



DESARROLLO DE ARQUITECTURA SoPC PARA ROBOT MÓVIL BASADO EN FPGA

DEVELOPMENT OF SoPC ARCHITECTURE FOR FPGA-BASED MOBILE ROBOT

¹Andrés-David Suárez-Gómez, ²Jorge Orlando Bareño Quintero

^{1,2,3}Universidad Nacional Abierta y a Distancia, Colombia

Recibido: 20/10/2023 Aprobado 20/11/2023

RESUMEN

En el contexto del rápido avance en el campo de la robótica móvil, la necesidad de sistemas eficientes y con alto poder de procesamiento embebido ha surgido como un desafío fundamental. En este artículo se aborda dicha problemática mediante el uso de FPGAs, dispositivos subutilizados en la literatura para el desarrollo de arquitecturas con co-diseño H/S (hardware/software) en robots móviles. Se destaca la idoneidad de las FPGAs para este propósito, ya que ofrecen un equilibrio entre eficiencia y bajo consumo de energía. El artículo se centra en la implementación de una arquitectura basada en SoPC (System on Programmable Chip) diseñada específicamente para un robot móvil de tracción diferencial basado en FPGA. Los aspectos más relevantes del diseño se presentan mediante diagramas que ilustran la disposición y la interconexión de los componentes clave. La implementación práctica de esta arquitectura se llevó a cabo en un robot móvil cuya unidad de procesamiento es una tarjeta de desarrollo DE0 Nano equipada con una FPGA Cyclone® IV. Los resultados obtenidos revelan un comportamiento exitoso de la arquitectura en términos de interfaz con sensores, actuadores y procesamiento de datos. La eficiencia del diseño se refleja en la ocupación equilibrada de recursos en la FPGA, destacando el 67 % de área lógica aún disponible para futuras implementaciones de módulos para aceleración de hardware. Este artículo demuestra la viabilidad y eficacia de las FPGAs en el diseño de arquitecturas avanzadas para robots móviles, proporcionando una base sólida para futuras investigaciones y desarrollos en el campo de la robótica

Palabras clave: co-diseño H/S, FPGA, robot móvil, SoPC.

Citación: Suárez Gómez, A. D. ., & Bareño Quintero, J. O. . (2023). Desarrollo de arquitectura SoPC para robot móvil basado en FPGA. *Publicaciones E Investigación*, 17(4). <https://doi.org/10.22490/25394088.7506>

¹andresd.suarez@unad.edu.co / <https://orcid.org/0000-0002-3576-4428>

²jorge.bareno@unad.edu.co / <https://orcid.org/0000-0002-4821-8194>

<https://doi.org/10.22490/25394088.7506>

ABSTRACT

In the context of the rapid progress in the field of mobile robotics, the need for efficient systems with high embedded processing power has emerged as a fundamental challenge. This paper addresses this issue by using FPGAs, devices underutilized in the literature for the development of architectures with H/S (hardware/software) co-design in mobile robots. The suitability of FPGAs for this purpose is highlighted, as they offer a balance between efficiency and low power consumption. The paper focuses on the implementation of a SoPC (System on Programmable Chip) based architecture designed specifically for an FPGA-based differential drive mobile robot. The most relevant aspects of the design are presented by means of diagrams illustrating the layout and interconnection of key components. The practical implementation of this architecture was carried out on a mobile robot whose processing unit is a DE0 Nano development board equipped with a Cyclone® IV FPGA. The results obtained reveal a successful behavior of the architecture in terms of interfacing with sensors, actuators, and data processing. The efficiency of the design is reflected in the balanced occupation of resources in the FPGA, highlighting the 67 % of logical area still available for future implementations of modules for hardware acceleration. This paper demonstrates the feasibility and efficiency of FPGAs in the design of advanced architectures for mobile robots, providing a solid foundation for future research and development in the field of robotics.

Keywords: H/S co-design, FPGA, Mobile Robot, SoPC.



