

Tipo de artículo: Artículo original

Implementación hardware-software de un sistema visualizador de la información de señales de radar

Hardware-software implementation of a radar signal information display system

Lisvan Guevara Trujillo^{1*} , <https://orcid.org/0000-0002-1830-2045>

Ariel Hernández Reyes¹ , <https://orcid.org/0009-0009-1133-3779>

Marcelino Sánchez Posada¹ , <https://orcid.org/0000-0001-8906-1149>

Wenny Hojas-Mazo² , <https://orcid.org/0000-0002-8298-3439>

¹ Centro de Investigación y Desarrollo Naval. Calle Estrada Palma No. 13, Casa Blanca, Regla, La Habana, Cuba.

² Departamento de Inteligencia Artificial e Infraestructura de Sistemas Informáticos, Facultad de Informática, Universidad Tecnológica de La Habana “José Antonio Echeverría”. Calle 114 No. 11901, Marianao, La Habana, Cuba.

* Autor para correspondencia: lisvan94trujillo@gmail.com

Resumen

Conceptualmente, un sistema de radar consta de cinco componentes: un generador, un receptor, un amplificador, un procesador y un visualizador. En la actualidad existen radares de aproximación de precisión basados en tecnología analógica, lo cual genera inestabilidad durante su funcionamiento. En este artículo se presenta una implementación combinada hardware-software de un sistema visualizador digital para un radar de aproximación de precisión. En la implementación se emplean los siguientes dispositivos disponibles comercialmente: un kit FPGA y un ordenador de escritorio. El kit FPGA posibilita la recepción y conversión de la señal analógica a digital, así como implementar algoritmos de procesamiento de los datos del radar. La representación de la información se realizó en el ordenador de escritorio. La comunicación entre estos dispositivos se efectuó por la interfaz de alta velocidad PCI Express. La solución procesa digitalmente los datos y disminuye los mantenimientos técnicos causados por el largo período de explotación y la obsolescencia de las piezas. Se logra una sustitución tecnológica de los componentes de procesamiento y visualización analógicos, que cumple con el tiempo de actualización de la información impuesta por el radar de aproximación de precisión. Se obtiene una solución modular, genérica y flexible.

Palabras clave: procesador de señales de radar; visualizador de radar; radar de aproximación de precisión.

Abstract

Conceptually, a radar system consists of five components: a generator, a receiver, an amplifier, a processor and a display. Currently there are precision approach radars based on analog technology, which generates instability during operation. This article presents a combined hardware-software implementation of a digital display system for a precision approach radar. The following commercially available devices are used in the implementation: an FPGA kit and a desktop computer. The FPGA kit enables the reception and conversion of the analog signal to digital, as well as implement radar data processing algorithms. The representation of the information was performed on the desktop computer. Communication between these devices was carried out through the high-speed PCI Express interface. The solution digitally processes the data and reduces technical maintenance caused by the long period of operation and the obsolescence of the parts. A technological replacement of the analog processing and display components is achieved, which complies with the information update time imposed by the precision approach radar. A modular, generic and flexible solution is obtained.



Esta obra está bajo una licencia *Creative Commons* de tipo **Atribución 4.0 Internacional**
(CC BY 4.0)

Keywords: *radar signal processor; radar display; precision approach radar.*

Recibido: 10/09/2023

Aceptado: 28/10/2023

En línea: 01/11/2023

Introducción

Conceptualmente, un sistema de radar consta de cinco componentes: un generador, un receptor, un amplificador, un procesador y un visualizador (Acosta *et al.*, 2006; Acosta *et al.*, 2007). El procesador permite obtener de las señales recibidas la información deseada. El visualizador es un dispositivo electrónico que se utiliza para representar en un formato adecuado la información obtenida del procesador (Kavyashree *et al.*, 2017). Los visualizadores de radar deben presentar al observador una imagen gráfica continua, que permita comprender de forma fácil la posición relativa de los objetos detectados, lo que contribuye a la eficiencia del proceso de toma de decisión por los operadores (Kaushik, 2014; Sulistyaningsih *et al.*, 2019). Los componentes empleados para la transmisión y recepción en radiofrecuencia se ofrecen habitualmente como dispositivos COTS (*commercial off the shelf*), mientras que el procesamiento de señal radar, es el encargado de brindar valor agregado (Gómez, 2017).

Actualmente, Cuba cuenta con radares de aproximación de precisión que datan de la década del 50, construidos con tecnología analógica. En estos dispositivos son deficitarios los transformadores, rectificadores del circuito de alto voltaje y las bobinas de los circuitos que forman los barridos entrelazados (Trujillo and Méndez, 2022). Además son altos consumidores de energía eléctrica, poseen gran volumen y peso, sus piezas están obsoletas y existe incapacidad de adquirir los repuestos en el mercado internacional (Trujillo *et al.*, 2022). También, están sometidos a largos períodos de mantenimiento provocado fundamentalmente por el envejecimiento de sus partes y piezas. La inestabilidad en el funcionamiento y los altos costos que provocaría la modernización del sistema de procesamiento y visualización analógico en países extranjeros, determinan la necesidad de obtener un sistema procesador y visualizador que cumpla con las exigencias operativas del radar de aproximación de precisión.

