

UNIVERSIDAD NACIONAL DE MAR DEL PLATA

Trabajo Final de Carrera

Optimización del hardware para medición de radiación utilizando sensores de imagen CMOS

Autor: Gaspar Carlos Domé
Carrera: Ingeniería en Electrónica

17 de agosto de 2025



RINFI es desarrollado por la Biblioteca de la Facultad de Ingeniería de la Universidad Nacional de Mar del Plata.

Tiene como objetivo recopilar, organizar, gestionar, difundir y preservar documentos digitales en Ingeniería, Ciencia y Tecnología de Materiales y Ciencias Afines.

A través del Acceso Abierto, se pretende aumentar la visibilidad y el impacto de los resultados de la investigación, asumiendo las políticas y cumpliendo con los protocolos y estándares internacionales para la interoperabilidad entre repositorios



Esta obra está bajo una [Licencia Creative Commons Atribución- NoComercial-CompartirIgual 4.0 Internacional](https://creativecommons.org/licenses/by-nc-sa/4.0/).

UNIVERSIDAD NACIONAL DE MAR DEL PLATA

Trabajo Final de Carrera

Optimización del hardware para medición de radiación utilizando sensores de imagen CMOS

Autor: Gaspar Carlos Domé
Carrera: Ingeniería en Electrónica

17 de agosto de 2025

*A la memoria de mi abuela, Anna
A mi familia, por su apoyo incondicional.*

Agradecimientos

A mi Directora Luciana De Micco, por toda su paciencia, predisposición y compromiso. Todos sus aportes, comentarios y puntos de vista fueron vitales para el desarrollo de este proyecto final.

A Lucas Andrés Rabioglio, por acompañarme con paciencia y comprensión en cada etapa de este proceso. Su apoyo constante, sus consejos oportunos y la generosidad con su tiempo fueron de gran ayuda al atravesar los momentos más difíciles. Este trabajo no habría sido posible sin su presencia incondicional.

A Martín Pérez, José Lipovetzky y Maximiliano Antonelli, por acompañarme a lo largo de este proyecto final, aclarando dudas y proveyendo tanto de conocimientos como de materiales.

A Dante Cecchetti, Jeremías Olivero y a otros colegas, por compartir ideas y sugerencias valiosas durante las pruebas, y por estar presentes también en las charlas y el apoyo cotidiano, hicieron que este recorrido fuera mucho más llevadero.

Resumen

En este trabajo se desarrolla un sistema de bajo costo para la detección de radiación ionizante, diseñado para operar en entornos espaciales utilizando sensores CMOS comerciales como el MT9M001C12STM y una FPGA de bajo costo de la marca Lattice semiconductor. El objetivo principal es comparar el rendimiento de este sistema con diseños previos, destacando las decisiones tomadas para optimizar el uso de hardware sin comprometer significativamente la calidad de los resultados.

El sistema implementado mide eventos ionizantes y calcula parámetros como el flujo de partículas y el Linear Energy Transfer (LET). El sistema procesa las imágenes capturadas por el sensor CMOS, filtra píxeles muertos y genera histogramas que caracterizan la intensidad de cada evento.

Durante el desarrollo, se analizaron compromisos entre eficiencia y rendimiento. Logrando una implementación altamente eficiente en el uso del hardware disponible, los resultados obtenidos muestran que, a pesar de las limitaciones de hardware, el sistema logra un desempeño comparable al diseño previo, con la ventaja de reducir drásticamente los recursos utilizados.

Palabras clave: Radiación ionizante, FPGA, CMOS, Algoritmo de detección, Satélite.

Abstract

In this work, a low-cost system for ionizing radiation detection is developed, designed to operate in space environments using commercial CMOS sensors such as the MT9M001C12STM and a low-cost FPGA from Lattice Semiconductor. The main objective is to compare the performance of this system with previous designs, highlighting the decisions made to optimize hardware usage without significantly compromising the quality of the results.

The implemented system measures ionizing events and calculates parameters such as particle flux and Linear Energy Transfer (LET). The system processes images captured by the CMOS sensor, filters dead pixels, and generates histograms to characterize the intensity of each event.

During the development, trade-offs between efficiency and performance were analyzed. Achieving a highly efficient implementation within the available hardware, the results show that despite hardware limitations, the system achieves performance comparable to the previous design, with the advantage of drastically reducing the resources used.

Keywords: Ionizing radiation, FPGA, CMOS, Detection algorithm, Satellite.

Índice general

Agradecimientos	4
Resumen	5
Abstract	6
1. Introducción	9
1.1. Objetivos	10
1.1.1. Objetivos específicos	10
2. Anteproyecto	11
2.1. Idea inicial	11
2.2. Justificación	11
3. Proyecto	13
3.1. Primera etapa: Programación del dispositivo y manejo del sensor	13
3.2. Segunda etapa: Algoritmo y memorias	16
3.2.1. Optimización del algoritmo	17
3.2.2. Mapeo de memoria	18
3.2.3. Histograma	20
3.2.4. Controlador	20
3.2.5. Pruebas sobre el sistema completo	20
4. Conclusiones	21
4.1. Análisis de Resultados	21
4.2. Limitaciones	21
4.3. Trabajo Futuro	21
Bibliografía	21
Apéndices	24
A. Detalle maquina de estado protocolo de comunicación I2C	25
A.1. programador	25
A.2. Controlador	25
B. Funcionamiento del algoritmo desarrollado en el trabajo previo	27
C. Funcionamiento del histograma	29

D. Especificación de requerimientos	30
Anexo D: Especificación de requerimientos	30
E. Plan de proyecto	39
Anexo E: Plan de proyecto	39
F. Especificación funcional	45
Anexo F: Especificación funcional	45
G. Especificación técnica	53
Anexo G: Especificación técnica	53
H. Plan de pruebas	106
Anexo H: Plan de pruebas	106

Capítulo 1

Introducción

La detección de radiación ionizante en entornos espaciales es un desafío para la protección de los componentes electrónicos de satélites y otros dispositivos. Los componentes electrónicos en el espacio son altamente susceptibles a la radiación ionizante debido a la reducción drástica de la atenuación que provee la atmósfera, a medida que se incrementa la distancia a la superficie de la Tierra. Los efectos de la radiación en sistemas electrónicos son la desintegración de materiales, cambios en el rendimiento, errores suaves (soft errors) y el latch-up que alteran el comportamiento de los circuitos, provocan fallos funcionales y reducen la vida útil del hardware. Existen diversas tecnologías de detección, como detectores de semiconductores, contadores Geiger y cámaras de ionización, pero presentan limitaciones en términos de costo, tamaño, peso, consumo de energía y durabilidad, lo que los hace menos adecuados para aplicaciones espaciales, donde el peso y la potencia son importantes. [\[1\]](#)[\[2\]](#)[\[3\]](#)[\[4\]](#)[\[5\]](#)[\[6\]](#)

Este trabajo tiene como objetivo optimizar un algoritmo previamente desarrollado que requiere un uso intensivo de recursos lógicos, adaptándolo a una FPGA de bajo costo para implementar un sistema de detección eficiente y económico. Se utiliza un sensor comercial CMOS y una FPGA económica de la marca Lattice, específicamente el modelo ICE40HX4K. Esta solución permite el diseño de un sistema compacto, con un consumo energético reducido, siendo de solo 2W, con la posibilidad de ser mejorado a través de la fabricación de sistemas en chip (SOC) en el futuro.

El objetivo principal del proyecto es demostrar que es posible implementar un sistema de bajo costo, optimizando el uso de recursos limitados, sin comprometer la calidad de la detección. A diferencia de sistemas existentes que emplean componentes especializados y costosos, se aprovechan los sensores CMOS de producción en masa y las ventajas de las FPGA de Lattice, los cuales son ampliamente utilizadas en el campo aeroespacial. Además, las FPGA de Lattice permiten un prototipado rápido y son programables de forma permanente mediante el uso de fusibles, dándoles más resistencia en entornos de alta radiación.

Se caracteriza la intensidad de los eventos de radiación mediante histogramas y se evalúa el riesgo de daño en los materiales mediante el cálculo del parámetro LET (Transferencia de Energía Lineal). El sistema desarrollado es capaz de detectar las partículas de radiación y analizar su intensidad, siendo una herramienta nueva para la protección de los equipos en sistemas espaciales.

1.1. Objetivos

Optimizar un algoritmo de detección de radiación ionizante previamente desarrollado, adaptándolo a una FPGA de bajo costo e integrándolo con sensores CMOS comerciales, con el propósito de desarrollar un sistema compacto, eficiente y de bajo costo, adecuado para aplicaciones en entornos espaciales.

1.1.1. Objetivos específicos

- Optimizar el algoritmo de detección con el objetivo de reducir el uso de recursos de hardware, permitiendo su implementación en una FPGA de bajo costo sin comprometer la precisión del sistema.
- Desarrollar el sistema utilizando un sensor CMOS comercial (MT9M001C12STM) y la FPGA Lattice ICE40HX4K, que permita medir eventos de radiación y calcular histogramas.
- Que el sistema sea capaz de comunicarse con el satélite
- Que el sistema se adapte a las limitaciones postuladas por la plataforma LabOsat
- Que el sistema y todos sus periféricos tengan un consumo igual o menor a 2 W

Capítulo 2

Anteproyecto

2.1. Idea inicial

El proyecto actual se concibe como una continuación del trabajo inicial realizado por los ingenieros Rolón y García [7]. Este trabajo demuestra la viabilidad de utilizar sensores ópticos CMOS para la detección de radiación ionizante. En él se desarrolló un algoritmo capaz de detectar y clasificar eventos, validando la teoría propuesta por los Dres. Lipovetzky y Pérez [8]. Sin embargo, [7] presenta un elevado consumo de recursos de hardware y una dependencia significativa de la síntesis de alto nivel (HLS). El presente proyecto busca optimizar dicho algoritmo, reduciendo el uso de unidades lógicas sin comprometer la funcionalidad y diseñándolo para ser compatible con una amplia gama de FPGAs, con un enfoque específico en la plataforma de hardware abierto EDU-CIAA-FPGA, basada en la FPGA Lattice ICE40HX4K

2.2. Justificación

Los efectos de la radiación sobre dispositivos CMOS pueden clasificarse en dos categorías principales: efectos puntuales, como los *Single Event Upsets* (SEUs), y la degradación acumulativa. Los SEUs alteran transitoriamente el estado lógico de los circuitos, y suelen mitigarse mediante técnicas de redundancia o corrección de errores. En cambio, la degradación acumulativa representa un riesgo mayor, ya que puede provocar cambios permanentes en las características eléctricas de los transistores. Por ejemplo, la acumulación de carga en las capas de óxido puede generar un aumento en la corriente de fuga o una pérdida de control en la compuerta, lo que afecta directamente la funcionalidad del dispositivo. A largo plazo, estas alteraciones pueden causar fallos irreversibles [?].

En este contexto, contar con un sistema de detección que no solo registre eventos ionizantes, sino que también permita caracterizarlos y analizar su impacto, se vuelve fundamental. Este tipo de herramienta puede ser útil tanto en tareas de validación en tierra como en aplicaciones en órbita, donde se requieren soluciones confiables, de bajo consumo y bajo costo, integrables en plataformas de escala reducida u otros dispositivos científicos.

En trabajos previos se demostró que es posible utilizar sensores CMOS comerciales para detectar eventos de radiación [8, 7]. Sin embargo, la solución propuesta requiere una cantidad considerable de recursos lógicos, lo que limita su uso en plataformas con restricciones de consumo, tamaño y costo. Este proyecto busca superar esa limitación mediante

la optimización del procesamiento y la implementación de un sistema más eficiente, utilizando una FPGA de bajo costo como la Lattice ICE40HX4K, integrada en la plataforma abierta EDU-CIAA-FPGA.

Capítulo 3

Proyecto

Este capítulo describe el desarrollo progresivo del sistema, comenzando desde la configuración del sensor y adquisición de imágenes, hasta la implementación del algoritmo de procesamiento, el uso de memorias y el diseño del circuito impreso (PCB). El objetivo general fue implementar una solución funcional, optimizada y sencilla que cumpla con los requerimientos establecidos para operar en entornos espaciales.

3.1. Primera etapa: Programación del dispositivo y manejo del sensor

Los objetivos de esta etapa son:

- Configurar los registros del sensor.
- Obtener una imagen válida desde el sensor.

Estos corresponden al requisito funcional (RF) 1.

Durante esta fase inicial, se evaluaron las herramientas disponibles para el diseño y la programación de la FPGA, tanto en la EDU-CIAA-FPGA [9] como en la plataforma de Lattice [10], optándose por esta última debido a la mayor facilidad de uso de su entorno de desarrollo.

Se implementó un primer bloque encargado de configurar los registros del sensor a través de una interfaz de comunicación serie similar a I2C, pero restringida a ser punto a punto, por lo que la inclusión de un id del dispositivo resulta redundante, siendo el primer dato una instrucción de escritura o lectura. El bloque se desarrolló priorizando simplicidad y bajo consumo de lógica. Permite modificar la ganancia y no incluye lectura de registros. La Figura 3.1 muestra el protocolo funcionando en un osciloscopio. El Anexo A contiene detalles de la máquina de estados embebida.

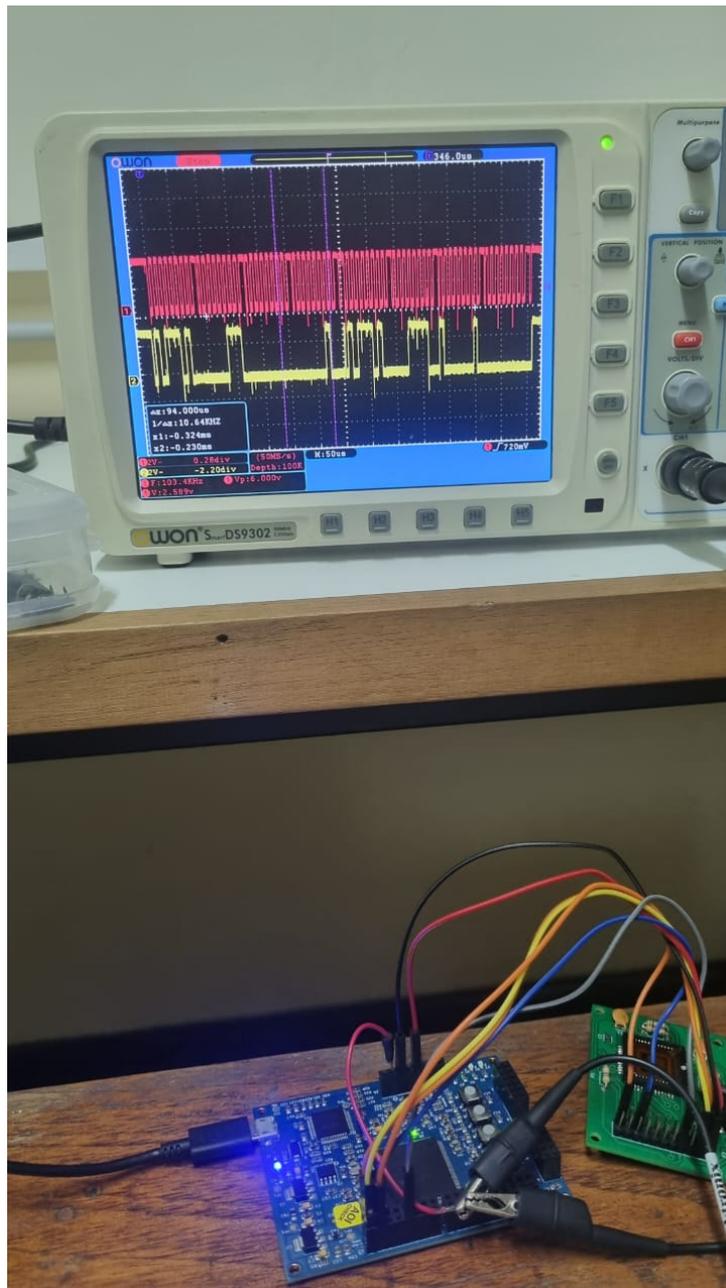


Figura 3.1: Protocolo de comunicación en el osciloscopio.

El sensor entrega datos mediante las siguientes señales:

- **Frame Valid:** Señala que la imagen no se encuentra en zonas de blanqueo vertical.
- **Line Valid:** Señala que la línea está activa.
- **PixCLK:** Reloj que sincroniza los datos.
- **Trigger:** Señal de inicio de adquisición.
- **d9–d2:** Bus de datos de 8 bits por pixel indicando el brillo.

La salida digital del sensor posee 10 bits de resolución, los datos se truncan a 8 bits descartando los dos bits menos significativos, para simplificar el diseño, reducir el uso de memoria y realizar un filtrado previo.

La FPGA tiene un reloj externo de 12 MHz el cual es llevado a 50.5 MHz mediante un PLL para disponer de una mayor cantidad de ciclos de reloj durante el procesado en tiempo real. Para el sensor se genera un reloj de 2 MHz, y dado que existen señales provenientes de dominios distintos, es necesario implementar un sincronizador que prevenga problemas de metastabilidad.

Para probar el correcto funcionamiento del bloque capturador se decidió recibir los datos sin procesar en una PC y luego reconstruirlos. El sensor genera 8 bits en cada ciclo de reloj, por lo que la tasa de datos resulta:

$$bps = 8 * f$$

$$bps = 8 * 2MHz = 16Mbps$$

Como el sistema utiliza un enlace de comunicación serie entre la FPGA y la PC, la tasa de datos generada resulta demasiado alta para ser transmitida directamente. Para resolver este problema, se emplea la placa de desarrollo DE2-115, que incorpora una memoria SRAM de 2 MB. Esta memoria permite almacenar temporalmente los píxeles a la velocidad necesaria durante la adquisición, y luego enviarlos a la PC a una tasa reducida, compatible con la interfaz serie.

Inicialmente, las pruebas no fueron satisfactorias dando como resultado la siguiente imagen:



Figura 3.2: Reconstrucción de la primer imagen capturada.

Luego de solucionar los problemas que provienen de la metaestabilidad se obtuvo la primera imagen satisfactoria.



Figura 3.3: Reconstrucción de la primera imagen exitosa capturada.

3.2. Segunda etapa: Algoritmo y memorias

Luego de lograr capturar una imagen, se reescribió completamente el algoritmo, ya que en el trabajo previo se había trabajado con High Level Synthesis (HLS) y no fue posible reutilizar ningún bloque.

El objetivo del algoritmo es detectar los eventos, categorizarlos, conocer el tamaño y la energía, con el requisito de no almacenar toda la imagen para su procesamiento.

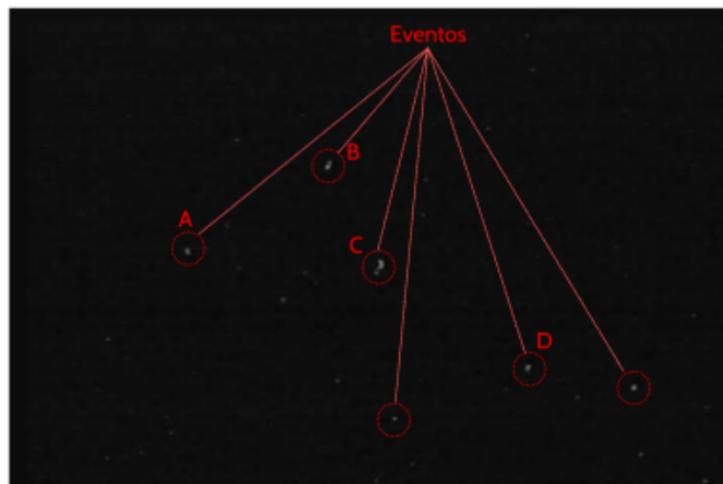


Figura 3.4: Ejemplo de conjunto de eventos.

Los objetivos de esta etapa son:

- Reconstruir el algoritmo para la iCE40 HX4K.
- Lograr una funcionalidad similar a la del desarrollo anterior.
- Completar el sistema incluyendo un protocolo de comunicación compatible con el satélite.

Desarrollo del algoritmo:

- En MATLAB, se diseñaron y probaron varias versiones del algoritmo, teniendo en cuenta las limitaciones de recursos de la FPGA.
- Una vez definido el algoritmo, se tradujo a VHDL como una máquina de estados.
- Se realizaron pruebas simulando el flujo de datos desde una PC y comparando los resultados procesados por la FPGA con los obtenidos en MATLAB.

3.2.1. Optimización del algoritmo

Como punto de inicio se tomó el desarrollo del trabajo previo; para una explicación detallada sobre el funcionamiento del algoritmo, ver anexo [B](#). Como el sistema fue desarrollado utilizando HLS y no se tenían restricciones en la cantidad de hardware a utilizar, el sistema presentaba deficiencias en el uso de las unidades lógicas, utilizando un estimado 80 % del hardware solamente como flip-flop para las distintas memorias internas que se necesitaban.

Cuando se identificaron las falencias, se prosiguió a esbozar un nuevo enfoque para implementar el algoritmo sin la necesidad de utilizar tantos flip-flops.

El algoritmo previo presenta como principal limitación la necesidad de modificar el vector de entrada en muy pocos ciclos de reloj, debido a que no se dispone de almacenamiento suficiente para guardar toda la imagen, lo que obliga a realizar el procesamiento en tiempo real. Esto implica actualizar 1024 datos de 8 bits en el menor número de ciclos posible. En la implementación anterior, este requerimiento se resuelve utilizando registros. Sin embargo, el enfoque actual propone el uso de bloques de memoria BRAM, los cuales no permiten modificar 1024 posiciones de memoria en el breve intervalo de tiempo entre la llegada de un píxel y el siguiente, lo que vuelve esta estrategia impráctica bajo las mismas condiciones de operación. Como solución se decide eliminar los casos problemáticos y analizar los efectos que esto tiene en los resultados, donde cada color significa un evento diferente. Idealmente, cada grupo debería tener un único color.



(a) Algoritmo considerando todos los casos.



(b) Algoritmo sin considerar todos los casos.

Figura 3.5: Comparación de los dos algoritmos.

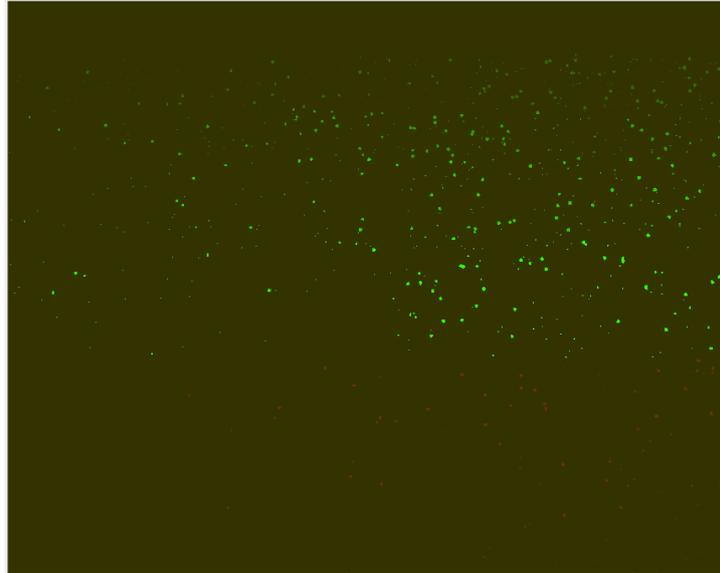


Figura 3.6: Nuevo algoritmo sobre datos reales.

Luego del análisis de los datos, se concluye que, en las imágenes de tortura¹, el nuevo algoritmo presenta una mayor cantidad de fallas en comparación con el diseño previo. Sin embargo, bajo condiciones reales, utilizando imágenes obtenidas a partir de fuentes reales de radiación, no se observan diferencias significativas en el rendimiento. Por lo tanto, se considera que el nuevo algoritmo es válido para aplicaciones prácticas con fuentes reales de radiación.

Como banco de prueba se incorporaron dos bloques UART como entrada y salida de datos al bloque que manejaba el algoritmo. Se enviaban desde la PC imágenes de prueba y se comparaban con los resultados esperados.

Se comparan las dos implementaciones

Componente	Sistema previo	Sistema actual	Reducción %
Unidades Lógicas (LU)	24,792	1,945	92.16
Flip-Flops (FF)	27,043	1,024	96.22

Cuadro 3.1: Comparativa de recursos entre sistemas.

3.2.2. Mapeo de memoria

Una vez demostrado que es posible implementar el algoritmo en la EDU-CIAA-FPGA utilizando una cantidad de unidades lógicas inferior al máximo disponible, el siguiente paso consiste en evaluar si la memoria disponible en la plataforma es suficiente para el funcionamiento del sistema.

Se propone el uso de tres bloques de memoria relacionados con el procesamiento: una memoria circular de entrada, que almacena los índices de los eventos correspondientes a la fila actual de la imagen; una memoria para registrar la energía acumulada por evento; y una memoria adicional para almacenar la cantidad de píxeles asociados a cada evento.