1. INTRODUCCIÓN

La robótica móvil ha avanzado significativamente en los últimos años, usando cada vez algoritmos más complejos y que necesitan un procesamiento en tiempo real, esta complejidad en el procesamiento de datos y los altos requerimientos computacionales suponen un gran reto para su aplicación en tareas del mundo real ya que se necesita procesar una enorme cantidad de datos de múltiples sensores heterogéneos (Wan *et al.*, 2021). Adicionalmente, el sistema robótico suele disponer de recursos limitados, como memoria, ancho de banda y capacidad de cálculo, lo que dificulta el cumplimiento de los requisitos en tiempo real. Por otra parte, suelen tener estrictas limitaciones de potencia que no pueden soportar la cantidad de computación necesaria para realizar tareas como localización, navegación y planificación de trayectorias. Por tanto, la complejidad computacional, así como las limitaciones de tiempo real y potencia del sistema robótico dificultan su aplicación en escenarios de latencia crítica o potencia limitada (Liu *et al.*, 2021; Wan *et al.*, 2021).

Teniendo en cuenta lo anterior, es esencial elegir una unidad de procesamiento adecuada para el sistema robótico, siendo las CPUs y las GPUs las más utilizadas. La CPU está diseñada para manejar una amplia gama de tareas con rapidez y flexibilidad (Gu *et al.*, 2015; Wan *et al.*, 2021). La GPU está diseñada con miles de núcleos de procesamiento permitiendo paralelismo (Wan *et al.*, 2021). Sin embargo, las CPU y GPU convencionales suelen consumir cantidades de energía muy superiores a la disponible en los sistemas robóticos de recursos limitados (Liu *et al.*, 2021; Wan *et al.*, 2021). Las FPGAs se han convertido en alternativas competitivas ya que pueden superar a la CPU y la GPU en rendimiento, paralelismo y eficiencia energética (Correa-Caicedo *et al.*, 2021; Liu *et al.*, 2021; Plancher *et al.*, 2021; Shi *et al.*, 2018; Wan *et al.*, 2021, 2022; Wu *et al.*, 2021).

Las FPGAs ofrecen ventajas significativas para aplicaciones en sistemas robóticos como: bajo consumo de energía, rendimiento, velocidad de procesamiento,

paralelismo, aceleración de hardware, reconfigurabilidad, versatilidad y baja latencia (Al-Khalidy *et al.*, 2020; Chand *et al.*, 2022; Liu *et al.*, 2021; Miyagi *et al.*, 2021; Nilay *et al.*, 2020; Wan *et al.*, 2021, 2022). Sin embargo, el uso de FPGAs aún no está muy extendido debido a una complejidad mayor con respecto a la programación en software y la integración de componentes de hardware y software por medio del codiseño (Podlubne & Göhringer, 2022; Wan *et al.*, 2021, 2022).

Dentro de la literatura existen esfuerzos en el uso de FPGAs en robótica, algunos ejemplos de los trabajos más actuales son los siguientes: (i) recolector de fruta robótico con base móvil y brazo de tres grados de libertad con aceleración de redes neuronales en FPGA (Nilay *et al.*, 2020), (ii) robot omnidireccional de tres ruedas con control inteligente adaptativo en FPGA (Al-Khalidy *et al.*, 2020), (iii) robot de rescate con control en FPGA (Sudhakar *et al.*, 2023), (iv) robot con base móvil y dos brazos robóticos con planificación y control de movimiento en FPGA (Chand *et al.*, 2022), (v) robot autónomo con procesamiento de imágenes en FPGA (Kojima, 2022), (vi) robot móvil con aparcamiento autónomo en FPGA (Divya Vani *et al.*, 2022), (vii) robot de rescate con detección en FPGA (Sun *et al.*, 2014), (viii) robot móvil con controlador cinemático en FPGA (Tsai *et al.*, 2010), y (ix) robot móvil con procesamiento de imágenes en FPGA (Miyagi *et al.*, 2021).

Este trabajo se enmarca en el contexto descrito, explorando el potencial de las FPGAs en el diseño de arquitecturas SoPC para robots móviles. Este artículo se enfoca en abordar la complejidad computacional inherente de la robótica mediante el codiseño hardware/software, proporcionando así una solución innovadora a los desafíos previamente expuestos. En este sentido, se alinea con los esfuerzos actuales en la literatura, en donde se busca la aplicación efectiva de las FPGAs en contextos de robótica, y más específicamente, de robótica móvil.