En la actualidad, gracias a los avances en la tecnología digital, es posible procesar y presentar la información de señales de radar con el empleo de medios de cómputo (Jevtić and Stamatović, 2009; Stamatovic *et al.*, 2012; Stamatović *et al.*, 2013), siendo esta una de las vías para la solución de los problemas planteados. Para la implementación de un sistema de procesamiento digital de señal de radar se suelen emplear dispositivos electrónicos programables como microcontroladores, DSPs (*digital signal processing*), FPGAs (*field-programmable gate array*) y CPLDs (*Complex Programmable Logic Device*) (Gómez, 2017). Varios autores han abordado la implementación de



sistemas de procesamiento y visualizadores digitales cumpliendo con las exigencias de tiempo y procesamiento que impone el radar, empleándose diversas plataformas hardware y software (Jevtić and Stamatović, 2009; Stamatovic *et al.*, 2012; Stamatović *et al.*, 2013).

Los autores de (Acosta *et al.*, 2006; Acosta *et al.*, 2007) diseñaron e implementaron una modernización de los componentes procesador y visualizador del radar, capaz de realizar el proceso de adquisición, interpretación y presentación en tiempo real. El sistema permite implementar algoritmos de procesamiento de información y reconocimiento de objetivos. Para realizar el procesamiento de la señal del radar se auxiliaron de dispositivos FPGA. En (Jevtić and Stamatović, 2009) se presenta un sistema de adquisición, procesamiento y visualización de datos de radar. Los procedimientos de procesamiento y visualización de los datos de radar se implementaron como módulos software que se ejecutan en una PC (*personal computer*) disponible comercialmente. Los autores fundamentan el empleo de una PC, por la disminución de los costos de desarrollo, mantenimiento del sistema y por el aumento de la flexibilidad, pues al implementarse los algoritmos como componentes software se pueden modificar fácilmente. En este trabajo se plantea que el procesamiento y la visualización de datos de radar son operaciones computacionalmente muy exigentes. Esto conllevó aprovechar las capacidades de procesamiento del CPU (*central processing unit*) para lograr un rendimiento en tiempo real.

Por su parte en (Stamatovic *et al.*, 2012; Stamatović *et al.*, 2013) se desarrolló un visualizador digital de la situación aérea para remplazar uno antiguo basado en tubo de rayos catódicos. Por razones de rendimiento la aplicación fue escrita en C++ y la representación de la situación aérea se realizó en un hilo independiente. Para simplificar el desarrollo, se utilizó el *framework* Qt.

A partir del análisis de estos trabajos se apreció el uso de los kit FPGA, ya que en el desarrollo de las nuevas generaciones de radar, las plataformas de procesamiento de señales son ampliamente utilizadas (López Estrada, 2009; Gómez, 2017). Entre las ventajas que posee la recepción de la señal digital está la posibilidad del procesamiento automático de la información, además de aumentar la cantidad de información a representar en pantalla, facilitando la comprensión y toma de decisiones a los operadores (Trujillo *et al.*, 2020; Trujillo *et al.*, 2022).

El objetivo de este trabajo es implementar un sistema de procesamiento y visualización digital que cumpla con las exigencias operativas del radar de aproximación de precisión.

Materiales y métodos

Para la solución de los problemas presentes en el radar analógico, el sistema debe cumplir con varios requisitos. Uno de los requerimientos más importantes del sistema es realizar la adquisición, procesamiento y presentación de la



información en tiempo real. Además, debe ser un sistema flexible ante futuras modificaciones que permita poder ser actualizado, tanto en lo que respecta al dominio de la aplicación como a las necesidades de los usuarios. La arquitectura del diseño posee gran importancia para que el sistema pueda ser implementado con éxito. Una variante factible en la implementación de estos sistemas lo constituye la combinación de un FPGA con una unidad de procesamiento de un ordenador personal (PC, por sus siglas en inglés), sobre todo por su accesibilidad en el mercado y su potencia de cálculo.

La selección de estos dispositivos se fundamenta en el cumplimiento de los requerimientos de fiabilidad, rendimiento en tiempo real y flexibilidad ante posibles cambios. Una implementación flexible garantizaría poder actualizar el sistema para corregir errores detectados. De lo contrario, sería necesario reemplazar todo el hardware, lo que implica mayores costes de mantenimiento (Novo, 2019).

La combinación de un FPGA con un procesador generalmente se dificulta en el establecimiento de un enlace de datos de alta velocidad entre el FPGA y el procesador. En la revisión bibliográfica se identificó la interfaz *PCI Express* como una de las vías de comunicación a alta velocidad entre un kit FPGA y un ordenador (Boyang, 2019; Khelif *et al.*, 2019; 2020; Guerra *et al.*, 2022a; Guerra *et al.*, 2022b).

La Figura 1 muestra el esquema del sistema de procesamiento y visualización de señales de radar propuesto, así como el enlace de comunicación entre el kit FPGA y el ordenador de escritorio a través de la interfaz *PCI Express*.

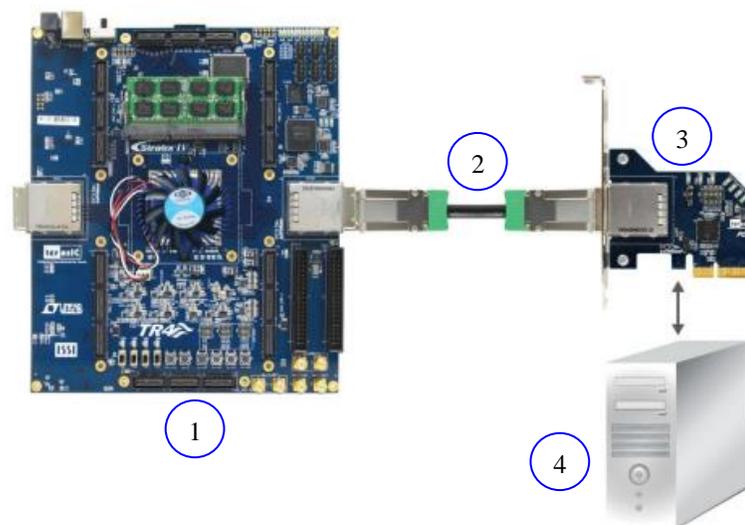


Figura 1. Conexión del kit FPGA al ordenador de escritorio

La Figura 1 muestra los cuatro elementos fundamentales que intervienen en la comunicación. El primero es el kit TR4 de Terasic, el cual cuenta con un FPGA Stratix IV GX EP4SGX230. El segundo es el cable de conexión PCIe x4. El



