

NIOS II: Un procesador embebido para el desarrollo de sistemas en *Field-Programmable Gate Arrays*¹

NIOS II: An embedded processor for developing systems in Field-Programmable Gate Arrays

Página | 1

NIOS II: Um processador incorporado para o desenvolvimento de sistemas em Field-Programmable Gate Arrays

Pedro Selencio Landaeta Herrera*²

Universidad Politécnica Territorial del Estado Aragua “Federico Brito Figueroa”, Venezuela*

Fecha de Recepción: 16-06-2025 Fecha de Aceptación: 11-11-2025

Autor de correspondencia: Pedro Selencio Landaeta Herrera, landaetabiz@gmail.com

Cómo citar:

Landaeta H., P. S. (2025). NIOS II: Un procesador embebido para el desarrollo de sistemas en *Field-Programmable Gate Arrays*. *Revista Científica Cuadernos de Investigación*, 3, e56, 1-15. <https://doi.org/10.59758/rcci.2025.3.e56>

Resumen

Este ensayo analizó el papel del microprocesador embebido NIOS II, desarrollado por Intel, como una alternativa estratégica dentro de la computación reconfigurable. Se abordó su capacidad para posibilitar una digitalización personalizada en plataformas de arreglos de compuertas, lo que favoreció el avance tecnológico en el ámbito de los programas informáticos. El estudio se sustentó en una revisión documental de fuentes académicas y técnicas, complementada con un análisis interpretativo de los fundamentos del procesador, su arquitectura interna y las herramientas empleadas en el procesamiento de sistemas embebidos. Los hallazgos mostraron que la aplicación del NIOS II facilitó la comprensión de conceptos complejos de arquitectura de hardware y software, al tiempo que promovió la innovación y el aprendizaje activo. Asimismo, su incorporación en contextos educativos contribuyó a reducir la distancia entre teoría y práctica, fortaleciendo la transferencia tecnológica y el desarrollo de competencias en diseño digital. Finalmente, la evolución hacia el NIOS V, basado en una arquitectura abierta, amplió las perspectivas de investigación e innovación tanto en entornos académicos como industriales.

Palabras clave: NIOS II; programa informático; digitalización; microprocesador; procesamiento.

¹ Copyright: © 2025, Landaeta. Este es un artículo de acceso abierto distribuido bajo la licencia **Creative Commons de Atribución No Comercial 4.0**. Permite su uso sin restricciones, su distribución y reproducción por cualquier medio, siempre que no se haga con fines comerciales y el trabajo original sea fielmente citado.

² <https://orcid.org/0000-0002-8843-4991>

Abstract

This essay analyzed the role of the embedded microprocessor NIOS II, developed by Intel, as a strategic alternative within reconfigurable computing. Its capacity to enable customized digitalization on gate array platforms was addressed, highlighting how it fostered technological progress in the field of computer programs. The study was based on a documentary review of academic and technical sources, complemented by an interpretative analysis of the processor's fundamentals, its internal architecture, and the tools used for embedded system processing. The findings showed that the application of NIOS II facilitated the understanding of complex concepts in hardware and software architecture while promoting innovation and active learning. Likewise, its incorporation in educational contexts helped to bridge the gap between theory and practice, strengthening technological transfer and the development of competencies in digital design. Finally, the evolution toward NIOS V, based on an open architecture, broadened the perspectives of research and innovation in both academic and industrial environments.

Página | 2

Keywords: NIOS II; computer program; digitization; microprocessor; processing.

Resumo

Este ensaio analisou o papel do microprocessador embutido NIOS II, desenvolvido pela Intel, como uma alternativa estratégica dentro da computação reconfigurável. Abordou-se sua capacidade de possibilitar uma digitalização personalizada em plataformas de matrizes de portas lógicas, o que favoreceu o avanço tecnológico no campo dos programas informáticos. O estudo baseou-se em uma revisão documental de fontes acadêmicas e técnicas, complementada por uma análise interpretativa dos fundamentos do processador, de sua arquitetura interna e das ferramentas utilizadas no processamento de sistemas embarcados. Os resultados mostraram que a aplicação do NIOS II facilitou a compreensão de conceitos complexos de arquitetura de hardware e software, ao mesmo tempo em que promoveu a inovação e a aprendizagem ativa. Da mesma forma, sua incorporação em contextos educacionais contribuiu para reduzir a distância entre teoria e prática, fortalecendo a transferência tecnológica e o desenvolvimento de competências em design digital. Finalmente, a evolução para o NIOS V, baseado em uma arquitetura aberta, ampliou as perspectivas de pesquisa e inovação tanto em ambientes acadêmicos quanto industriais.

Palavras Chave: NIOS II; programa informático; digitalização; microprocessador; processamento.

Introducción

En el contexto actual del diseño digital, las FPGAs (*Field-Programmable Gate Arrays*) se han consolidado como herramientas clave para el desarrollo de sistemas embebidos flexibles, eficientes y altamente personalizables. Este ensayo tiene como objetivo principal introducir al lector en el uso del procesador embebido NIOS II, explorando sus fundamentos técnicos, su arquitectura interna y las herramientas necesarias para su implementación.

A diferencia de los microcontroladores tradicionales o sistemas con procesadores dedicados, las FPGAs permiten a los ingenieros definir la arquitectura del sistema directamente en hardware mediante lenguajes de descripción como VHDL (*Very High Speed Integrated Circuit Hardware Description Language*) o Verilog, lo que habilita el desarrollo de soluciones digitales completamente a medida. Xu et al. (2022), los autores advierten cómo la inteligencia artificial (IA) está transformando las fábricas tradicionales en entornos de producción personalizados y flexibles. Presentan tecnologías clave como el aprendizaje automático, sistemas multiagente, Internet de las Cosas (IoT), *big data* y computación en la nube/borde, y aunque no mencionan directamente a NIOS II, el enfoque del artículo encaja perfectamente con los principios de diseño embebido y reconfigurable de los *softcores* en FPGAs como NIOS II o NIOS V. A continuación, se presentan nombres de diferentes tecnologías donde se aplica el uso de FPGA y NIOS II, como por ejemplo Personalización de *hardware*, *Edge Computing*, IA y *Deep Learning*, Ontologías y sistemas multiagente, Redes Industriales y Sensado Inteligente.