2. MATERIALES Y MÉTODOS

La tecnología SoPC en FPGAs ha transformado el codiseño H/S, permitiendo la integración de circuitos de hardware, procesadores y módulos IP en un solo dispositivo lógico programable. Esta tecnología ha revolucionado el co-diseño H/S combinando la flexibilidad del software y la aceleración y paralelismo del hardware (Tsai *et al.*, 2010). Esta flexibilidad única se materializa en el diseño de una arquitectura SoPC específicamente concebida para un robot móvil basado en FPGA. La arquitectura, representada visualmente en la Figura 1, se caracteriza por su capacidad para interactuar con sensores y actuadores, aprovechar módulos IP para aceleración en hardware, y ejecutar algoritmos en software mediante un procesador soft-core Nios II con el sistema operativo en tiempo real FreeRTOS.

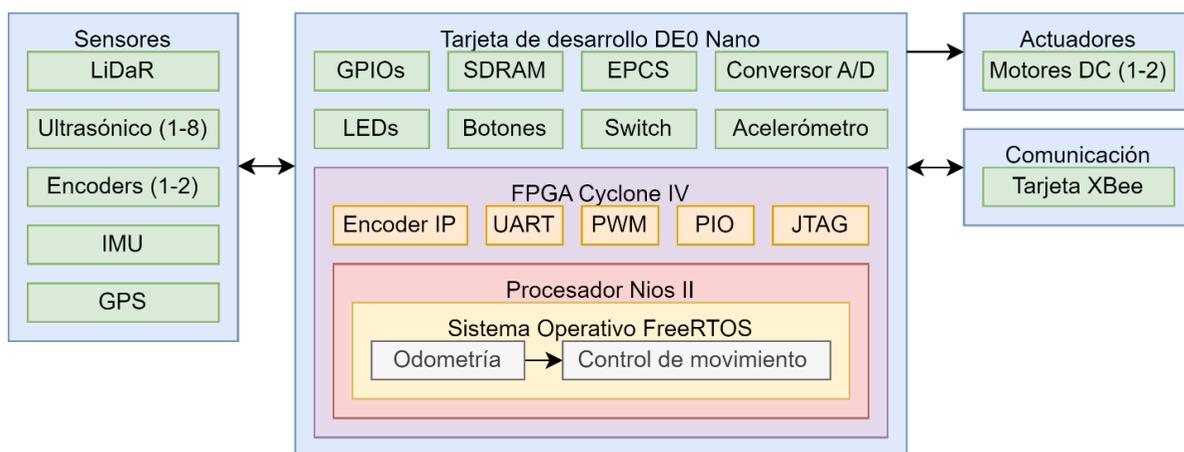


Figura 1. Representación de la arquitectura SoPC para un robot móvil basado en FPGA.

En el núcleo de la arquitectura se encuentra la adaptación del módulo IP para la decodificación de encoders de cuadratura, previamente desarrollado en Suárez-Gómez & Pérez-Holguín (2020). Este módulo desempeña un papel crucial al facilitar la interpretación precisa de las señales de los encoders, contribuyendo así a la precisión en la retroalimentación de posición del robot. Además, se implementa el sistema de odometría basado en codiseño H/S desarrollado en Suárez-Gómez & Pérez-Holguín (2022). Esta estrategia de odometría no solo proporciona una estimación precisa de la posición del robot, sino que también demuestra la sinergia entre los elementos de hardware y software en la arquitectura.

La elección de FreeRTOS como sistema operativo en tiempo real se fundamenta en su eficacia para

gestionar tareas de manera eficiente en un entorno multitarea, aportando estabilidad y previsibilidad al sistema. La arquitectura se beneficia significativamente de las capacidades de gestión de tareas en tiempo real de FreeRTOS, empleándolas para la ejecución de algoritmos críticos de control y comunicación. Para facilitar la transmisión de datos, se utiliza una tarjeta Xbee, estableciendo una comunicación efectiva entre el robot móvil y una interfaz de usuario gráfica (GUI) implementada en un PC.

La Figura 2 ofrece una representación visual del robot móvil que sirve como plataforma de implementación para la arquitectura. La Figura 3 representa la comunicación robot-PC y la Figura 4 presenta la interfaz gráfica para la comunicación con el robot.



Figura 2. Robot móvil de tracción diferencial basado en FPGA.

