

UNA BREVE REVISIÓN DEL *HARDWARE* EVOLUTIVO

Federico Fernández y Diego Pinto
Facultad Politécnica – Universidad Nacional de Asunción
Campus Universitario San Lorenzo, Paraguay

RESUMEN

En todos los ámbitos tecnológicos hay una evolución vertiginosa hacia la inteligencia artificial implicando esto la necesidad de hardware más veloz, paralelismo, reconfiguración del hardware, ejecución de algoritmos evolutivos, optimización, multiprocesamiento, ahorro de consumo de energía, etc. Los microprocesadores y microcontroladores clásicos han respondido hasta hace poco tiempo a estas exigencias, pero la aparición de las FPGAs han supuesto una evolución notable desde el punto de vista del diseño de circuitos electrónicos ya que en estos dispositivos lo que se programa es hardware. Por tanto a diferencia de los circuitos clásicos se puede realizar un diseño a medida de las necesidades y volver a diseñar para otra aplicación o su optimización. Pero esta ventaja ha tenido una evolución aún más grande con la aparición de la reconfiguración parcial dinámica, lo que ha supuesto un salto notable en las tecnologías relacionadas al mantenimiento y corrección de circuitos que no pueden ser accedidos fácilmente ya sea por su ubicación geográfica, por estar en una zona peligrosa o de difícil acceso, etc. La aparición del hardware evolutivo es un paso más en ya que además de la utilización de las capacidades de la reconfiguración parcial dinámica propios de las FPGAs, se utilizan algoritmos de inteligencia artificial para facilitar la búsqueda de diseños electrónicos óptimos según especificaciones requeridas de circuitos electrónicos o para adaptarse a cambios en el entorno del mismo por lo que su funcionalidad debe adaptarse a ello. En este trabajo presentamos una breve revisión de los conceptos asociados con el hardware reconfigurable y el hardware evolutivo, para finalmente describir algunas aplicaciones que se están implementando en la actualidad con esta tecnología.

PALABRAS CLAVES

Hardware Evolutivo, Reconfiguración, FPGA, Inteligencia Artificial

1. INTRODUCCIÓN

Dentro de las tecnologías que más han evolucionado en las últimas décadas se destacan las relacionadas a los dispositivos reconfigurables o FPGAs (Field Programmable Gate Array), por presentar ventajas que carecen otras tecnologías utilizadas en los sistemas digitales modernos como la disponibilidad de herramientas para reducir los tiempos de diseño, prueba y depuración del circuito final. También ofrece la ventaja que se puede modificar un diseño realizando los cambios sobre el circuito final debido a que las FPGA lo que se programa no es software sino hardware, pudiendo el circuito final ser objeto de una optimización constante a medida que pasa el tiempo. Esta capacidad de modificar un diseño dentro de una FPGA se denomina reconfiguración pudiendo ser, según el modelo, estáticas o dinámicas. El hardware evolutivo es un tipo particular de hardware cuya arquitectura, estructura y funciones cambia dinámicamente y autónomamente para mejorar el rendimiento de un circuito electrónico. La aparición de este nuevo campo ha sido influenciada por el progreso en el hardware reconfigurable y la computación evolutiva. En la computación evolutiva se simula la evolución natural para resolver problemas complejos de optimización o clasificación. El hardware evolutivo es un tipo particular de hardware cuya arquitectura, estructura y funciones cambia dinámicamente y autónomamente para mejorar el rendimiento de un circuito electrónico. La aparición de este nuevo campo ha sido influenciada por el progreso en el hardware reconfigurable y la computación evolutiva. Este trabajo está organizado de la siguiente forma: En la sección II se hace una breve descripción del hardware reconfigurable. La sección III describe los conceptos relacionados al hardware evolutivo. En la Sección IV se describe las aplicaciones del hardware evolutivo y en la sección V las conclusiones.