tercero es la tarjeta PCA3 que se conecta mediante la interfaz PCIe (*peripheral component interconnect express*) al cuarto elemento que es el ordenador. La tarjeta PCA3 puede soportar hasta la tercera generación de PCIe x4. El ordenador de escritorio posee interfaz PCI Express que permite realizar la lectura de los datos almacenados en el kit FPGA. La transmisión de datos a alta velocidad mantiene un rendimiento y latencia aceptables para operar en tiempo real. En la solución se emplea tecnología COTS (*Commercial off the shelf*) para reducir recursos, esfuerzo y tiempo de desarrollo, debido a la posibilidad de adquirirlos en el mercado para su uso inmediato y no tener que desarrollarlos (Trujillo and Méndez, 2022).

El kit FPGA se emplea para la recepción de la señal analógica, el procesamiento de la señal y de los datos. Este dispositivo garantiza una simple modificación ante futuros cambios con respecto a cómo se realiza el procesamiento, el flujo de datos y el control de las interfaces de entradas/salidas. Estos dispositivos son particularmente adecuados para la implementación de algoritmos paralelos, garantizando un bajo tiempo de ejecución de los mismos (Gómez, 2017). El procesamiento que se realiza en este dispositivo se encarga de llevar un cúmulo relativamente grande de datos a la cantidad de píxeles a representar en la pantalla. Véase el procesamiento de señales que se realiza en la Figura 2.

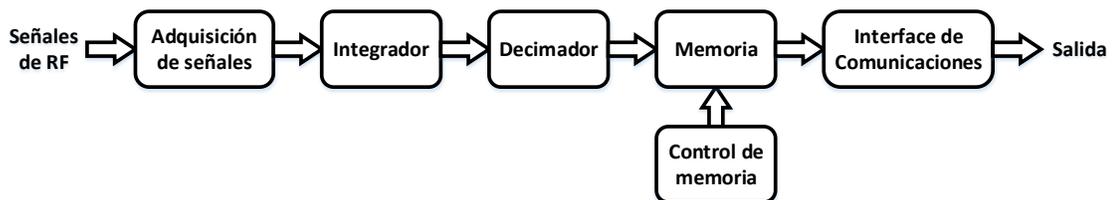


Figura 2. Diagrama en bloques del procesamiento de señales en la aplicación

La Figura 2 muestra como el dispositivo FPGA recibirá las señales de radiofrecuencia provenientes del radar. A partir del empleo del conversor de señal analógico a digital que posee el kit FPGA, la señal de entrada fue muestreada, dando la posibilidad de ser procesada digitalmente. Teniendo en cuenta que la distancia máxima que se desea representar en el visualizador es de 37500 metros y el tiempo que demora el pulso en recorrer esa distancia es de 250 microsegundos, al ser muestreada a 25 MHz se obtuvieron 6250 muestras por barrido. Luego, estas muestras se entregaron al integrador que alisa la señal y disminuye la influencia de picos ruidosos. Para esto se realizaron promediaciones cada 5 muestras. Posteriormente, se decima la señal, al tomarse el mayor valor del resultado de 2 promediaciones, que corresponden a 2 grupos de 5 muestras, es decir 10 muestras. El integrador disminuye la amplitud de los picos de ruido y el decimador evita una reducción mayor del valor de amplitud del blanco. Como resultado del empleo del integrador y el decimador, al tomar 1 valor cada 10 muestras de los 6250 valores, se obtienen 625 valores por barrido. Los 80000 datos obtenidos como resultado de los 128 barridos que realiza el radar en una



exploración del espacio aéreo, se almacenan en una memoria del kit FPGA. En el FPGA se creó un espacio de memoria que se comparte entre el kit FPGA y la PC; además de un bloque encargado de escribir los valores de salida del decimador, donde cada espacio de memoria tiene un tamaño de 256 bits. Se necesitaría 2500 espacios de memoria para almacenar 80000 datos de 8 bits. La escritura de los datos en memoria se realiza a través de un control de memoria que se encarga de organizar la estructura de los datos a escribir. Para garantizar que no se realice la lectura desde la PC en el mismo instante de tiempo que el FPGA esté escribiendo, la PC lee un espacio de memoria del FPGA que le permite saber si ya se realizó la escritura de todos los datos y se puede leer la memoria.

Luego de almacenar los datos del procesamiento en direcciones de memoria del kit FPGA, se leen desde una PC mediante la interfaz de comunicaciones *PCI Express*. Se emplea una biblioteca de enlace dinámico de 64 bits propia de Terasys (fabricante del kit FPGA que se utiliza) para la lectura de las direcciones de memoria. Una biblioteca de enlace dinámico (DLL, por sus siglas en inglés *Dynamic Link Library*), es un archivo ejecutable que contiene funciones que se podrán llamar desde cualquier aplicación Windows (Torres, 2017; Álvarez, 2021). El uso de este tipo de archivos ofrece una serie de ventajas que se pueden resumir en:

- Una función, definida dentro de una DLL, se encuentra disponible para cualquier aplicación Windows.
- Reducción del tamaño de las aplicaciones que utilizan la DLL.
- Mejora en el tiempo de compilación y/o carga de la aplicación.
- Ahorro de espacio en disco.
- Las DLLs son independientes de la aplicación.
- Brindan mayor flexibilidad frente a cambios, al solucionar pequeños errores con solo distribuir una nueva versión de la biblioteca dinámica.