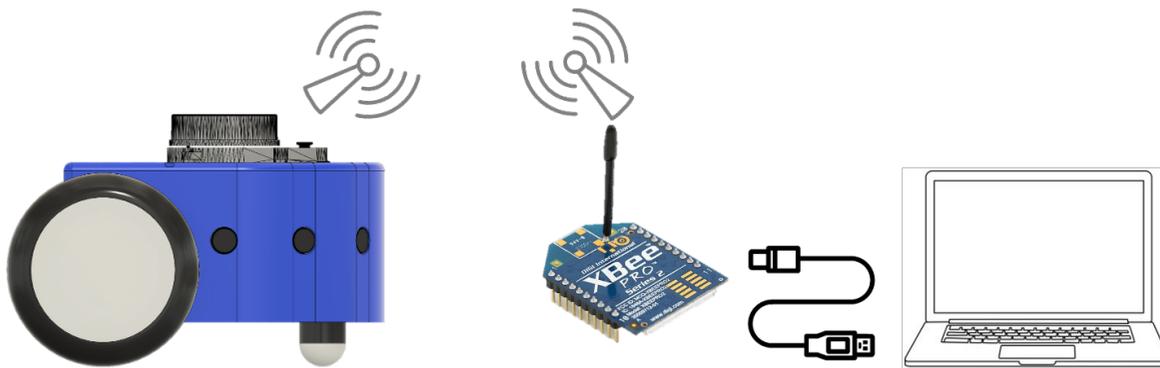


Figura 3. Comunicación de robot móvil basado en FPGA con PC.



Figura 4. GUI para comunicación con robot móvil basado en FPGA.

3. DESARROLLO

La meta principal de este trabajo radica en la concepción y materialización de una arquitectura SoPC para un robot móvil basado en FPGAs. Este diseño se ha construido cuidadosamente mediante la aplicación de técnicas avanzadas de codiseño H/S, permitiendo la integración sinérgica de componentes de hardware y software. La aceleración de hardware se convierte en un pilar fundamental, dotando al sistema de la capacidad de interactuar de manera eficiente con sensores y actuadores, así como de ejecutar algoritmos de procesamiento de datos y control de manera optimizada usando módulos IP.

Este enfoque es muy poco utilizado en la literatura, ya que las soluciones convencionales suelen basarse únicamente en la programación de software en dispositivos como microcontroladores o microprocesadores, la arquitectura descrita se erige como una alternativa innovadora que permite no solo la reconfiguración sino también la adición de módulos IP según las

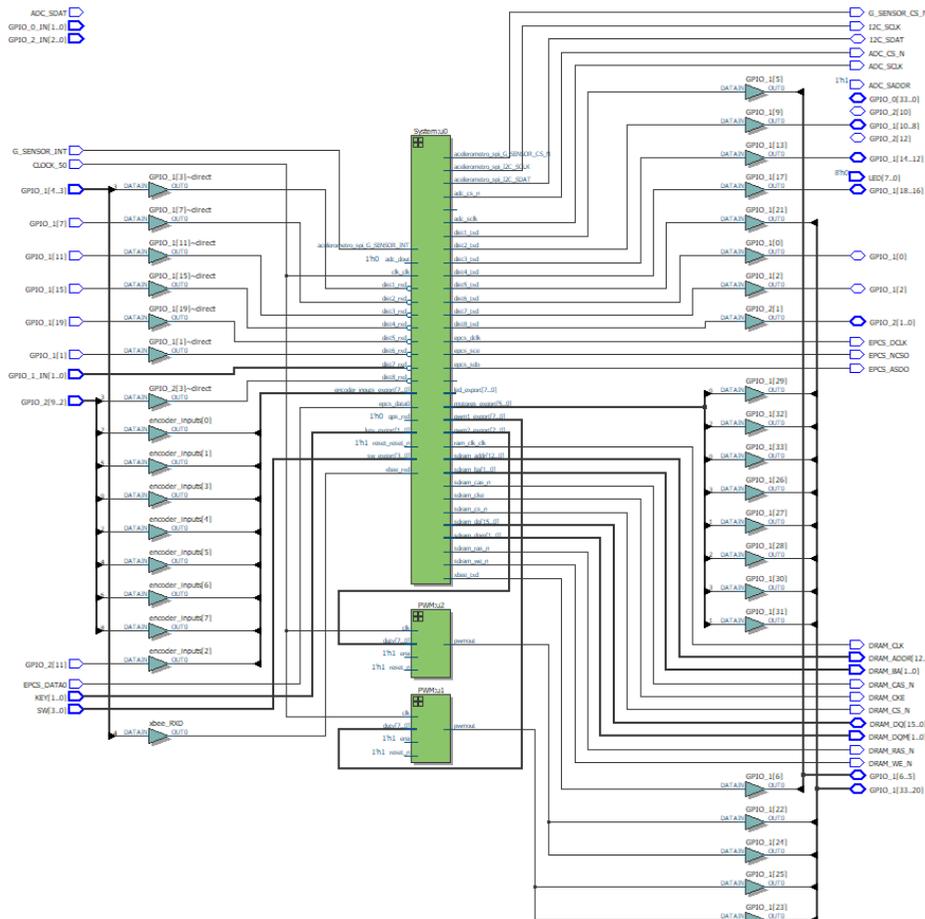
necesidades específicas de la aplicación. La capacidad de reconfiguración y flexibilidad incorporadas no solo satisfacen las demandas actuales, sino que también anticipan las futuras necesidades de sistemas robóticos en constante evolución.