Por otro lado, Huseynli et al. (2022) proponen un modelo de proceso integrado para la ingeniería de innovación digital, combinando la metodología de *Design Science Research* (DSR) con estrategias de innovación. La metodología permite la creación sistemática de artefactos de sistemas de información innovadores, facilitando su descripción, comparabilidad y reutilización. Como se advierte, el desarrollo de soluciones digitales completamente a medida, es un mundo que está en constante crecimiento.

En el caso del presente ensayo, la innovación y el aporte en lo digital va dirigido a microprocesadores [NIOS II – CPU - FPGA] con la meta de impactar en lo académico. La innovación y los aportes en lo digital no se apartan del NIOS II. Por ejemplo, Villamizar et al. (2024) desarrollaron un controlador digital completo para aplicaciones de calentamiento por inducción utilizando FPGA Altera Cyclone y procesador NIOS II, destacando su precisión y eficiencia en el control de potencia. Dos investigaciones que añaden el poder de uso del NIOS II en aplicaciones de alta tecnología son la de Santucci et al. (2022) que aplica el NIOS II para construir una Tarjeta de Adquisición de Datos (DAQ en inglés, *Data Acquisition System*) conectada a un controlador vía *Ethernet*, y la de Silva et al. (2014) y, quienes realizaron una experiencia educativa en programación de procesadores *multicore* utilizando plataformas didácticas embebidas en FPGA, mejorando la comprensión de la arquitectura de computadoras en entornos académicos.

Dentro de este ecosistema, el procesador embebido NIOS II representa una de las alternativas más potentes y accesibles para implementar sistemas de propósito general en plataformas reconfigurables. Se trata de un procesador *softcore*, es decir, una CPU que no está físicamente grabada en silicio, sino que se describe en código y se implementa dentro de la FPGA. Esta característica permite un nivel de personalización inigualable, facilitando la creación de sistemas digitales flexibles y personalizados sin necesidad de hardware adicional. Cao & Meyer-Baese (2022) hablan de este tema, y expresan que su relevancia académica radica en su contribución al diseño y prototipado rápido de sistemas embebidos, ofreciendo una herramienta valiosa para la enseñanza y la investigación en áreas relacionadas con la arquitectura de computadoras, sistemas digitales y diseño de hardware/software co-diseñado.

El NIOS II se integra perfectamente con las herramientas de desarrollo de Intel, particularmente Quartus II, a través de su módulo QSys (hoy *Platform Designer*), que permite el diseño visual del sistema embebido

mediante el ensamblaje modular de componentes: procesador, memoria, periféricos, buses y controladores. Una vez generado el hardware, su programación se realiza en lenguaje C/C++, utilizando el entorno Eclipse y los NIOS II *Software Build Tools*, lo que acerca su uso tanto a ingenieros digitales como a desarrolladores de software tradicionales. Faycal et al. (2021), presentan un sistema de segmentación de imágenes basado en FPGA utilizando el algoritmo ISODATA y el procesador NIOS II. El desarrollo del hardware se realizó con Quartus II, integrando el sistema NIOS II mediante SOPC *Builder* (QSys), y la programación del software se llevó a cabo en lenguaje C++ utilizando el entorno Eclipse y los NIOS II *Software Build Tools*. El sistema completo se implementó en una única FPGA, demostrando la eficacia de estas herramientas en el desarrollo de sistemas embebidos personalizados.

Finalmente, se abren perspectivas de futuro al considerar la transición hacia el procesador NIOS V, basado en la arquitectura RISC-V, y la potencialidad de construir sistemas *multicore* dentro de una misma FPGA, una línea de investigación prometedora que está ganando tracción en contextos académicos e industriales. Islam & Kise (2022), expresan que la relevancia académica de dicho trabajo radica en su contribución al campo de la investigación en sistemas embebidos y SoC, proporcionando una base sólida para la exploración de arquitecturas *multicore* y la adopción de estándares abiertos como RISC-V en entornos académicos e industriales.

A pesar del creciente interés en los procesadores *softcore* y en las arquitecturas reconfigurables, existe una brecha entre el desarrollo técnico del NIOS II y la disponibilidad de materiales introductorios que faciliten su aprendizaje y aplicación en entornos educativos. Aunque numerosos estudios abordan sus aspectos funcionales e industriales, son escasos los trabajos que presentan su uso de manera formativa y accesible para estudiantes o investigadores en formación.

A partir de esta perspectiva, la pregunta que orienta el trabajo es: ¿cómo puede el procesador embebido NIOS II contribuir al fortalecimiento de la enseñanza y la investigación aplicada en el ámbito académico del diseño digital reconfigurable?

En este sentido, el presente ensayo busca responder a esa necesidad, introduciendo al lector en el uso del procesador embebido NIOS II, un *softcore* desarrollado por Intel (antiguamente Altera) para plataformas FPGA, y mostrando cómo su implementación puede fortalecer la enseñanza y la investigación aplicada en el diseño de sistemas digitales reconfigurables.

Desarrollo

El desarrollo de este ensayo se estructura en varios apartados que abordan, de manera progresiva y analítica, los principales aspectos técnicos y formativos relacionados con el procesador embebido NIOS II. En primer lugar, se examina su utilización dentro del ámbito de los sistemas embebidos y su relevancia como alternativa reconfigurable frente a procesadores convencionales. Posteriormente, se analizan las posibilidades de diseño de sistemas digitales flexibles sin *hardware* adicional, así como la comparación con otros procesadores *softcore* desarrollados por distintas compañías. A continuación, se revisan diversas tesis y experiencias académicas que han empleado el NIOS II como herramienta de enseñanza e investigación. Asimismo, se describen los procesos técnicos asociados a su configuración

mediante QSys (*Platform Designer*) y la programación en C++ dentro del entorno Eclipse, complementando con una explicación de los fundamentos técnicos y la arquitectura interna del procesador. Finalmente, se presentan las perspectivas futuras vinculadas a la evolución hacia el NIOS V y las arquitecturas *multicore*, destacando los retos y oportunidades que plantean para la ingeniería embebida moderna.

Uso del procesador embebido NIOS II

Página | 5

El procesador NIOS II, perteneciente a la familia de procesadores *softcore* de Intel diseñados específicamente para FPGAs, representa una de las soluciones más versátiles y configurables dentro del ámbito de los sistemas embebidos. Su naturaleza programable y su integración directa dentro del *hardware* lo convierten en una herramienta estratégica para desarrollar aplicaciones personalizadas, eliminando la dependencia de microcontroladores externos. Esta característica ha sido ampliamente reconocida por la comunidad académica y profesional, consolidando al NIOS II como un punto de referencia en la evolución de los sistemas digitales reconfigurables. Una de sus principales fortalezas radica en la capacidad de fusionar *hardware* y *software* en una misma plataforma, lo que permite un control total del rendimiento y una optimización profunda del diseño.