Para acceder a los datos ubicados en las direcciones de memoria del FPGA a través de *PCI Express*, se debe cargar primero la DLL en memoria. Luego se llama la función *PCIE_Open()* de la DLL para abrir el controlador *PCI Express*. Al transferir los datos a alta velocidad por DMA (*Direct Memory Access*), se empleó la función *PCIE_DmaRead()* de la DLL, la cual lee datos desde la dirección de memoria de destino del FPGA.

A continuación de la lectura de los datos que corresponden a las muestras de amplitud de señal del radar, se representa la información en pantalla. Para la representación se empleó la indicación de tipo E. La indicación de tipo E es una representación bidimensional en la que los objetivos aparecen como puntos brillantes indicando la distancia a la que se encuentra en el eje vertical y el ángulo de elevación por la horizontal. Por ejemplo, al realizar el radar 128 barridos por exploración, se le asignó al indicador un ancho de 256 píxeles y se representaron 2 columnas de píxeles por barrido. En el eje vertical del indicador se visualizarán cada uno de los barridos del radar, los cuales están



compuestos por 625 valores. En la Figura 3 se aprecia cada 2 columnas del mismo color la representación de un barrido de 625 valores y al recibirse los 128 barridos (una exploración del radar) se representaría una imagen de 256 x 625 píxeles en la indicación de tipo E.

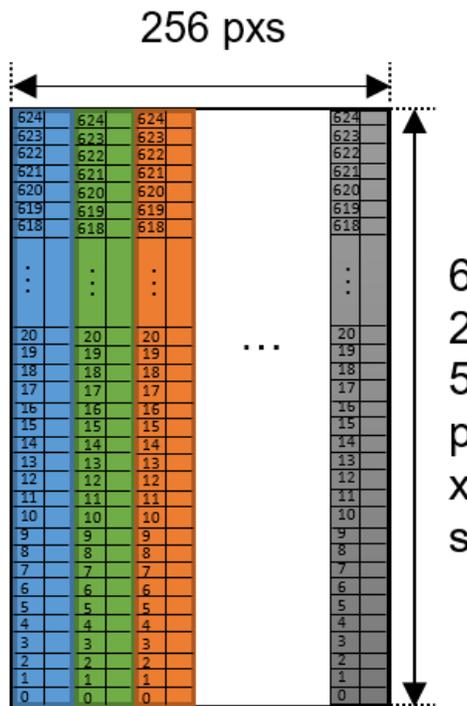


Figura 3. Dimensiones de la indicación de tipo E

Los píxeles que componen la imagen representan la intensidad de las muestras de la señal. El formato de la imagen que se empleó fue el RGB (*Red, Green, Blue*) de 32 bits, compuesto por 4 octetos, tres de ellos corresponden a las componentes RGB y el cuarto al canal alfa, el cual se le mantuvo su valor en 255. El proceso de pintar en la imagen consiste en acceder a cada uno de los 625 valores de 8 bits de cada barrido y generar el color que se le asignará al píxel correspondiente. Para la generación del color de cada pixel se empleó el formato RGB y al ponerse los valores de las tres componentes (rojo, verde y azul) con el mismo valor, se obtiene un color en escala de grises (Mesa *et al.*, 2018). Si el valor de las tres componentes es 255, el resultado será el blanco y será negro si las componentes tienen un valor 0 (Mesa *et al.*, 2018). Al conocerse el número del barrido a representar, se le asigna el color generado al píxel correspondiente de la imagen. La Figura 4 muestra el resultado de la representación de la situación aérea en imágenes.



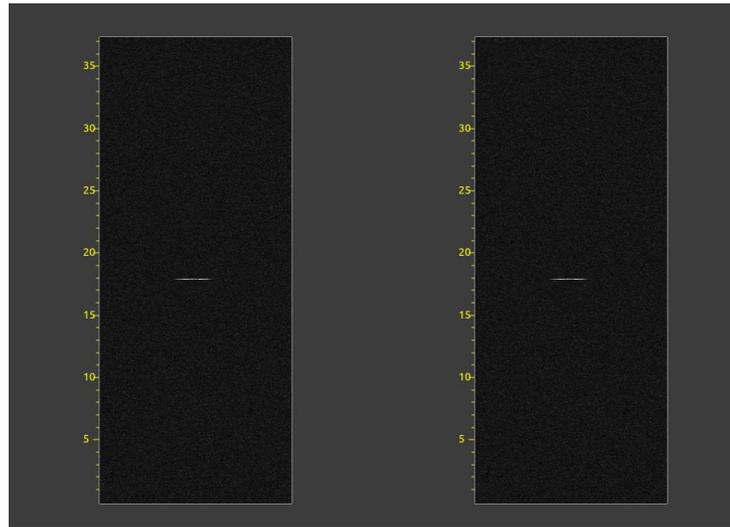


Figura 4. Representación de la situación aérea en la indicación tipo E

La Figura 4 muestra los picos de ruido y un objeto aéreo que se puede identificar por poseer una mayor intensidad, lo que produce un color blanco en la imagen. Además, muestra la representación de la situación aérea en dos indicadores de tipo E como resultado de la adquisición de las señales del radar de aproximación de precisión, su procesamiento en el kit FPGA y su representación en un ordenador de escritorio.