La implementación de la arquitectura se llevó a cabo en el software Quartus II v13.1 mediante el uso de la herramienta Qsys. Este proceso implica la configuración y conexión de los diferentes componentes necesarios para la arquitectura SoPC. La Figura 5 muestra visualmente la disposición de estos elementos en el entorno de desarrollo. La Figura 6 muestra el diagrama RTL generado para la arquitectura SoPC del robot móvil.

Una vez completada la etapa de diseño en software, se procedió a la implementación física en el robot móvil de tracción diferencial que se muestra en la Figura 2. Este robot utiliza una unidad de procesamiento basada en una tarjeta de desarrollo DE0 Nano equipada con una FPGA Cyclone® IV.

Use	Connections	Name	Description	Export	Clock	Base	End	IRQ
		CPU	Nios II Processor					
		clk	Clock Input	Double-click to export	sys_clk			
		reset_n	Reset Input	Double-click to export	[clk]			
		data_master	Avalon Memory Mapped Master	Double-click to export	[clk]			IRQ 0
		instruction_master	Avalon Memory Mapped Master	Double-click to export	[clk]			
		itag_debug_module_f...	Reset Output	Double-click to export	[clk]			
		itag_debug_module...	Avalon Memory Mapped Slave	Double-click to export	[clk]			
		custom_instruction_m...	Custom Instruction Master	Double-click to export	[clk]	0x0400_1800	0x0400_1fff	
		ITAG	ITAG UART		sys_clk	0x0400_2270	0x0400_2277	
		SDRAM	SDRAM Controller		sys_clk	0x0200_0000	0x03ff_ffff	
		EPCS	EPCS(EPCQx1 Serial Flash Controller		sys_clk	0x0400_1000	0x0400_1fff	
		LED	PID (Parallel I/O)		sys_clk	0x0400_2260	0x0400_226f	
		SW	PID (Parallel I/O)		sys_clk	0x0400_2250	0x0400_225f	
		KEY	PID (Parallel I/O)		sys_clk	0x0400_2240	0x0400_224f	
		ADC_D00	DES-Hero ADC Controller		sys_clk	0x0400_2160	0x0400_217f	
		ACCELEROMETRO_SPI	Accelerometer SPI Mode		sys_clk	0x0400_2284	0x0400_228b	
		XBEE	UART (RS-232 Serial Port)		sys_clk			
		clk	Clock Input	Double-click to export	[clk]			
		reset	Reset Input	Double-click to export	[clk]			
		external_connection	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x0400_2120	0x0400_213f	
		external_connection	Conduit	xbxee				
		DIST1	UART (RS-232 Serial Port)		sys_clk			
		clk	Clock Input	Double-click to export	[clk]			
		reset	Reset Input	Double-click to export	[clk]			
		s1	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x0400_2100	0x0400_211f	
		external_connection	Conduit	dist1				
		DIST2	UART (RS-232 Serial Port)		sys_clk	0x0400_20e0	0x0400_20ef	
		DIST3	UART (RS-232 Serial Port)		sys_clk	0x0400_20d0	0x0400_20df	
		DIST4	UART (RS-232 Serial Port)		sys_clk	0x0400_20a0	0x0400_20af	
		DIST5	UART (RS-232 Serial Port)		sys_clk	0x0400_2080	0x0400_209f	
		DIST6	UART (RS-232 Serial Port)		sys_clk	0x0400_2060	0x0400_207f	
		DIST7	UART (RS-232 Serial Port)		sys_clk	0x0400_2040	0x0400_205f	
		DIST8	UART (RS-232 Serial Port)		sys_clk	0x0400_2020	0x0400_203f	
		MOTORES	PID (Parallel I/O)		sys_clk			
		clk	Clock Input	Double-click to export	[clk]			
		reset	Reset Input	Double-click to export	[clk]			
		s1	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x0400_2200	0x0400_220f	
		external_connection	Conduit	motores				
		PWM1	PID (Parallel I/O)		sys_clk			
		clk	Clock Input	Double-click to export	[clk]			
		reset	Reset Input	Double-click to export	[clk]			
		s1	Avalon Memory Mapped Slave	Double-click to export	[clk]	0x0400_21f0	0x0400_21ff	
		external_connection	Conduit	pwm1				
		PWM2	PID (Parallel I/O)		sys_clk	0x0400_21e0	0x0400_21ef	
		ENCODER_READING	QUADRATURE_ENCODER_IP		[clock_smk]	0x0400_2280	0x0400_2283	
		avalon_slave_0	Avalon Memory Mapped Slave	Double-click to export	[clock_smk]			
		clock_smk	Clock Input	Double-click to export	[clock_smk]			
		reset_smk	Reset Input	Double-click to export	[clock_smk]			
		conduit_end	Conduit	encoder_inputs				
		GPS	UART (RS-232 Serial Port)		sys_clk			
		clk	Clock Input	Double-click to export	[clk]			
		reset	Reset Input	Double-click to export	[clk]			