Con el fin de estimar la cantidad mínima de memoria necesaria, se utiliza una serie de videos de irradiación real sobre el sensor, provistos por el Laboratorio de Bajas Tem-

¹Imágenes de tortura: conjunto de imágenes generadas específicamente para probar casos límite

peraturas del Instituto Balseiro. A cada secuencia se le aplica el algoritmo propuesto, y se analizan los resultados para caracterizar el uso de memoria bajo condiciones reales. Durante este proceso, se registran los siguientes datos estadísticos:

- Promedio del tamaño máximo de evento detectado en cada video.
- Promedio de la energía máxima registrada por evento en cada video. ²
- Promedio de la cantidad máxima de eventos detectados simultáneamente en cada video.

Con estos valores se establecen los parámetros necesarios para dimensionar cada una de las memorias utilizadas en el sistema:

- **Memoria circular de entrada:** La memoria se dimensiona considerando dos factores principales. Por un lado, la cantidad de palabras está definida por el ancho del sensor, es decir, la cantidad de píxeles que componen una fila, a la cual se suman tres píxeles adicionales necesarios por el funcionamiento del algoritmo. Por otro lado, la profundidad de la memoria se establece en función de la cantidad promedio de eventos detectados.
- **Memoria de cantidad de píxeles por evento:** El tamaño promedio de los eventos permite estimar cuántos bits se requieren para almacenar la cantidad de píxeles que conforman un evento. Este valor define la profundidad necesaria de esta memoria y la cantidad de palabras está definida por la cantidad promedio de eventos detectados.
- **Memoria de energía por evento:** La energía promedio de los eventos permite estimar la cantidad de bits requeridos para representar adecuadamente la suma de los valores de los píxeles asociados a un evento. Esto define la profundidad de esta memoria y la cantidad de palabras está definida por la cantidad promedio de eventos detectados.

A cada uno de estos valores se les agrega un bit adicional de profundidad y cantidad de palabras como margen de seguridad para evitar errores por saturación o eventos fuera del promedio.

Memoria	Tamaño
Entrada	$2048 \times 12 \text{ bits} \approx 3 \text{ kB}$
Energía	$2048 \times 14 \text{ bits} \approx 3 \text{ kB}$
Cantidad	$2048 \times 6 \text{ bits} \approx 2 \text{ kB}$
Total	8 kB

Cuadro 3.2: Tamaño de memorias

Como se puede ver en la tabla ^{3.2} el total de memoria necesario solo para el algoritmo es de aproximadamente 8 kilo bytes, por lo tanto es menos de los 10 disponibles en la FPGA ICE40HX4k.

²Energía del evento: se define como la suma de los valores de los píxeles que lo componen.

3.2.3. Histograma

Luego de validar el correcto funcionamiento del algoritmo, se diseñó el histograma. Para ello, se consultó con el usuario final acerca de sus necesidades. Se definió realizar un histograma con 32 bins de ancho variable para, posteriormente, en la calibración, determinar el ancho óptimo. El restante de la memoria BRAM se utiliza para almacenar los histogramas de manera circular, esperando la solicitud de parte del satélite para la descarga de datos. Para generar un solo histograma se toman 60 imágenes durante un minuto que se vuelcan en un solo histograma para luego ser guardado en memoria.

Para una descripción más detallada del proceso de elaboración del histograma, se remite al anexo [C](#)

3.2.4. Controlador

Con todos los bloques capturador, algoritmo, banco de memoria y generador de histogramas se diseña un controlador que coordine todas las máquinas de estado y maneje la comunicación con el satélite, esperando un comando para reiniciar si fuese necesario y otro para descargar toda la información.

3.2.5. Pruebas sobre el sistema completo

Para verificar el correcto funcionamiento del sistema, se diseña un banco de pruebas que permite comparar el histograma generado por la FPGA con un histograma de referencia generado a partir de los mismos datos en una computadora.

En esta etapa se utiliza la plataforma de desarrollo DE2-115, principalmente debido a que cuenta con suficiente capacidad de memoria para almacenar una imagen completa capturada por el sensor. Lo cual es fundamental, ya que permite garantizar que los datos enviados a la computadora para su procesamiento corresponden exactamente a la misma imagen procesada en la FPGA, condición necesaria para que la comparación de histogramas tenga validez.

Se desarrollaron bloques específicos para el manejo de la memoria, también se implementó un bloque de coordinación que sincroniza el almacenamiento de la imagen y su posterior transmisión a la computadora.

Adicionalmente, se implementa un programa en MATLAB encargado de recibir los datos provenientes de la FPGA, reconstruir la imagen, aplicar el mismo algoritmo utilizado en la FPGA respetando las limitaciones de resolución en bits del hardware, y generar el histograma de referencia. Todo este proceso asegura una comparación válida entre el histograma generado por la FPGA y el histograma de referencia obtenido en la computadora.

Capítulo 4

Conclusiones

4.1. Análisis de Resultados

En este trabajo, se demostró que es posible optimizar el algoritmo desarrollado previamente, logrando mejoras superiores al 90 % en términos de eficiencia de hardware, sin afectar significativamente los resultados en escenarios reales.

Se validó que el sistema es adecuado para la ICE40HX4K, utilizando el 100 % de la memoria y menos del 60 % de las unidades lógicas, incorporando todos los periféricos necesarios para el control del sensor, el procesamiento, el almacenamiento y la comunicación.

Además, se diseñó un PCB que cumple con todas las especificaciones actuales, asegurando un sistema compacto, flexible y de bajo consumo.

Según las estimaciones, el sistema consume menos de 1W, cumpliendo ampliamente con los requisitos establecidos.

Se ha logrado un sistema completo, listo para iniciar las pruebas de campo.

4.2. Limitaciones

- El sistema no permite la lectura de registros del sensor, ya que esta funcionalidad no fue implementada al considerarse innecesaria para los objetivos del proyecto.
- No se fabricó el PCB.
- No se realizaron pruebas en condiciones de radiación.

4.3. Trabajo Futuro

Se espera avanzar con la fabricación del sistema y pruebas en campo. Es importante que quien continúe tenga conocimientos en máquinas de estado y efectos de la radiación en dispositivos CMOS. Las pruebas permitirán evaluar la resiliencia del sistema y explorar mejoras como el control de errores.

Además, se presenta la opción de llevar a cabo mejoras en el hardware, como la incorporación de una fuente conmutada síncrona de mayor frecuencia, lo cual permitiría incrementar la eficiencia y reducir aún más los requerimientos de potencia.

Bibliografía

- [1] R. Dosimetry, “Ventajas y desventajas de los contadores geiger-müller,” <https://www.radiation-dosimetry.org/es/que-es-la-ventaja-y-la-desventaja-de-los-contadores-geiger-mueller-definicion/>, s.f., accedido: abril 2025.
- [2] C. N. de Energía Atómica (CNEA), “Caracterización de dispositivos para uso en entornos espaciales,” <https://www.argentina.gob.ar/cnea/investigacion-y-desarrollo/energia-solar/caracterizacion-de-dispositivos-para-uso-en-entornos>, s.f., accedido: abril 2025.
- [3] R. Dosimetry, “Cámara de ionización vs. detector de semiconductores,” <https://www.radiation-dosimetry.org/es/que-es-la-camara-de-ionizacion-frente-al-detector-de-semiconductores-definicion/>, s.f., accedido: abril 2025.
- [4] Profesional Review, “Efectos de la radiación en semiconductores: Dosis, seu, set, sel, emp,” <https://www.profesionalreview.com/2024/08/04/efectos-de-la-radiacion-en-semiconductores-dosis-seu-set-sel-emp/>, agosto 2024, accedido: abril 2025.
- [5] J. R. Schwank, M. R. Shaneyfelt, D. M. Fleetwood, J. A. Felix, P. E. Dodd, P. Paillet, and V. Ferlet-Cavrois, “Radiation effects in mos oxides,” *IEEE Transactions on Nuclear Science*, vol. 55, no. 4, pp. 1833–1853, 2008. [Online]. Available: <https://www.osti.gov/servlets/purl/1147853>
- [6] A. H. Johnston, “Radiation damage of electronic and optoelectronic devices in space,” *4th International Workshop on Radiation Effects on Semiconductor Devices for Space Application*, 2000, accedido: abril 2025. [Online]. Available: https://nepp.nasa.gov/docuploads/D41D389D-04D4-4710-BBCFF24F4529B3B3/Dmg_Space-00.pdf
- [7] M. R. y Mathías Sebastian García, “Detector de radiación ionizante para aplicaciones satelitales,” Tesis de grado, Universidad Nacional de Mar del Plata (UNMDP), Mar del Plata, Buenos Aires, Argentina, 2021.
- [8] M. Pérez, J. Lipovetzky, M. Sofo Haro, I. Sidelnik, J. J. Blostein, F. Alcalde Bessia, and M. G. Berisso, “Particle detection and classification using commercial off the shelf cmos image sensors,” *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 827, pp. 171–180, 2016. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0168900216302844>

-
- [9] CIAA - Computadora Industrial Abierta Argentina, “EDU-CIAA-FPGA - Computadora Industrial Abierta Argentina,” <https://www.proyecto-ciaa.com.ar/devwiki/doku.php?id=edu-ciaa:fpga:start>, accedido el 13 de abril de 2025.
- [10] Lattice, “Lattice FPGA Design Software,” <https://www.latticesemi.com/software>, accedido el 17 de abril de 2025.

Apéndice

Apéndice A

Detalle maquina de estado protocolo de comunicación I2C

El sistema encargado de configurar el sensor que incluye el modulo I2C consiste de dos maquinas de estados, el programador y el controlador, el programador es una maquina de estados que se encarga de enviar una palabra en su entrada por los pines de salida, mientras que el controlador es la maquina de estados encargada de enviarle al programador las palabras y el orden en que esas palabras son enviadas.

A.1. programador

El programador es el encargado de replicar el protocolo I2C, como en esta etapa solo se requiere de escribir no se implementa la funcionalidad para leer los registros.

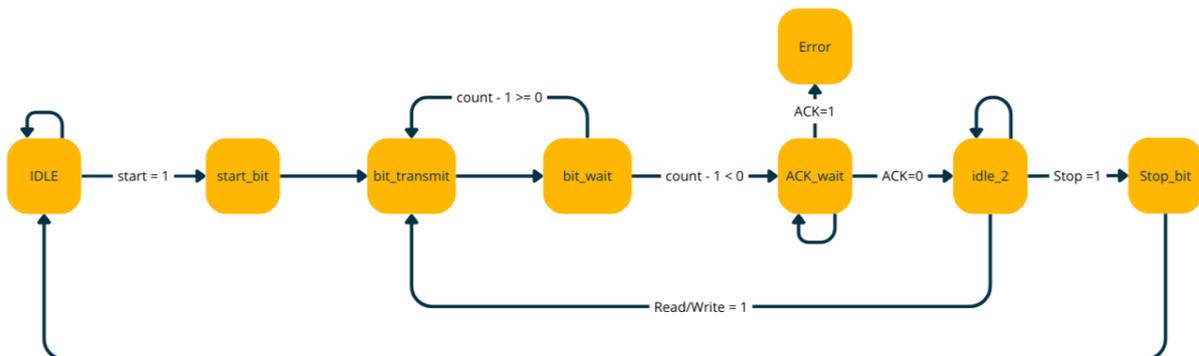


Figura A.1: Maquina de estados I2C

A.2. Controlador

La segunda maquina de estados encargada de alimentar la maquina de estados de la figura [A.1](#) es mucho mas simple, el primer estado espera recibir una señal de inicio luego le presenta uno a uno los bytes a enviar y los bytes de stop en el momento adecuado.

la lógica que utiliza el sensor para escribir los registros es la siguiente:

- Enviar byte especifico de escritura xBA.
- Enviar byte del registro a escribir.

- Enviar uno o dos bytes dependiendo el registro para el dato.
- Si se quiere escribir el registro próximo al que se acaba de escribir se puede seguir enviando bytes de datos ya que internamente la cámara incrementa el registro
- Si se quiere escribir otro registro no contiguo se envía una señal de stop y se inicia nuevamente el proceso

Apéndice B

Funcionamiento del algoritmo desarrollado en el trabajo previo

El algoritmo enfrenta el desafío de no contar con suficiente memoria para almacenar toda la imagen. Por eso, clasifica los eventos a medida que recibe los píxeles. Para cada píxel nuevo, analiza si alguno de sus vecinos ya fue categorizado: el píxel anterior, el superior, el superior derecho o el superior izquierdo. Si al menos uno ya está clasificado, se le asigna al píxel la misma categoría. Si ninguno lo está, se crea un nuevo número de evento.

Se pueden generar errores cuando un nuevo píxel conecta dos regiones que fueron clasificadas como eventos distintos, aunque en realidad forman parte del mismo. En el trabajo previo, ante esta situación, se volvía a recorrer toda la memoria de entrada para corregir la categorización y unificar los eventos, esto requiere de acceder y modificar los datos en pocos ciclos de reloj, para lograrlo utilizaron memorias fabricadas con Flip-Flops

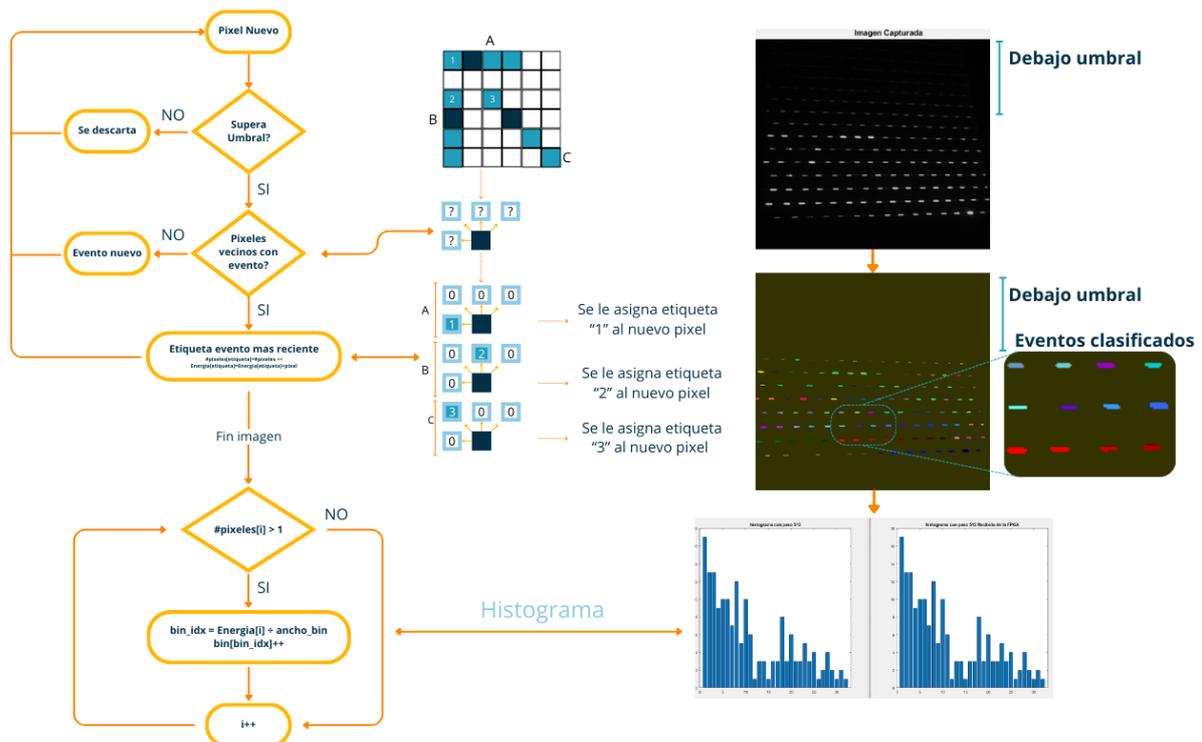


Figura B.1: Diagrama simplificado algoritmo e histograma

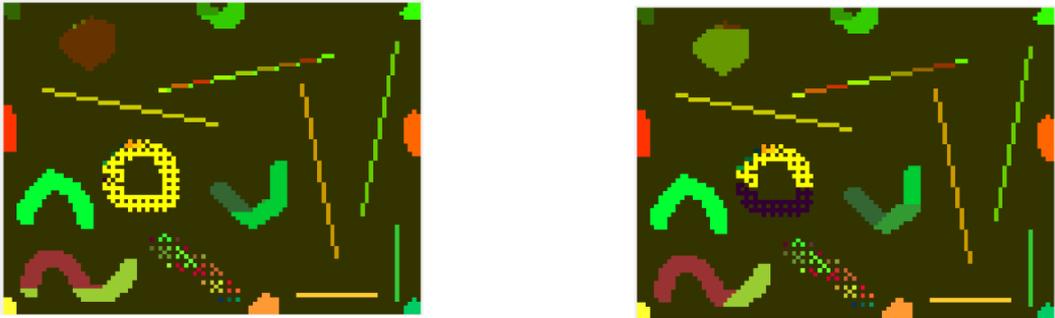
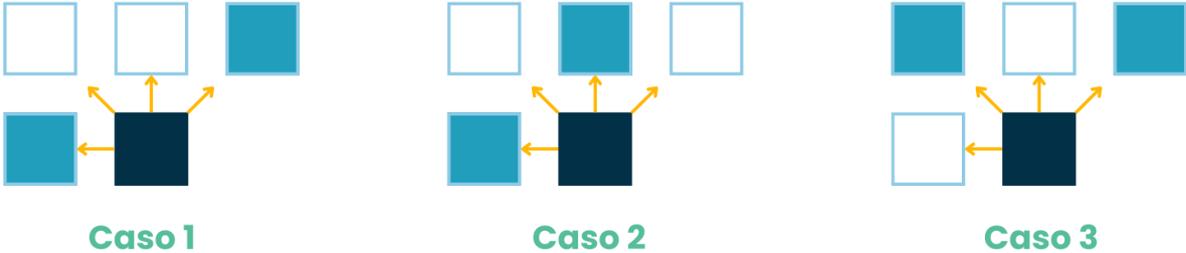


Figura B.2: Casos donde se categoriza mal y comparación entre el algoritmo viejo y nuevo

Apéndice C

Funcionamiento del histograma

El histograma categoriza los eventos según su energía asociada. En este caso se utilizan 32 bins de un ancho regulable según la cantidad de bits que se utilizan de la energía, por ejemplo si se utilizan los 14 bits de resolución de la energía el ancho de bin es

$$\text{ancho del bin} = \frac{2^{\text{cantidad de bits}}}{32} = \frac{2^{14}}{32} = 512$$

El procesamiento del histograma comienza una vez que finaliza la ejecución del algoritmo de detección de eventos. En esta etapa, se recorre la memoria que almacena la cantidad de píxeles por evento. Si un evento tiene más de un píxel asociado, se accede a su correspondiente valor en la memoria de energía y se clasifica dentro del bin correspondiente del histograma.

Se impone la condición de que el evento tenga más de un píxel porque, en caso contrario, se considera que el evento es nulo. Esto se debe a que un evento de un solo píxel generalmente indica un píxel "muerto" no una interacción real de radiación.

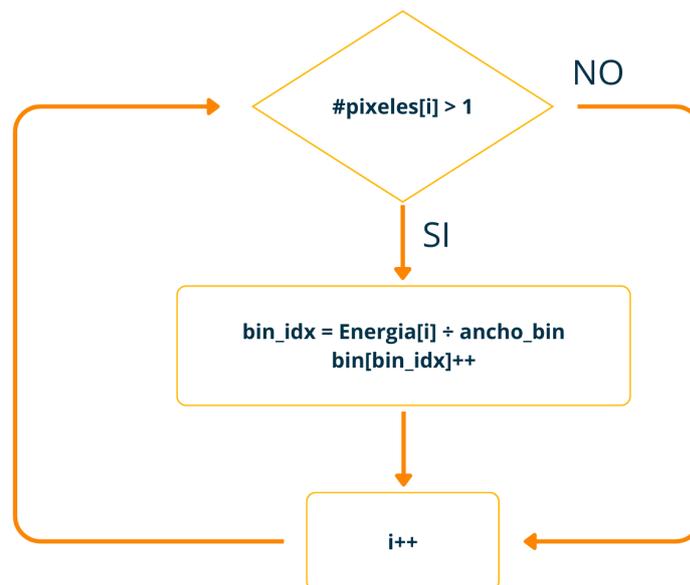


Figura C.1: Diagrama del funcionamiento del histograma

Apéndice D

Especificación de requerimientos

UNMDP-FI

Carrera: Ingeniería Electrónica / en Computación

Proyecto:

Definición del Tema de Investigación

Versión 1.0

Autor/a
Gaspar Carlos Domé

Definición del Tema de Investigación

Fecha	Versión	Descripción	Autor/a
7/4/2024	1.0	Versión inicial	Gaspar Carlos Domé

Índice

1.	Introducción	4
2.	Proyecto	4
2.1	Personal involucrado	4
2.2	Definiciones, acrónimos y abreviaturas	4
2.3	Referencias	5
2.4	Resumen	5
3.	Problema de investigación	5
3.1	Surgimiento de la idea de investigación	5
3.2	Preguntas de investigación	5
3.3	Objetivos de la investigación	5
3.4	Justificación y viabilidad de la investigación	5
3.5	Enfoque	5
3.6	Alcance	5

Definición del Tema de Investigación

1. Introducción

En este documento se propone establecer y detallar las necesidades funcionales y no funcionales del dispositivo que se va a desarrollar como proyecto final, perteneciente a la carrera de Ingeniería Electrónica. Este documento constituye la documentación necesaria para obtener la aprobación del trabajo final. Está destinado a los autores del documento, al estudiante futuro que seguirá con el proyecto y al personal de LabOSat.

2. Proyecto

2.1 Personal involucrado

Nombre	Gaspar Carlos Domé
Rol	Desarrollador
Categoría Profesional	Estudiante
Responsabilidad	Desarrollo y diseño del sistema
Información de contacto	gaspar.c.dome@gmail.com

Nombre	José Lipovetzky
Rol	Investigador del proyecto SELFIE
Categoría Profesional	Investigador independiente
Responsabilidad	Proveer información técnica especializada sobre partículas
Información de contacto	joselipo@gmail.com

Nombre	Martín Pérez
Rol	Integrante del proyecto SELFIE
Categoría Profesional	Investigador asistente
Responsabilidad	Proveer información técnica especializada sobre partículas
Información de contacto	ing.perezmartin@gmail.com

Nombre	Luciana De Mico
Rol	Directora
Categoría Profesional	Investigador categoría IV
Responsabilidad	Tutelar y orientar en el diseño y seguimiento del desarrollo del proyecto
Información de contacto	lucianadm55@gmail.com

Nombre	Maximiliano Antonelli
Rol	Codirector
Categoría Profesional	Investigador categoría IV

Definición del Tema de Investigación

Responsabilidad	Tutelar y orientar en el diseño y seguimiento del desarrollo del proyecto
Información de contacto	maxanto@fi.mdp.edu.ar

2.2 Definiciones, acrónimos y abreviaturas

Nombre	Descripción
FPGA	Field Programmable Gate Array
FSM	Máquina de estados finita
TFEC	Trabajo final de las carreras de grado de Ingeniería Electrónica e Ingeniería en Computación
VHDL	Very High Speed Integrated Circuit Hardware Description Language

2.3 Referencias

Título del Documento	Referencia
[1] Detector de Radiación Ionizante para aplicaciones satelitales	Ing. Mariano Rolón- Ing. Mathías Sebastian García
[2] EDU-CIAA-FPGA	https://gitlab.com/RamadrianG/wiki---fpga-para-todos/-/wikis/FPGA-para-Todos

2.4 Resumen

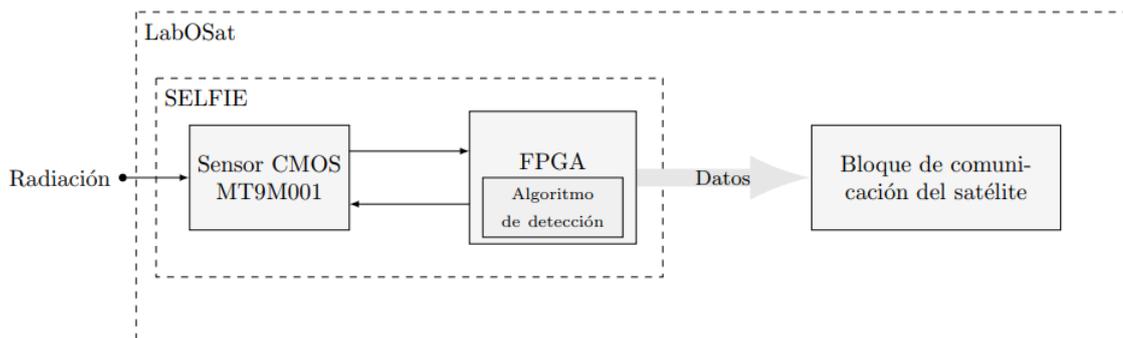
3. Descripción general

3.1 Perspectiva del producto

El proyecto busca ser una continuación, optimización y simplificación del detector desarrollado por el Ing. Mariano Rolón en conjunto con el Ing. Mathías Sebastian García, en dicho proyecto realizado para el laboratorio de bajas temperaturas del instituto balseiro, el Laboratorio de Sistemas Caóticos de la UNMdP y el ICyTE [1] se desarrolló un sistema diseñado para trabajar en entornos espaciales, integrado a satélites. En particular, parte de la plataforma LabOSat, el sistema recibe y cuantifica la radiación ionizante impactando el área del sensor MT9M001C12STM producido por *ONSEMI*® controlado por la FPGA de marca Xilinx y de la familia Zynq 7000.