2. *HARDWARE RECONFIGURABLE*

Entre las tecnologías que más han evolucionado en las últimas décadas se destacan las relacionadas a los dispositivos reconfigurables o FPGAs, siendo estas un conjunto de recursos hardware más una memoria de configuración que determina cómo se conectan entre sí los diferentes recursos para ejecutar una tarea. Algunos modelos incorporan la posibilidad de configurar uno o más procesadores empotrados aumentando aún más sus posibilidades de diseño y potencia de cálculo. Estos dispositivos presentan ventajas de las que carecen otras tecnologías utilizadas en los sistemas digitales modernos, entre ellas se pueden mencionar la posibilidad de realizar diseños a medida en poco tiempo, la realización de pruebas de depuración y control de errores gracias a las herramientas que cuentan los entornos de desarrollo, como asimismo la posibilidad de realizar ensayos de simulación en la fases de prueba y depuración de un diseño complejo brindando ventajas en el tiempo de desarrollo final previa a la fase de producción masiva (Boutros, A., 2021) (Mencer, O. et al, 2020) (Sakulin, H., 2022) (AMD, 2023). Esta tecnología ha evolucionado un paso más con respecto a otras, ya que es la única que brinda la posibilidad de poder ser reconfiguradas tanto estática como dinámicamente posibilitando que un diseño pueda ser modificado. Posteriormente aparecieron nuevos modelos que permiten modificar una configuración determinada sin necesidad de borrar todo el diseño original, permitiendo modificar solamente aquellos módulos de interés sin la necesidad de detener el funcionamiento de todo el sistema. A este procedimiento se le denomina reconfiguración parcial dinámica. En este tipo de reconfiguración se presentan problemas asociados como la fragmentación y el retardo que esto produce en el tiempo de finalización en la ejecución de un diseño que tiene numerosas tareas que se van cargando y liberando de la FPGA. Estos problemas van aumentando a medida que el diseño crece en complejidad. En ese sentido la reconfiguración dinámica es aun más ventajosa que la estática, debido a que permite modificar una o varias partes de un modulo en funcionamiento sin necesidad de interrumpir el funcionamiento del proceso brindándole robustez al diseño (Aithal, M., 2016) (Carrow, M., 2022) (Siva, S., 2021) (Vipin, K., 2019).

3. *HARDWARE EVOLUTIVO*

El hardware evolutivo es un paso adelante en el desarrollo tecnológico que promete mejoras todavía más sofisticadas con la utilización de la reconfiguración parcial dinámica cuya arquitectura, estructura y funciones pueden cambiar de manera dinámica y autónoma para mejorar su rendimiento en tareas específicas, aumentando la capacidad de los circuitos electrónicos para evolucionar y adaptarse a través de algoritmos evolutivos, en lugar de ser diseñados de manera convencional a partir de especificaciones fijas. Este enfoque permite la optimización y adaptación a entornos físicos reales, siendo este campo influenciado por los avances en el hardware reconfigurable y computación evolutiva. Esto se consigue mediante la aplicación de técnicas de computación evolutiva en el diseño de hardware aprovechando que el mismo puede adaptarse en línea mediante la reconfiguración de su arquitectura, incluyendo la evolución simulada, que puede ser impulsada por algoritmos genéticos y el hardware electrónico, que puede ser digital, analógico o híbrido siendo clave el papel que juegan los circuitos reconfigurables. El proceso comienza con la generación de una población inicial de bits de arquitectura, que se evalúan en FPGAs. Se utilizan operaciones genéticas como el cruce y la mutación para generar nuevos cromosomas que reemplazan a los antiguos, formando nuevas generaciones.

El uso del hardware evolutivo puede llevar a circuitos más eficientes y adaptativos, capaces de ajustarse a diferentes condiciones operativas, siendo este factor relevante en aplicaciones donde la flexibilidad y la adaptabilidad son muy importantes. A pesar de sus promesas iniciales, el hardware evolutivo presenta varios desafíos. Uno de los principales es la dificultad para definir funciones de fitness adecuadas y criterios de parada en el proceso evolutivo. Además, la mantenibilidad y comprensibilidad de los circuitos evolucionados son problemáticas, ya que a menudo se comportan como cajas negras, difíciles de entender. Uno de los campos más prometedores es la auto-reparación de hardware, donde los sistemas evolutivos puedan detectar y corregir sus propios fallos de manera dinámica (Yao, X., 1999) (Yao, X., 1999).