Intel (2024), a través del *Nios II Software Developer Handbook*, demuestra la madurez tecnológica de este entorno, ofreciendo guías exhaustivas para la creación, depuración y despliegue de aplicaciones sobre FPGAs. Esta integración, en la práctica, reduce los costos de desarrollo y acelera los tiempos de diseño, permitiendo soluciones embebidas robustas sin necesidad de componentes físicos adicionales. Aun así, diversos estudios como los de Deng et al. (2022) y Tuncel et al. (2022) señalan que el NIOS II, pese a su flexibilidad, presenta limitaciones frente a procesadores físicos dedicados o arquitecturas de propósito general más recientes.

Al depender de la lógica programable de la FPGA, el rendimiento puede verse restringido por la cantidad de recursos disponibles o por la latencia introducida en sistemas de gran escala. Dichas observaciones sugieren que, aunque la versatilidad es una virtud, también introduce una complejidad adicional en la gestión del *hardware*, que puede ser un desafío para desarrolladores con menor experiencia. Por otro lado, varios autores defienden que estas limitaciones son inherentes a cualquier enfoque reconfigurable, ver los trabajos de Cao & Meyer-Baese (2022) y Gazziro et al. (2024) que señalan que el verdadero valor del NIOS II reside en su capacidad de adaptación a las necesidades del usuario. En contextos educativos, investigativos y de prototipado, la posibilidad de modificar el procesador y su arquitectura según la aplicación no solo amplía las competencias técnicas de los estudiantes, sino que estimula la creatividad y la innovación en el diseño digital. Así, el NIOS II no se concibe como un reemplazo del *hardware* convencional, sino como una alternativa flexible para entornos donde la personalización y el control del diseño son prioritarios.

Un ejemplo ilustrativo es el trabajo de Magdaleno et al. (2014), quienes implementaron un servidor web embebido con un núcleo Nios II sobre una FPGA Cyclone, integrando μ CLinux y el servidor Boa para monitorear sensores inteligentes. Su propuesta demuestra la capacidad del procesador para centralizar tareas de comunicación, control y procesamiento dentro de un mismo chip, un logro que difícilmente puede alcanzarse con procesadores convencionales sin comprometer modularidad o eficiencia. De manera complementaria, Pyrgas et al. (2016) desarrollaron un sistema abierto de procesamiento de imágenes basado en NIOS II y la tarjeta Altera DE2-70, evidenciando la versatilidad del procesador para tareas intensivas en

cómputo y su idoneidad para entornos embebidos donde se requiere alta integración entre *hardware* y *software*. En síntesis, el uso del NIOS II revela cómo la convergencia entre *hardware* y *software* reconfigurable redefine los límites de la ingeniería embebida moderna. Más que una solución transitoria, representa un modelo vigente de integración tecnológica, especialmente en proyectos educativos e industriales que valoran la experimentación, la eficiencia y la autonomía en el diseño.

Diseño de sistemas digitales flexibles y personalizados sin hardware adicional

El procesador NIOS II permite diseñar sistemas digitales altamente flexibles y adaptables, donde la totalidad de la lógica de control y procesamiento puede integrarse dentro de una FPGA, eliminando la necesidad de componentes externos como microcontroladores, memorias discretas o buses físicos. Esta integración no solo simplifica el diseño, sino que también reduce costos y riesgos de fallas asociadas a interconexiones externas. La posibilidad de configurar desde cero los periféricos, temporizadores, puertos de comunicación y aceleradores de *hardware* otorga al desarrollador un control preciso sobre la arquitectura del sistema, adaptándola a los requerimientos específicos de cada aplicación (Kumar, 2023; Zhang et al., 2013). No obstante, esta misma flexibilidad conlleva desafíos. La dependencia de los recursos de la FPGA puede limitar el rendimiento en sistemas complejos, y la gestión de configuraciones personalizadas requiere un conocimiento profundo tanto de *hardware* como de *software*. Esa idea coincide con las conclusiones de Khan et al. (2024), quienes discuten cómo el rendimiento de las implementaciones FPGA depende de la disponibilidad de recursos lógicos y de la experiencia técnica del diseñador para manejar configuraciones específicas.

Por ejemplo, al definir varios periféricos y controladores, el diseñador debe equilibrar cuidadosamente la utilización de LUTs (*Look-Up Table* o Tabla de Consulta), bloques de memoria y rutas de comunicación para evitar cuellos de botella o problemas de temporización. Sin embargo, la adaptabilidad que ofrece esta arquitectura permite iterar y optimizar el diseño según las restricciones y necesidades específicas, algo que no es posible en sistemas cerrados con *hardware* fijo (Hossain, 2024; Tuncel et al., 2022). El valor de integrar todo el sistema en una sola FPGA se evidencia en la capacidad de prototipado rápido y en la eficiencia de producción. Al eliminar el *hardware* adicional, se reducen los costos de fabricación y mantenimiento, se minimizan los puntos de fallo y se facilita la actualización de funciones mediante reprogramación. Estudios recientes muestran que esta aproximación es especialmente útil en dispositivos portátiles, sistemas embebidos industriales y redes de sensores inteligentes, donde la optimización de área, energía y rendimiento es crítica (Cabrera et al., 2023; Ren et al., 2025). Aunque algunos críticos argumentan que la dependencia de FPGAs introduce limitaciones frente a sistemas basados en procesadores físicos de alto rendimiento, la experiencia práctica demuestra que estas restricciones son compensadas por la capacidad de adaptar la arquitectura a la aplicación (Choi et al., 2024; Michon et al., 2025).

El enfoque permite diseñar sistemas *multicore*, integrar módulos de comunicación y controlar de manera precisa recursos de *hardware*, todo dentro del mismo chip. Esto convierte al NIOS II en una solución escalable y eficiente para aplicaciones donde el costo, el espacio físico y la adaptabilidad son factores determinantes (Baklouti & Abid, 2014). Un ejemplo concreto de esta flexibilidad se observa en la implementación de sistemas de control embebido que combinan lectura de sensores, procesamiento de señales y control de actuadores en tiempo real, todo dentro de la misma FPGA, sin requerir tarjetas adicionales. Esto demuestra que la arquitectura

basada en NIOS II no solo es eficiente desde el punto de vista técnico, sino que también permite innovar en la forma en que se diseñan sistemas embebidos, ofreciendo alternativas que serían inviables con procesadores y periféricos separados (Kumar, 2023; Zhang et al., 2013).