Definición del Tema de Investigación

El proyecto se implementará usando la “EDU-CIA-FPGA” desarrollada por la UTN-FRH de Haedo y utilizará el mismo sensor que el trabajo previo. La optimización debe integrar de manera simplificada el algoritmo de detección de manera que pueda ser implementada en el área disponible de la FPGA, pudiendo integrar un módulo de memoria RAM si fuese necesario, el sistema debe ser capaz de comunicarse con la plataforma LabOSat, además se diseñará un PCB que integre todos los componentes necesarios debiendo cumplir con todas las restricciones y requisitos de la plataforma y el ambiente.



3.2 Funcionalidad del producto

El sistema debe poder procesar la imagen del sensor CMOS, filtrar los píxeles muertos, detectar los eventos, su tamaño e intensidad y posteriormente comunicarse con el satélite para enviar los datos a tierra.

se evaluará la necesidad de implementar un testeo automático que determine si el sistema opera de la forma esperada, debiendo ser posible disparar con una señal proveniente del satélite y regresando una señal de reporte

Se evaluará agregar un chequeo de píxeles muertos o quemados sujeto a disponibilidad de área.

3.3 Características de los usuarios

- Futuro estudiante: es el responsable de continuar con el desarrollo del instrumento, para que pueda ser integrado al satélite. Deberá contar con conocimiento en desarrollo de PCB de baja y alta frecuencia, protocolos de comunicación, limitaciones y consideraciones de diseño para entornos espaciales.
- Dr. Ing. Martín Pérez y Dr. Ing. José Lipovetzky: interpretarán los datos enviados por el satélite a tierra.
- Personal de LabOSat: son los responsables del satélite en el que irá conectado el instrumento. Deberán contar con conocimiento técnico para conectar el instrumento al satélite

3.4 Restricciones

El desarrollo debe realizarse utilizando la FPGA “EDU-CIAA-FPGA” y el sensor MT9M001C12STM.

Definición del Tema de Investigación

3.5 Suposiciones y dependencias

Este trabajo supone que el área proporcionada por la EDU-CIAA-FPGA será suficiente para albergar todas las dependencias para poder implementar el algoritmo desarrollado en el trabajo previo en su forma más básica.

4. Requisitos específicos

4.1 Interfaces

El dispositivo debe contar con una interfaz compatible con el estándar elegido por los integrantes del equipo encargado del desarrollo del LabOSat

4.2 Funciones

ID requerimiento	RF1
Nombre requerimiento	Escritura de los registros del sensor
Descripción	Deberán escribirse todos los registros disponibles del sensor.
Prioridad	

ID requerimiento	RF2
Nombre requerimiento	Detección de eventos
Descripción	El sistema debe ser capaz de leer el valor de los píxeles y analizar la cantidad de eventos, su tamaño y su intensidad.
Prioridad	

ID requerimiento	RF3
Nombre requerimiento	Consumo
Descripción	El sistema debe, en conjunto, consumir una potencia menor a 1W
Prioridad	Alta

ID requerimiento	RF4
Nombre requerimiento	Compatibilidad
Descripción	El diseño propuesto debe ser compatible con los requisitos y restricciones impuestas por el equipo de LabOSat
Prioridad	Alta

4.3 Requisitos de rendimiento

El instrumento debe ser capaz de trabajar en tiempo real

Definición del Tema de Investigación

4.4 Restricciones de diseño

- Lenguaje VHDL
- Consumo máximo de 1W
- Limitación de hardware: utilizar 3520 celdas lógicas o menos, 80 Kb de ram o menos .
- Interfaz con el satélite: la computadora de a bordo del satélite LabOSat tiene disponibles como interfaces Ethernet, SPI, GPIO y UART.
-

4.5 Atributos del sistema

No aplica

4.6 Otros requisitos

Se debe proponer un diseño de PCB del sistema completo, teniendo en cuenta todas las restricciones de diseño particulares del entorno

- La montura de los circuitos integrados no debe ser BGA
- El estaño se sublima en condiciones de baja presión
- Conectores
- Velocidades (FALTA REUNION CON LOS DEL LABOSAT)
- dimensiones físicas

El sistema debe ser capaz de reportar en qué estado se encuentra el sistema, operativo o no operativo(pixeles muertos), sujeto a la incorporación del chequeo está la limitación de área.

5. Apéndice

No aplica

Apéndice E

Plan de proyecto

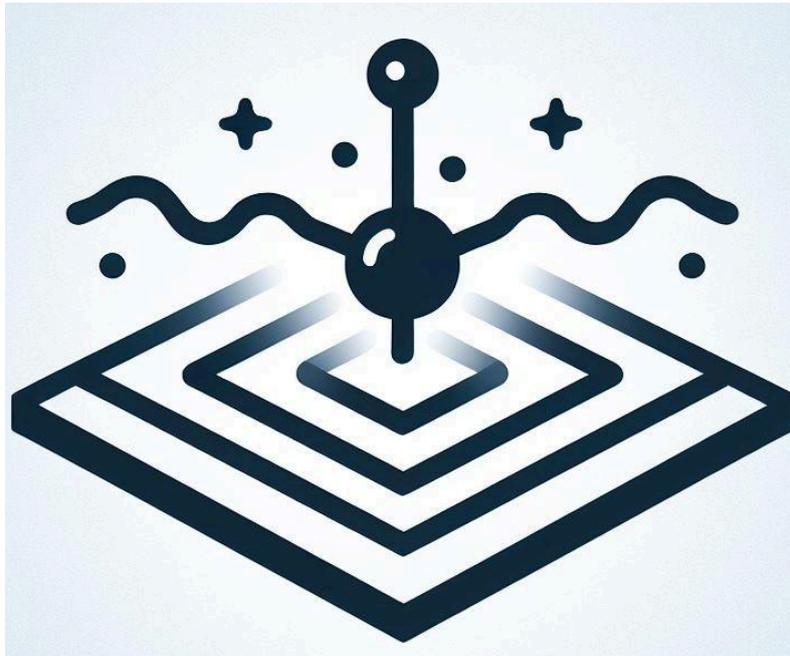
UNMDP-FI

Carrera: Ingeniería electrónica

Seminario-Taller para el Diseño de Soluciones Tecnológicas

TBTSELFIE

Optimización de Algoritmo SELFIE para adecuarlo a placa CIAA FPGA



Versión 1.1

Domé Gaspar Carlos
gaspar.c.dome@gmail.com

Fecha: 07/05/2024

Versión 1.1

Contenido

INTRODUCCIÓN:	3
TAREAS REALIZADAS PRE-PLANIFICACIÓN	3
Cronograma propuesto	3
Riesgos identificados	4

Fecha	Versión	Descripción	Autor/a
2/5/2024	1.0	Versión inicial	Gaspar Carlos Domé
7/5/2024	1.1	Revisión ortográfica, Resumen	Gaspar Carlos Domé

Introducción:

Título propuesto: *Optimización de Algoritmo SELFIE para adecuarlo a placa CIAA FPGA.*

Estudiantes: Domé Gaspar Carlos. Legajo 14793.

Directores: Dra. Ing. Luciana De Mico
Dr. Ing. Maximiliano Antonelli

Lugar de desarrollo: Laboratorio de Sistemas Caóticos (LSC)
Instituto de Investigaciones Científicas y Tecnológicas en
Electrónica (ICyTE)
Conicet - UNMDP

El siguiente documento corresponde al plan de proyecto, lo componen 3 partes:

- Listado de tareas pre-planificación, donde se detallan las distintas actividades que fueron realizadas antes de hacer una planificación formal.
- Cronograma preliminar propuesto
- Estimación de riesgos

Tareas realizadas pre-planificación

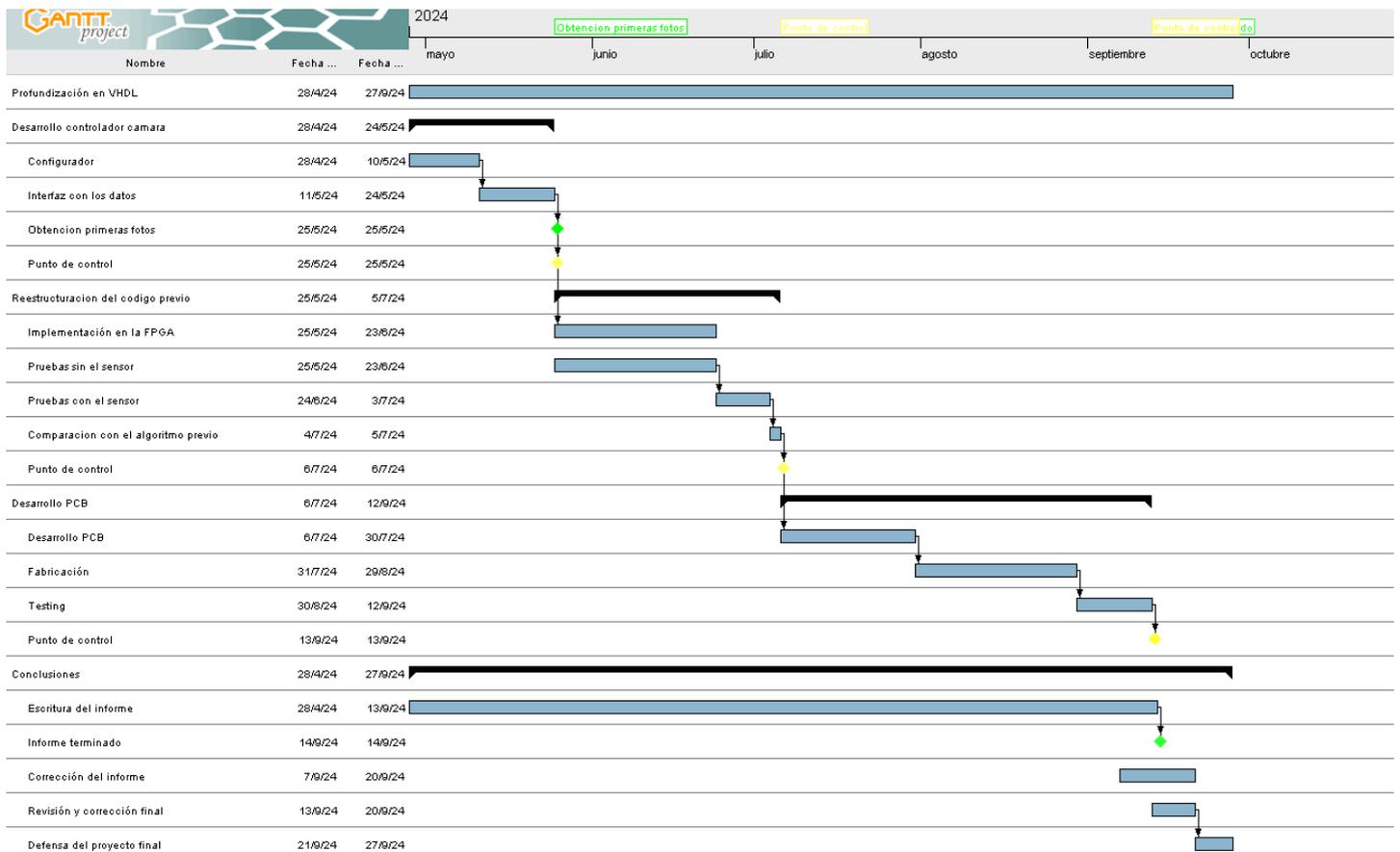
Listado de tareas realizadas previas a la creación de este documento, realizado a partir de la bitácora personal, la fecha oficial de inicio de las actividades fue el 1 marzo del año 2024.

- Investigación trabajo previo : inicio 1 marzo 2024 , duración 1 semana.
- Instalación de las herramientas : inicio 21 marzo 2024, duración 2 semana.
- Investigación sobre el Hardware desarrollado en el trabajo previo: inicio 10 de abril de 2024, duración 2 días
- Exploración sobre alternativas a las herramientas instaladas para tener mayor control sobre la FPGA: inicio 19 de abril 2024, duración 1 semana

Cronograma propuesto

El diagrama de Gantt se divide en 4 etapas distintivas, Desarrollo del controlador de la cámara, Reestructuración del código previo, desarrollo del PCB, conclusiones, entre las etapas se encuentran puntos de control, en amarillo, donde se reúne con los directores y los interesados al proyecto, se presentan los avances y se genera un nuevo plan para cumplir las metas. Los hitos fueron marcados en verde.

La fecha de inicio del diagrama de Gantt es el 28 de abril de 2024, y la fecha tentativa de finalización es el 27 de septiembre de 2024.



Riesgos identificados

El cronograma se realizó teniendo en cuenta tiempos extra en algunas tareas, con el fin de cubrirse de la aparición de inconvenientes con el fin de minimizar los efectos que estos puedan ocasionar en el cronograma.

En la siguiente tabla se detallan los riesgos considerados junto con la severidad, que es una medida del retraso que generaría en el cronograma. También se indica la probabilidad de que ese inconveniente aparezca. La combinación entre severidad y probabilidad da la medida de prioridad siendo los que tengan mayor severidad y mayor probabilidad una prioridad más alta.

Riesgo	Probabilidad de aparición	Severidad	Plan de respuesta	Prioridad
Inicio de un trabajo	Media	Alta	Se flexibilizan los plazos	baja
Rotura sensor CMOS	Baja	Alta	Conseguir nuevos sensores, en su defecto adaptación del sistema a un nuevo sensor CMOS	baja
Rotura de la FPGA	Baja	media	Conseguir una nueva placa de desarrollo CIAA, en su defecto	baja

			limitarse a las celdas lógicas disponibles en la CIAA FPGA e implementarlo en una FPGA alternativa	
Demoras en la fabricación	Media	Media	Se buscará otro fabricante	baja

Apéndice F

Especificación funcional

UNMDP-FI

Carrera: Ingeniería Electrónica

Proyecto: ***Optimización de Algoritmo SELFIE para adecuarlo a placa CIAA FPGA***

Especificación Funcional

Versión 1.0

Domé Gaspar Carlos
gaspar.c.dome@gmail.com

Optimización de Algoritmo SELFIE para adecuarlo a placa CIAA FPGA
Especificación funcional

Fecha	Versión	Descripción	Autor/a
2/5/2025	1.0	Versión inicial	Gaspar Carlos Domé

Índice

1. Introducción	4
1.1 Propósito del documento	4
1.2 Alcance del proyecto	4
1.3 Personal involucrado	4
1.4 Definiciones, acrónimos y abreviaturas	5
1.5 Referencias	5
1.6 Resumen	6
2. Descripción del dispositivo	6
2.1 Bloque 1	6
2.2 Bloque 2	7
3. Especificaciones funcionales	7
3.1 RF01: Escritura de los registros del sensor	7
3.2 RF02: Detección de eventos	7

Optimización de Algoritmo SELFIE para adecuarlo a placa CIAA FPGA

Especificación funcional

1. Introducción

Este documento corresponde a la Especificación Funcional para la Optimización de Algoritmo SELFIE para adecuarlo a placa CIAA FPGA. Esta especificación se ha estructurado basándose en la información mencionada en el documento de Especificación de Requerimientos (ER) [1]

1.1 Propósito del documento

El presente documento tiene como propósito proveer información detallada sobre cómo funcionará el sistema, cuáles serán sus comportamientos deseados y cómo se deberá construir, con base en los requerimientos anteriormente definidos en la ER.

Está dirigido a:

- Los desarrolladores del instrumento, quienes lo construirán.
- Los directores y solicitantes del proyecto, así como a quienes lo evaluarán, con el fin de corroborar su funcionamiento
- Los futuros estudiantes, que a partir del desarrollo, construirán el PCB

1.2 Alcance del proyecto

Este proyecto tiene como objetivo el diseño y desarrollo de un sistema de adquisición de imágenes basado en la FPGA iCE40HX4K y el sensor óptico CMOS MT9M001C12STM, destinado a operar en entornos espaciales como parte de la carga útil de un satélite. El sistema debe ser capaz de capturar y procesar imágenes en condiciones adversas, incluyendo exposición a radiación ionizante, temperaturas extremas y restricciones estrictas en consumo energético.

La arquitectura desarrollada se encarga del control del sensor, la sincronización de lectura, y el procesamiento de las imágenes adquiridas para la generación de histogramas. Asimismo, gestiona la comunicación con el sistema principal del satélite. Todas las funcionalidades se implementan en lenguaje VHDL, complementadas por el diseño de una placa de circuito impreso (PCB) que cumple con los requisitos eléctricos y de potencia definidos por la plataforma LabOSat.

1.3 Personal involucrado

Nombre	Gaspar Carlos Domé
Rol	Desarrollador
Categoría Profesional	Estudiante
Responsabilidad	Desarrollo y diseño del sistema
Información de contacto	gaspar.c.dome@gmail.com

Nombre	José Lipovetzky
Rol	Investigador del proyecto SELFIE
Categoría Profesional	Investigador independiente CONICET

Optimización de Algoritmo SELFIE para adecuarlo a placa CIAA FPGA
Especificación funcional

Responsabilidad	Proveer información técnica especializada sobre partículas
Información de contacto	joselipo@gmail.com

Nombre	Martín Pérez
Rol	Integrante del proyecto SELFIE
Categoría Profesional	Investigador asistente CONICET
Responsabilidad	Proveer información técnica especializada sobre partículas
Información de contacto	ing.perezmartin@gmail.com

Nombre	Luciana De Mico
Rol	Directora
Categoría Profesional	Investigadora adjunta CONICET, Prof. Asociada UNMDP, categoría 3.
Responsabilidad	Tutelar y orientar en el diseño y seguimiento del desarrollo del proyecto
Información de contacto	lucianadm55@gmail.com

Nombre	Maximiliano Antonelli
Rol	Codirector
Categoría Profesional	Investigador asistente CONICET, Prof. Adjunto UNMDP, categoría 4.
Responsabilidad	Tutelar y orientar en el diseño y seguimiento del desarrollo del proyecto
Información de contacto	maxanto@fi.mdp.edu.ar

1.4 Definiciones, acrónimos y abreviaturas

Nombre	Descripción
RF	Requerimiento Funcional
RNF	Requerimiento No Funcional
FPGA	Field-programmable gate array

1.5 Referencias

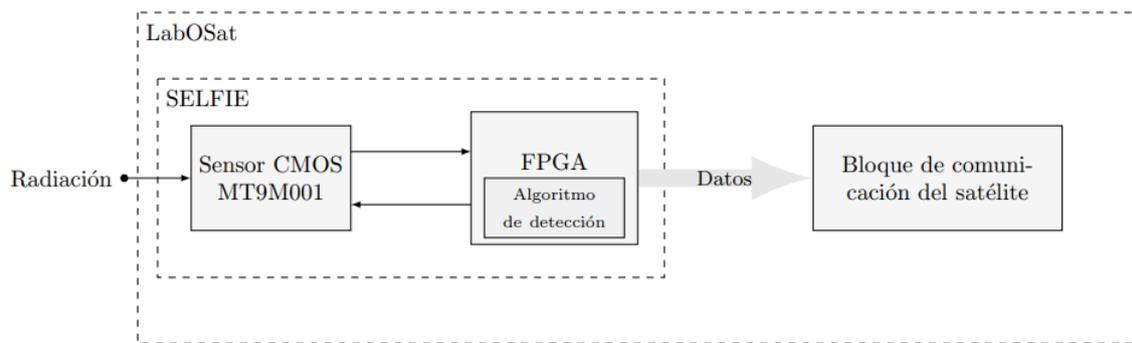
Título del Documento
[1] Optimización de Algoritmo SELFIE para adecuarlo a placa CIAA FPGA - Especificación de requerimientos
[2] Particle detection and classification using commercial off the shelf CMOS image sensors - Jose Lipovetzky y Martin Pérez

[3] Detector de Radiación Ionizante para aplicaciones satelitales(SELFIE) - Mariano Rolón y Mathías Sebastian García

1.6 Resumen

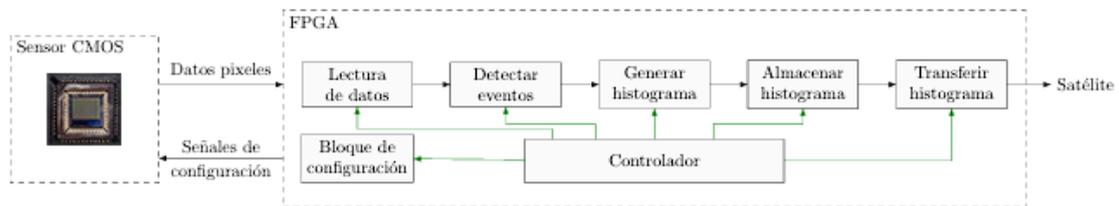
2. Descripción del dispositivo

El dispositivo, se compone de un sensor CMOS MT9M001 y una FPGA encargado de realizar el procesamiento y comunicación con la plataforma de la que será parte, el LabOSat



El principio lógico de funcionamiento se basa en el efecto que produce la radiación al interactuar con el sensor. Cuando una partícula cargada impacta sobre la superficie del sensor, deposita una cierta cantidad de carga que se manifiesta como una región con mayor brillo en comparación con los píxeles circundantes. De este modo, al exponer el sensor a un flujo de radiación, se genera un patrón de puntos y trazas que representan las trayectorias de las partículas cargadas a través del sensor.

El objetivo del sistema es detectar estos eventos individuales y clasificarlos según su tamaño y nivel de brillo. A partir de esta categorización, se genera un histograma que permite, siguiendo lo desarrollado por los doctores José Lipovetzky y Martín Pérez [2], estimar tanto el tipo como la cantidad de partículas que atravesaron el sensor durante una irradiación, luego enviarlo al satélite



2.1 Bloque 1

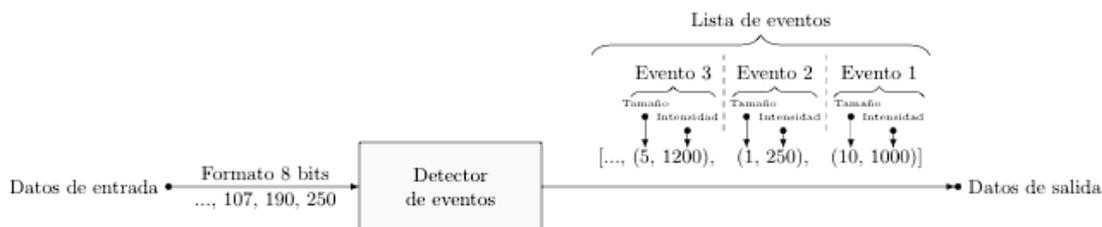
Bloque de configuración del sensor, se encarga de generar los datos necesarios para configurar al sensor CMOS MT9M001 para funcionar correctamente según el diseño, no permitirá la lectura de los registros del sensor, ya que no es necesario debido a que se puede volver a configurar sin necesidad de leer los registros, pudiendo en una eventual falla reiniciar el sensor y volverlo a configurar

2.2 Bloque 2

Bloque de adquisición de píxeles, bloque encargado de recibir las señales de control proveniente del sensor y leer los valores de los píxeles, reportar al resto del sistema que se tiene un dato disponible y en caso de perder un dato enviar una señal de error al controlador

2.3 Bloque 3

Bloque detector de eventos es el encargado de aplicar el algoritmo desarrollado en el trabajo de Rolón y Garcia[3] y categorizar los eventos, tanto en tamaño y energía, luego de la categorización, enviar una señal para realizar el histograma a partir de los datos categorizados .



2.4 Bloque 4

Bloque generador de histogramas, encargado de generar el histograma a partir de los datos provenientes del bloque detector de eventos, luego guardarlos en una memoria circular para su posterior envío al satélite

2.5 Bloque 5

Bloque de control y comunicación, encargado de coordinar las distintas máquinas de estado y recibir comandos del satélite, pudiendo restablecer todos los datos a cero, en un estado inicial si se percibiera algún error ya sea por la pérdida de datos o violación en los tiempos de respuesta

2.6 Bloque 6

Fuente de alimentación, encargado de proveer los 3.3 V y 1.2 V necesarios para el correcto funcionamiento de la FPGA y el sensor, debe tener una alta eficiencia. Para cumplir con RF3, debe ser una fuente conmutada ya que se espera el satélite provea una tensión máxima de 12 V y mínima de 9 V y se desestima el uso de fuentes lineales.

3. Especificaciones funcionales

3.1 RF01: Escritura de los registros del sensor

Deberán poderse escribir los registros internos del sensor, siendo capaz de configurarlo para su correcto funcionamiento, además de dar flexibilidad a futuros usuarios para su calibración en distintos entorno de radiación

3.2 RF02: Detección de eventos

El sistema debe ser capaz de recibir los píxeles, procesarlos con un algoritmo basado en el desarrollado en el trabajo de Rolón y Garcia [3] pero adaptado a las nuevas necesidades de la FPGA iCE40HX4k, luego de la categorización deber poder generar histogramas y almacenarlos para su posterior descarga al satélite

3.3 RF03: Consumo

El sistema debe tener un consumo total, incluyendo las pérdidas por alimentación, la FPGA con el sistema cargado y el sensor menor a 2 W, siendo compatible con las tensiones provistas por la plataforma LabOSat que se prevén sean entre 9 y 12 V.