4. APLICACIONES DEL *HARDWARE* EVOLUTIVO

En (Zhang, J.B., 2020) se presenta una técnica innovadora de diseño de circuitos autoadaptativos que utiliza hardware evolutivo para mejorar la tolerancia a fallos en sistemas electrónicos. Los métodos tradicionales de tolerancia a fallos en sistemas electrónicos suelen basarse en técnicas de redundancia, lo que limita su adaptabilidad debido a estructuras fijas en el diseño de circuitos pudiendo resultar en fallos en sistemas complejos. Se propone una técnica de diseño de circuitos que permite la autoorganización y autoadaptación dinámica en respuesta a cambios en el entorno y a los fallos del circuito. Esto se logra mediante la configuración dinámica de los mismos. El circuito tiene la capacidad de resistir interferencias en estados redundantes (RRSI), lo que es fundamental para el diseño autoadaptativo. Se investigan factores como el consumo de recursos de hardware y el número de iteraciones de convergencia. La efectividad y superioridad de la técnica propuesta se verifican a través de simulaciones, mostrando que puede mejorar la eficiencia del diseño de sistemas electrónicos y facilitar la adaptación a condiciones cambiantes. La técnica tiene amplias perspectivas de aplicación en sistemas electrónicos, especialmente en entornos donde los fallos son comunes debido a condiciones adversas. En conclusión, el estudio destaca la importancia del desarrollo de circuitos que puedan adaptarse a cambios en el entorno y responder a los fallos, lo que representa un avance significativo en el diseño de sistemas electrónicos más robustos y eficientes.

En (Borrett, F., 2023) se presenta un estudio que compara dos enfoques para controlar el movimiento de un robot hexápodo: un controlador de hardware evolutivo y una red neuronal artificial, evaluando la eficacia de un controlador de hardware evolutivo en comparación con una red neuronal artificial para la evolución del patrón de marcha de un robot hexápodo. Se diseñó un sistema de hardware evolutivo y se implementó en un robot hexápodo. Se utilizaron señales de entrada que incluían información sobre la fase de la pierna (en el suelo o en el aire) y un contador de marcha, verificándose que ambos controladores lograron resultados similares en términos de rendimiento. Sin embargo, el controlador de hardware evolutivo mostró una mayor eficiencia evolutiva, lo que significa que requirió menos generaciones para alcanzar un rendimiento óptimo. Como resultado se considera que el hardware evolutivo puede ser una alternativa viable y más eficiente en comparación con las redes neuronales para el control de robots en aplicaciones del mundo real.

En (Manjith, B. C., 2019) se presenta un sistema basado en hardware evolutivo para optimizar el algoritmo AES (Advanced Encryption Standard), centrándose en la evolución de las operaciones básicas del AES, que incluyen Substitute Byte, Shift Rows, Mix Column y Add Round Key, utilizando un algoritmo genético. Este enfoque permite crear nuevos algoritmos AES a partir de unidades básicas sin intervención humana. El sistema propuesto se basa en la evolución a nivel funcional, donde las operaciones del AES se consideran unidades reconfigurables virtuales. Se aplican funciones de fitness que evalúan la cofusión, difusión y aleatoriedad de los algoritmos generados. Los resultados muestran que los nuevos algoritmos evolucionados son más eficientes, con un 37% de mejora en comparación con implementaciones anteriores, y logran una alta tasa de éxito en pruebas de aleatoriedad, superando el 98%. En (Yao, R., et al, 2023) se presenta una metodología de diseño para sistemas adaptativos basados en la evolución directa de bloques de configuración (CB) en FPGAs. Se enfoca en la co-evolución de recursos lógicos y de enrutamiento mediante la manipulación en línea de los CBs, permitiendo reducir la dependencia de herramientas de diseño externas y disminuir el consumo de recursos, tiempo y energía del sistema. El enfoque propuesto mejora la adaptabilidad del sistema modelando la relación entre los recursos de FPGA y los CBs. Se discuten las estructuras de recursos de las FPGAs, incluyendo bloques lógicos configurables (CLBs), matrices de conmutación (SMs) y bloques de entrada/salida programables (IOBs). Se menciona que el tamaño del área de evolución y la memoria RAM de bloques pueden personalizarse según la funcionalidad a implementar, utilizando programación genética cartesiana para una conectividad más flexible.