Resultados y discusión

Una vez implementada la arquitectura hardware-software del sistema de procesamiento y visualización, se hace necesario verificar si se cumple con la restricción temporal que impone el radar de aproximación de precisión. No completar las operaciones en 40 milisegundos puede resultar en una pérdida grave de la calidad proporcionada por el sistema. Se realiza un diseño de experimentos con el objetivo de determinar la configuración óptima del sistema visualizador y determinar los factores que influyen de manera significativa en el tiempo de ejecución. El proceso experimental a analizar es la ejecución del software encargado de la lectura de los datos ubicados en las direcciones de memoria del FPGA a través de *PCI Express* y la representación de los datos del radar. Se define como rendimiento del proceso, el tiempo en milisegundos que emplea el sistema en realizar la lectura y representación de los datos correspondientes a una exploración del radar.

Los factores controlables juegan un rol primordial en la caracterización de un proceso. Los factores controlables identificados son descritos a continuación:



- **Afinidad del procesador:** Característica del sistema operativo que permite vincular un proceso a uno o varios procesadores específicos, de forma tal que el mismo solo pueda ser ejecutado por esos procesadores.
- **Prioridad del proceso:** El sistema operativo le asigna una prioridad a cada uno de los procesos. Los procesos con mayor prioridad se ejecutan antes que los procesos con menor prioridad.

El factor no controlable identificado es:

- **Procesos ejecutados por el sistema operativo:** En el momento en que se ejecuta el software hay procesos ejecutándose en el sistema operativo, que consumen recursos arbitrariamente y no pueden ser detenidos.

La unidad experimental sobre la cual se ejecutarán todos los tratamientos es un ordenador que tiene las siguientes características:

- Procesador: Intel Core i3-2120 a 3.30 GHz
- Memoria: 4 GB RAM
- Sistema operativo: Windows 7 Enterprise 64 bits

Para cada uno de los factores seleccionados en el experimento existen dos niveles, uno de ellos se considera como el valor bajo y otro como el alto. En la Tabla 1 se muestran los factores con los niveles asociados a cada uno:

Tabla 1. Niveles de los factores seleccionados para realizar los experimentos

Factor	Nivel bajo	Nivel alto
Afinidad del procesador	2	4
Prioridad del proceso	Por debajo de lo normal (DN)	Por encima de lo normal (EN)

Como se busca la configuración óptima se eligió como nivel alto para la afinidad 4, que es la máxima afinidad permitida por el ordenador donde se realizarán los experimentos y el nivel bajo 2. En el caso de la prioridad del proceso se seleccionó para el nivel alto por encima de lo normal (EN) y para el nivel bajo por debajo de lo normal (DN).

Para realizar el diseño de experimentos se decidió hacer un diseño factorial completo, ya que se desea incluir todas las combinaciones posibles de los niveles de los factores. Al tener el factor controlable afinidad del procesador 2 niveles y prioridad del proceso 2 niveles, se realizaron 4 combinaciones y cada combinación será replicada 5 veces para disminuir el error experimental, por lo que en total se realizaron 20 corridas. Al utilizarse el mismo número de repeticiones en cada tratamiento, provoca que el diseño sea balanceado. Para que el ambiente en el que se aplican los tratamientos sea lo más uniforme posible, se realizará en orden aleatorio.



Los tratamientos correspondientes al diseño señalado con anterioridad, fueron generados en el software Minitab 19. La Tabla 2 muestra los valores de rendimiento en milisegundos de las 20 ejecuciones realizadas como resultado de los cambios intencionados en los factores del proceso.

Tabla 2. Tratamientos generados por el software Minitab 19 y tiempo de ejecución

No. ejecución	Afinidad del procesador	Prioridad del proceso	Tiempo (ms)
1	4	DN	24,0
2	4	DN	22,5
3	4	EN	15,0
4	2	EN	21,0
5	2	DN	28,5
6	4	EN	17,5
7	4	DN	25,0
8	2	DN	30,0
9	2	EN	19,5
10	4	DN	23,5
11	4	DN	22,0
12	4	EN	12,5
13	2	EN	22,0
14	2	DN	27,0
15	2	EN	22,0
16	4	EN	19,0
17	2	DN	27,5
18	2	EN	20,0
19	4	EN	14,5
20	2	DN	31,5

Luego de obtener los valores de rendimiento para los tratamientos generados, se procedió a generar un conjunto de gráficas para obtener conclusiones sobre los experimentos realizados. Para ello se empleó el software Minitab 19. El objetivo es investigar los efectos de las variables de entrada sobre la variable de respuesta. El análisis a los datos recolectados permitirá la determinación del factor más influyente para el cumplimiento del tiempo de ejecución y la determinación de la configuración de factores que optimiza los resultados.

El diagrama de Pareto permite detectar los efectos determinantes de factores e interacciones en el rendimiento del proceso o sistema (Antony, 2014). Este diagrama muestra el valor absoluto de los efectos y se traza una línea de referencia en el gráfico. Cualquier efecto que sobrepase esta línea de referencia es potencialmente importante para el rendimiento del proceso (Antony, 2014). Los efectos son organizados según su valor en orden decreciente. Además,



permite visualizar las variables que tienen un mayor impacto sobre la variable de respuesta. Como se aprecia en la Figura 5 la afinidad del procesador y prioridad del proceso influyen en el tiempo de ejecución del software.