3.4 RF04: Comparabilidad

El sistema debe adecuarse a los requisitos impuestos por la plataforma LabOSat tanto en la elección de los componentes, las restricciones en volumen, la posibilidad de blindar el dispositivo ante situaciones de extrema radiación y conectores.

Apéndice G

Especificación técnica

UNMDP-FI

Carrera: Ingeniería Electrónica

Proyecto: ***Optimización del hardware para medición de radiación utilizando sensores de imagen CMOS***

Especificación Técnica

Versión 1.0

Domé Gaspar Carlos
gaspar.c.dome@gmail.com

Optimización del hardware para medición de radiación utilizando sensores de imagen CMOS

Especificación técnica

Fecha	Versión	Descripción	Autor/a
2/5/2025	1.0	Versión inicial	Gaspar Carlos Domé

**Optimización del hardware para medición de
radiación utilizando sensores de imagen CMOS**

Especificación técnica

Índice

1. Alcance.....	4
2. Referencias.....	4
2.1 Definiciones, acrónimos y abreviaturas.....	4
2.2 Referencias.....	4
3. Diseño de hardware.....	5
3.1 FPGA iCE40HX4K.....	5
3.1.1 Alimentación.....	5
3.1.2 Reloj.....	6
3.1.3 Programación.....	6
3.1.4 PCB del programador completa.....	10
3.1.5 Bancos.....	10
3.2 Sensor óptico MT9M001C12STM.....	11
3.3 Fuente de alimentación.....	12
3.3.1 Fuente basada en el LM2576.....	12
3.3.2 Selección del inductor.....	13
3.3.3 Selección de capacitores.....	14
3.3.4 Sistema de protección.....	15
3.3.5 Fuente lineal.....	16
3.3.6 PCB completa.....	17
4. Procedimiento de construcción.....	18
5. Código.....	18
5.1 Configurador.....	18
5.1.1 Máquina de estados protocolo “Programador”.....	18
5.1.2 Máquina de estados controlador del “Programador”.....	21
5.2 Capturador de píxeles.....	27
5.3 Algoritmo/Histograma.....	30
5.4 Memoria.....	36
5.5 Selector de acceso a memoria.....	38
5.6 UART RX.....	40
5.7 UART TX.....	41
5.8 Controlador.....	47
6. Simulaciones y pruebas.....	52

Optimización del hardware para medición de radiación utilizando sensores de imagen CMOS

Especificación técnica

1. Alcance

Este proyecto tiene como objetivo el diseño y desarrollo de un sistema de adquisición de imágenes basado en la FPGA iCE40HX4K y el sensor óptico CMOS MT9M001C12STM, destinado a operar en entornos espaciales como parte de la carga útil de un satélite. El sistema debe ser capaz de capturar y procesar imágenes en condiciones adversas, incluyendo exposición a radiación ionizante, temperaturas extremas y restricciones estrictas en consumo energético.

La arquitectura desarrollada se encarga del control del sensor, la sincronización de lectura, y el procesamiento de las imágenes adquiridas para la generación de histogramas. Asimismo, gestiona la comunicación con el satélite. Todas las funcionalidades se implementan en lenguaje VHDL, complementadas por el diseño de una placa de circuito impreso (PCB).

2. Referencias

2.1 Definiciones, acrónimos y abreviaturas

Nombre	Descripción
RF	Requerimiento Funcional
FPGA	Field-programmable gate array

2.2 Referencias

Título del Documento
[1] Optimización de Algoritmo SELFIE para adecuarlo a placa CIAA FPGA - Especificación de requerimientos
[2] Particle detection and classification using commercial off the shelf CMOS image sensors - Jose Lipovetzky y Martin Pérez
[3] Detector de Radiación Ionizante para aplicaciones satelitales(SELFIE) - Mariano Rolón y Mathías Sebastian García
[4]iCE40™ LP/HX/LM Family Handbook-Lattice Semiconductor
[5] Proyecto CIAA - EDU -CIAA- FPGA - Mg. Ing. Facundo Larosa, Ing. Martín Heredia, Ing. Ramiro Ghignone, Sr. Federico Vazquez Saraullo, Sr. Hernán Mendes Gouveia y Sr. Martín Fernández
[6] FT PROG - [link]
[7]Basic USB Type-C™ Upstream Facing Port Implementation - [link]
[8]What's the role of CC pin in Type-C solution - [link]

Optimización del hardware para medición de radiación utilizando sensores de imagen CMOS

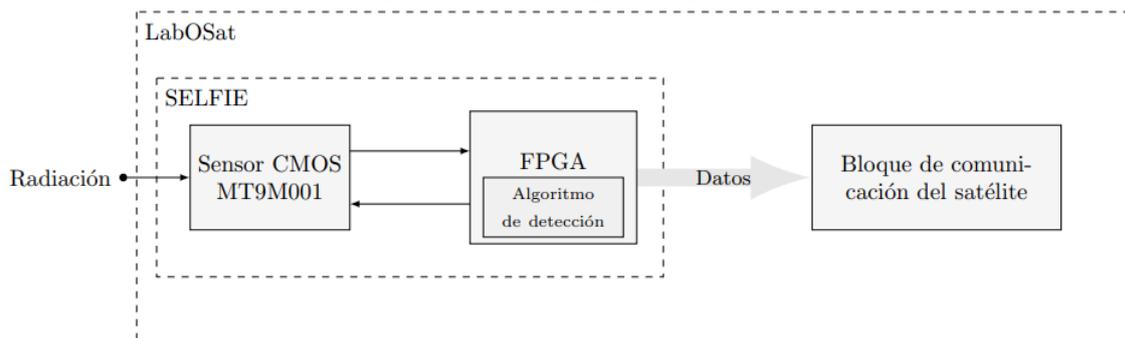
Especificación técnica

[9]Demystifying the USB Type C Connector – Tyler Ward - [\[link\]](#)

[10]LM2576xx Series SIMPLE SWITCHER® - [\[link\]](#)

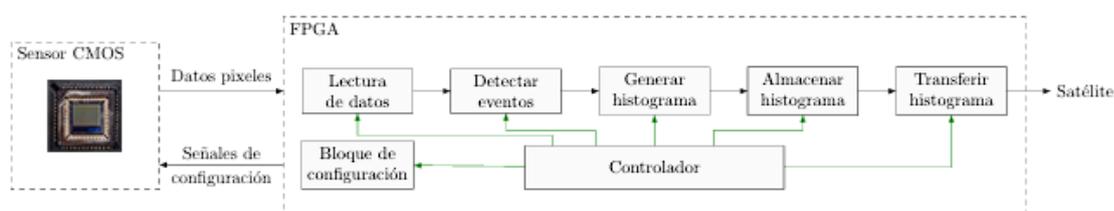
2.3 Descripción del dispositivo

El dispositivo, se compone de un sensor CMOS MT9M001 y una FPGA encargado de realizar el procesamiento y comunicación con la plataforma de la que será parte, el LabOSat



El principio lógico de funcionamiento se basa en el efecto que produce la radiación al interactuar con el sensor. Cuando una partícula cargada impacta sobre la superficie del sensor, deposita una cierta cantidad de carga que se manifiesta como una región con mayor brillo en comparación con los píxeles circundantes. De este modo, al exponer el sensor a un flujo de radiación, se genera un patrón de puntos y trazas que representan las trayectorias de las partículas cargadas a través del sensor.

El objetivo del sistema es detectar estos eventos individuales y clasificarlos según su tamaño y nivel de brillo. A partir de esta categorización, se genera un histograma que permite, siguiendo lo desarrollado por los doctores José Lipovetzky y Martín Pérez [2], estimar tanto el tipo como la cantidad de partículas que atravesaron el sensor durante una irradiación, luego enviarlo al satélite



3. Diseño de hardware

3.1 FPGA iCE40HX4K

Se eligió la FPGA iCE40HX4K por dos motivos principales. En primer lugar, existen versiones idénticas de esta FPGA que pueden ser programadas de forma permanente mediante tecnología antifusible, lo que las hace más robustas frente a los efectos de la radiación en ambientes hostiles como el espacio. En segundo lugar, esta FPGA forma parte de la plataforma abierta de hardware EDU-CIAA-FPGA, lo que facilita el desarrollo al tener acceso a diseños previos y documentación técnica en su comunidad.

Optimización del hardware para medición de radiación utilizando sensores de imagen CMOS

Especificación técnica

Sus características clave incluyen:

- Bajo costo
- Número reducido de unidades lógicas, lo que impone un desafío de optimización en el diseño del algoritmo
- Empaquetado 144-LQFP, adecuado para aplicaciones espaciales debido a la mayor flexibilidad de los pines comparado con BGA

3.1.1 Alimentación

Se incorporan capacitores de desacople a la alimentación de la FPGA los valores se eligen siguiendo los *guidelines* del datasheet[4] y las recomendaciones del proyecto EDU-CIAA FPGA[5]

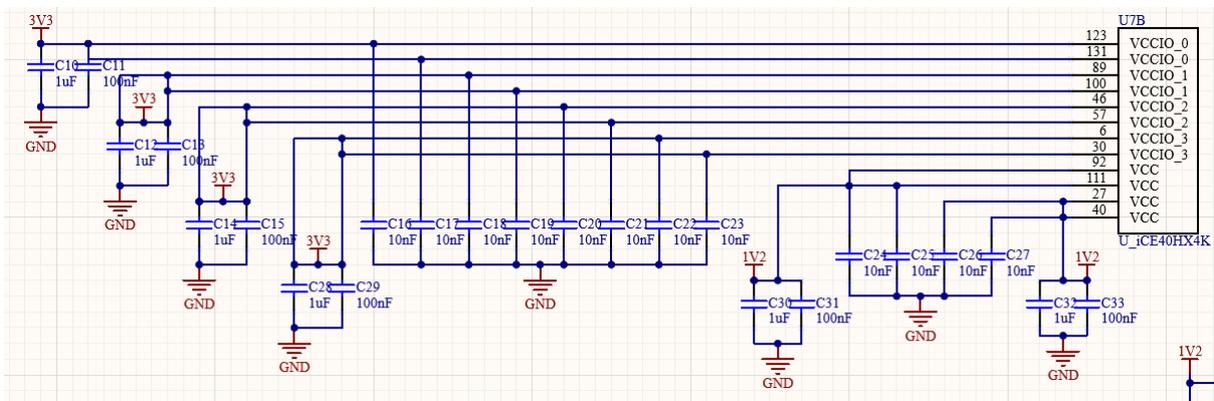


Figura 1. Alimentación FPGA

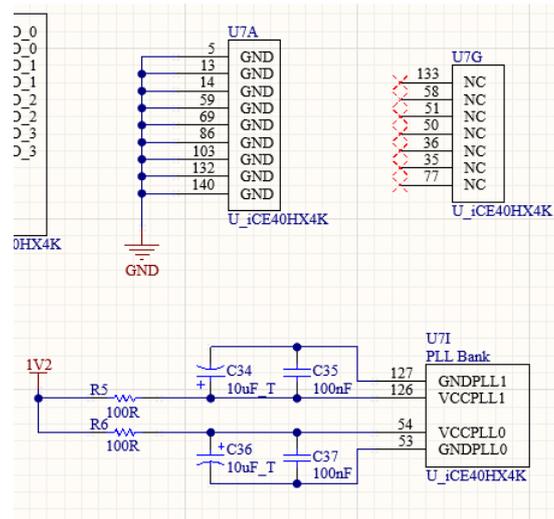


Figura 2. Alimentación PLL

3.1.2 Reloj

Se siguen las recomendaciones del proyecto EDU-CIAA-FPGA en la selección del reloj, este es un chip de cuatro conexiones al que se lo alimenta y entrega una señal de 12 MHz, el ECS-3225MV es el dispositivo

Optimización del hardware para medición de radiación utilizando sensores de imagen CMOS

Especificación técnica

recomendado, siguiendo las recomendaciones del datasheet se incorpora un capacitor de desacople de 100 nF.

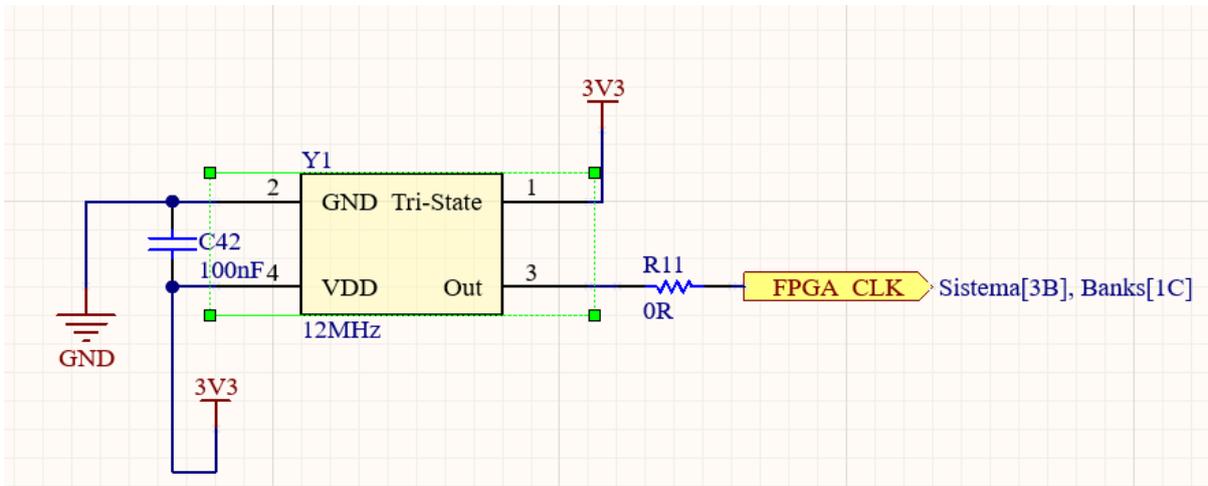


Figura 3. Reloj 12 MHz para la FPGA

3.1.3 Programación

La programación del dispositivo se realiza con un PCB extra que incorpora el integrado FT2232H, que se encarga de la conversión entre el protocolo USB y SPI, la programación se realiza sobre una memoria flash serie la W25X40CLSNIG ambos dispositivos son recomendados por el *hanbook*[4].

la conexión se realiza según lo mostrado en la siguiente figura, donde a la izquierda se muestra una bornera para conectar el programador, que va directo a la memoria flash, luego cuando la FPGA es alimentada lee la memoria y se autoconfigura

Optimización del hardware para medición de radiación utilizando sensores de imagen CMOS

Especificación técnica

La placa de programación utiliza el chip FT2232H y requiere de una pequeña EEPROM para la configuración de los puertos del chip, en este caso se utiliza la M93C66, se debe utilizar el FT PROG para configurarla previo a la utilización del programador[6]

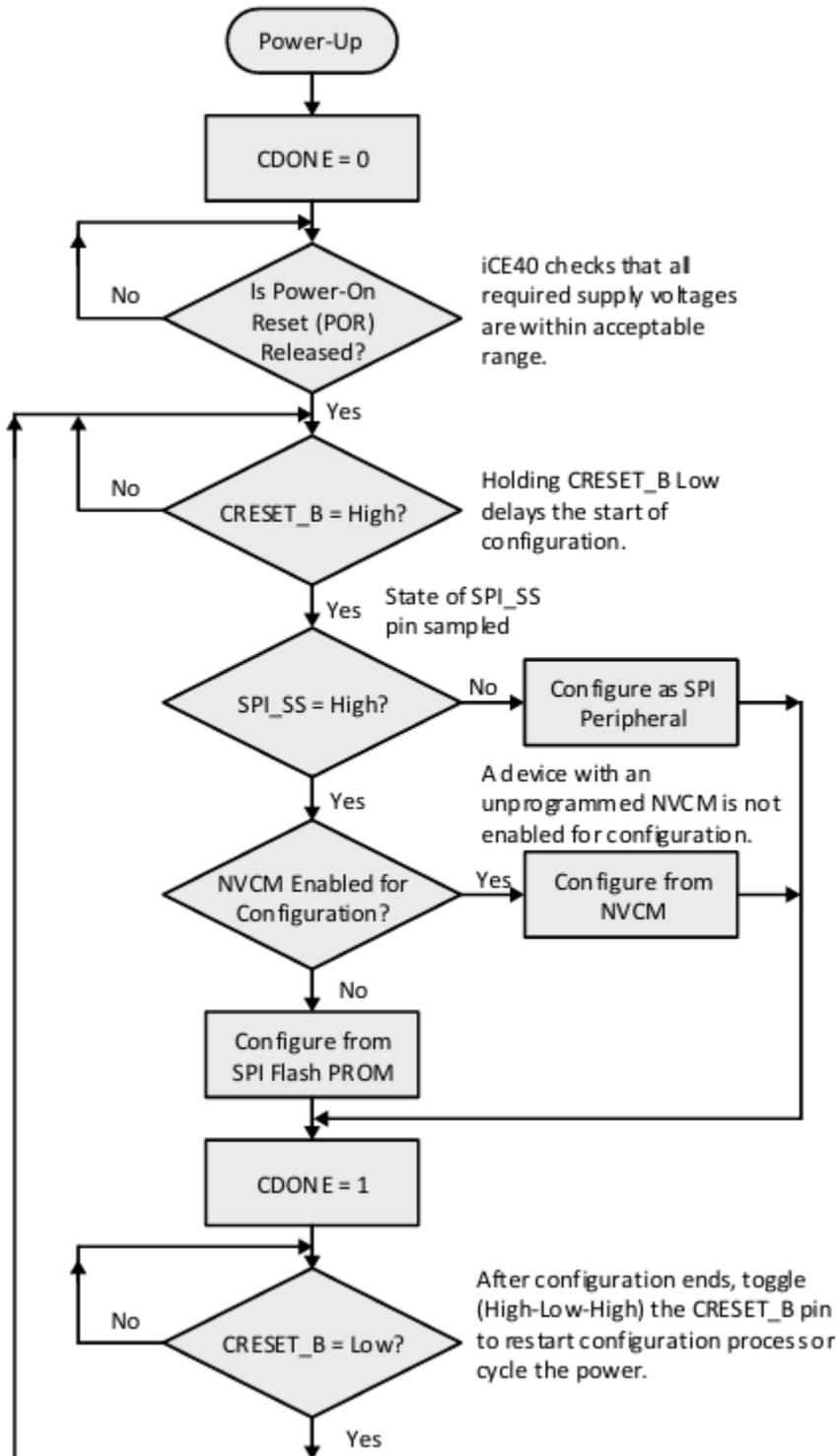


Figure 3.1. iCE40 Device Configuration Control Flow

Optimización del hardware para medición de radiación utilizando sensores de imagen CMOS

Especificación técnica

Figura 6. Proceso de configuración de la FPGA

como conector principal se utiliza un conector USB tipo C

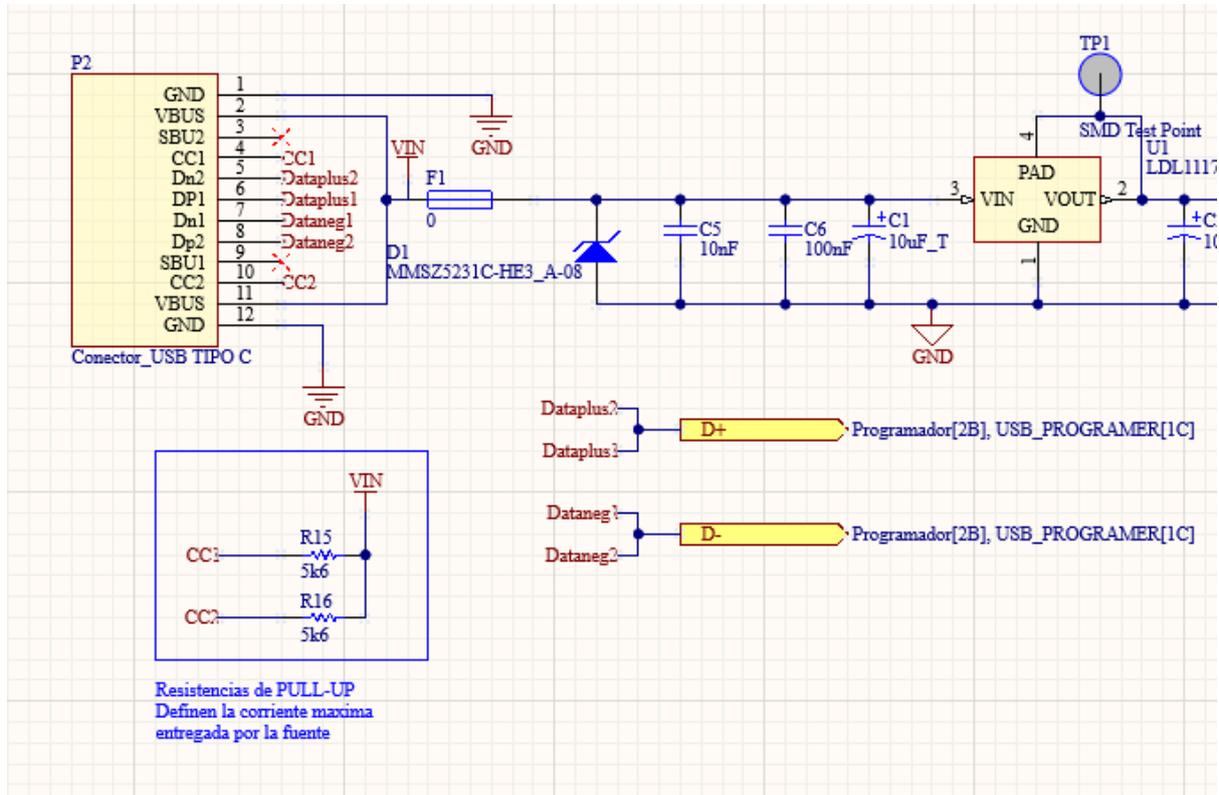


Figura 7. Conector USB tipo C

donde todos los pines vienen duplicados ya que el conector no tiene un sentido único de conexión, los detalles de conexión se extrajeron de las fuentes [7,8,9].

3.1.4 PCB del programador completa

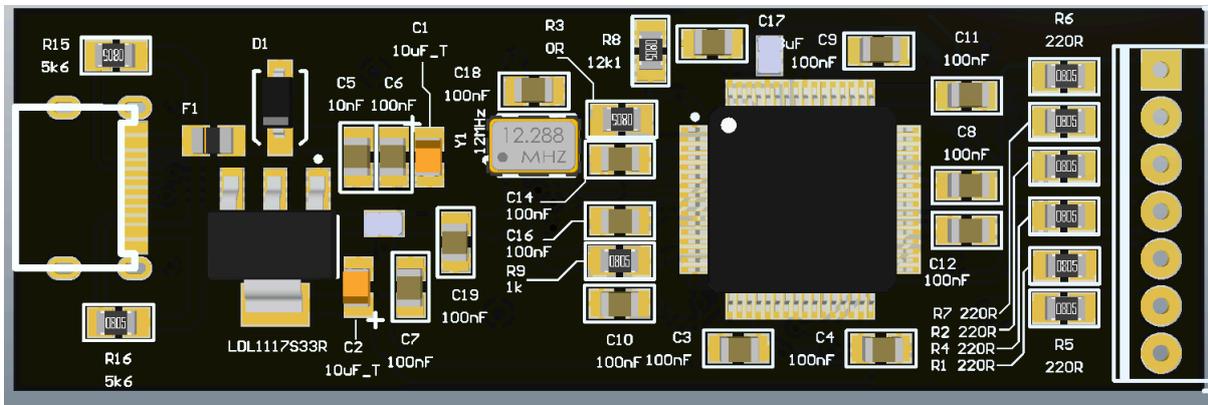


Figura 8. Placa del programador

Optimización del hardware para medición de radiación utilizando sensores de imagen CMOS

Especificación técnica

3.1.5 Bancos

Los bancos restantes de la FPGA se conectan al sensor, a un led, a una *tactile switch* y a una bornera para posibles futuras expansiones o modificaciones, el resto de pines no utilizado se deja sin conectar

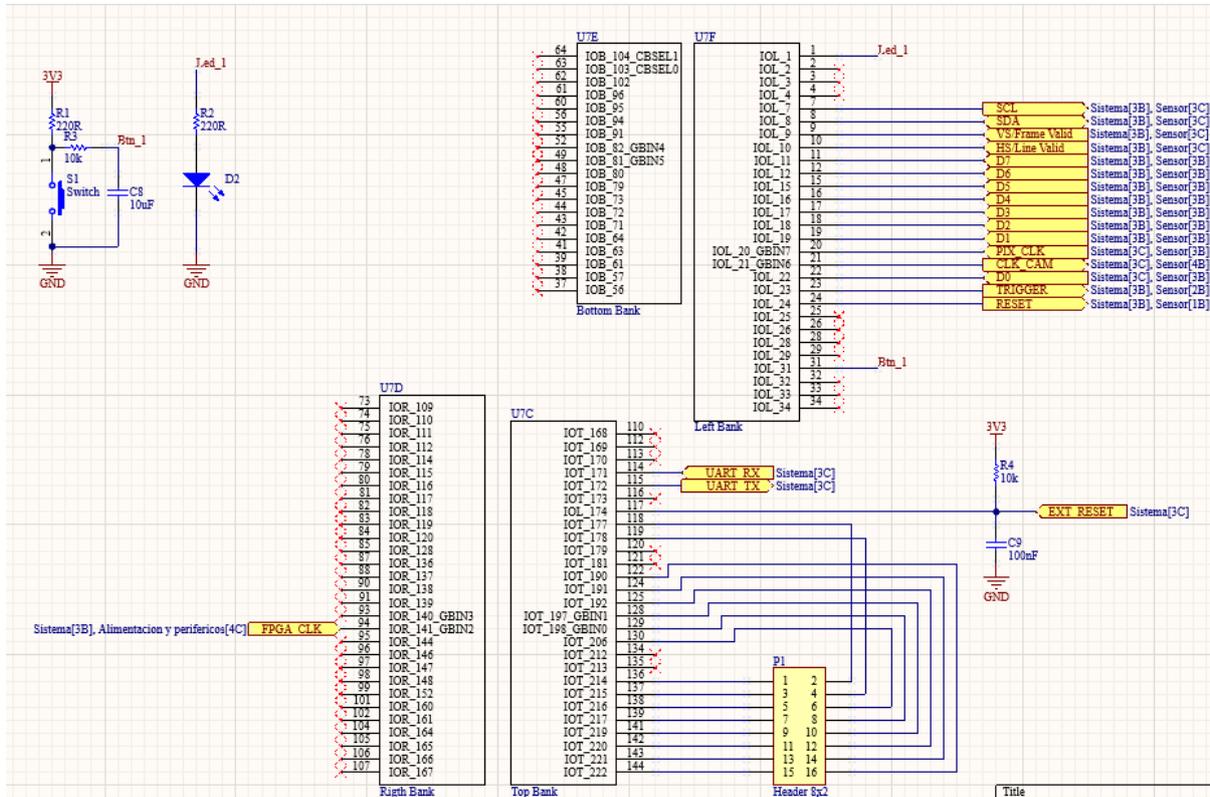


Figura 9. Bancos de la FPGA

El banco B es utilizado para conectar el sensor a la FPGA

3.2 Sensor óptico MT9M001C12STM

Este sensor fue seleccionado por continuidad con un trabajo anterior sobre el cual se amplía el presente proyecto, constituyendo una restricción de diseño.