Figura 5. Implementación en Qsys de la arquitectura diseñada.



4. DISCUSIÓN

La implementación de la arquitectura SoPC en el robot móvil basado en FPGA demostró un comportamiento conforme a las expectativas, facilitando la interacción con sensores, actuadores y comunicación. Además, se logró realizar tareas de procesamiento de datos, localización y control en

el procesador mediante técnicas de codiseño H/S. La Figura 7 muestra una representación visual del módulo IP para la decodificación de encoders de cuadratura para determinar la posición del robot por medio de software programado en el procesador soft-core Nios II. Este enfoque permite aprovechar las capacidades de procesamiento tanto del hardware como del software de manera integrada.

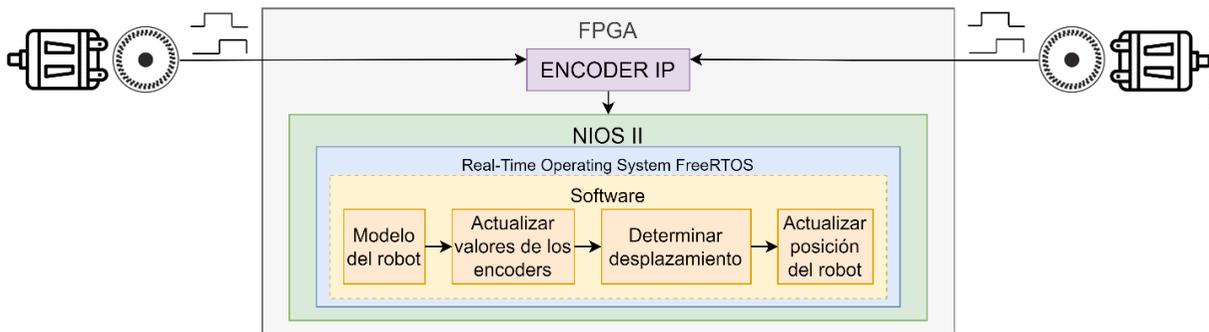


Figura 7. Representación del codiseño H/S para determinar la posición del robot.

Flow Summary	
Flow Status	Successful - Fri Sep 29 09:08:08 2023
Quartus II 64-Bit Version	13.1.4 Build 182 03/12/2014 SJ Web Edition
Revision Name	ROB3D
Top-level Entity Name	ROB3D
Family	Cyclone IV E
Device	EP4CE22F17C6
Timing Models	Final
Total logic elements	7,408 / 22,320 (33 %)
Total combinational functions	6,399 / 22,320 (29 %)
Dedicated logic registers	4,696 / 22,320 (21 %)
Total registers	4765
Total pins	154 / 154 (100 %)
Total virtual pins	0
Total memory bits	73,472 / 608,256 (12 %)
Embedded Multiplier 9-bit elements	4 / 132 (3 %)
Total PLLs	1 / 4 (25 %)

Figura 8. Resumen de los elementos usados en la FPGA del robot móvil.

