hz
t=85294011976 a in=0.000 rising=1.0 falling=0.0 freq=196.08 Hz
t=85294011976 a in=0.000 rising=0.0 falling=1.0 freq=196.08 Hz
t=88288023952 a in=1.000 rising=0.0 falling=1.0 freq=196.08 Hz
t=88288023952 a in=1.000 rising=1.0 falling=0.0 freq=167.00 Hz
t=91282035928 a in=0.000 rising=1.0 falling=0.0 freq=167.00 Hz
t=91282035928 a in=0.000 rising=0.0 falling=1.0 freq=167.00 Hz
t=94276047904 a in=1.000 rising=0.0 falling=1.0 freq=167.00 Hz
t=94276047904 a in=1.000 rising=1.0 falling=0.0 freq=167.00 Hz
t=97270059880 a in=0.000 rising=1.0 falling=0.0 freq=167.00 Hz
t=97270059880 a in=0.000 rising=0.0 falling=1.0 freq=167.00 Hz
```

Fig. 9. Resultado de onda cuadrada con una frecuencia de 167 Hz.

Se realizaron pruebas con frecuencias desde los 50 Hz hasta los 200 Hz, para evaluar la diferencia entre las frecuencias esperadas y la calculada. A partir de esto se calculó el error porcentual de cada una de las pruebas realizadas. Los resultados fueron los siguientes: para las señales cuadradas el error porcentual absoluto fue de 0.00%. En señales senoidales, el error promedio fue de 0.32%, con un máximo de 0.73%. En señales de diente de cierra, el promedio fue de 0.35% también con un error máximo de 0.73%. Estos resultados permiten caracterizar el desempeño del DUT frente a distintas formas de onda, evidenciando su alta precisión en señales con transiciones abruptas y una mayor sensibilidad a pendientes suaves en señales analógicas. Se utilizó SimVision para realizar un análisis visual de la verificación realizada.

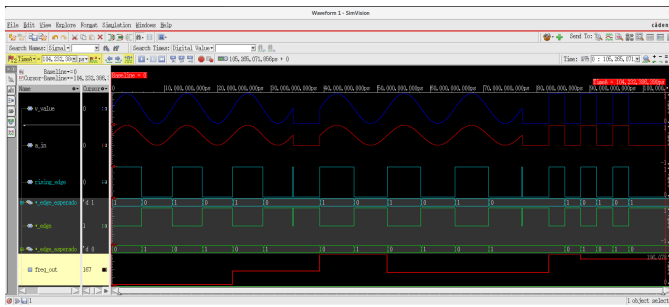


Fig. 10. Resultado de la verificación en forma gráfica por medio de SimVision

Se realizó un análisis de cobertura de pruebas del ambiente de verificación, los resultados se muestran en seguida. Se definieron tres frecuencias de prueba 50 Hz, 100 Hz y 200 Hz, estableciendo así el rango de frecuencias que se deseaba analizar.

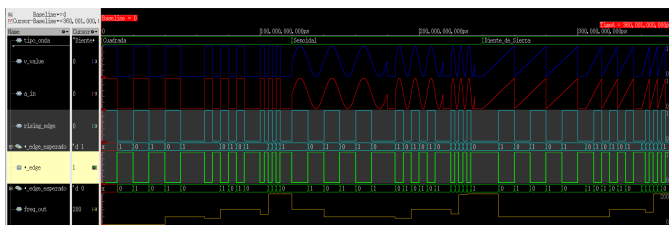


Fig. 11. Visualización del total de casos de prueba ejecutados por el ambiente de verificación sobre el DUT

V. CONCLUSIONES

Se logró el desarrollo de un ambiente de verificación funcional basado en la metodología de Real Number Modeling (RNM), gracias al uso de la herramienta Xcelium de Cadence.

La adaptación del detector de flancos desde su versión digital hasta su implementación compatible con RNM fue exitosa.

Los resultados obtenidos tanto a nivel de consola como de forma gráfica mediante SimVision, confirmaron el funcionamiento esperado para las frecuencias en el rango de los 50-200 Hz. En particular se observó alta precisión en señales cuadradas, mientras que en señales más suaves como las senoidales, se identificaron errores pequeños en la estimación de la frecuencia.

Se concluye que las discrepancias no se deben a fallos estructurales del DUT, sino a la interacción entre la pendiente de la señal de entrada y el paso temporal de simulación, esto afecta el momento exacto en que se detectan los cruces del umbral. A pesar de que no se realizó un análisis cuantitativo del tiempo de detección de flancos los resultados obtenidos reflejan un comportamiento estable y predecible.

Finalmente, se recomienda para futuras investigaciones realizar pruebas con señales analógicas que incluyan ruido o

transitorios con el objetivo de evaluar la robustez del DUT. Así también validar el modelo RNM contra una implementación basada en SPICE para verificar que el modelo abstracto reproduce fielmente el comportamiento de un hardware físico.

REFERENCES

- [1] G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," *Phil. Trans. Roy. Soc. London*, vol. A247, pp. 529–551, April 1955.
- [2] Universal verification methodology (uvm) 1.1 user's guide. Acellera, 2011.
- [3] Mixed-signal verification. <https://www.cadence.com/en-US/home/solutions/mixed-signal-solutions/mixed-signal-verification.html>. Accessed: 2025-03-17
- [4] Mixed-signal verification. <https://www.cadence.com/en-US/home/training/all-courses/86218.html>. Accessed: 2025-03-17.
- [5] Antonio Toro Frías. Diseño de circuitos analógicos y de señal mixta con consideraciones de diseño físico y variabilidad. PhD thesis, Universidad de Sevilla, 2021. Accessed: 2025-06-11.
- [6] B. Wile, J. Goss, and W. Roesner. *Comprehensive Functional Verification: The Complete Industry Cycle*. Elsevier, San Francisco, 2005.
- [7] M. Ferdjallah. *Introduction to Digital Systems: Modeling, Synthesis, and Simulation using VHDL*. John Wiley and Sons, Inc., New Jersey, 2011.
- [8] Accellera Systems Initiative Inc. *Verilog-AMS Language Reference Manual*, February 2024. VAMS-2023 Standard, Accessed: 2025-06-11.
- [9] Universal verification methodology for mixed-signal standard (uvm-ms) version 1.0, January 2025. Accessed: 2025-06-11.
- [10] Jr. Charles H. Roth, Larry L. Kinney, and Eugene B. John. *Fundamentals of Logic Design*. Cengage Learning, 7th, enhanced edition, 2021. Accessed: 2025-06-11.
- [11] Cadence Design Systems. *Demystifying mixed-signal simulation for verification engineers*. Technical report, 2020. Accessed: 2025-06-11.
- [12] Cadence Design Systems. *Xcelium simulator*. <https://www.cadence.com/en-US/home/tools/system-design-and-verification/simulation-and-testbench-verification/xcelium-simulator.html>, 2025. Accessed: 2025-06-11.
- [13] Cadence Design Systems. *Virtuoso layout suite datasheet*. <https://www.cadence.com,2023>. Accessed: 2025-06-11.
- [14] Nikolaos Georgouloupoulos and Alkiviadis Hatzopoulos. Real number modeling of a flash adc using systemverilog. *Conference/Workshop Paper*, 2020. Accessed: 2025-06-11.