Características técnicas:

- Resolución de 1280x1024 píxeles activos
- Salida paralela de 8 bits que representa el brillo de cada píxel
- Señales de sincronización: Clock de lectura, Frame Valid y Line Valid
- Interfaz de configuración serie I2C

Los componentes complementarios al sensor fueron seleccionados de acuerdo a lo desarrollado en el trabajo de los ingenieros Rolon y Garcia[3], el mismo incluye capacitores de desacople y resistencias de pull-up para la interfaz de configuración

Optimización del hardware para medición de radiación utilizando sensores de imagen CMOS

Especificación técnica

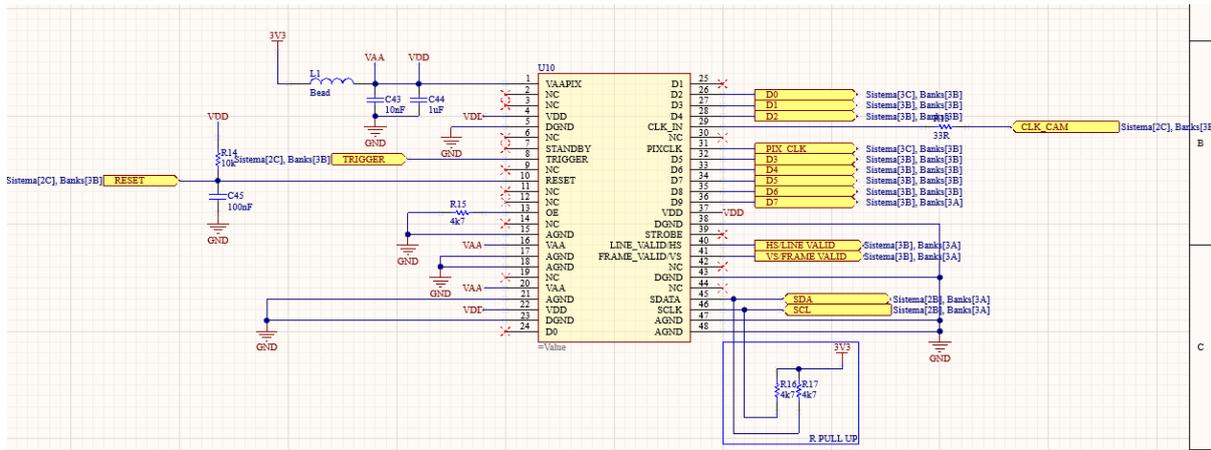


Figura 10. Sensor

3.3 Fuente de alimentación

El sistema recibe alimentación de 12V o 9V desde el satélite. Esta tensión es reducida mediante una fuente DC-DC basada en el regulador LM2576 para obtener 3.3V. Luego, se utiliza un regulador lineal adicional para obtener los 1.2V necesarios para el núcleo de la FPGA.

Además se incorporan distintas protecciones por sobretensión y sobrecorriente

3.3.1 Fuente basada en el LM2576

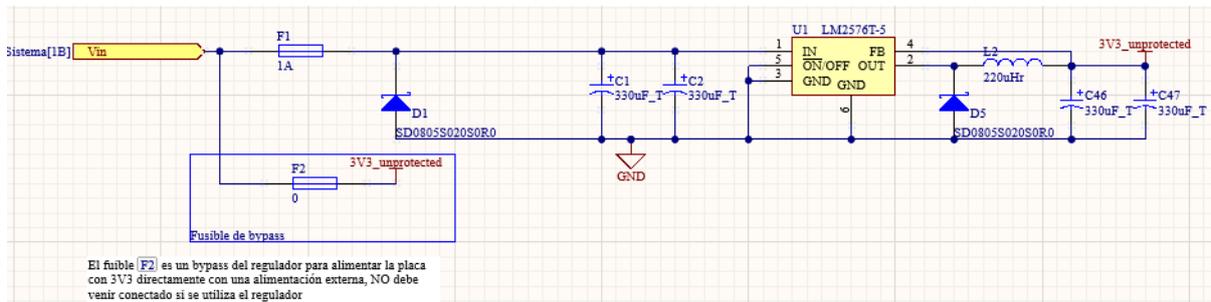


Figura 11. Fuente conmutada LM2576

los fusibles se incorporan para poder seleccionar si se quiere utilizar la fuente o no, los diodos schottky se seleccionan por recomendación del fabricante [10]

Optimización del hardware para medición de radiación utilizando sensores de imagen CMOS

Especificación técnica

Table 8-3. Diode Selection Guide

V _R	SCHOTTKY		FAST RECOVERY	
	3 A	4 A to 6 A	3 A	4 A to 6 A
20 V	1N5820 MBR320P SR302	1N5823	The following diodes are all rated to 100-V 31DF1 HER302	The following diodes are all rated to 100-V 50WF10 MUR410 HER602
30 V	1N5821 MBR330 31DQ03 SR303	50WQ03 1N5824		
40 V	1N5822 MBR340 31DQ04 SR304	MBR340 50WQ04 1N5825		
50 V	MBR350 31DQ05 SR305	50WQ05		
60 V	MBR360 DQ06 SR306	50WR06 50SQ060		

Figura 12. Selección del diodo

Como la tensión y la corriente va a ser mínima según estimaciones de consumo, se selecciona una alternativa SMD al 1N5820, se selecciona el SD0805S020S1R0 que cumple con especificaciones similares al 1N5820.

3.3.2 Selección del inductor

Para la selección del inductor se utiliza la figura 12

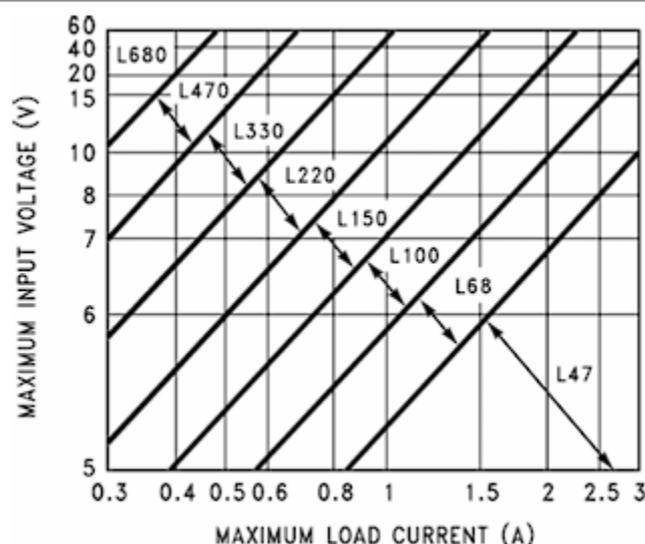


Figure 8-4. LM2576(HV)-3.3

Figura 13. Selección del inductor

Optimización del hardware para medición de radiación utilizando sensores de imagen CMOS

Especificación técnica

Suponiendo una carga máxima de 1A, que supera más de 2 veces el máximo consumo estimado, se requiere un L220 que utilizando la tabla de la figura 13 nos indica un inductor de 220 uHr

INDUCTOR CODE	INDUCTOR VALUE	SCHOTT ⁽¹⁾	PULSE ENG. ⁽²⁾	RENCO ⁽³⁾
L47	47 µH	671 26980	PE-53112	RL2442
L68	68 µH	671 26990	PE-92114	RL2443
L100	100 µH	671 27000	PE-92108	RL2444
L150	150 µH	671 27010	PE-53113	RL1954
L220	220 µH	671 27020	PE-52626	RL1953
L330	330 µH	671 27030	PE-52627	RL1952
L470	470 µH	671 27040	PE-53114	RL1951
L680	680 µH	671 27050	PE-52629	RL1950
H150	150 µH	671 27060	PE-53115	RL2445
H220	220 µH	671 27070	PE-53116	RL2446
H330	330 µH	671 27080	PE-53117	RL2447
H470	470 µH	671 27090	PE-53118	RL1961
H680	680 µH	671 27100	PE-53119	RL1960
H1000	1000 µH	671 27110	PE-53120	RL1959
H1500	1500 µH	671 27120	PE-53121	RL1958
H2200	2200 µH	671 27130	PE-53122	RL2448

3.3.3 Selección de capacitores

para que sea estable se debe cumplir lo siguiente

1. The value of the output capacitor together with the inductor defines the dominate pole-pair of the switching regulator loop. For stable operation, the capacitor must satisfy Equation 11:

$$C_{OUT} \geq 13,300 \frac{V_{IN(Max)}}{V_{OUT} \times L(\mu H)} (\mu F)$$

por lo tanto $C_{OUT} \geq 220 \mu F$

$$\text{Output Ripple Voltage} = (\Delta I_{IND}) (\text{ESR of } C_{OUT})$$

para mantener el ripple aceptable (<1%) se seleccionan capacitores con un ESR bajo

$$\text{Cálculo de } \Delta i_L: \Delta i_L = \frac{V_O \cdot (1-D) \cdot T_S}{L}$$

por lo tanto si $D = \frac{V_{out}}{V_{in}} = 0.275$ a 0.36 dependiendo la tensión de entrada, $T_S = \frac{1}{52kHz} \rightarrow \Delta I_L = 0.18$ a $0.2A$, seleccionando capacitores de tantalio, para reducir ESR, se selecciona el TPME337*016#0050, un capacitor de 330 uF que presenta 50 mΩ de ESR quedando $\Delta V = 10 \text{ mV} \rightarrow \sim 0.3\%$ de la tensión, cumpliendo los requisitos.

Optimización del hardware para medición de radiación utilizando sensores de imagen CMOS

Especificación técnica

Como medida adicional de estabilidad, el datasheet recomienda utilizar capacitores en paralelo en condiciones de temperaturas bajas, como podrían ser condiciones espaciales

If the operating temperature range includes temperatures below -25°C , the input capacitor value can need to be larger. With most electrolytic capacitors, the capacitance value decreases and the ESR increases with lower temperatures and age. Paralleling a ceramic or solid tantalum capacitor increases the regulator stability at cold temperatures. For maximum capacitor operating lifetime, the RMS ripple current rating of the capacitor must be greater than:

Por lo tanto se adiciona otro capacitor TPME337*016#0050 en paralelo para reducir los efectos de ESR y mejorar el comportamiento bajo diferentes condiciones termicas.

3.3.4 Sistema de protección

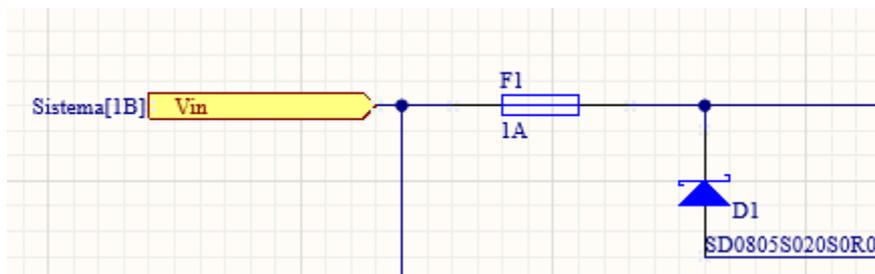


Figura 14. Protecciones alimentación

Como primera protección se incorpora un fusible y un diodo para protección contra sobre corriente y tensión inversa

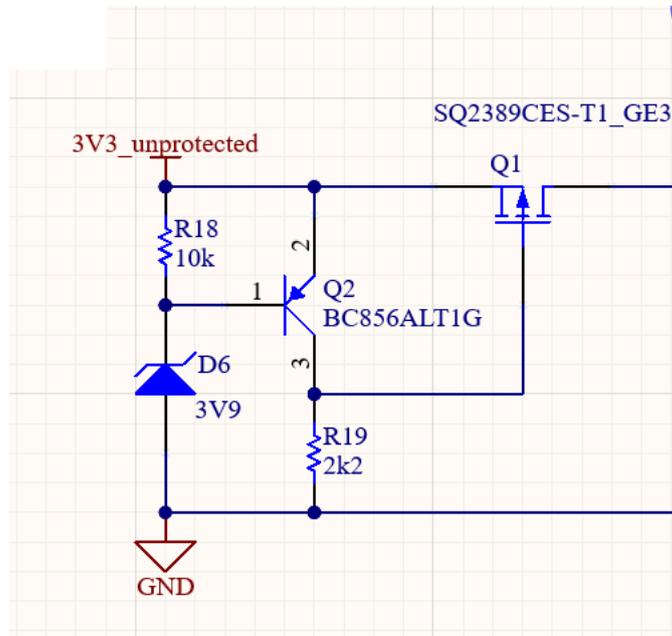


Figura 15. Protección sobretensión

Optimización del hardware para medición de radiación utilizando sensores de imagen CMOS

Especificación técnica

Posterior a la fuente conmutada se incorpora una protección por sobretensión, si por algún motivo la fuente conmutada fallara en corto, se proporcionarán 12 o 9 volts directamente a la entrada de la alimentación de 3,3 V rompiendo la FPGA y el sensor, para evitar esto se incorpora el circuito de protección de la figura 15, que se activa si se sobrepasa la tensión del zener desactivando el mosfet de alimentación hasta que la tensión disminuya, una mejora a este circuito podría ser incorporar un zener configurable y disminuir la tensión de protección a 3.4/3.5 V.

3.3.5 Fuente lineal

Adicionalmente a los 3,3V que necesita la FPGA se requiere 1,2 V, esto es proporcionado por un regulador lineal

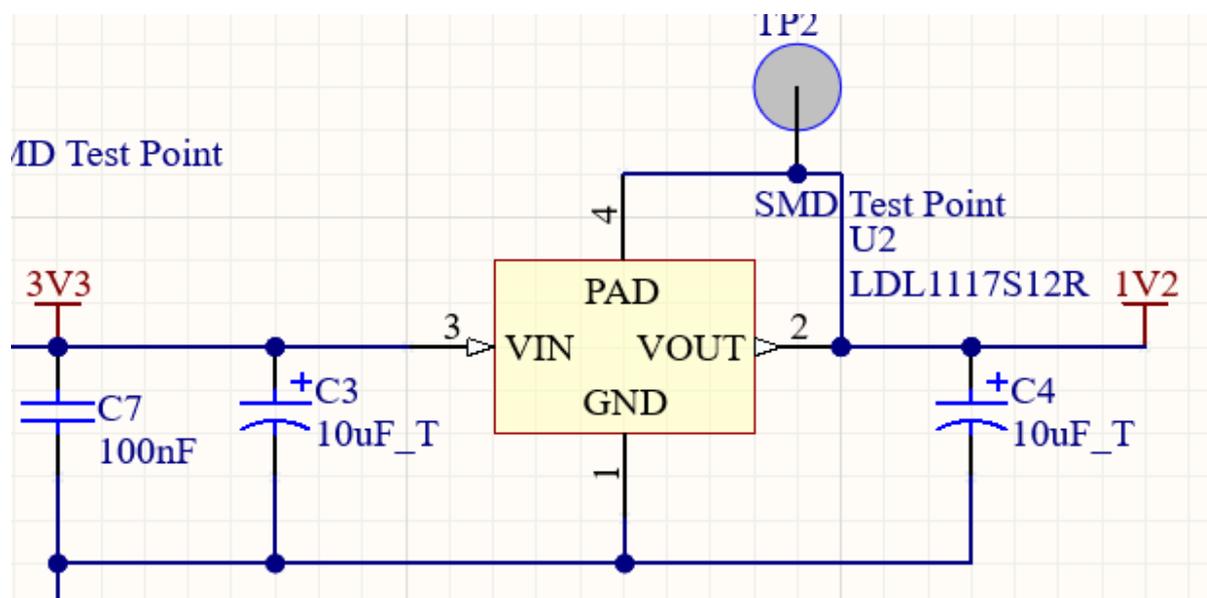


Figura 16. Regulador lineal LDL1117S12R

Se siguen las recomendaciones del *datasheet* para la selección de los valores de los capacitores.

Optimización del hardware para medición de radiación utilizando sensores de imagen CMOS

Especificación técnica

3.3.6 PCB completa

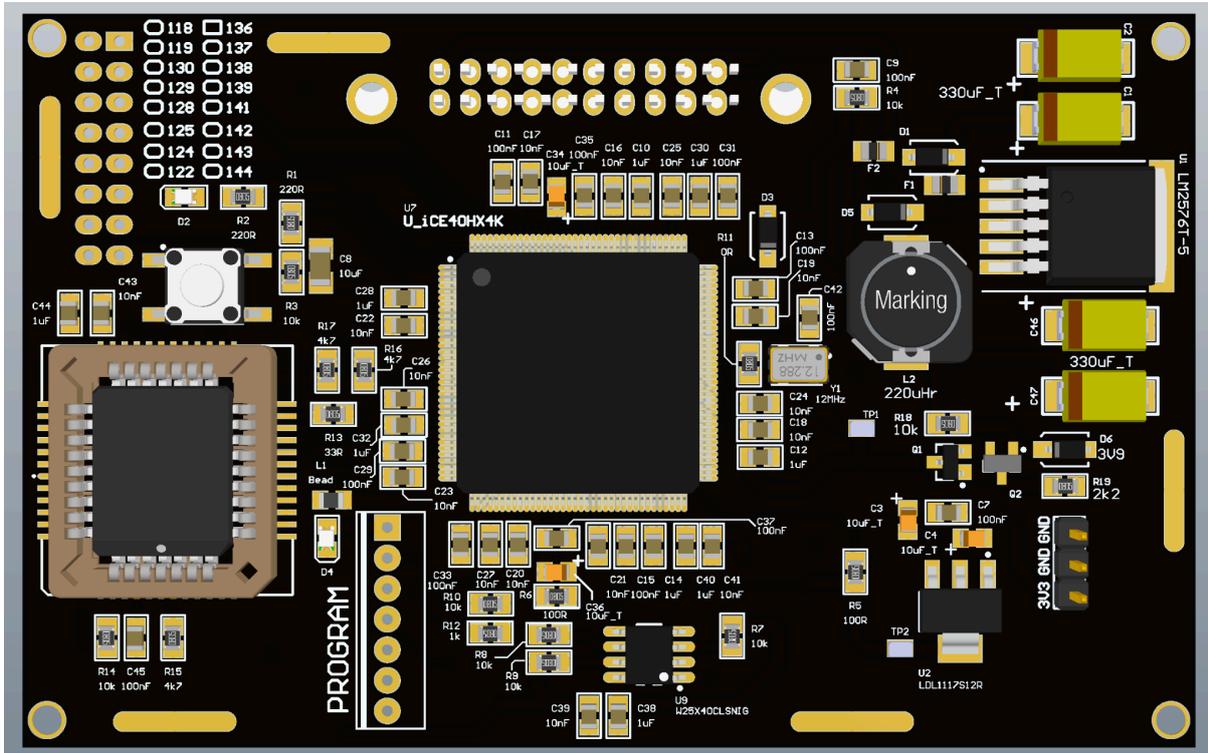


Figura 17. PCB completo frente

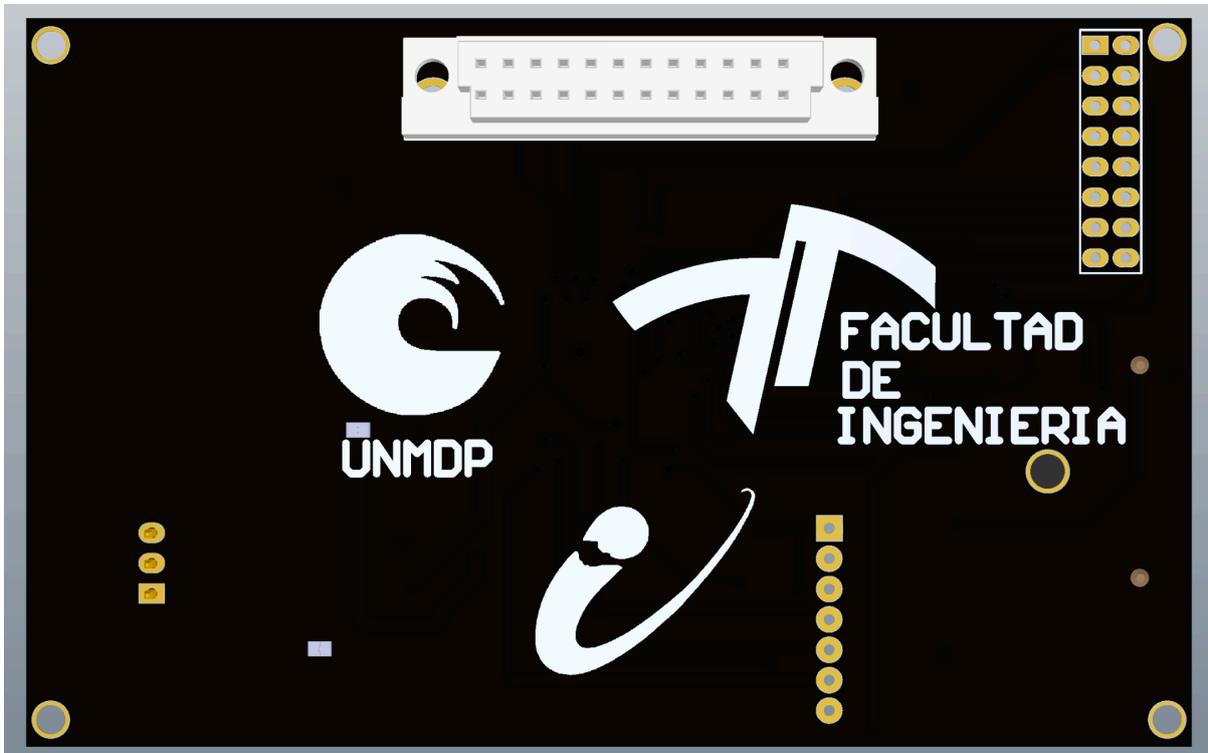


Figura 18. PCB completo contrafrente

El PCB completo se divide en 3 secciones a la derecha la alimentación, en el centro de arriba hacia abajo el conector, la FPGA y por debajo la memoria flash, a la izquierda se encuentra de

Optimización del hardware para medición de radiación utilizando sensores de imagen CMOS

Especificación técnica

arriba hacia abajo, una bornera de expansión junto con un botón y led, por debajo el sensor y a la derecha la bornera de programación

Se incorpora además bandas expuestas a tierra en todo el borde del PCB para soldar *shielding* en caso de ser necesario.

4. Procedimiento de construcción

Se diseñó un PCB propio a partir del trabajo anterior y aprovechando las experiencias documentadas en la plataforma EDU-CIAA-FPGA. El diseño fue realizado utilizando la herramienta Altium Designer.

El procedimiento de verificación de construcción se realiza en varias etapas:

1. Soldadura de la fuente de alimentación y verificación de tensiones de salida para las dos posibles tensiones de entrada (12V o 9V), incluyendo protecciones.
2. Soldadura del resto de los componentes y ejecución de pruebas básicas para verificar el correcto funcionamiento de la FPGA.
3. Carga de un proyecto de prueba para verificar que la FPGA pueda comunicarse correctamente con el sensor óptico.
4. Verificación completa mediante pruebas de campo con configuración y calibración del sistema.