En términos de recursos utilizados en la FPGA, la arquitectura ocupa porcentajes específicos de área lógica, memoria, multiplicadores embebidos y PLLs. Con un 33 % de área lógica ocupada, el diseño demuestra eficiencia en términos de espacio. La asignación del 25

% de los PLLs indica una gestión adecuada del reloj, mientras que el 12 % de la memoria y el 3 % de los multiplicadores embebidos sugieren un uso equilibrado de estos recursos esenciales. En la Figura 8 se muestra el resumen de los recursos utilizados en la FPGA.

Un aspecto destacado es que existe un espacio considerable sin utilizar en la FPGA, con un 67 % de área lógica disponible. Esto abre la posibilidad de ampliar la arquitectura en el futuro y diseñar módulos IP adicionales con co-diseño H/S. Esta capacidad de expansión permite descargar tareas del procesador Nios II hacia el hardware de la tarjeta de desarrollo, optimizando así el rendimiento del sistema y preparándolo para posibles mejoras. Además, se tiene la posibilidad de adaptarlo a diferentes aplicaciones que pueden ser más complejas o requerir una mayor capacidad de procesamiento de datos y control más avanzado.

5. CONCLUSIONES

Las FPGAs emergen como dispositivos con ventajas significativas en comparación con CPUs y GPUs, marcando un avance lógico para lograr sistemas robóticos con un buen rendimiento computacional y un consumo de energía eficiente. En este artículo, se presentó el diseño e implementación de una arquitectura SoPC para un robot móvil basado en FPGA, aprovechando técnicas avanzadas de codiseño H/S y aceleración de hardware para la interacción con sensores y actuadores, así como el procesamiento de datos y control. La singularidad de la arquitectura propuesta es evidente en el contexto actual de la robótica, donde prevalecen soluciones basadas exclusivamente en la programación de software. Al optar por un enfoque que incorpora FPGAs, se logra una flexibilidad que permite la reconfiguración de la arquitectura y la adición de módulos IP de manera adaptable a diversas aplicaciones. La capacidad de reconfiguración y la flexibilidad para integrar módulos IP específicos según las necesidades de la aplicación añaden una capa adicional de adaptabilidad a la arquitectura, destacando su versatilidad en entornos dinámicos y cambiantes.

REFERENCIAS

Al-Khalidy, M. M. M., Ebrahim, R. N., & Al-Khalidy, A. (2020). Artificial intelligent and FPGAs for omnidirectional 3-wheeled mobile robot. *3rd Smart Cities Symposium (SCS 2020)*, 2020, 124–129. <https://doi.org/10.1049/icp.2021.0821>