En conclusión, el diseño de sistemas digitales sin hardware adicional mediante NIOS II representa un avance significativo en la ingeniería de sistemas embebidos. La integración de toda la funcionalidad dentro de una FPGA ofrece flexibilidad, eficiencia y control detallado sobre el sistema, a la vez que fomenta la innovación en prototipado y desarrollo de aplicaciones especializadas. Esta aproximación reafirma la relevancia de los procesadores softcore en el contexto de la electrónica reconfigurable y consolida su papel como herramienta estratégica en el diseño de sistemas digitales modernos (Cabrera et al., 2023; Ren et al., 2025).

SoftCore diseñados por otras compañías distintas a Intel

Aunque el NIOS II ha consolidado su posición como referencia en procesadores softcore para FPGAs, no es la única opción disponible. Otras compañías han desarrollado arquitecturas propietarias y abiertas que buscan objetivos similares, como el MicroBlaze de Xilinx y el LatticeMico32 de Lattice Semiconductor, optimizados para sus respectivas familias de FPGA (Lattice Semiconductor, s.f.; Lysecky & Vahid, 2009). La existencia de estas alternativas amplía el panorama tecnológico, ofreciendo a los diseñadores la posibilidad de seleccionar un procesador que se ajuste a requerimientos específicos de rendimiento, costo o compatibilidad con herramientas de desarrollo. Desde una perspectiva crítica, algunos autores sostienen que los softcores de Xilinx y Lattice presentan ventajas sobre el NIOS II en términos de eficiencia de recursos y soporte para aplicaciones industriales (Lysecky & Vahid, 2009). Además, la adopción creciente de arquitecturas open-source como RISC-V ha generado debate sobre la sostenibilidad y libertad de diseño frente a procesadores con licencias cerradas como el NIOS II (Islam & Kise, 2022). La comunidad académica, en particular, valora la capacidad de RISC-V para permitir modificaciones profundas en la arquitectura, promoviendo la innovación educativa y la experimentación en investigación.

No obstante, la madurez del ecosistema NIOS II sigue siendo una ventaja significativa. Estudios recientes destacan que la plataforma de Intel ofrece herramientas de desarrollo consolidadas, documentación extensiva y soporte técnico confiable, factores que facilitan la enseñanza, la integración de sistemas y la depuración de aplicaciones complejas (Amiri et al., 2017; Gazziro et al., 2024). Esto sugiere que, aunque la flexibilidad de otros *softcores* pueda ser mayor, la estabilidad y soporte integral del NIOS II representan un valor agregado que no debe subestimarse. La coexistencia de múltiples arquitecturas *softcore*, por tanto, genera un equilibrio entre innovación y confiabilidad. Por un lado, procesadores como MicroBlaze, LatticeMico32 o RISC-V estimulan la competencia tecnológica y permiten diseños altamente personalizados. Por otro, el NIOS II mantiene su relevancia gracias a la robustez de su ecosistema y su amplia adopción en entornos educativos y de investigación. Esta dinámica de ventajas y limitaciones demuestra que la elección de un *softcore* depende en gran medida del contexto de aplicación, los objetivos del proyecto y el nivel de experiencia del desarrollador, reafirmando la necesidad de evaluar críticamente cada alternativa antes de su implementación.

El NIOS II como herramienta pedagógica en la formación en ingeniería

El interés académico en los procesadores *softcore* ha crecido de manera notable, siendo el NIOS II uno de los más estudiados y aplicados en proyectos universitarios. Tesis de grado, como las de Arcentales (2016) y Castro (2021), evidencian su utilidad en la enseñanza de sistemas embebidos y automatización, promoviendo un aprendizaje experimental mediante la implementación real de microprocesadores configurables. La principal fortaleza de estas investigaciones radica en que los estudiantes no solo adquieren conocimientos teóricos, sino también competencias prácticas en diseño digital, programación y depuración de sistemas embebidos. Esto evidencia que las tesis universitarias funcionan como espacios de innovación y transferencia tecnológica, donde los futuros ingenieros desarrollan habilidades críticas para enfrentar problemas reales de la industria. Sin embargo, Huseynli et al. (2022) argumentan que la naturaleza experimental de estos proyectos limita su impacto práctico fuera del entorno educativo. Dado que los prototipos desarrollados rara vez se transfieren a entornos industriales, la aplicabilidad inmediata puede ser cuestionable. Esta observación sugiere que, aunque los proyectos son útiles para la formación, podrían carecer de escalabilidad o relevancia comercial.

Página | 8

En contrapartida, defensores de estas prácticas educativas como Cabrera et al. (2023) y Silva et al. (2014) sostienen que el verdadero valor de las tesis radica en la formación integral del estudiante, más que en su impacto industrial inmediato. La experiencia práctica adquirida a través del NIOS II fortalece competencias como resolución de problemas, diseño modular y adaptación de *hardware* y *software* a necesidades específicas. Además, la experimentación fomenta la creatividad y la innovación, habilidades que son altamente valoradas en entornos de investigación y desarrollo tecnológico. Un ejemplo concreto se encuentra en proyectos que integran NIOS II con FPGAs para implementar sistemas de control, comunicación o automatización en laboratorios universitarios. Dichos trabajos demuestran que, aunque los resultados puedan no replicarse inmediatamente en la industria, generan conocimiento aplicable y metodologías transferibles, sirviendo de puente entre la educación y la innovación tecnológica.

En síntesis, el estudio del NIOS II en tesis de grado evidencia una tensión entre impacto educativo y relevancia práctica, donde la crítica apunta a la limitada aplicación industrial, pero la contracrítica resalta que el aprendizaje experimental y la formación de competencias técnicas profundas son objetivos igualmente valiosos. Esta dinámica reafirma que las investigaciones en universidades no solo producen conocimiento, sino que preparan a los estudiantes para la transición hacia la investigación aplicada y la ingeniería profesional.