5. Código

En esta sección se hablará sobre los distintos bloques que componen el sistema en detalle, posteriormente se dará una visión general sobre como funciona todo junto

5.1 Configurador

Para utilizar el sensor para el sistema primero se debe configurar en modo *snapshot*, para que tome una sola imagen si se le envía una señal de *trigger* para simplificar este proceso se diseñan dos máquinas de estado, una encargada de implementar el protocolo de comunicación y otra encargada de presentarle la información necesaria a al bloque de comunicación

5.1.1 Máquina de estados protocolo “Programador”

Puertos

Nombre del puerto	Dirección	Tipo	Descripción
clk	in	std_logic	Señal de clock que debe ser 5 veces menor que la señal de clock enviada a la cámara.
s_t	in	std_logic	Señal de start que comienza la máquina de estados.
r_w	in	std_logic	Señal que le indica a la máquina de estados que puede escribir el próximo byte luego de

Optimización del hardware para medición de radiación utilizando sensores de imagen CMOS

Especificación técnica

			estar en <i>idle_2</i> . Debemos haber cambiado los datos de entrada antes de accionar esta señal.
stop	in	std_logic	Señal que indica que comience el proceso de <i>stop</i> .
reset	in	std_logic	Señal que reinicia la máquina de estados.
data_in	in	std_logic_vector (7 downto 0)	Señal paralela del dato a enviar.
sda_o	inout	std_logic	Señal serie de salida de datos con un buffer tristate.
sca_o	out	std_logic	Señal de salida de <i>clk</i> para la salida de datos serie.
err	out	std_logic	Señal de error cuando no se tiene la señal de <i>ack</i> .
buisy	out	std_logic	Señal que indica que se está mandando un byte y que se deben mantener los datos.

Señales internas

Nombre	Tipo	Descripción
state	state_type	Estado de la máquina de estados.
count	integer range 0 to 7	Contador que indica cuántos bits se enviaron.
data	std_logic_vector (7 downto 0)	Buffer interno de datos.
sda	std_logic	Señal interna de datos serie.
sca	std_logic	Señal interna de <i>clock</i> serie.
ack_lect	std_logic	Señal interna para recibir el <i>ACK</i> .

Estados

Nombre	Descripción
idle	Estado inicial que espera la señal de <i>start</i> activo alto.
start	Estado que envía la primera parte de la señal de <i>start</i> en la línea de comunicación serie.

Optimización del hardware para medición de radiación utilizando sensores de imagen CMOS

Especificación técnica

start_2	Estado que envía la segunda parte de la señal de <i>start</i> en la línea de comunicación serie.
b_trans	Estado donde solo pulsamos la señal de <i>clock</i> .
b_write_1	Primer estado donde presentamos el bit a escribir.
b_write_2	Segundo estado donde presentamos el bit a escribir.
b_write_3	Tercer estado donde presentamos el bit a escribir.
ack_1	Primer estado donde pulsamos la señal de <i>clock</i> y esperamos recibir el <i>ACK</i> .
ack_2	Segundo estado donde pulsamos la señal de <i>clock</i> y esperamos recibir el <i>ACK</i> .
ack_3	Tercer estado donde pulsamos la señal de <i>clock</i> y esperamos recibir el <i>ACK</i> .
ack_fin	Cuarto estado donde pulsamos la señal de <i>clock</i> y esperamos recibir el <i>ACK</i> .
error	Estado donde no se recibe <i>ACK</i> .
idle_2	Estado intermedio que espera recibir la señal de <i>r_w</i> para escribir el siguiente byte o la señal de <i>stop</i> para enviar la señal de <i>stop</i> .
stop_1	Primer estado de envío de la señal <i>stop</i> .
stop_2	Último estado de envío de la señal <i>stop</i> .

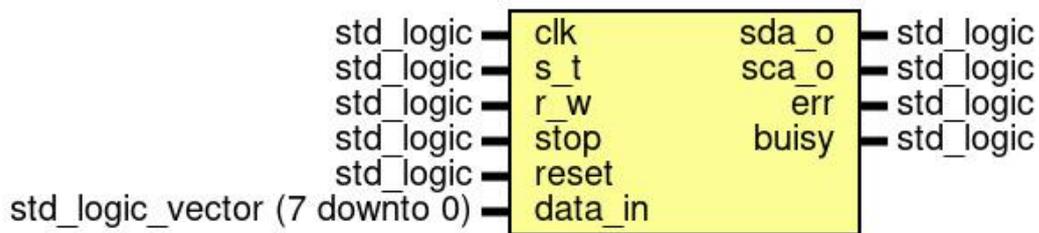


Figura 19. Máquina de estados programador

Optimización del hardware para medición de radiación utilizando sensores de imagen CMOS

Especificación técnica

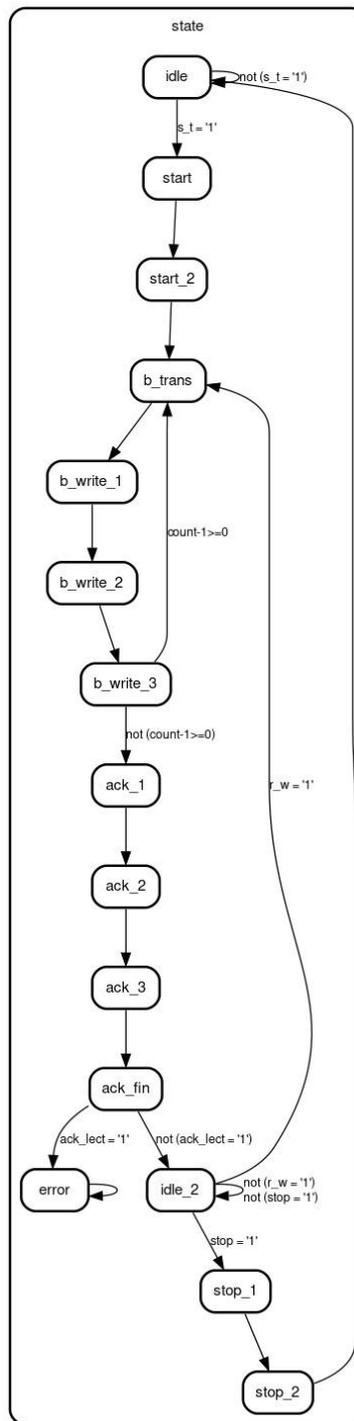


Figura 20. Diagrama de flujo máquina de estados "Programador"

5.1.2 Máquina de estados controlador del "Programador"

Esta máquina de estados controla la máquina de estados que envía los bits ("Programador"), juntos son el bloque de configuración del sensor

Optimización del hardware para medición de radiación utilizando sensores de imagen CMOS

Especificación técnica

Genéricos

Nombre del genérico	Tipo	Valor	Descripción
Ganancia	integer	4	Genérico que controla la ganancia del sensor

Puertos

Nombre del puerto	Dirección	Tipo	Descripción
clk	in	std_logic	Clock de entrada.
start	in	std_logic	Señal que comienza a enviar los bytes que están cargados.
reset	in	std_logic	Señal que reinicia la máquina de estados.
buisy	in	std_logic	Señal que le indica al controlador que la máquina de estados que envía los bytes está ocupada.
err	in	std_logic	Señal de error proveniente de la máquina de estados "programador" que reinicia la máquina de estados.
done_s	out	std_logic	Señal de control que indica que el sensor ya está programado.
clk_o	out	std_logic	Señal de reloj de salida para alimentar al programador.
r_s	out	std_logic	Señal de control del programador para reiniciarlo.
s_t	out	std_logic	Señal de control del programador que le indica que puede enviar la señal de <i>start</i> .
w	out	std_logic	Señal de control del programador que le indica que puede seguir enviando bytes.

Optimización del hardware para medición de radiación utilizando sensores de imagen CMOS

Especificación técnica

stop	out	std_logic	Señal de control del programador que le indica que debe enviar la señal de <i>stop</i> .
data	out	std_logic_vector (7 downto 0)	Señal de datos paralelos para la máquina de estados para enviar.

Señales internas

Nombre	Tipo	Descripción
state	state_type	Estado actual de la máquina de estados.
count	integer range 0 to 3	Contador para enviar los datos por más de un ciclo de reloj, ya que las máquinas trabajan con diferentes velocidades.
clk_int	std_logic	Señales de reloj internas para generar diferentes relojes.
clk_int_2	std_logic	Señales de reloj internas para generar diferentes relojes.

Estados

Nombre	Descripción
idle	Estado inicial que espera la señal de <i>start</i> .
dp_BA	Envía el byte xBA que le indica al sensor una escritura.
dp_07	Estado que presenta el byte x07 a la máquina de estados que envía los bytes por 4 ciclos de reloj.
dw_07	Estado que espera a que la máquina de estados que envía los bytes termine de enviar el dato x07 .
dp_00	Estado que presenta el byte x00 a la máquina de estados.
dw_00	Estado que espera a que la máquina de estados termine de enviar el dato x00 .
dp_02	Estado que presenta el byte x02 a la máquina de estados.
dw_02	Estado que espera a que la máquina de estados termine de enviar el dato x02 .
stop_1	Estado que envía la señal de <i>stop</i> .
dp_BA_2	<i>(sin descripción)</i>
dp_1E	<i>(sin descripción)</i>
dw_1E	<i>(sin descripción)</i>
dp_81	<i>(sin descripción)</i>

Optimización del hardware para medición de radiación utilizando sensores de imagen CMOS

Especificación técnica

dw_81	(sin descripción)
dp_00_2	(sin descripción)
dw_00_2	(sin descripción)
stop_2	(sin descripción)
dp_BA_3	(sin descripción)
dp_35	(sin descripción)
dw_35	(sin descripción)
dp_00_3	(sin descripción)
dw_00_3	(sin descripción)
dp_ganancia	(sin descripción)
dw_ganancia	(sin descripción)
stop_3	(sin descripción)
done	Estado final que indica que el sensor y a está configurado.

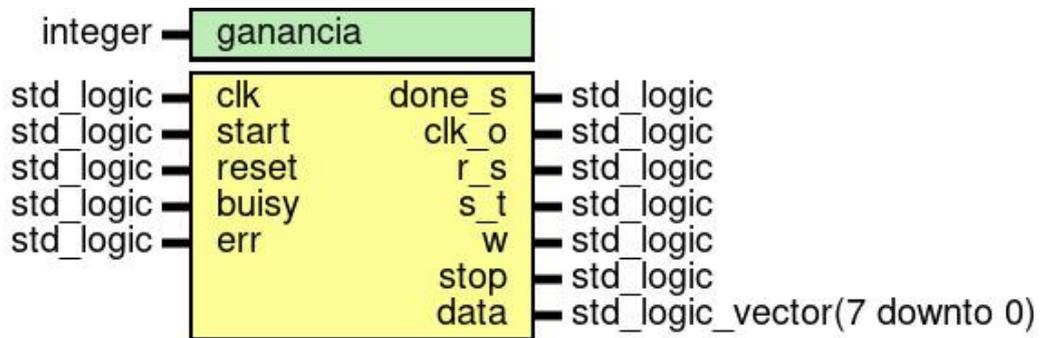
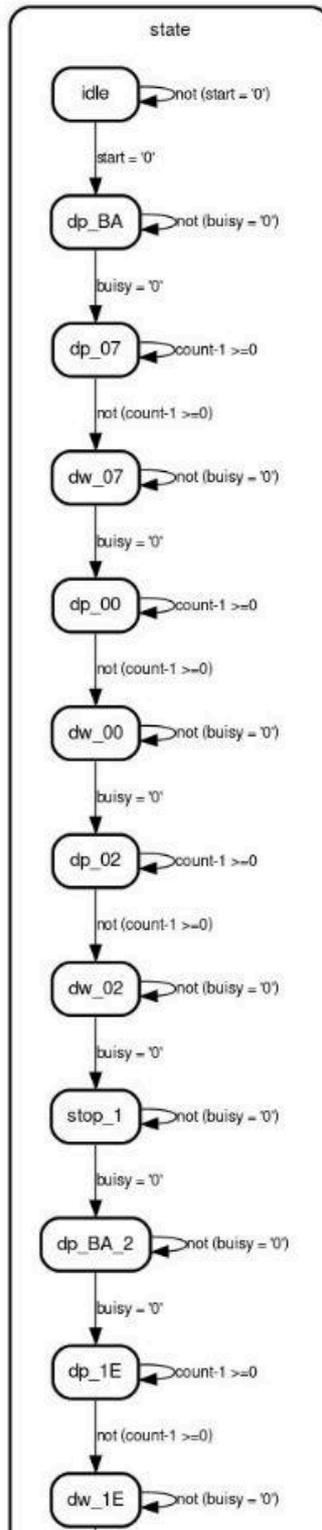


Figura 21. Maquina de estados

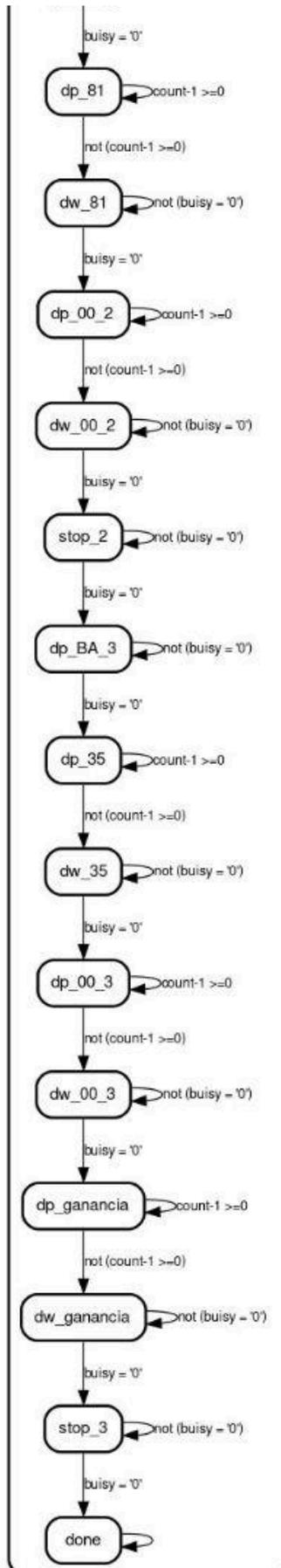
Optimización del hardware para medición de radiación utilizando sensores de imagen CMOS

Especificación técnica



Optimización del hardware para medición de radiación utilizando sensores de imagen CMOS

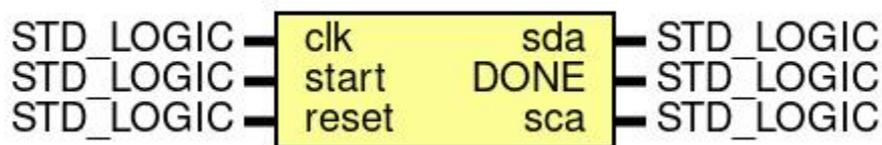
Especificación técnica



Optimización del hardware para medición de radiación utilizando sensores de imagen CMOS

Especificación técnica

Ambas máquinas de estado se juntan en un único bloque "Programador_controlador"

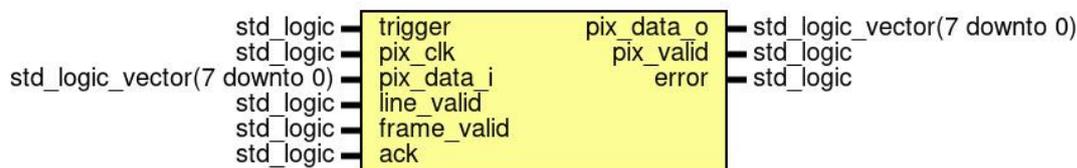


Se le proporciona una señal de reloj, una señal de start y reset y luego se conecta al sensor con SDA y SCA yendo a las conexiones de comunicación serie, cuando se termina de programar el sensor la señal DONE se activa en alto indicando una programación exitosa.

5.2 Capturador de píxeles

Luego de configurar el sensor se genera una máquina de estados que se encarga de recibir los datos del sensor. El sensor cuenta con 3 señales de control Line_valid(HS), Frame_valid(VS) y pixel_clock cuando HS y VS están en alto significa que estamos dentro de la zona activa del sensor y que en cada flanco negativo de pixel_clock se tiene un nuevo dato valido, la máquina de estado recibe estas señales y coloca en un buffer el dato recibido, levanta una bandera de dato válido y espera una señal de ACK para validar la correcta recepción del dato, si se recibe un nuevo dato y no se recibe un ACK del dato previo, se levanta una bandera de error que indica un dato perdido, el controlador luego reinicia todas las máquinas de estado y vuelve a comenzar el proceso.

Puertos



Nombre del puerto	Dirección	Tipo	Descripción
trigger	in	std_logic	Señal de <i>trigger</i> interno enviado por el controlador para limpiar el buffer y esperar una nueva captura.
pix_clk	in	std_logic	Señal de control enviada por el sensor; indica el momento de lectura del dato.
pix_data_i	in	std_logic_vector (7 downto 0)	Señal de los datos paralelos entrantes enviados por el sensor.
line_valid	in	std_logic	Señal de control enviada por el sensor; indica sincronización horizontal.

Optimización del hardware para medición de radiación utilizando sensores de imagen CMOS

Especificación técnica

frame_valid	in	std_logic	Señal de control enviada por el sensor; indica sincronización vertical.
ack	in	std_logic	Señal de control recibida de la máquina de estados que aplica el algoritmo; indica una correcta recepción del dato.
pix_data_o	out	std_logic_vector (7 downto 0)	Señal de salida del dato válido.
pix_valid	out	std_logic	Señal de control que indica que el dato es válido.
error	out	std_logic	Señal de control que indica que se ha perdido un dato.

Señales internas

Nombre	Tipo	Descripción
register_0	std_logic_vector(7 downto 0)	Buffer interno del dato.
leer	std_logic	Señal generada cuando <code>pix_valid</code> y <code>line_valid</code> son 1.
pix_valid_int	std_logic	Señal interna que marca que el pixel es válido.

codigo

```

library ieee;
    use ieee.std_logic_1164.all;
    use ieee.numeric_std.all;

entity captura_pixeles is
    port (
        --entradas
        trigger          : in std_logic;
        pix_clk          : in std_logic;
        pix_data_i       : in std_logic_vector(7 downto 0);
        line_valid       : in std_logic;
        frame_valid      : in std_logic;
        ack              : in std_logic;
        --salidas
        pix_data_o       : out std_logic_vector(7 downto 0);
    );
end entity;

```

Optimización del hardware para medición de radiación utilizando sensores de imagen CMOS

Especificación técnica

```
        pix_valid      : out std_logic;
        error          : out std_logic
    );
end entity;

architecture arch of captura_pixeles is
    signal register_0 : std_logic_vector(7 downto 0) := (others=>'0');
    signal leer, pix_valid_int : std_logic;
begin

    line_frame : process(all)
    begin
        if frame_valid='1' and line_valid='1' then
            leer <= '1';
        else
            leer <= '0';
        end if;
    end process;

    clk_lento:process(all)
    begin
        if trigger = '1' then
            register_0<=(others=>'0');
            error <= '0';
            pix_valid_int <= '0';
        elsif ack = '1' then
            pix_valid_int <= '0';
        elsif falling_edge(pix_clk) and pix_valid_int = '0' then
            if leer = '1' then --en el caso de que se haya leído el dato y
este dispuesto a recibir otro
                register_0<=pix_data_i;
                pix_valid_int <= '1';
                error <= '0';
            end if;
        elsif falling_edge(pix_clk) and pix_valid_int = '1' then --en el
caso de que no se haya leído el dato entonces hay que esperar
            error <= '1';
        end if;
    end process;

    pix_valid <= pix_valid_int;
    pix_data_o <= register_0;
```

Optimización del hardware para medición de radiación utilizando sensores de imagen CMOS

Especificación técnica

```
end architecture;
```

5.3 Algoritmo/Histograma

La máquina de estados que se encarga de implementar el algoritmo y el histograma es una maquina de estados mas grande del sistema que se encarga del manejo de memoria, la implementación del algoritmo y la generación del histograma

Genéricos

Nombre	Tipo	Valor	Descripción
umbral	integer	0	Umbral que determina qué brillo de píxeles es válido para considerarlo evento o no.
pixels	integer	7081	Número de píxeles activos que tiene el sensor (alto x ancho).
ancho	integer	97	Ancho horizontal de los píxeles activos del sensor.
num_bits	integer	13	Número de bits utilizados para el histograma; determina el ancho del bin como 2 elevado al número de bits / 32.

Puertos

Nombre del Puerto	Dirección	Tipo	Descripción
clk	in	std_logic	Señal de reloj global.
trigger	in	std_logic	Señal interna de trigger, controlada por el controlador.
reset	in	std_logic	Señal de reset global, controlada por el controlador.
pix_data	in	std_logic_vector(7 downto 0)	Datos de brillo del píxel proveniente del captador de píxeles.
pix_valid	in	std_logic	Señal que indica que un píxel es válido.
data_ram_i	in	std_logic_vector(10 downto 0)	Datos de la memoria de entrada de píxeles clasificados.
data_ram_o	out	std_logic_vector(10 downto 0)	Datos salientes de la memoria de entrada de píxeles clasificados.

Optimización del hardware para medición de radiación utilizando sensores de imagen CMOS

Especificación técnica

addr_ram	out	std_logic_vector(10 downto 0)	Dirección de memoria de los datos clasificados.
we	out	std_logic	Habilitación de escritura/lectura.
escritura_hist_signal	in	std_logic	Señal que habilita escritura del histograma en memoria.
errase_histograma	in	std_logic	Señal para borrar el histograma actual.
dir_escritura_hist	in	std_logic_vector(9 downto 0)	Dirección inicial para escritura de histograma.
data_ram_energia_i	in	std_logic_vector(13 downto 0)	Datos desde memoria de energía.
data_ram_energia_o	out	std_logic_vector(13 downto 0)	Datos hacia memoria de energía.
addr_ram_energia	out	std_logic_vector(10 downto 0)	Dirección de memoria de energía.
data_ram_cantidad_i	in	std_logic_vector(5 downto 0)	Datos desde memoria de cantidad de píxeles.
data_ram_cantidad_o	out	std_logic_vector(5 downto 0)	Datos hacia memoria de cantidad de píxeles.
data_ram_histograma_o	out	std_logic_vector(13 downto 0)	Datos hacia memoria de histogramas.
addr_ram_histograma_o	out	std_logic_vector(9 downto 0)	Dirección para escritura de bin del histograma.
selector_ram_histograma	out	std_logic	Selección de acceso a RAM del histograma.
we_histograma	out	std_logic	Señal para lectura/escritura de RAM del histograma.
fin_borrado	out	std_logic	Indica que las memorias fueron borradas y está lista la nueva captura.
fin_histograma	out	std_logic	Indica que se generó el histograma.

Optimización del hardware para medición de radiación utilizando sensores de imagen CMOS

Especificación técnica

fin_escritura	out	std_logic	Indica que el histograma fue escrito.
ack	out	std_logic	Confirmación de recepción del dato.

Señales internas

Nombre	Tipo	Descripción
state	state_type	Estado actual de la máquina de estados.
data_a_escribir	integer range 0 to 2047	Clasificación del píxel actual.
dir_mem, dir_mem_1/2/3	integer range 0 to 2047	Direcciones vecinas para clasificación.
pix_count_int	integer range 0 to 1_310_720	Cuenta total de píxeles recibidos.
reg_anterior	integer range 0 to 2047	Registro del dato anterior.
reg_ancho_1/2/3	integer range 0 to 2047	Registros de datos vecinos.
ignorar_ancho_1	boolean	Ignorar vecino ancho +1.
ignorar_anterior	boolean	Ignorar vecino anterior.
indice	integer range 0 to 2047	Número de evento actual.
pix_data_reg	integer range 0 to 2047	Brillo actual.
dir_energia	integer range -2 to 2047	Dirección de energía.
cantidad_temp	integer range 0 to 63	Valor temporal de cantidad.
energia_temp	integer range 0 to 16383	Valor temporal de energía.
dir_histograma_int	integer range 0 to 1023	Dirección actual del histograma.
indice_histograma	integer range 1 to 32	Número de bin.
histogram	hist_array	Arreglo que guarda el histograma.

Optimización del hardware para medición de radiación utilizando sensores de imagen CMOS

Especificación técnica

Constantes

Nombre	Tipo	Valor	Descripción
ancho_bin	integer	$(2^{**} \text{ num_bits})/32$	Ancho de bin del histograma.