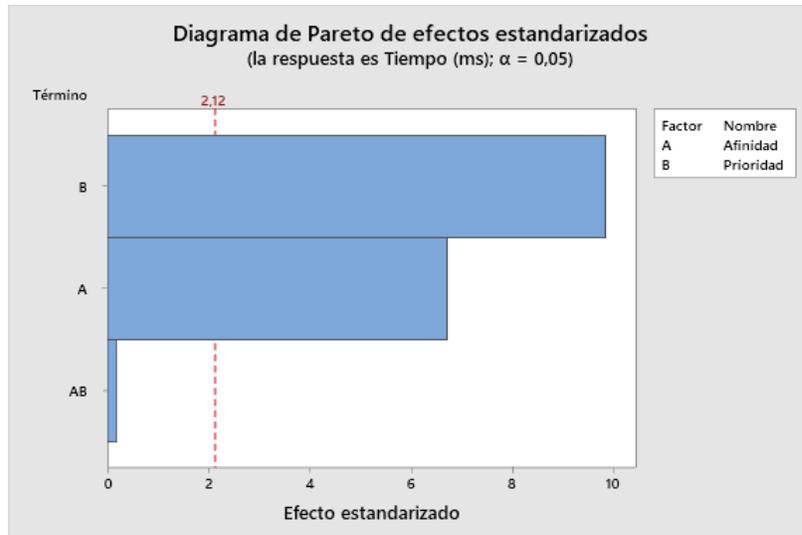


Figura 5. Diagrama de Pareto

Puesto que el diagrama de Pareto muestra el valor absoluto de los efectos, se determinó que el efecto prioridad del proceso es el más significativo, pero no se pudo determinar cuáles efectos aumentan o disminuyen la respuesta. Se utilizó la gráfica de probabilidad normal de los efectos estandarizados para examinar la magnitud y dirección de los efectos. Los efectos no influyentes tienden a estar sobre la línea en la zona central, mientras que los efectos influyentes tienden a estar alejados de la línea, en los extremos de la misma. Véase la Figura 6.

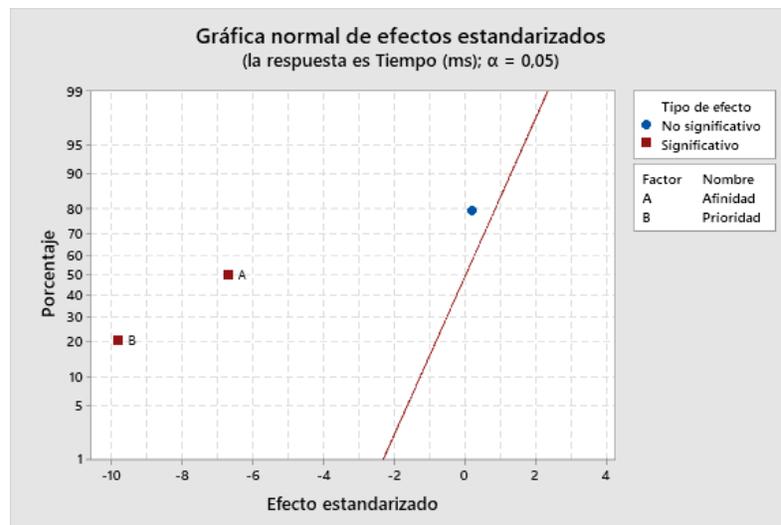


Figura 6. Gráfica de probabilidad normal



La gráfica de probabilidad normal corrobora lo obtenido en el diagrama de Pareto, pues el factor prioridad del proceso es el más significativo al estar más distante de 0 en el eje X. Además, se aprecia que ambos factores tienen un efecto estandarizado negativo debido a que reducen la repuesta (tiempo de ejecución) cuando la configuración de cada factor cambia del nivel bajo al nivel alto.

Para analizar la magnitud del efecto de cada factor y comparar el efecto entre distintos factores, se emplea la gráfica de efectos principales (Antony, 2014). Un efecto principal es la diferencia en la respuesta media entre los niveles de un factor. En esta gráfica se muestran puntos azules que indican las medias de tiempo al utilizar los dos niveles de afinidad del procesador y las medias de tiempo al utilizar las dos prioridades. Las líneas discontinuas horizontales muestran la media de tiempo de todas las corridas. En la Figura 7 se muestra el efecto que tiene cada uno de los factores por separado en el rendimiento.

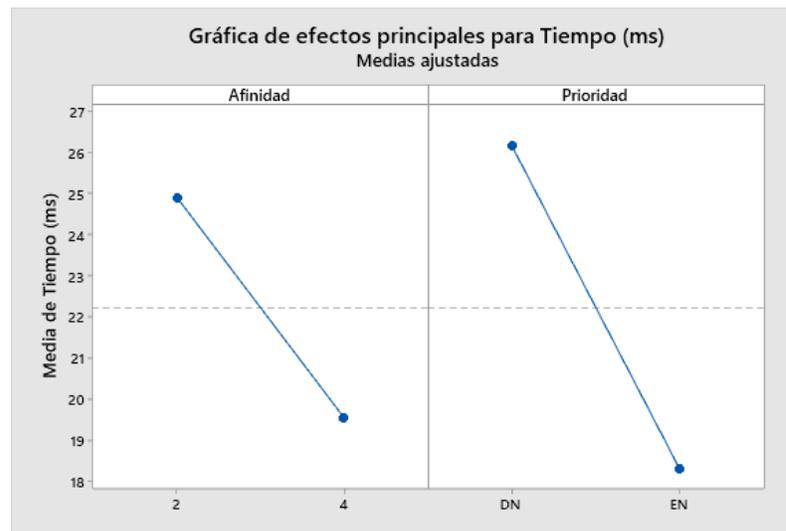


Figura 7. Gráfica de efectos principales

El factor prioridad del proceso al poseer la mayor pendiente en la gráfica, indica que la magnitud del efecto principal es mayor. El nivel en el que se obtienen los mejores valores de rendimiento se caracteriza por tener prioridad por encima de lo normal (EN), con una media de 18.3 ms y un efecto directo sobre el rendimiento. En el caso del factor afinidad del procesador, el nivel que obtiene una disminución del tiempo se caracteriza por tener 4 procesadores y una media de 19.55 ms. Los dos factores tienen un efecto negativo. Con esta gráfica se corrobora los resultados anteriores del análisis del diagrama de Pareto y la gráfica de probabilidad normal de los efectos estandarizados. Ambos factores influyen en el tiempo de ejecución del software y el más significativo es la prioridad del proceso, ya que existe una diferencia entre las medias de los niveles de 1.25 ms.