Proceso de configuración del procesador mediante QSys (Platform Designer)

El entorno QSys, actualmente conocido como *Platform Designer*, constituye una herramienta fundamental para la configuración y ensamblaje de sistemas basados en el procesador NIOS II. Su interfaz gráfica permite integrar periféricos, módulos de memoria y componentes de comunicación de manera visual, lo que reduce significativamente la complejidad del diseño y acelera los tiempos de desarrollo (Al Rayahi, 2008). Esta automatización facilita que los desarrolladores, incluso con experiencia limitada en *hardware*, puedan construir sistemas embebidos complejos sin necesidad de programar manualmente la interconexión de cada módulo. No obstante, Gutiérrez (2015) sostiene que el uso de QSys puede generar una dependencia hacia el ecosistema de Intel, limitando la interoperabilidad con plataformas abiertas y dificultando la migración de proyectos a otras

arquitecturas o procesadores *softcore*. Esta observación indica que, aunque QSys simplifica la creación de sistemas, podría restringir la libertad del diseñador al estar ligado a herramientas propietarias. En contraposición, Intel ha buscado mitigar estas limitaciones mediante la compatibilidad con estándares industriales y herramientas de código abierto, promoviendo un equilibrio entre la automatización propia de QSys y la flexibilidad requerida en entornos más amplios (Intel, 2023).

Página | 9

Esta estrategia permite que los desarrolladores mantengan control sobre la arquitectura y la integración de periféricos, mientras se aprovechan las ventajas de un entorno consolidado y probado. Desde una perspectiva crítica, algunos estudios sugieren que la dependencia de herramientas gráficas puede reducir la comprensión profunda de la arquitectura subyacente, generando diseñadores menos preparados para problemas de optimización compleja o depuración avanzada (Muttillio et al., 2017; Nane et al., 2016). Sin embargo, la contracrítica indica que QSys no reemplaza la necesidad de conocimiento técnico, sino que facilita el prototipado rápido y la validación de conceptos, liberando al diseñador de tareas repetitivas y permitiéndole concentrarse en la optimización y personalización del sistema (LaForest, 2015). En conclusión, QSys representa un equilibrio entre facilidad de uso y control técnico, donde las críticas sobre dependencia y comprensión profunda se ven compensadas por la eficiencia, reducción de errores y capacidad de integración que ofrece la herramienta. Este enfoque demuestra que la configuración de sistemas NIOS II mediante QSys no solo agiliza el desarrollo, sino que también contribuye al aprendizaje práctico y a la creación de soluciones embebidas robustas y escalables.

Programación del NIOS II en lenguaje C++ mediante Eclipse

El desarrollo de *software* para el procesador NIOS II se realiza principalmente en **C** o **C++**, empleando el entorno Eclipse proporcionado por Intel. Esta combinación permite integrar de manera eficiente el *hardware* programable de la FPGA con la lógica de *software*, facilitando la implementación de controladores, algoritmos e interfaces de usuario sobre plataformas personalizadas (Keat et al., 2015). La principal ventaja de este enfoque radica en que los programadores pueden aprovechar la modularidad del lenguaje C++ y la reutilización de código, acelerando el desarrollo de sistemas embebidos complejos y escalables. No obstante, algunos autores argumentan que Eclipse puede resultar pesado y lento en equipos con recursos limitados, lo que puede obstaculizar el aprendizaje inicial y la depuración de proyectos pequeños o experimentales (Kumar, 2023) (Sáenz, 2018). Esta limitación técnica puede generar frustración en estudiantes o desarrolladores que no cuentan con *hardware* potente, reduciendo la accesibilidad del entorno de desarrollo.

Desde la contrapartida, se sostiene que las ventajas de un entorno unificado y depurador integrado superan estas limitaciones. Eclipse permite acceder a herramientas de análisis de código, compatibilidad con librerías estándar y soporte directo de Intel para NIOS II, lo que facilita la depuración, optimización y escalabilidad de los proyectos embebidos (Intel, 2024). Además, la programación en C++ potencia la integración de paradigmas orientados a objetos, lo que contribuye a la creación de *software* más mantenible y adaptable a diferentes configuraciones de *hardware*. Críticamente, se podría señalar que depender de un entorno específico como Eclipse puede generar cierta rigidez en el aprendizaje, limitando la exposición del estudiante a otras plataformas de desarrollo o lenguajes de programación (Ahid & Givargis, 2002). Sin embargo, la contracrítica indica que Eclipse sirve como puente pedagógico y profesional, permitiendo que los estudiantes se familiaricen

con herramientas de la industria mientras desarrollan competencias transferibles a otros entornos y lenguajes. En síntesis, la programación del NIOS II mediante C++ en Eclipse representa un balance entre eficiencia y complejidad: aunque existen limitaciones en rendimiento y dependencia del entorno, las ventajas en integración *hardware-software*, depuración y escalabilidad consolidan a esta combinación como una opción robusta para la formación académica y el desarrollo de sistemas embebidos avanzados (He & Huang, 2017).

Fundamentos técnicos, arquitectura interna y ejemplos básicos

El NIOS II se fundamenta en una arquitectura RISC (*Reduced Instruction Set Computer*), lo que le otorga simplicidad y velocidad en la ejecución de instrucciones. Su diseño modular permite configurar desde versiones básicas hasta sistemas con multiplicadores, controladores de interrupciones y memorias cache, adaptándose a diferentes niveles de complejidad según la aplicación (Ahmad et al., 2018). Esta modularidad facilita que los desarrolladores optimicen el uso de recursos de la FPGA y ajusten el rendimiento del procesador a las necesidades específicas del proyecto. Por otro lado, críticos de la arquitectura RISC sostienen que su conjunto reducido de instrucciones puede limitar la capacidad de procesamiento en aplicaciones de alta demanda o en sistemas que requieren cálculos intensivos (Balwaik & Shailja, 2013) y (Sáenz et al., 2018). Esta limitación podría hacer que, en ciertos contextos industriales, la arquitectura RISC se perciba como menos potente frente a procesadores CISC o arquitecturas más complejas. En contrapartida, investigaciones recientes indican que la combinación de RISC con módulos personalizados y aceleradores de *hardware* permite lograr un desempeño competitivo incluso frente a arquitecturas más complejas Santucci et al. (2022) y Cao & Meyer-Baese (2022).