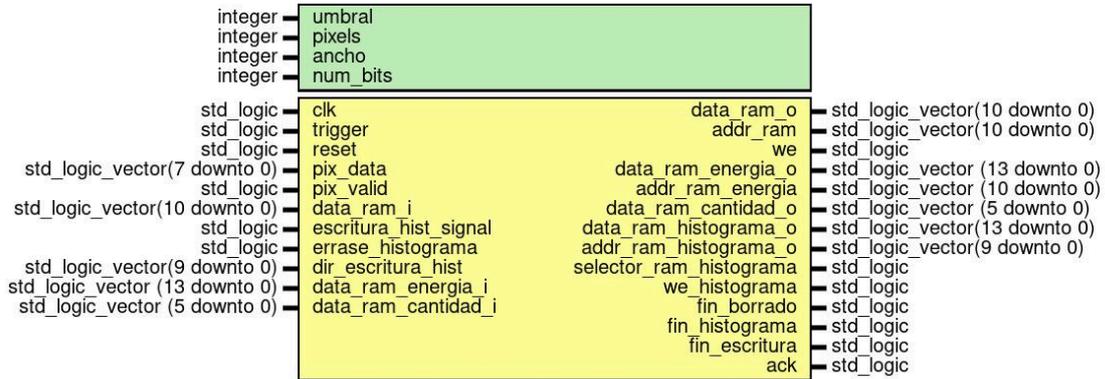
Descripción de los estados

Nombre	Descripción
erase	Borra memorias iniciales.
escritura_erase_1/2	Pasos intermedios del borrado.
nuevo_pix	Espera un nuevo píxel y verifica umbral.
dir_anterior / lectura_anterior	Dirección y lectura del dato anterior.
dir_ancho_1/2/3	Dirección de píxeles vecinos.
lectura_ancho_1/2/3	Lectura de vecinos.
casos	Clasificación de píxel.
dir_cantidad_energia	Dirección para leer cantidad/energía.
lectura_cantidad_energia	Lectura de cantidad/energía.
escritura_1/2	Escritura de memoria y actualización.
incremento_indice	Incrementa número de evento.
dir_memorias_energia_histograma	Prepara dirección para histograma.
lectura_memorias_histograma	Lee energía y cantidad.
escritura_histograma_1/2	Escribe datos del histograma.
incremento_dir_histograma	Cambia dirección de bin.
filtro	Filtro para píxeles muertos.

Optimización del hardware para medición de radiación utilizando sensores de imagen CMOS

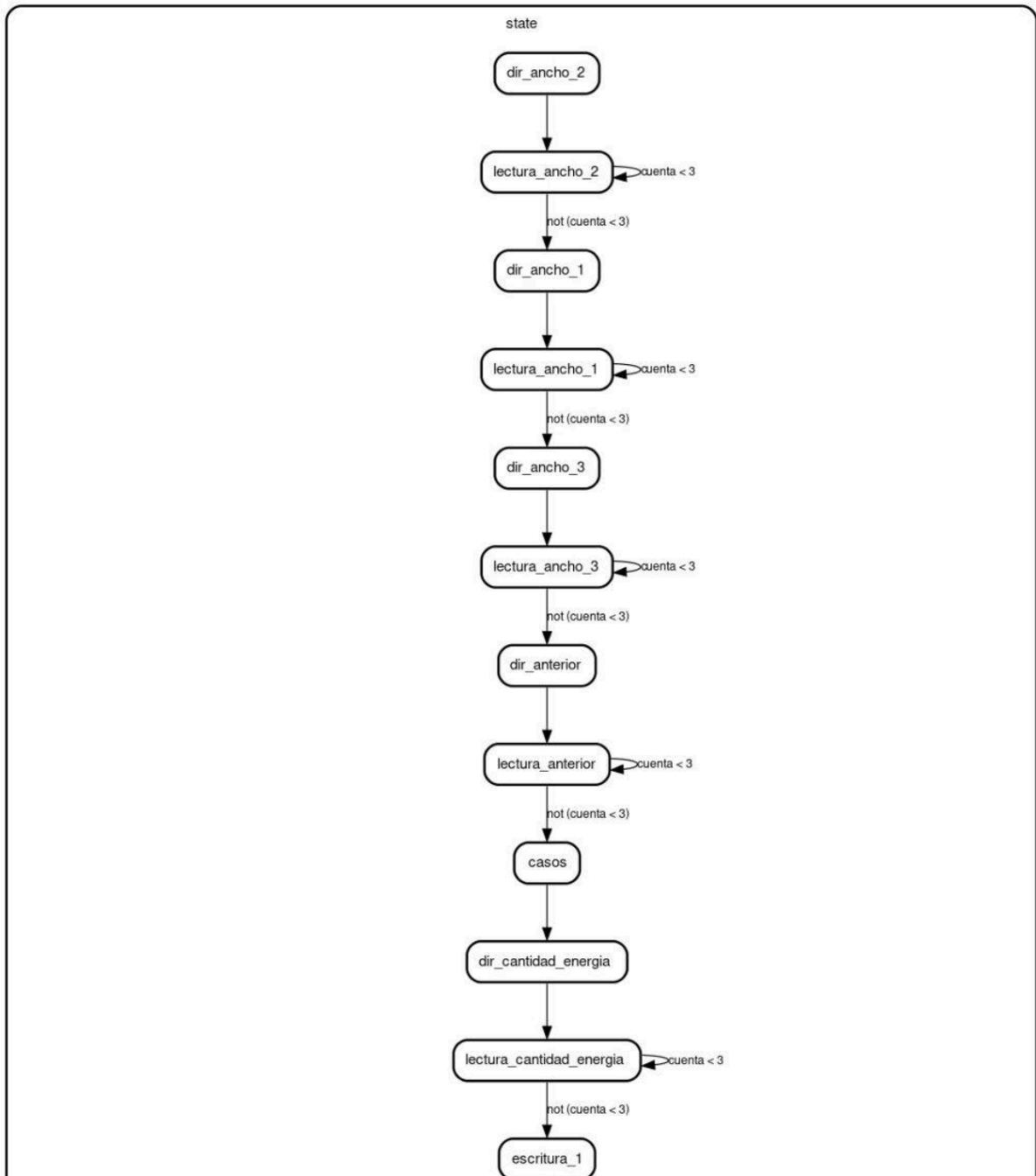
Especificación técnica

histograma_gen	Generación de histograma.
end_histograma	Fin de generación.
fin_escritura_histograma	Fin de escritura del histograma.



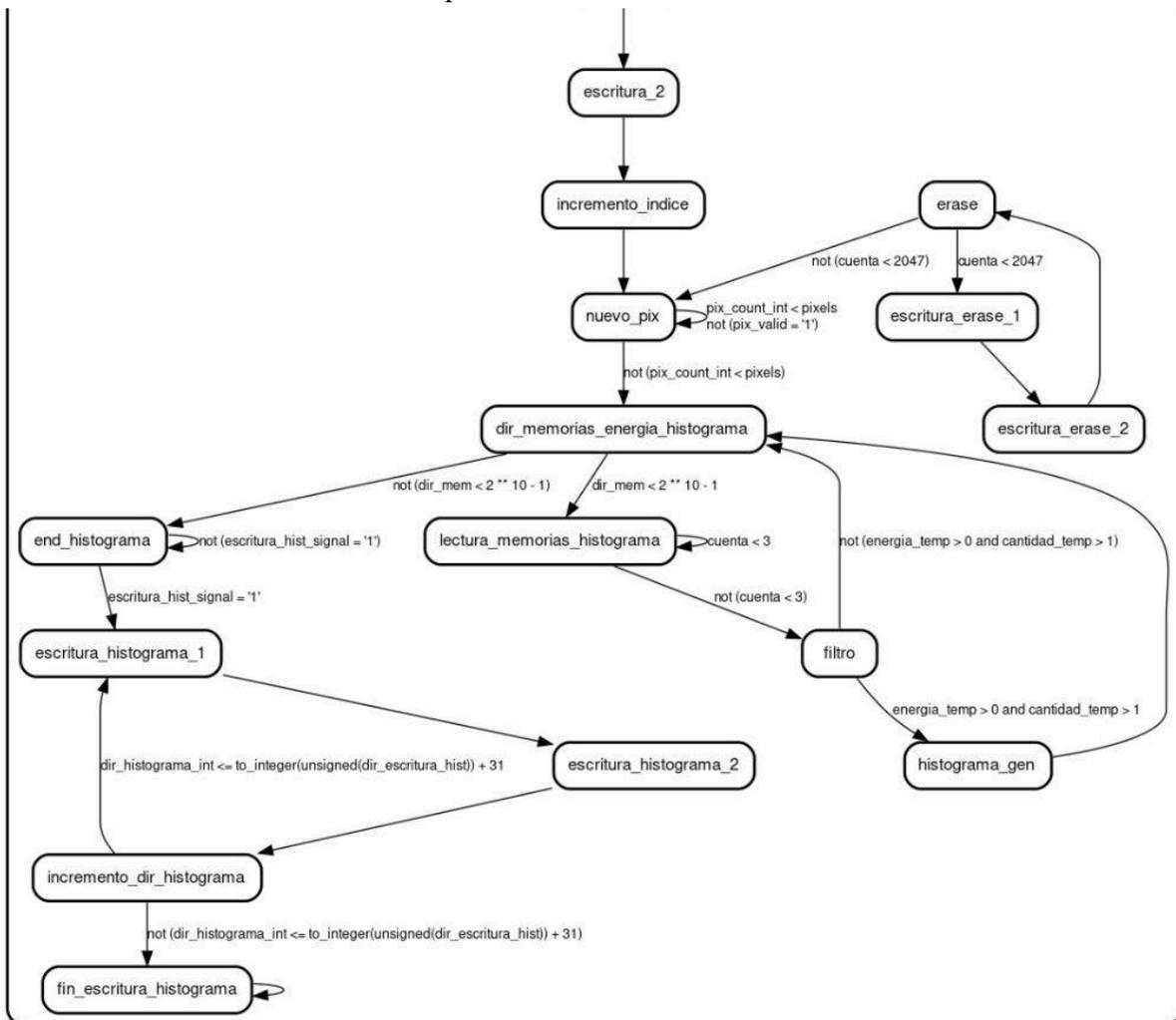
Optimización del hardware para medición de radiación utilizando sensores de imagen CMOS

Especificación técnica



Optimización del hardware para medición de radiación utilizando sensores de imagen CMOS

Especificación técnica



5.4 Memoria

La memoria es un bloque descriptivo que le indica al compilador que debe instanciar una memoria, el bloque está escrito en verilog por la simplicidad de lectura, el mismo bloque escrito en VHDL trae complejidades innecesarias al diseño

Puertos

Nombre	Dirección	Tipo	Descripción
addr	input	[addr_width-1:0]	Dirección de la memoria, ajustable con el parámetro <code>addr_width</code>
din	input	[data_width-1:0]	Datos de entrada de la memoria, ajustable con el parámetro <code>data_width</code>
write_en	input		Habilitación de lectura/escritura
clk	input		Reloj

Optimización del hardware para medición de radiación utilizando sensores de imagen CMOS

Especificación técnica

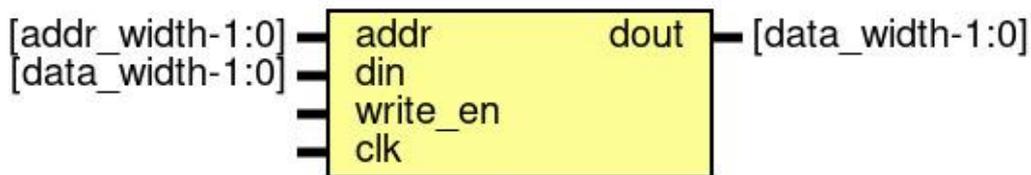
dout	output	[data_width-1:0]	Datos de salida de la memoria, ajustable con el parámetro <code>data_width</code>
------	--------	------------------	---

Señales internas

Nombre	Tipo	Descripción
dout	reg [data_width-1:0]	Registro de datos de salida
mem [(1<<addr_width)-1:0]	reg [data_width-1:0]	Instanciación de la BRAM

Genericos

Nombre	Tipo	Valor	Descripción
addr_width		9	Genéricos para ajustar el tamaño de la memoria
data_width		8	Genéricos para ajustar el tamaño de la memoria



Codigo

```

module ram (din, addr, write_en, clk, dout); // 512x8
    parameter                addr_width = 9;
    parameter                data_width = 8;
    input    [addr_width-1:0] addr;
    input    [data_width-1:0] din;
    input                write_en, clk;
    output    [data_width-1:0] dout;
    reg    [data_width-1:0] dout; // Register for output.
    reg [data_width-1:0] mem [(1<<addr_width)-1:0];

    always @(posedge clk)
    begin
        if (write_en)
            mem[addr] <= din;
        dout = mem[addr]; // Output register controlled by clock.
    end
endmodule

```

Optimización del hardware para medición de radiación utilizando sensores de imagen CMOS

Especificación técnica

5.5 Selector de acceso a memoria

El selector de acceso a la memoria no es más que un demultiplexor específico para las memorias BRAM ya que varias máquinas de estado necesitan acceso a estas memorias y los compiladores en general no aceptan que múltiples bloques manejen una memoria, por lo tanto es un bloque necesario desde el punto de vista de la compilación.

Puertos

Nombre	Dirección	Tipo	Descripción
selector	in	std_logic	Puerto de control que elige entre el puerto de entrada uno y el puerto dos; un '1' en esta entrada selecciona el puerto 2
data_1	in	std_logic_vector(data_width - 1 downto 0)	Entrada de datos puerto 1
addr_1	in	std_logic_vector(addr_width - 1 downto 0)	Entrada de dirección puerto 1
r_w_1	in	std_logic	Entrada de lectura/escritura puerto 1
addr_2	in	std_logic_vector(addr_width - 1 downto 0)	Entrada de dirección puerto 2
r_w_2	in	std_logic	Entrada de lectura/escritura puerto 2
data_o	out	std_logic_vector(data_width - 1 downto 0)	Salida de datos hacia la memoria
addr_o	out	std_logic_vector(addr_width - 1 downto 0)	Salida de dirección hacia la memoria
r_w_o	out	std_logic	Salida de lectura/escritura hacia la memoria

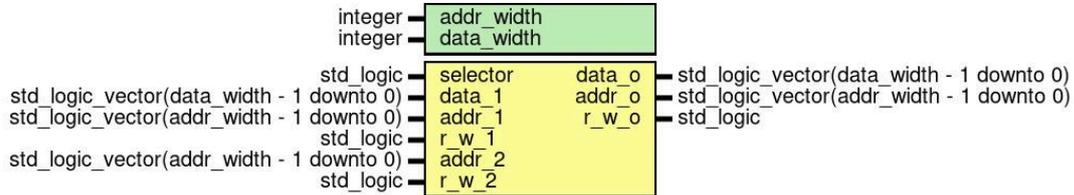
Genéricos

Nombre del genérico	Tipo	Valor	Descripción
addr_width	integer	10	Genérico que modifica el tamaño del bus de direcciones, debe coincidir con la memoria

Optimización del hardware para medición de radiación utilizando sensores de imagen CMOS

Especificación técnica

data_width	integer	16	Genérico que modifica el tamaño del bus de datos, debe coincidir con la memoria
------------	---------	----	---



Código

```

library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity access_ram is
  generic (
    addr_width : integer := 10;
    data_width  : integer := 16
  );
  port (
    selector : in std_logic;
    data_1   : in std_logic_vector(data_width - 1 downto 0);
    addr_1   : in std_logic_vector(addr_width - 1 downto 0);
    r_w_1    : in std_logic;
    addr_2   : in std_logic_vector(addr_width - 1 downto 0);
    r_w_2    : in std_logic;
    --salidas
    data_o   : out std_logic_vector(data_width - 1 downto 0);
    addr_o   : out std_logic_vector(addr_width - 1 downto 0);
    r_w_o    : out std_logic
  );
end entity access_ram;

architecture rtl of access_ram is
begin
  process (all)
  begin
    if selector = '1' then
      addr_o <= addr_2;
      data_o <= (others => '0');
      r_w_o  <= r_w_2;
    end if;
  end process;
end architecture rtl;

```

Optimización del hardware para medición de radiación utilizando sensores de imagen CMOS

Especificación técnica

```

else
    addr_o <= addr_1;
    data_o <= data_1;
    r_w_o <= r_w_1;
end if;
end process;
end architecture;

```

5.6 UART RX

Bloque encargado de recibir datos del satélite por puerto serie UART velocidad configurable mediante la señal generica `g_CLKS_PER_BIT`

Genéricos

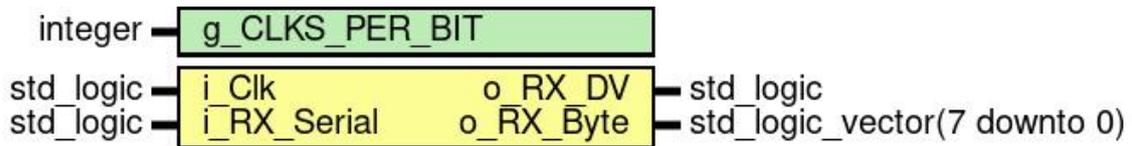
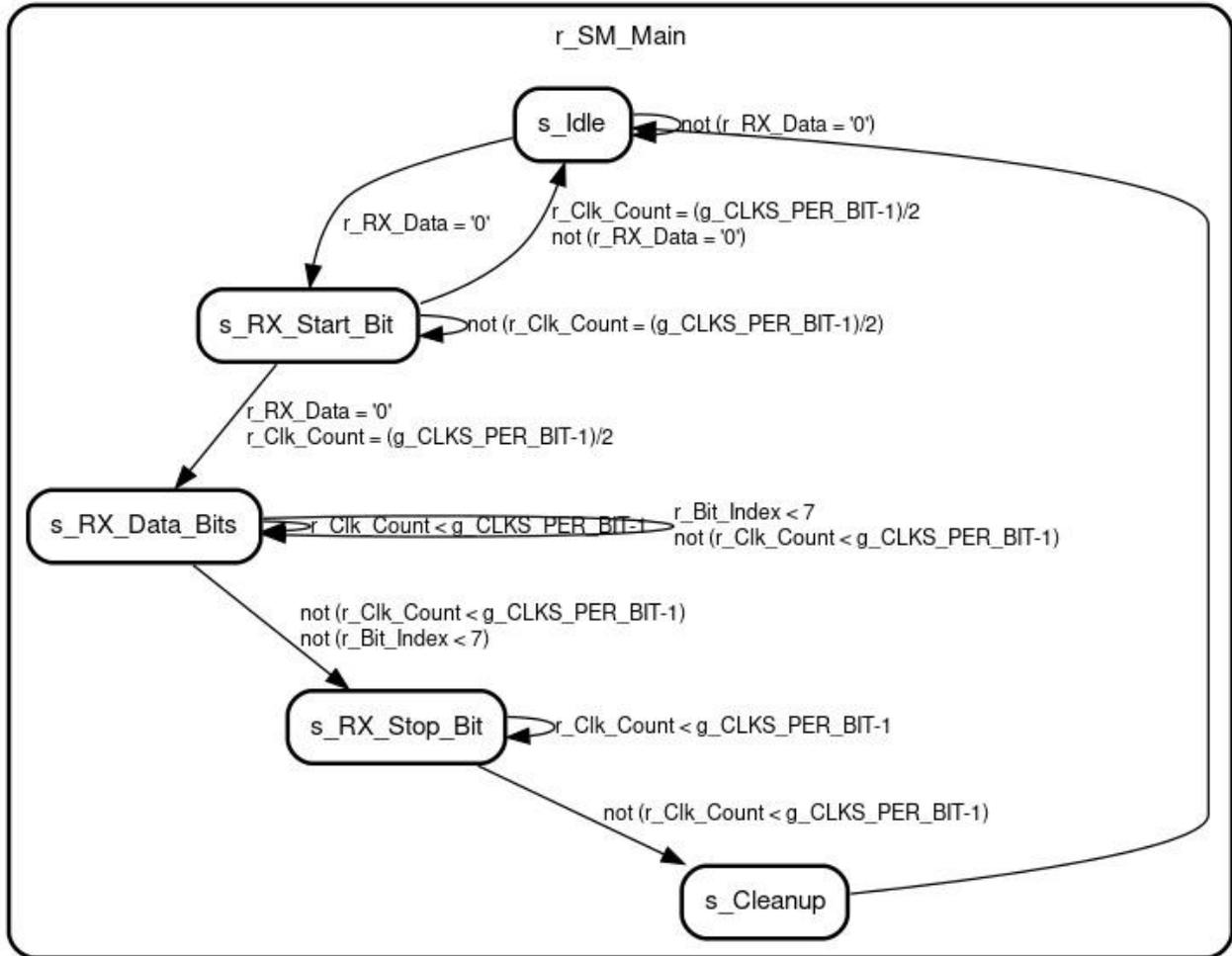
Nombre del genérico	Tipo	Valor	Descripción
<code>g_CLKS_PER_BIT</code>	integer	104	Genérico de configuración del baudrate: <code>g_CLKS_PER_BIT = (Frecuencia de i_Clk)/(Frecuencia UART)</code>

Puertos

Nombre	Dirección	Tipo	Descripción
<code>i_Clk</code>	in	std_logic	Señal de reloj entrante
<code>i_RX_Serial</code>	in	std_logic	Señal de dato de entrada
<code>o_RX_DV</code>	out	std_logic	Señal de control indicando dato nuevo
<code>o_RX_Byte</code>	out	std_logic_vector(7 downto 0)	Señal paralela del dato recibido

Optimización del hardware para medición de radiación utilizando sensores de imagen CMOS

Especificación técnica



5.7 UART TX

Bloque encargado de enviar datos por UART al satélite

Genéricos

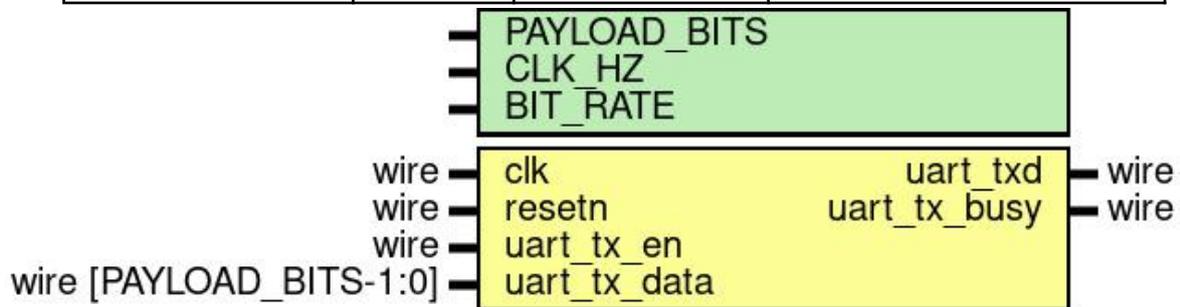
Nombre del genérico	Tipo	Valor	Descripción
PAYLOAD_BITS	–	8	Cantidad de bits por palabra
CLK_HZ	–	50_000_000	Frecuencia del reloj de entrada
BIT_RATE	–	115200	Baudrate

Optimización del hardware para medición de radiación utilizando sensores de imagen CMOS

Especificación técnica

Puertos

Nombre del puerto	Dirección	Tipo	Descripción
clk	input	wire	Reloj de entrada
resetrn	input	wire	Reset asincrónico manejado por el controlador
uart_txd	output	wire	Señal de salida de datos serie
uart_tx_busy	output	wire	Señal de control indicando que el bloque esta enviando un dato
uart_tx_en	input	wire	Habilitación para enviado de dato
uart_tx_data	input	wire [PAYLOAD_BITS-1:0]	Entrada de datos paralelos para enviar



Código

```

//
// Module: uart_tx
//
// Notes:
// - UART transmitter module.
//
module uart_tx #(
    parameter    PAYLOAD_BITS    = 8,
    parameter    CLK_HZ          = 50_000_000,

```

Optimización del hardware para medición de radiación utilizando sensores de imagen CMOS

Especificación técnica

```
parameter BIT_RATE      = 115200 // bits / sec
)
(
input  wire      clk      , // Top level system clock input.
input  wire      resetn   , // Asynchronous active low reset.
output wire      uart_txd , // UART transmit pin.
output wire      uart_tx_busy, // Module busy sending previous item.
input  wire      uart_tx_en , // Send the data on uart_tx_data
input  wire [PAYLOAD_BITS-1:0]  uart_tx_data // The data to be sent
);

//
-----
// External parameters.
//
//
// Input bit rate of the UART line.

localparam BIT_P      = 1_000_000_000 * 1/BIT_RATE; // nanoseconds

//
// Clock frequency in hertz.

localparam CLK_P      = 1_000_000_000 * 1/CLK_HZ; // nanoseconds

//
//
// Number of stop bits indicating the end of a packet.
parameter STOP_BITS   = 1;

//
-----
// Internal parameters.
//
//
// Number of clock cycles per uart bit.
localparam CYCLES_PER_BIT = BIT_P / CLK_P;

//
```

Optimización del hardware para medición de radiación utilizando sensores de imagen CMOS

Especificación técnica

```
// Size of the registers which store sample counts and bit durations.
localparam COUNT_REG_LEN = 1+$clog2(CYCLES_PER_BIT);

//
-----

// Internal registers.
//
//
// Internally latched value of the uart_txd line. Helps break long timing
// paths from the logic to the output pins.
reg txd_reg;

//
// Storage for the serial data to be sent.
reg [PAYLOAD_BITS-1:0] data_to_send;

//
// Counter for the number of cycles over a packet bit.
reg [COUNT_REG_LEN-1:0] cycle_counter;

//
// Counter for the number of sent bits of the packet.
reg [3:0] bit_counter;

//
// Current and next states of the internal FSM.
reg [2:0] fsm_state;
reg [2:0] n_fsm_state;

localparam FSM_IDLE = 0;
localparam FSM_START= 1;
localparam FSM_SEND = 2;
localparam FSM_STOP = 3;

//
-----

// FSM next state selection.
//
```