Para analizar la interacción entre los dos factores se debe examinar la gráfica de interacción que se muestra en la Figura 8. La gráfica de interacciones es una herramienta que grafica la respuesta media de dos factores para todas las posibles combinaciones de sus niveles. Cada punto de la gráfica de interacción muestra el tiempo de procesamiento medio con diferentes combinaciones de los niveles de los factores.

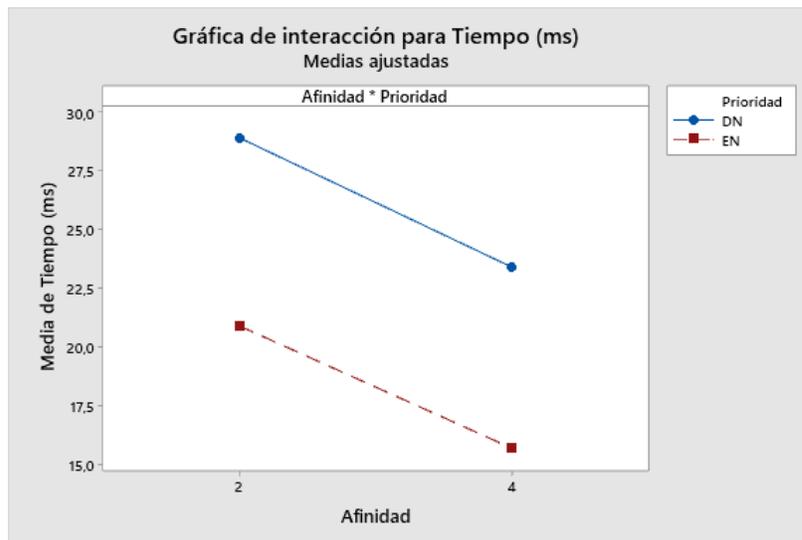


Figura 8. Gráfica de interacción

La Figura 8 muestra que todas las combinaciones de los niveles de los factores cumplen con la restricción temporal. La gráfica de interacción corrobora que con la ejecución del software con prioridad por debajo de lo normal se aumenta el tiempo de ejecución. Al ser las líneas paralelas, no existe una interacción entre los factores, corroborando lo obtenido en la gráfica de Pareto. Al no existir interacción entre los factores implica que el cambio en la respuesta media del nivel bajo a alto de un factor, no depende del nivel del otro factor. La gráfica de interacción indica que al ejecutar el software con prioridad por encima de lo normal y con una afinidad de 4 procesadores, el software se ejecutó con menor tiempo, con una media de 15.7 ms, siendo esta última la configuración óptima de los factores.

Conclusiones

La arquitectura hardware-software propuesta logra una sustitución tecnológica del sistema de procesamiento y visualización, cumpliendo con el período de actualización del radar de aproximación de precisión. Se propone una solución de hardware modular, genérico y flexible, sin la necesidad de diseño y fabricación de placas de circuito impreso personalizadas, permitiendo realizar el procesamiento digital de las señales de radar y una fácil modificación ante futuros cambios. Los resultados experimentales demostraron que al ejecutarse el software de representación de la



información del radar con prioridad por encima de lo normal y con una afinidad de 4 procesadores se logró un mejor aprovechamiento de los recursos de cómputo y el menor tiempo de ejecución del software. El sistema hardware-software contribuyó a la obtención de una solución de alto rendimiento y fácil implementación.

Conflictos de intereses

Los autores no poseen conflictos de intereses.

Contribución de los autores

1. Conceptualización: Lisvan Guevara Trujillo y Ariel Hernández Reyes.
2. Curación de datos: Lisvan Guevara Trujillo y Ariel Hernández Reyes.
3. Análisis formal: Marcelino Sánchez Posada.
4. Adquisición de fondos: Ariel Hernández Reyes y Marcelino Sánchez Posada.
5. Investigación: Lisvan Guevara Trujillo, Ariel Hernández Reyes y Marcelino Sánchez Posada.
6. Metodología: Ariel Hernández Reyes y Lisvan Guevara Trujillo.
7. Administración del proyecto: Lisvan Guevara Trujillo y Marcelino Sánchez Posada.
8. Recursos: Ariel Hernández Reyes y Marcelino Sánchez Posada.
9. Software: Lisvan Guevara Trujillo, Ariel Hernández Reyes y Marcelino Sánchez Posada.
10. Supervisión: Wenny Hojas-Mazo.
11. Validación: Marcelino Sánchez Posada y Wenny Hojas-Mazo.
12. Visualización: Lisvan Guevara Trujillo y Ariel Hernández Reyes.
13. Redacción – borrador original: Lisvan Guevara Trujillo.
14. Redacción – revisión y edición: Wenny Hojas-Mazo.

Financiamiento

La investigación no requirió fuente de financiamiento externa.