Esto demuestra que, aunque la simplicidad del conjunto de instrucciones podría ser una limitación teórica, la personalización y flexibilidad de la arquitectura compensan dicha restricción, permitiendo adaptarla a aplicaciones que demandan un rendimiento alto sin sacrificar eficiencia. Críticamente, se podría argumentar que la modularidad y personalización del NIOS II requieren un conocimiento profundo de *hardware* y *software*, lo que representa una barrera para desarrolladores menos experimentados (Zagan & Gaitan, 2023). Sin embargo, la contracrítica señala que esta característica también es una fortaleza pedagógica y profesional, ya que fomenta la adquisición de habilidades avanzadas en diseño digital, depuración y optimización, competencias que son altamente valoradas en entornos académicos e industriales (Adam, 2022). En conclusión, los fundamentos técnicos y la arquitectura interna del NIOS II muestran un equilibrio entre simplicidad, flexibilidad y rendimiento. Aunque su diseño RISC presenta limitaciones teóricas frente a arquitecturas más complejas, la posibilidad de personalizar módulos y optimizar recursos convierte al NIOS II en una solución práctica y adaptable para sistemas embebidos modernos, consolidando su papel en la enseñanza, investigación y prototipado de aplicaciones digitales.

Perspectivas futuras: NIOS V y sistemas multicore

Las tendencias actuales en procesadores softcore apuntan hacia el desarrollo de arquitecturas *multicore* y abiertas, como el NIOS V, basado en la arquitectura RISC-V (Intel, 2021). Este avance promete mayor rendimiento, paralelismo y compatibilidad con entornos open-source, lo que facilita la experimentación, la innovación y la integración de sistemas embebidos más complejos. La adopción de NIOS V representa una

evolución natural del NIOS II, orientada a sistemas que requieren procesamiento paralelo, eficiencia energética y escalabilidad, características esenciales en la computación moderna y en aplicaciones industriales avanzadas. No obstante, algunos investigadores advierten que la transición hacia RISC-V y sistemas *multicore* podría fragmentar los ecosistemas propietarios existentes, generando desafíos en compatibilidad y migración de proyectos previos basados en NIOS II (Dakić et al., 2024) y (Tuncel et al., 2022). Esta fragmentación podría incrementar la curva de aprendizaje para desarrolladores y estudiantes, y limitar la reutilización inmediata de software y periféricos diseñados para NIOS II. En contraposición, estudios recientes destacan que las ventajas de procesamiento paralelo y arquitectura abierta superan estas limitaciones.

El enfoque *multicore* permite ejecutar múltiples tareas simultáneamente, optimizar el uso de recursos de la FPGA y adaptar la plataforma a diferentes aplicaciones, desde sistemas embebidos industriales hasta laboratorios educativos (Cao & Meyer-Baese, 2022; Villamizar et al., 2024). En esta misma línea, Müller et al. (2024) plantean un enfoque de diseño generativo de arquitecturas multiprocesador, que automatiza la síntesis y configuración de plataformas dentro de FPGAs, fortaleciendo la eficiencia y escalabilidad de los sistemas basados en NIOS V y otras arquitecturas RISC-V. Asimismo, la compatibilidad con RISC-V democratiza el acceso a la innovación tecnológica, facilitando que más desarrolladores puedan modificar, optimizar y ampliar el procesador según sus necesidades específicas. Desde una perspectiva crítica, algunos expertos sostienen que la adopción de arquitecturas abiertas puede complicar la integración con *software* propietario y herramientas tradicionales, lo que podría ralentizar la adopción en entornos industriales consolidados (Dakić et al., 2024; Islam & Kise., 2022).

Sin embargo, la contracrítica indica que esta apertura refuerza la sostenibilidad y la flexibilidad a largo plazo, permitiendo que la tecnología evolucione con menor dependencia de proveedores específicos y fomentando la colaboración en la comunidad académica y profesional (Pozzana, 2020). En conclusión, las perspectivas futuras de los procesadores softcore reflejan un equilibrio entre innovación y compatibilidad. Aunque existen retos en la transición hacia sistemas multicore y RISC-V, los beneficios en rendimiento, escalabilidad y libertad de diseño consolidan a NIOS V como una evolución estratégica de NIOS II, posicionando a los procesadores softcore como protagonistas en la computación reconfigurable del futuro.

Conclusiones

El procesador NIOS II se erige como una herramienta esencial en la ingeniería digital moderna, al posibilitar el diseño de sistemas embebidos personalizados sobre plataformas FPGA con altos niveles de flexibilidad, eficiencia y control. Su arquitectura configurable permite integrar hardware y software en un mismo entorno, reduciendo la dependencia de componentes externos y fomentando el desarrollo de soluciones adaptadas a necesidades específicas.

Desde la perspectiva educativa y tecnológica, el NIOS II trasciende el ámbito de la experimentación académica al convertirse en un vehículo efectivo para la transferencia de conocimiento. Facilita la comprensión de conceptos complejos de arquitectura y diseño digital, permitiendo a estudiantes y docentes experimentar con la creación de sistemas reales y fortalecer la relación entre la teoría y la práctica. En este sentido, el procesador

contribuye a reducir la brecha entre el conocimiento teórico y la aplicación práctica en el campo del diseño digital reconfigurable, donde aún existe una limitada disponibilidad de experiencias formativas accesibles.

Asimismo, la posibilidad de programar el procesador en lenguajes de alto nivel como C/C++ fortalece la integración entre hardware y software, ampliando las capacidades del desarrollador para implementar soluciones embebidas eficientes. Esta característica lo consolida como un entorno ideal para la formación en co-diseño y desarrollo de sistemas digitales integrados.

En cuanto a la proyección tecnológica, la evolución hacia el NIOS V basado en la arquitectura RISC-V representa un paso decisivo hacia la apertura y escalabilidad del ecosistema Intel. Dicha transición anuncia una nueva etapa de investigación e innovación, orientada a la creación de plataformas heterogéneas, energéticamente eficientes y compatibles con estándares abiertos.

En síntesis, el NIOS II se consolida como un procesador de referencia en el ámbito académico y profesional, no solo por su capacidad técnica, sino por su impacto formativo y su potencial para cerrar brechas tecnológicas en la educación y la investigación. Su relevancia radica en integrar ciencia, aprendizaje e innovación tecnológica en un mismo marco de desarrollo, constituyendo un modelo de enseñanza aplicada que impulsa la evolución de la ingeniería digital contemporánea.

Conflicto de Intereses

Esta investigación está libre de cualquier conflicto de intereses.