Optimización del hardware para medición de radiación utilizando sensores de imagen CMOS

Especificación técnica

```
assign uart_tx_busy = fsm_state != FSM_IDLE; //1 cuando la maquina de
estados esta en cualquier estado que no sea idle
assign uart_txd      = txd_reg;

wire next_bit      = cycle_counter == CYCLES_PER_BIT;
wire payload_done  = bit_counter   == PAYLOAD_BITS  ;
wire stop_done     = bit_counter   == STOP_BITS    && fsm_state == FSM_STOP;

//
// Handle picking the next state.
always @(*) begin : p_n_fsm_state
    case(fsm_state)
        FSM_IDLE : n_fsm_state = uart_tx_en    ? FSM_START: FSM_IDLE ;
        FSM_START: n_fsm_state = next_bit      ? FSM_SEND  : FSM_START;
        FSM_SEND  : n_fsm_state = payload_done ? FSM_STOP  : FSM_SEND ;
        FSM_STOP  : n_fsm_state = stop_done    ? FSM_IDLE  : FSM_STOP ;
        default   : n_fsm_state = FSM_IDLE;
    endcase
end

//
-----
// Internal register setting and re-setting.
//
//
// Handle updates to the sent data register.
integer i = 0;
always @(posedge clk) begin : p_data_to_send
    if(!resetn) begin
        data_to_send <= {PAYLOAD_BITS{1'b0}};
    end else if(fsm_state == FSM_IDLE && uart_tx_en) begin
        data_to_send <= uart_tx_data;
    end else if(fsm_state == FSM_SEND && next_bit ) begin
        for ( i = PAYLOAD_BITS-2; i >= 0; i = i - 1) begin
            data_to_send[i] <= data_to_send[i+1];
        end
    end
end
end

//
```

Optimización del hardware para medición de radiación utilizando sensores de imagen CMOS

Especificación técnica

```
// Increments the bit counter each time a new bit frame is sent.
always @(posedge clk) begin : p_bit_counter
    if(!resetn) begin
        bit_counter <= 4'b0;
    end else if(fsm_state != FSM_SEND && fsm_state != FSM_STOP) begin
        bit_counter <= {COUNT_REG_LEN{1'b0}};
    end else if(fsm_state == FSM_SEND && n_fsm_state == FSM_STOP) begin
        bit_counter <= {COUNT_REG_LEN{1'b0}};
    end else if(fsm_state == FSM_STOP&& next_bit) begin
        bit_counter <= bit_counter + 1'b1;
    end else if(fsm_state == FSM_SEND && next_bit) begin
        bit_counter <= bit_counter + 1'b1;
    end
end

//
// Increments the cycle counter when sending.
always @(posedge clk) begin : p_cycle_counter
    if(!resetn) begin
        cycle_counter <= {COUNT_REG_LEN{1'b0}};
    end else if(next_bit) begin
        cycle_counter <= {COUNT_REG_LEN{1'b0}};
    end else if(fsm_state == FSM_START ||
                fsm_state == FSM_SEND ||
                fsm_state == FSM_STOP ) begin
        cycle_counter <= cycle_counter + 1'b1;
    end
end

//
// Progresses the next FSM state.
always @(posedge clk) begin : p_fsm_state
    if(!resetn) begin
        fsm_state <= FSM_IDLE;
    end else begin
        fsm_state <= n_fsm_state;
    end
end
```

Optimización del hardware para medición de radiación utilizando sensores de imagen CMOS

Especificación técnica

```
//  
// Responsible for updating the internal value of the txd_reg.  
always @(posedge clk) begin : p_txd_reg  
    if(!resetn) begin  
        txd_reg <= 1'b1;  
    end else if(fsm_state == FSM_IDLE) begin  
        txd_reg <= 1'b1;  
    end else if(fsm_state == FSM_START) begin  
        txd_reg <= 1'b0;  
    end else if(fsm_state == FSM_SEND) begin  
        txd_reg <= data_to_send[0];  
    end else if(fsm_state == FSM_STOP) begin  
        txd_reg <= 1'b1;  
    end  
end  
  
endmodule
```

5.8 Controlador

Esta máquina de estados es la encargada de coordinar todas las máquinas de estado, activar la programación, enviar triggers, habilitar escrituras, enviar datos al satélite, recibir los comandos, etc.

Puertos

Nombre del puerto	Dirección	Tipo	Descripción
clk	in	std_logic	Reloj de entrada
reset	in	std_logic	Señal de reset general, enviado por elemento externo
fin_programador	in	std_logic	Señal de control proveniente del programador del sensor
fin_histograma	in	std_logic	Señal de control proveniente del algoritmo/histograma
fin_escritura_histograma	in	std_logic	Indica que se ha escrito la memoria que almacena los histogramas

Optimización del hardware para medición de radiación utilizando sensores de imagen CMOS

Especificación técnica

fin_borrado_algo	in	std_logic	Indica que las memorias fueron borradas y se puede realizar una nueva captura
datos_histograma	in	std_logic_vector(13 downto 0)	Datos de la memoria de histogramas a enviar por UART
uart_tx_busy_i	in	std_logic	Indica que se está enviando un byte por UART
trigger_externo	in	std_logic	Trigger externo que controla cuándo capturar
selector_ram_histograma	out	std_logic	Selección de acceso a la RAM de histograma
trigger_intern	out	std_logic	Trigger hacia máquinas internas
trigger_cam_o	out	std_logic	Trigger hacia el sensor para iniciar captura
start_program	out	std_logic	Comienza la programación del sensor
escritura_hist	out	std_logic	Indica que se debe escribir el histograma actual
errase_histograma	out	std_logic	Indica que se debe borrar el histograma actual
dir_mem_hist	out	std_logic_vector(9 downto 0)	Dirección de escritura en la RAM de histograma
reset_decod	out	std_logic	Reset de la máquina que decodifica comandos por UART
uart_tx_en_o	out	std_logic	Habilitación del envío por UART
uart_byte_o	out	std_logic_vector(7 downto 0)	Byte a enviar por UART
addr_histograma	out	std_logic_vector(9 downto 0)	Dirección de lectura del histograma
we_histograma	out	std_logic	Control de lectura de histograma

Optimización del hardware para medición de radiación utilizando sensores de imagen CMOS

Especificación técnica

trigger_wait_led	out	std_logic	Señal de espera de trigger (indicación visual)
------------------	-----	-----------	--

Señales internas

Nombre	Tipo	Descripción
state	state_type	Señal de estado actual
addr_histograma_int	integer range 0 to 1023	Dirección interna de histograma
img	integer range 0 to 60	Número de captura actual
dir_mem_hist_int	integer range 0 to 1023	Dirección de escritura en memoria interna
reg_histograma	std_logic_vector(13 downto 0)	Registro del dato leído de la memoria

Estados

Nombre	Descripción
reset_todo	Estado que reinicia todas las máquinas de estado
esp_borrado_1	Espera que el algoritmo/histograma borre las memorias RAM
fin_prog	Espera que el sensor esté configurado
trigger_algoritmo	Envía el trigger a las máquinas de estado internas
esp_borrado_2	Igual que esp_borrado_1, segunda instancia
trigger_cam	Envía el trigger al sensor
esp_fin_histograma	Espera que el histograma sea generado
enable_histograma_escritura	Habilita al algoritmo para escribir datos en memoria
esp_fin_escritura	Espera que el algoritmo termine de escribir
reset_histograma	Borra los contenidos de la máquina de histograma
trigger_wait	Espera señal de trigger externo para iniciar la captura
lectura_histograma	Lee los contenidos del histograma

Optimización del hardware para medición de radiación utilizando sensores de imagen CMOS

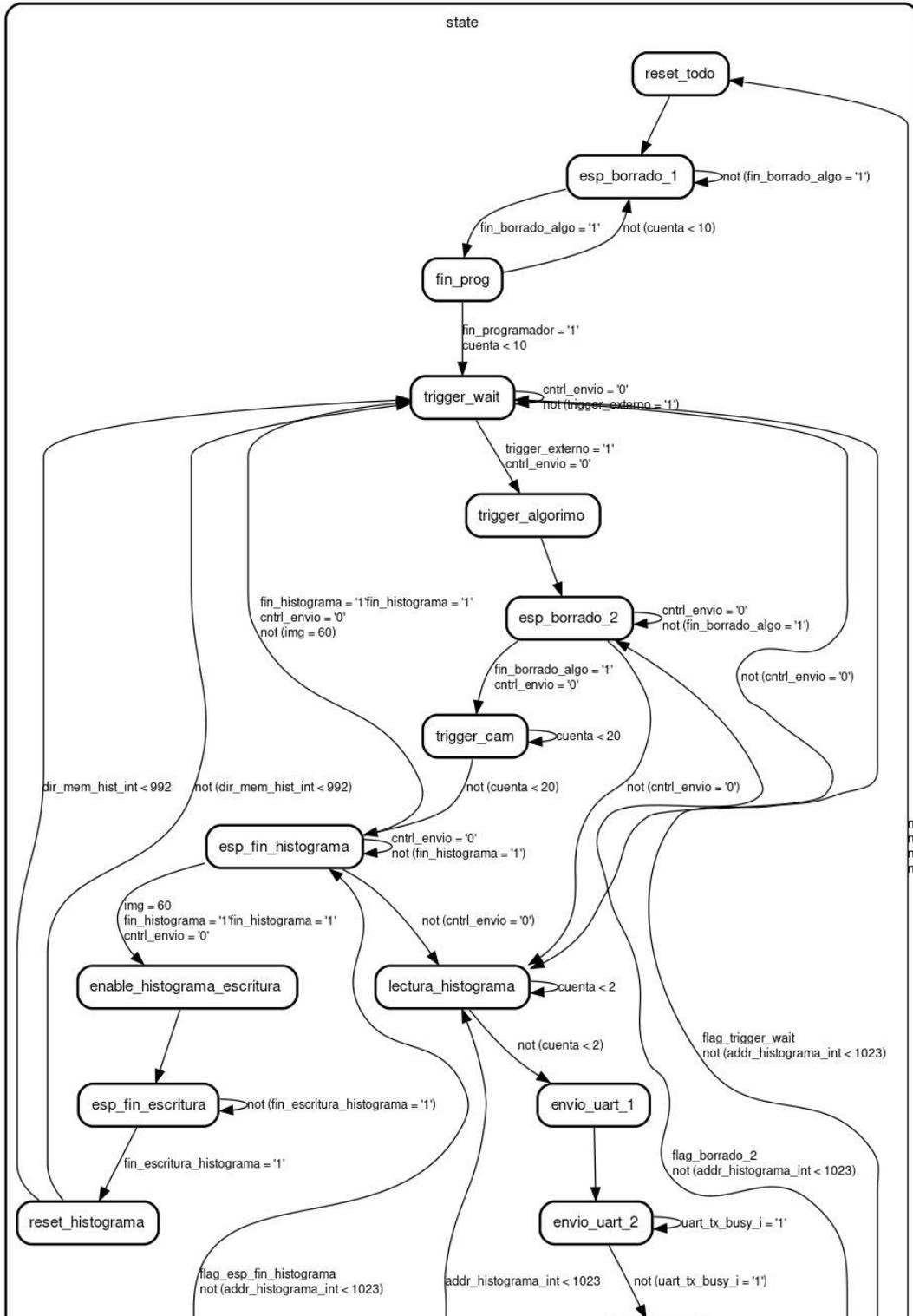
Especificación técnica

envio_uart_1	Envía la parte alta del histograma por UART
envio_uart_2	Igual que envio_uart_1
envio_uart_3	Envía la parte baja del histograma por UART
envio_uart_4	Igual que envio_uart_3
incremento_addr_histograma_envio	Incrementa dirección de memoria para el siguiente envío por UART
borrado_mem_histograma	Borra la memoria de histograma
escritura_borrado_hist_1	Primer paso del borrado del histograma
escritura_borrado_hist_2	Segundo paso del borrado del histograma

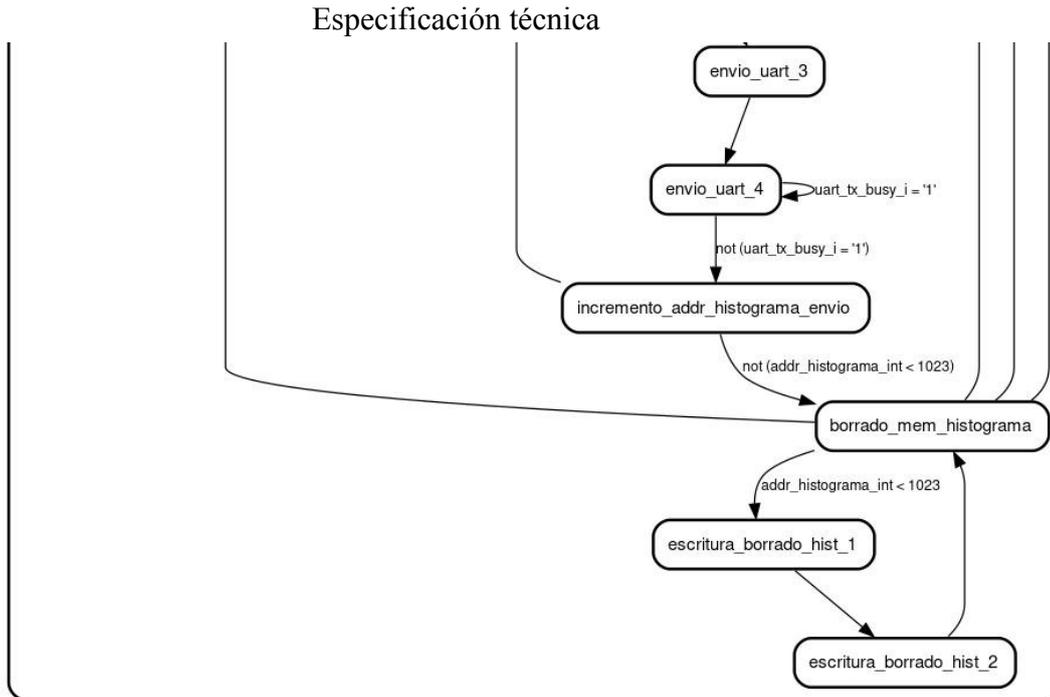


Optimización del hardware para medición de radiación utilizando sensores de imagen CMOS

Especificación técnica



Optimización del hardware para medición de radiación utilizando sensores de imagen CMOS



6. Simulaciones y pruebas

Se realizaron simulaciones en bloques simples del diseño donde era fácil verificar que el comportamiento era el esperado. La verificación del sistema completo se llevó a cabo directamente en la FPGA, utilizando la plataforma DE2-115.

Las pruebas se realizaron conectando la FPGA a una PC a través de UART. Desde la PC se enviaron datos de prueba conocidos, y se comparó la respuesta del sistema con la salida esperada para validar el correcto funcionamiento de los bloques. Para el procesamiento y generación de los datos de prueba, se utilizó MATLAB.

No se planificaron pruebas en condiciones de radiación real. La validación del sistema se realizó mediante la comparación de la salida del histograma contra patrones conocidos, asegurando así el correcto funcionamiento en condiciones de laboratorio.

Apéndice H

Plan de pruebas

UNMDP-FI

Carrera: Ingeniería Electrónica

Proyecto: ***Optimización del hardware para medición de radiación utilizando sensores de imagen CMOS***

Plan de pruebas

Versión 1.0

Domé Gaspar Carlos
gaspar.c.dome@gmail.com

Alcance

El siguiente plan de pruebas abarca desde las pruebas individuales de los módulos de hardware y software, hasta las pruebas integradas del sistema completo de adquisición de imágenes, con el objetivo de verificar su correcto funcionamiento en condiciones de laboratorio representativas del entorno espacial.

Ambientes de prueba

Los ambientes de prueba definidos fueron los siguientes:

- PC (simulación y pruebas de software).
- PC + FPGA DE2-115.
- PC + EDU-CIAA-FPGA.
- PC + DE2-115 + Sensor Aptina.
- PC + EDU-CIAA-FPGA + Sensor Aptina.

Recursos, herramientas e instrumentos

- Placa de desarrollo FPGA DE2-115.
- Placa EDU-CIAA-FPGA.
- Sensor CMOS Aptina MTM9001.
- PC.
- Osciloscopio.
- Lente.
- Entorno de desarrollo Lattice.
- Entorno de desarrollo Quartus.

Políticas de trabajo

Todas las pruebas que no requieran el uso de instrumental como el osciloscopio podrán realizarse tanto en el Laboratorio como en los hogares de los integrantes del proyecto, según disponibilidad, manteniendo la comunicación constante.

Las pruebas que requieran instrumental específico o condiciones de laboratorio serán realizadas exclusivamente en las instalaciones del Laboratorio, asegurando el correcto uso y cuidado de los equipos involucrados.

Estrategia de comunicación

Los resultados de las pruebas serán comunicados de forma regular a los directores del proyecto, junto con las recomendaciones y observaciones de los investigadores del Instituto Balseiro, con el fin de guiar y optimizar el desarrollo del sistema.

Módulo	ID	prueba	tipo de prueba	procedimiento	resultados esperados	instrumental	fecha	observaciones	Resultados obtenidos
Instalación del entorno de desarrollo	1	Validación del correcto funcionamiento del entorno	Unitaria	Se genera un proyecto de prueba simple y se carga a la FPGA	Se espera que un led parpadee	FPGA,PC	25/3/24	-	Acorde a lo esperado
Programador	2	Validación funcional	Unitaria	Se genera un banco de pruebas para el Módulo luego se compara con los resultados esperados	Se espera visualizar la señal de comunicación y que en el estado de reposo se encuentre en alta impedancia	PC	1/5/24	-	Acorde a lo esperado
Programador	2	Validación funcional en FPGA	Unitaria	Se cargan el programador y el controlador y se espera ver el sensor ya programado por parte de un led indicador	se espera ver el sensor ya programado por parte de un led indicador	PC,FPGA, Osciloscopio	29/5/24	Controlador no entrega clk a programador, diferencia de velocidad de clk entre las máquinas de estado generan error,programador avanza a pesar de no recibir ACK	No cumple con las expectativas
Programador	2	Validación funcional en FPGA	Unitaria	Se cargan el programador y el controlador y se espera ver el sensor ya programado por parte de un led indicador	se espera ver el sensor ya programado por parte de un led indicador	PC,FPGA, Osciloscopio	30/5/24	Sensor no envía ACK	No cumple con las expectativas
Programador	2	Validación funcional en FPGA	Unitaria	Se cargan el programador y el controlador y se espera ver el sensor ya programado por parte de un led indicador	Se espera ver en el osciloscopio que el sensor envía un ACK	PC,FPGA, Osciloscopio	5/6/24	El sensor envía ACK y se encuentra programado, cumpliendo la escritura del sensor	Acorde a lo esperado
Capturador	3	Validación funcional en FPGA y PC	Unitaria	Se carga en la FPGA el banco de prueba del capturador de pixeles, se programa el sensor y se envía la señal de captura, se capturan todos los pixeles, luego se envía al PC para ser procesado con MATLAB	Se espera ver una imagen en blanco y negro	PC,DE2-115,Sensor	4/9/24	La imagen es recibida pero de manera desordenada	No cumple con las expectativas
Capturador	3	Validación funcional en FPGA y PC	Unitaria	Se carga en la FPGA el banco de prueba del capturador de pixeles, se programa el sensor y se envía la señal de captura, se capturan todos los pixeles, luego se envía al PC para ser procesado con MATLAB	Se espera ver una imagen en blanco y negro	PC,DE2-115,Sensor,conversor UART	6/9/24	Imagen recibida ordenada	Acorde a lo esperado
Prueba de BRAM	4	Validación funcional de las memorias internas de la FPGA	Unitaria	Se carga el banco de pruebas a la FPGA y se descarga a la PC una secuencia de números cargada en la memoria BRAM a diferentes velocidades y con distinta cantidad de ciclos de reloj para probar las capacidades de la memoria	Se espera recibir una secuencia predeterminada de números descargada de la memoria sin que falte ninguno y de forma ordenada	PC, FPGA, Conversor UART	15/9/24	-	Acorde a lo esperado
Pruebas del funcionamiento del algoritmo	5	Validación de la máquina de estados encargada de realizar el algoritmo	Unitaria	Se carga el banco de pruebas del algoritmo a la FPGA luego se envía por uart los valores de los pixeles de una imagen de prueba conocida y se recibe por uart el valor del índice asignado, luego se reconstruye en la PC la imagen clasificada por la FPGA	Se espera que la imagen clasificada por la FPGA y la clasificada por PC coincidan exactamente igual	PC,FPGA,Conversor UART	24/9/24	La clasificación presentada por la FPGA es errónea, parece que no se encuentra el dato asignado al pixel previo al que se está analizando, esto se debe a que el índice no se está manejando de forma correcta	No cumple con las expectativas
Pruebas del funcionamiento del algoritmo	5	Validación de la máquina de estados encargada de realizar el algoritmo	Unitaria	Se carga el banco de pruebas del algoritmo a la FPGA, luego se envía por uart los valores de los pixeles de una imagen de prueba conocida y se recibe por uart el valor del índice asignado, luego se reconstruye en la PC la imagen clasificada por la FPGA	Se espera que la imagen clasificada por la FPGA y la clasificada por PC coincidan exactamente igual	PC,DE2-115,Conversor UART	4/10/24	La clasificación se realiza correctamente, no se valida el funcionamiento en la EDU-CIAA-FPGA	No cumple con las expectativas
Pruebas del funcionamiento del algoritmo	5	Validación de la máquina de estados encargada de realizar el algoritmo	Unitaria	Se carga el banco de pruebas del algoritmo a la FPGA, luego se envía por uart los valores de los pixeles de una imagen de prueba conocida y se recibe por uart el valor del índice asignado, luego se reconstruye en la PC la imagen clasificada por la FPGA	Se espera que la imagen clasificada por la FPGA y la clasificada por PC coincidan exactamente igual	PC,FPGA,Conversor UART	5/10/24	Luego de agregar cambios mínimos la prueba es satisfactoria con la EDU-CIAA-FPGA	Acorde a lo esperado

Pruebas de funcionamiento del histograma	6	Validación de la máquina de estados encargada de realizar el histograma	Unitaria	Se carga el banco de pruebas del histograma se le envían datos de prueba por UART y se recibe el histograma por UART en el PC luego se compara con el teórico	Se espera que el histograma generado por la FPGA y por el PC sean idénticos	PC,FPGA,Conversor UART	2/11/24	-	Acorde a lo esperado
Pruebas en el banco de prueba completo	7	Validar el funcionamiento y coordinación de todas las máquinas de estado	Integral	Se carga en la FPGA el banco de prueba simplificado y se espera recibir en el PC la imagen sin procesar y el histograma generado. Luego en el PC con la imagen se genera el histograma teórico con las limitaciones del sistema en mente, luego se comparan	Se esperan que ambos histogramas tengan menos de 3 cuentas de diferencia entre bins	PC,FPGA,DE2-115,Sensor,Conversor UART	21/11/24	No se logra una coordinación entre ambas FPGA y las máquinas de estado	No cumple con las expectativas
Pruebas en el banco de prueba completo	7	Validar el funcionamiento y coordinación de todas las máquinas de estado	Integral	Se carga en la FPGA el banco de prueba simplificado y se espera recibir en el PC la imagen sin procesar y el histograma generado. Luego en el PC con la imagen se genera el histograma teórico con las limitaciones del sistema en mente, luego se comparan	Se esperan que ambos histogramas tengan menos de 3 cuentas de diferencia entre bins	PC,FPGA,DE2-115,Sensor,Conversor UART	8/1/25	Se logra generar el histograma en la DE2-115 y compararlo con el PC de forma satisfactoria	No cumple con las expectativas
Pruebas en el banco de prueba completo	7	Validar el funcionamiento y coordinación de todas las máquinas de estado	Integral	Se carga en la FPGA el banco de prueba simplificado y se espera recibir en el PC la imagen sin procesar y el histograma generado. Luego en el PC con la imagen se genera el histograma teórico con las limitaciones del sistema en mente, luego se comparan	Se esperan que ambos histogramas tengan menos de 3 cuentas de diferencia entre bins	PC,FPGA,DE2-115,Sensor,Conversor UART	11/1/25	Se logra generar el histograma en la DE2-115 y compararlo con el PC de forma satisfactoria además ahora se genera con la EDU-CIMA-FPGA Confirmando el correcto funcionamiento del sistema	Acorde a lo esperado
Cálculos y simulación sobre el consumo de la placa de desarrollo sensor	8	Validar el consumo del sistema completo por simulación	Integral	Se generan estimaciones de consumo a partir del software del fabricante del sensor y la FPGA luego se modela como una resistencia, se generan los circuitos del regulador y se simula el consumo	Se espera que el consumo general del sistema sea menor a 2W	PC	25/2/25	-	Acorde a lo esperado