Referencias

ACOSTA, N.; TOSINI, M. A., *et al.* *Desarrollo de un Visualizador de Señales de Radar. XII Congreso Argentino de Ciencias de la Computación.* Argentina, 2006. 53-64.



Esta obra está bajo una licencia *Creative Commons* de tipo **Atribución 4.0 Internacional**
(CC BY 4.0)

- ACOSTA, N.; TOSINI, M. A., *et al.* Improving radar visualization system *Journal of Computer Science & Technology*, 2007, 7(1): 1-7.
- ÁLVAREZ, P. F. *Extracción y Análisis de Artefactos de Memoria de la Aplicación Telegram Desktop*. Departamento de Matemáticas. España, Universidad de León, 2021. 67. p.
- ANTONY, J. *Design of Experiments for Engineers and Scientists*. 2nd. Netherlands, Elsevier, 2014. 221 p. 978-0-08-099417-8.
- BOYANG, L. *Research and Implementation of XDMA High Speed Data Transmission IP Core Based on PCI Express and FPGA. 1st International Conference on Civil Aviation Safety and Information Technology (ICCASIT)*. Kunming, China, IEEE, 2019. 408-411.
- GÓMEZ, F. A. S. *Metodología de codiseño hardware-software para procesamiento de señales radar en sistemas embebidos*. Facultad de Ingeniería. Colombia, Pontificia Universidad Javeriana, 2017. 62. p.
- GUERRA, Y. I.; ZALDÍVAR, A. R., *et al.* Implementación Hardware-Software de un Sistema de Estimación de DOA con Súper Resolución *Ingeniería Electrónica, Automática y Comunicaciones*, 2022, 43(1): 16-32.
- GUERRA, Y. I.; ZALDÍVAR, A. R. R., *et al.* Sistema de estimación de dirección de arribo para un radio cognitivo consciente de la ubicación *Telemática*, 2022, 21(2): 12-26.
- JEVTIĆ, M. and STAMATOVIĆ, M. *Radar data processing and visualization on desktop platforms. 17th Telecommunications forum (TELFOR)*. Belgrade, Serbia, 2009. 1315-1318.
- KAUSHIK, P. Radar Displays *International Journal of Innovative Research in Technology*, 2014, 1(7): 472-476.
- KAVYASHREE, V.; MADHU CHAITRA, P. N., *et al.* A Radar Target Generator for Airborne Targets *International Journal of Advance Research and Innovative ideas in Education*, 2017, 2(15): 249-261.
- KHELIF, M. A.; LORANDEL, J., *et al.* Toward a hardware man-in-the-middle attack on PCIe bus *Microprocessors and Microsystems*, 2020, 77: 103198. Disponible en: <https://doi.org/10.1016/j.micpro.2020.103198>.
- KHELIF, M. A.; LORANDEL, J., *et al.* Toward a Hardware Man-in-the-Middle Attack on PCIe Bus for Smart Data Replay. *22nd Euromicro Conference on Digital System Design*. Kallithea, Greece, IEEE, 2019. 230-237.
- LÓPEZ ESTRADA, S. M. *Detección Automática en Señales de Radar Basada en Conocimiento*. Puebla, Mexico, Instituto Nacional de Astrofísica, Óptica y Electrónica, 2009. p.
- MESA, E. D. A.; VARGAS, H. S. T., *et al.* Proceso de extracción y almacenaje de características a partir de imágenes de huellas de mordida en el desarrollo de un software para la identificación de personas mediante procesamiento digital de imágenes *Revista Espacios*, 2018, 39(11): 2-15.



- NOVO, J. A. C. *Aceleración y optimización del consumo energético de clasificadores en cascada para la detección de rostros sobre arquitecturas asimétricas*. Departamento de Arquitectura y Tecnología de Computadores. Granada, Universidad de Granada, 2019. p.
- STAMATOVIĆ, M.; JEVTIĆ, M., *et al.* *Design and implementation of a modern radar display for air surveillance applications*. *20th Telecommunications Forum (TELFOR)*, IEEE, 2012. 1520-1523.
- STAMATOVIĆ, M.; JEVTIĆ, M., *et al.* *Modern Air Situation Picture Display for Air Surveillance Radar Applications* *Telfor Journal*, 2013, 5(1): 54-59.
- SULISTYANINGSIH; SAPUTERA, Y. P., *et al.* *Design of radar display of Indonesian airspace monitoring application* *TELKOMNIKA*, 2019, 17(3): 1176-1184. Disponible en: <https://doi.org/10.12928/TELKOMNIKA.v17i3.11778>.
- TORRES, J. J. R. *Detección y clasificación de software malicioso*. Centro de Investigación en Computación. México, Instituto Politécnico Nacional, 2017. 107. p.
- TRUJILLO, L. G. and MÉNDEZ, L. Z. *Sistema visualizador para un radar de seguimiento con reducción de costo, tamaño y consumo eléctrico*. Departamento de Automática y Computación. Cuba, Universidad Tecnológica de La Habana José Antonio Echeverría, 2022. 90. p.
- TRUJILLO, L. G.; MÉNDEZ, L. Z., *et al.* *Visualizador digital de un radar de seguimiento empleando Raspberry Pi 4* *Serie Científica de la Universidad de las Ciencias Informáticas*, 2022, 15(9): 45-64.
- TRUJILLO, L. G.; RODRÍGUEZ, A. M., *et al.* *Representación en tiempo real de señales de radar empleando Odroid XU4* *Elektron*, 2020, 4(2): 87-92. Disponible en: <https://doi.org/10.37537/rev.elektron.4.2.104.2020>.