Referencias

- Adam, G. (2022). *Co-Design of Multicore Hardware and Multithreaded Software on FPGA Platform*. *Computers*, 11(5), 76. <https://doi.org/10.3390/computers11050076>
- Ahid, F. & Givargis, T. (2002). *Embedded System Design: A Unified Hardware/Software Introduction*. John Wiley & Sons.
- Ahmad, W.; Abbasi, I.; Sanwal, U. & Mahmood, H. (2018). Accelerating Viterbi Algorithm using Custom Instruction Approach. *IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications, MESA*. <https://doi.org/10.1109/MESA.2018.8449144>
- Al Rayahi, O. (2008). *A CAD Tool for Synthesizing Optimized Variants of Altera's Nios II Soft-Core Processor*. University of Windsor. <https://scholar.uwindsor.ca/etd/8039/>
- Amiri, M.; Siddiqui, F. M.; Kelly, C.; Woods, R.; Rafferty, K. & Bardak, B. (2017). *FPGA-Based Soft-Core Processors for Image Processing Applications*. *Journal of Signal Processing Systems*, 89(2), 175–188. <https://doi.org/10.1007/s11265-016-1185-7>
- Arcentales, V. (2016). *Diseño de prácticas para el aprendizaje de sistemas embebidos basados en el procesador NIOS® II utilizando herramientas de Quartus II y la tarjeta DE0-Nano* (Proyecto integrador de grado). Escuela Superior Politécnica del Litoral. <https://www.dspace.espol.edu.ec/handle/123456789/38556>
- Baklouti, M. & Abid, M. (2014). Multi-Softcore Architecture on FPGA. *International Journal of Reconfigurable Computing*, 979327. <https://doi.org/10.1155/2014/979327>

- Balwaik, R. & Shailja N. (2013). Open-Source 32-Bit RISC Soft-Core Processors. *IOSR Journal of VLSI and Signal Processing*, 2, 43-46. <https://doi.org/10.9790/4200-0244346>
- Cabrera, A.; Garcés, L.; Cabrera, A.; Culman, R.; Sánchez, S.; Brox, P.; Ieno, E.; Cleber, T. & Bob, B. (2023). Contribuciones a la implementación de sistemas electrónicos digitales embebidas sobre hardware reconfigurable. *Anales de la Academia de Ciencias de Cuba*, 13(4), http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2304-01062023000400007&lng=es&tlng=es.
- Cao, H. & Meyer-Baese, U. (2022). XML-Based Automatic NIOS II Multi-Processor System Generation for Intel FPGAs. *Electronics*, 11(18), 2840. <https://doi.org/10.3390/electronics11182840>
- Castro, J. (2021). *Implementación de algoritmos de aprendizaje automático en procesadores softcores basados en ARM*. [Tesis de Licenciatura, Tecnológico de Costa Rica]. <https://n9.cl/gpv7ga>
- Choi, E.; Park, J.; Han, K. & Lee, W. (2024). AESware: Developing AES-enabled low-power multicore processors leveraging open RISC-V cores with a shared lightweight AES accelerator. *Journal of Electronic Science and Technology*, 60, 101894. <https://doi.org/10.1016/j.jestch.2024.101894>
- Dakić, V.; Mrsic, L.; Kunic, Z. & Dambic, G. (2024). Evaluating ARM and RISC-V Architectures for High-Performance and Energy-Efficient Systems. *Electronics*, 13(17), 3494. <https://doi.org/10.3390/electronics13173494>
- Deng, D.; Crookes, D.; Woods, R. & Siddiqui, F. (2022). A soft coprocessor approach for developing image and video processing applications on FPGAs. *Journal of Imaging*, 8(2), 42. <https://doi.org/10.3390/jimaging8020042>
- Faycal, R.; Lahcene, Z. & Nabil, B. (2021). ISODATA SOPC-FPGA Implementation of Image Segmentation Using NIOS-II Processor. *Indonesian Journal of Electrical Engineering and Computer Science*, 22(2), 818–825. <https://doi.org/10.11591/ijeecs.v22.i2.pp818-825>
- Gazziro, M.; Junior, J.; Junior, O.; Cavallari, M. & Carmo, J. (2024). Design and Evaluation of Open-Source Soft-Core Processors. *Electronics*, 13, 781, 1-20. <https://doi.org/10.3390/electronics13040781>
- Gutiérrez, J. (2015). *Implementación en una FPGA de un procesador básico basado en MIPS*. [Trabajo Fin de Grado para optar al grado de ingeniero, Universidad de Valladolid]. Repositorio Universidad de Valladolid. <http://uvadoc.uva.es/handle/10324/14472>
- He, N. & Huang, H. (2017). Use of model-based design to teach embedded systems programming. In *2017 IEEE International Conference on Electro Information Technology (EIT)* (pp. 135–139). IEEE. <https://doi.org/10.1109/EIT.2017.8053336>
- Hossain, M. (2024). *Adaptive Beyond Von-Neumann Computing Devices and Reconfigurable Architectures for Edge Computing Applications* [Doctoral dissertation, University of Central Florida]. <https://n9.cl/97191>
- Huseynli, M.; Bub, U. & Ogbuachi, M. (2022). Development of a Method for the Engineering of Digital Innovation Using Design Science Research. *Information*, 13, 573. <https://doi.org/10.3390/info13120573>
- Intel (2024). *NIOS II software developer handbook*. [Consulta: 30-3-2025]. <https://www.intel.com/content/www/us/en/docs/programmable/683525/21-3/eol.html>
- Intel (2023). *Nios® V Processor Overview*. [Consulta: 15-4-2025]. <https://www.intel.com/content/www/us/en/products/details/fpga/intellectual-property/processors-peripherals/niosv.html>
- Intel (2021). *Introducing Intel's Next-Generation of Nios® Soft-Core Processor: Nios® V Processor*. Intel Blog. [Consulta: 23-5-2025]. <https://community.intel.com/t5/Blogs/Products-and-Solutions/FPGA/Introducing-Intel-s-Next-Generation-of-Nios-Soft-Core-Processor/post/1342953>
- Islam, M. & Kise, K. (2022). An efficient resource shared RISC-V multicore architecture. *IEICE Transactions on Information and Systems*, 105(9), 1585–1595. <https://doi.org/10.1587/transinf.2021EDP7248>
- Khan, L.; Isanaka, S. P. & Liou, F. (2024). *FPGA-Based Sensors for Distributed Digital Manufacturing Systems: A State-of-the-Art Review*. *Sensors*, 24(23), 7709. <https://doi.org/10.3390/s24237709>