En (Zhu, M., et al, 2020) se discute la integración de algoritmos de aprendizaje por refuerzo (RL) con algoritmos genéticos (GA) para mejorar las operaciones de hardware evolutivo, destacándose que tiene la capacidad de adaptarse y reestructurarse de manera autónoma, factor de mucha utilidad en entornos complejos como la exploración espacial y submarina, donde la intervención humana es limitada. El estudio se centra en cómo el algoritmo RLGA (Reinforcement Learning Genetic Algorithm) puede mantener la diversidad de la población durante el proceso evolutivo, evitando la convergencia prematura que a menudo se observa en los algoritmos genéticos tradicionales. Los resultados experimentales muestran que el RLGA tiene una tasa de éxito y velocidad de convergencia significativamente mejores en comparación con el GA, lo que evidencia que la combinación de aprendizaje por refuerzo con algoritmos genéticos puede ser una estrategia efectiva para el

hardware evolutivo. En (Mora, J., 2019) se aborda la escalabilidad de las arquitecturas del hardware evolutivo mediante la comparación de dos topologías: arreglos sistólicos (SA) y programación genética cartesiana (CGP). Se determinan algunos puntos clave como que el hardware evolutivo permite generar circuitos adaptados a problemas específicos utilizando algoritmos evolutivos, y que la reconfiguración parcial dinámica de las LUTs de las FPGAs facilita la implementación de circuitos a gran escala en áreas reducidas. También se analizan las ventajas y desventajas que las arquitecturas SA y CGP presentan en términos de escalabilidad y uso de recursos. Se destaca que los arreglos sistólicos son más eficientes en el uso de recursos en comparación con la programación genética cartesiana. Se presentan sistemas de hardware evolutivo basados en ambas arquitecturas, evaluando su rendimiento y comparando los resultados obtenidos, sugiriéndose una modificación en la arquitectura SA para mejorar su rendimiento en tamaños grandes. En general, el artículo proporciona una visión integral sobre cómo las arquitecturas del hardware evolutivo pueden ser escaladas y optimizadas para su uso en sistemas embebidos con recursos limitados.

En (Tetteh, M., 2022) se aborda la evolución de circuitos digitales complejos utilizando la Evolución Gramatical (GE) en el contexto de SystemVerilog. Los autores identifican varios desafíos en la evolución de circuitos, como la necesidad de pruebas extensivas y las limitaciones de las estructuras de representación actuales. El enfoque propuesto utiliza GE para generar circuitos funcionales, como sumadores y multiplicadores, a un nivel de abstracción más alto permitiendo una mayor flexibilidad y adaptabilidad en el diseño de circuitos, facilitando la evolución de módulos parametrizados que pueden ajustarse a diferentes tamaños de entrada. El estudio demuestra que este método no solo mejora la eficiencia del proceso de diseño, sino que también abre nuevas posibilidades para la creación de circuitos más complejos y optimizados. En (Kojima, K., 2022) se presenta un enfoque innovador para la generación automática de descripciones en VHDL aplicadas a sistemas mecatrónicos. Se utilizan técnicas de computación evolutiva para diseñar controladores que optimicen el rendimiento de estos sistemas. El estudio destaca cómo la computación evolutiva puede facilitar el diseño de circuitos de control, permitiendo una mayor eficiencia y adaptabilidad en la implementación de soluciones mecatrónicas. A través de este enfoque, se busca mejorar la automatización en el diseño y la implementación de sistemas complejos, impactando positivamente en diversas aplicaciones industriales y tecnológicas.