- Chand, R., Chand, R. P., Assaf, M., Naicker, P. R., Narayan, S. V., & Hussain, A. F. (2022). Embedded FPGA-based Motion Planning and Control of a Dual-arm Car-like Robot. *2022 IEEE 7th Southern Power Electronics Conference (SPEC)*, 1–6. <https://doi.org/10.1109/SPEC55080.2022.10058252>
- Correa-Caicedo, P. J., Barranco-Gutiérrez, A. I., Guerra-Hernandez, E. I., Batres-Mendoza, P., Padilla-Medina, J. A., & Rostro-González, H. (2021). An FPGA-based architecture for a latitude and longitude correction in autonomous navigation tasks. *Measurement*, 182, 109757. <https://doi.org/https://doi.org/10.1016/j.measurement.2021.109757>
- Divya Vani, G., Rao, K. S., & Chinnaiah, M. C. (2022). Self-Automated Parking with FPGA-Based Robot. En T. R. Lenka, D. Misra, & A. Biswas (Eds.), *Micro and Nanoelectronics Devices, Circuits and Systems: Select Proceedings of MNDCS 2021* (pp. 459–470). Springer Singapore. https://doi.org/10.1007/978-981-16-3767-4_45
- Gu, M., Guo, K., Wang, W., Wang, Y., & Yang, H. (2015). An FPGA-based real-time simultaneous localization and mapping system. *2015 International Conference on Field Programmable Technology (FPT)*, 200–203.
- Kojima, A. (2022). Implementation and Improvement of Autonomous Robot Car using SoC FPGA with DPU. *2022 International Conference on Field-Programmable Technology (ICFPT)*, 1–4. <https://doi.org/10.1109/ICFPT56656.2022.9974526>
- Liu, S., Wan, Z., Yu, B., & Wang, Y. (2021). *Robotic computing on fpgas*. Springer.
- Miyagi, R., Takagi, N., Kinoshita, S., Oda, M., & Takase, H. (2021). Zytobot: FPGA integrated ros-based autonomous mobile robot. *2021 International Conference on Field-Programmable Technology (ICFPT)*, 1–4. <https://doi.org/10.1109/ICFPT52863.2021.9609883>
- Nilay, K., Mandal, S., Agarwal, Y., Gupta, R., Patel, M., Kumar, S., Shah, P., Dey, S., & Annanya. (2020). A Proposal of FPGA-Based Low Cost and Power Efficient Autonomous Fruit Harvester. *2020 6th International Conference on Control, Automation and Robotics (ICCAR)*, 324–329. <https://doi.org/10.1109/ICCAR49639.2020.9108079>
- Plancher, B., Neuman, S. M., Bourgeat, T., Kuindersma, S., Devadas, S., & Reddi, V. J. (2021). Accelerating Robot Dynamics Gradients on a CPU, GPU, and FPGA. *IEEE Robotics and Automation Letters*, 6(2), 2335–2342. <https://doi.org/10.1109/LRA.2021.3057845>
- Podlubne, A., & Göhringer, D. (2022). Modeling FPGA-based Architectures for Robotics. *2022 International Conference on Field-Programmable Technology (ICFPT)*, 1–4. <https://doi.org/10.1109/ICFPT56656.2022.9974412>
- Shi, X., Cao, L., Wang, D., Liu, L., You, G., Liu, S., & Wang, C. (2018). HERO: Accelerating Autonomous Robotic Tasks with FPGA. *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 7766–7772.
- Suárez-Gómez, A. D., & Pérez-Holguín, W. J. (2020). Módulo IP basado en FPGA para la decodificación de encoders de

- cuadratura. *Revista Politécnica*, 16(32), 68–76. <https://doi.org/10.33571/rpolitec.v16n32a6>
- Suárez-Gómez, A.-D., & Pérez-Holguín, W.-J. (2022). Sistema de odometría basado en codiseño H/S para un robot móvil diferencial. *IV Congreso Internacional de Ciencias Básicas e Ingeniería CICI2022*, 1–4.
- Sudhakar, K., Pirajin, S. S., Shanmugapriyan, J., & Sujeeth, S. (2023). Design and Implementation of FPGA based Rescue Bot. *2023 7th International Conference on Computing Methodologies and Communication (ICCMC)*, 1636–1643. <https://doi.org/10.1109/ICCMC56507.2023.10083599>
- Sun, H., Zhang, Y., Wu, Z., & Xue, J. (2014). The detecting robot based on SOPC. *2014 IEEE International Conference on Mechatronics and Automation*, 1935–1939. <https://doi.org/10.1109/ICMA.2014.6885998>
- Tsai, C.-C., Tai, F.-C., & Hsieh, S.-M. (2010). Unified motion controller design and FPGA-based implementation for nonholonomic mobile robots. *Proceedings of SICE Annual Conference 2010*, 2383–2389.
- Wan, Z., Lele, A., Yu, B., Liu, S., Wang, Y., Reddi, V. J., Hao, C., & Raychowdhury, A. (2022). Robotic Computing on FPGAs: Current Progress, Research Challenges, and Opportunities. *2022 IEEE 4th International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, 291–295. <https://doi.org/10.1109/AICAS54282.2022.9869951>
- Wan, Z., Yu, B., Li, T. Y., Tang, J., Zhu, Y., Wang, Y., Raychowdhury, A., & Liu, S. (2021). A Survey of FPGA-Based Robotic Computing. *IEEE Circuits and Systems Magazine*, 21(2), 48–74. <https://doi.org/10.1109/MCAS.2021.3071609>
- Wu, Y., Luo, L., Yin, S., Yu, M., Qiao, F., Huang, H., Shi, X., Wei, Q., & Liu, X. (2021). An FPGA Based Energy Efficient DS-SLAM Accelerator for Mobile Robots in Dynamic Environment. *Applied Sciences*, 11(4). <https://doi.org/10.3390/app11041828>