- Keat, L.; Jambek, A. & Hashim, U. (2015). Heart rate monitoring system design and analysis using a Nios II soft-core processor. *Journal of Electronics and Telecommunications*, 62(3), 283-288. <https://journals.pan.pl/dlibra/publication/116920/edition/101646/content>
- Kumar, B. (2023). Design of a Soft-Core Processor in FPGA. *Intrenational Journal of Engineering Research & Technology (IJERT)*, 1 (2), 107-114. <https://www.researchgate.net/publication/369233268>
- LaForest, C. E. (2015). *High-speed soft-processor architecture for FPGA overlays* [Tesis para optar al grado de Doctor, University of Toronto]. University of Toronto Research Repository. <https://fpgacpu.ca/publications/High-Speed-Soft-Processor-Architecture-for-FPGA-Overlays.pdf>
- Lattice Semiconductor (s.f.). *LatticeMico32 32-bit soft microprocessor core*. Lattice Semiconductor. [Consulta: 15-6-2025]. <https://n9.cl/u6hag>
- Lysecky, R. & Vahid, F. (2009). Design and implementation of a microblaze-based WARP processor. *ACM Transactions on Embedded Computing Systems*, 8 (3), 1-22. <https://doi.org/10.1145/1509288.1509294>
- Magdaleno, E.; Rodríguez, M.; Pérez, F.; Hernández, D. & García, E. (2014). A FPGA Embedded Web Server for Remote Monitoring and Control of Smart Sensors Networks. *Sensors*, 14(1), 416-430. <https://doi.org/10.3390/s140100416>
- Michon, R.; Ducceschi, M.; Cochard, P.; Skare, T.; Webb, C. J. & Russo, R. (2025). Evaluating CPU, GPU, and FPGA performance in the context of modal reverberation: A comparative analysis. *Frontiers in Signal Processing*, 5, 1522604. <https://doi.org/10.3389/frsip.2025.1522604>
- Müller, L.; Schumacher, N.; Steffen, L. & Haubelt, C. (2024). Generative Design of the Architecture Platform in Multiprocessor System Design. *Electronics*, 13(7), 1404. <https://doi.org/10.3390/electronics13071404>
- Muttillo, V.; Valente, G.; Federici, F.; Pomante, L.; Faccio, M.; Tieri, C. & Ferri, S. (2017). A design methodology for soft-core platforms on FPGA with SMP Linux, OpenMP support, and distributed hardware profiling system. *Journal of Embedded Systems*, 15, 1-14. <https://doi.org/10.1186/s13639-016-0051-9>
- Nane, R.; Sima, V-M.; Pilato, C.; Choi, J.; Fort, B.; Canis, A.; Chen, Y.; Hsiao, H.; Brown, S.; Ferrandi, F.; Anderson, J. & Bertels, K. (2016). A survey and evaluation of FPGA high-level synthesis tools. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 35(10), 1591-1604. <https://doi.org/10.1109/TCAD.2015.2513673>
- Pozzana, F. (2020). *RISC-V: An FPGA implementation for general purpose prototyping hardware*. [Tesis para optar al grado de Master, Politecnico di Torino]. Politecnico di Torino Institutional Repository. <https://webthesis.biblio.polito.it/16652/1/tesi.pdf>
- Pyrgas, L.; Kalantzopoulos, A. & Zigouris, E. (2016). Design and implementation of an open image processing system based on NIOS II and ALTERA DE2-70 board. *Journal of Engineering Science and Technology Review*, 9(5), 51-55. <https://doi.org/10.25103/jestr.095.06>
- Ren, Y.; He, Y.; Zhang, X.; Yen, A. & Li, G-P. (2025). A Cyber Manufacturing IoT System for Adaptive Machine Learning Model Deployment by Interactive Causality-Enabled Self-Labeling. *Machines*, 13(304). <https://doi.org/10.3390/machines13040304>
- Sáenz R., W., Rivera S., F. & Martínez S., F. (2018). 8-bit softcore microprocessor with dual accumulator designed to be used in FPGA. *Revista Tecnura*, 22(56), 40-50. <https://doi.org/10.14483/22487638.12976>
- Santucci, R., Diamond, J., Eddy, N., Semenov, A., & Voy, D. (2022). A modern Ethernet data acquisition architecture for Fermilab beam instrumentation. In *Proceedings of the 11th International Beam Instrumentation Conference (IBIC 2022)* (pp. 500-503). JACoW Publishing. <https://doi.org/10.18429/JACoW-IBIC2022-WEP40>
- Silva, I. S.; Luz, L. O.; Nepomuceno, R. & Dos Santos, J. C. (2014). Multi-Core Processor Programming: An Educational Experience Using FPGA Embedded Didactic Platforms. *International Journal of Computer Architecture Education*, 3 (1), 9-12. <https://doi.org/10.5753/ijcae.2014.4934>

- Tuncel, Y.; Krishnakumar, A.; Chithra, A. L.; Kim, Y. & Ogras, U. (2022). A Domain-Specific System-On-Chip Design for Energy Efficient Wearable Edge AI Applications. In *ACM/IEEE International Symposium on Low Power Electronics and Design (ISLPED '22)*, 20, 1–20. <https://doi.org/10.1145/3531437.3539711>
- Villamizar, J.; Borra, V. & Li, F. (2024). Building fully digital controller with Altera Cyclone FPGA and NIOS processor for induction heating application. *Discover Electronics*, 1(27). <https://doi.org/10.1007/s44291-024-00033-y>
- Xu, J.; Kovatsch, M.; Mattern, D.; Mazza, F.; Harasic, M.; Paschke, A. & Lucia, S. (2022). A review on AI for smart manufacturing: Deep learning challenges and solutions. *Applied Sciences*, 12(16), 8239. <https://doi.org/10.3390/app12168239>
- Zagan, I. & Gaitan G. (2023). *Soft-core processor integration based on different instruction set architectures and field programmable gate array custom datapath implementation*. *PeerJ Computer Science*, 9, e1300. <https://doi.org/10.7717/peerj-cs.1300>
- Zhang, J.; Guo, Y. & Mo, G. (2013). A Software Hardware Co-Design Approach for FPGAs on Nios II Soft-Core Processors. *Applied Mechanics and Materials*, 373–375, 1591–1594. <https://doi.org/10.4028/www.scientific.net/amm.373-375.1591>