En (Hoffmann, J., 2022) el hardware evolutivo se utiliza para aplicar algoritmos evolutivos al hardware con el fin de diseñar, mejorar o adaptar circuitos. Los enfoques que manipulan directamente el bitstream de las FPGAs se abandonaron debido a la falta de formatos de bitstream bien documentados. Los avances recientes en las cadenas de herramientas de FPGA de código abierto cambiaron fundamentalmente una vez más la viabilidad de la manipulación directa del flujo de bits. Sin embargo, las herramientas actuales son lentas y hacen perder un tiempo valioso llamando a herramientas externas. Se presenta un enfoque integrado que combina manipulación de bitstream, comunicación de bajo nivel y evaluación de hardware en un único marco llamado CoBEA. Además, el framework permite la compactación del flujo de bits y la configuración directa del dispositivo FPGA sin tener que programar la memoria flash. Esta propuesta logra una aceleración de 130 veces para la reconfiguración de FPGA permitiendo realizar experimentos complejos de hardware evolutivo. En (Mora, J., 2018) se describen múltiples optimizaciones aplicadas a un sistema de hardware evolutivo con el objetivo de acelerar su evolución. Se destacan mejoras en el hardware, algoritmos evolutivos (EA) y técnicas de paralelización, logrando un aumento total de la velocidad de evolución de 280 veces, reduciendo el tiempo de evolución de 480 segundos a solo 1.73 segundos, y alcanzando velocidades de 139,000 evaluaciones por segundo.

La reducción del tiempo de evolución proviene de la mejora del motor de reconfiguración y la optimización de la matriz sistólica (SA) mencionándose que la optimización del EA también proporciona un aumento significativo en la velocidad, inclusive la paralelización del sistema contribuye a la mejora del rendimiento. El artículo concluye que aunque alcanzar una alta velocidad de evaluación es crucial para reducir los tiempos de evolución, un EA eficiente también puede resultar en un gran aumento de velocidad sin necesidad de cambios en el hardware, sugiriendo que se debe considerar un análisis más fino de los parámetros del EA.

5. CONCLUSIONES

En este artículo se revisaron trabajos relacionados al hardware evolutivo, las características del hardware reconfigurable y algunas de las aplicaciones más recientes del hardware evolutivo basado en reconfiguración

parcial dinámica. El futuro se presenta prometedor por los crecientes desarrollos tecnológicos que permitirán su implementación y la demanda de los sistemas adaptables. La aparición de nuevos modelos de FPGAs que incluyen procesadores empotrados y la posibilidad de agregarle procesadores soft le brindan a estos sistemas un mayor poder de cálculo y paralelismo, fundamentales para los algoritmos utilizados en el hardware evolutivo. A medida que las técnicas de inteligencia artificial continúan evolucionando su integración con el hardware evolutivo podría dar lugar a sistemas más inteligentes y adaptativos, ejecutar mejores algoritmos para la computación evolutiva que pueden optimizar los diseños de hardware de manera más eficiente y efectiva, ampliando sus aplicaciones a otros campos.

REFERENCIAS

- Aithal, M. S. and Dr. Ambedkar, (2016). Institute of Technology. A Survey on Partial Reconfiguration Techniques. *International journal of engineering and computer science*. <https://doi.org/10.18535/ijecs/v5i5.47>
- AMD, (2023). *Programming an FPGA: An introduction to how it works*. LinkedIn.com. <https://www.linkedin.com/pulse/programming-fpga-introduction-how-works-ebics-fpga/>
- Borrett, F. and Beckerleg, M., (2023). A comparison of an evolvable hardware controller with an artificial neural network used for evolving the gait of a hexapod robot. *Genetic Programming and Evolvable Machines*, 24(1). <https://doi.org/10.1007/s10710-023-09452-4>
- Boutros, A. and Betz, V., (2021). FPGA Architecture: Principles and Progression. *IEEE circuits and systems magazine*, 21(2), 4–29. <https://doi.org/10.1109/mcas.2021.3071607>
- Carrow, M., (2022). FPGA Partial Reconfiguration. *Proposed for presentation at the Sandia SIP Symposium held July 27-28, 2022 in Albuquerque, NM United States of America*.
- Hoffmann, J., Fritzsche, C. and Bogdan, M., (2022). CoBEA: Framework for evolving hardware by direct manipulation of FPGA bitstreams. *Proceedings of the Genetic and Evolutionary Computation Conference Companion*.
- Kojima, K., (2022). Automatic VHDL description for mechatronics system by using evolutionary computation. *The 10th International Conference on Computer and Communications Management*.
- Manjith, B. C., (2019). Hardware evolution of AES algorithm on FPGA. *2019 Innovations in Power and Advanced Computing Technologies (i-PACT)*.
- Mencer, O. et al, (2020). The History, Status, and Future of FPGAs: Hitting a nerve with field-programmable gate arrays. *ACM Queue: Tomorrow's Computing Today*, 18(3), 71–82. <https://doi.org/10.1145/3411757.3411759>
- Mora, J. and de la Torre, E., (2018). Accelerating the evolution of a systolic array-based evolvable hardware system. *Microprocessors and Microsystems*, 56, 144–156. <https://doi.org/10.1016/j.micpro.2017.12.001>
- Mora, J., Salvador, R. and de la Torre, E., (2019). On the scalability of evolvable hardware architectures: comparison of systolic array and Cartesian genetic programming. *Genetic Programming and Evolvable Machines*, 20(2), 155–186. <https://doi.org/10.1007/s10710-018-9340-5>
- Sakulin, H., (2022). *Introduction to Field Programmable Gate Arrays. CERN/EP-CMD. 12th International School of Trigger and Data Acquisition (ISOTDAQ)*. Catania, Italy.
- Siva, S., et al, (2021). Partial dynamic reconfiguration framework for FPGA: A survey with concepts, constraints and trends. *Materials Today: Proceedings*, 46, 3704–3711. <https://doi.org/10.1016/j.matpr.2021.01.851>
- Tetteh, M., Dias, D. M. and Ryan, C., (2022). Grammatical Evolution of complex digital circuits in SystemVerilog. *SN Computer Science*, 3(3), 188. <https://doi.org/10.1007/s42979-022-01045-9>
- Vipin, K. and Fahmy, S. A., (2019). FPGA dynamic and partial reconfiguration: A survey of architectures, methods, and applications. *ACM Computing Surveys*, 51(4), 1–39. <https://doi.org/10.1145/3193827>
- Yao, R., et al, (2023). Efficient design methodology for adaptive system based on direct bitstream evolution. *IEICE Electronics Express*, 20(13), 20220518–20220518. <https://doi.org/10.1587/elex.20.20220518>
- Yao, X. and Higuchi, T., (1999). Promises and challenges of evolvable hardware. *IEEE transactions on systems, man and cybernetics. Part C, Applications and reviews: a publication of the IEEE Systems, Man, and Cybernetics Society*, 29(1), 87–97. <https://doi.org/10.1109/5326.740672>
- Yao, X., (1999). Following the path of evolvable hardware. *Communications of the ACM*, 42(4), 46–49. <https://doi.org/10.1145/299157.299169>
- Zhang, J.B., et al, (2020). A novel self-adaptive Circuit design technique based on evolvable hardware. *International Journal of Automation and Computing*, 17(5), 744–751. <https://doi.org/10.1007/s11633-016-1000-8>
- Zhu, M., et al, (2020). Research on RLGA-based hardware evolution optimization technology. *2020 15th IEEE Conference on Industrial Electronics and Applications (ICIEA)*.