

Plataforma Abierta para el Aprendizaje de Diseño Digital sobre sistemas basados en FPGA

Open Platform for Digital Design Learning on FPGA-based Systems

Martín A. Heredia, Facundo S. Larosa, Ramiro A. Ghignone, Federico A. Saraullo, Hernán P. Mendes

Grupo de Aplicaciones en Sistemas Embebidos (ASE)

Universidad Tecnológica Nacional, Facultad Regional Haedo

Haedo, Buenos Aires, Argentina

embebidos@frh.utn.edu.ar

Recibido: 17/10/20; Aceptado: 27/11/20

Abstract – *In embedded systems a wide variety of applications can be solved or approached using different technologies, such as microcontrollers, FPGA, Systems on Chip (SoC), among others. Even though the volume of information and solutions available that focus on low cost based microcontrollers is quite vast, FPGA and SoC do not have as many resources readily available despite being essential for critical applications and also those in which higher speeds for data processing are required. Not only that, but the high cost of development boards, discontinuity between contents taught in higher education and knowledge required in practical fields, and overall lack of learning resources all add difficulty to any professional looking to specialize themselves in this field of knowledge. It is for this reason that this work proposes an Open Platform for Digital Design Learning on FPGA-based Systems, which looks to facilitate the learning process for any student or professional that wants to get started in this field. For this purpose this work proposes the following: a low-cost development board alongside a set of learning tools and examples which can fit both self-taught students and lecture-based learning environments, and the creation of a national collaboration network to give support to students and teachers who want to join and participate in the project.*

Keywords – FPGA, VHDL, Verilog, Embedded Systems, SoC.

Resumen—*En los sistemas embebidos se puede encontrar una gran variedad de aplicaciones basadas en diferentes tecnologías: microcontroladores, FPGA, System on Chip (SoC), entre otras. A pesar de que el volumen de información y soluciones que existen para sistemas basados en microcontroladores de bajo costo es bastante extensa, no existen tantos recursos que se focalicen en FPGA y SoC, siendo estos dispositivos imprescindibles para ciertas aplicaciones de carácter crítico y también en aquellas que requieren una velocidad o capacidad de procesamiento de datos elevada. No solo eso, si no que el costo relativamente elevado de adquisición de placas para aprendizaje, la falta de recursos didácticos y la falta de continuidad entre los contenidos de las carreras de grado y los que se requieren en un ámbito profesional, agregan un gran número de dificultades a la formación inicial de profesionales en esta área de la electrónica digital. Por este motivo este trabajo propone una Plataforma Abierta para el Aprendizaje de Diseño Digital sobre sistemas basados en FPGA, que busca facilitar el aprendizaje para cualquier alumno o profesional que quiera capacitarse en esta temática. Para ello se propone: una placa de bajo costo inicial juntamente con una plataforma de herramientas pedagógicas para el autoaprendizaje o el dictado de clases, y la formación de una red de colaboradores a nivel nacional para dar soporte a los estudiantes y docentes que se incorporen al proyecto.*

Palabras clave— FPGA, VHDL, Verilog, Sistemas Embebidos, SoC

I. INTRODUCCIÓN

Las FPGA (Field Programmable Gate Array), son un conjunto de bloques lógicos dispuestos en un arreglo matricial que presenta la posibilidad de ser configurados para implementar una gran variedad de aplicaciones [1]. Además, existen soluciones compuestas por varias tecnologías, como FPGA y procesadores dentro del mismo chip, denominadas System on Chip (SoC). El uso de FPGA y SoC se puede encontrar en áreas tales como comunicaciones, broadcasting, defensa, investigación, aplicaciones aeroespaciales, medicina, electrónica de consumo, entre otras [2, 3]. Este tipo de tecnología presenta ventajas sobre otras tecnologías digitales, por ejemplo, una mayor capacidad de procesamiento y capacidad de paralelismo, aunque también presenta desventajas, tales como el costo y tiempo de desarrollo elevados. [4, 5]

Habida cuenta de la importancia estratégica de esta área de los sistemas embebidos, resulta de gran interés fomentar la formación de profesionales que se inicien en su aprendizaje en un nivel universitario de grado o incluso terciario o secundario. No obstante, existen algunos obstáculos para aquellos que se inician en esta tecnología: el costo relativamente elevado de los kits de desarrollo de nivel inicial, la escasez de bibliografía en español y la falta de una comunidad de usuarios que pueda dar impulso y soporte al aprendizaje. Si bien, a nivel local existen trabajos fundacionales y de gran aporte respecto al tema, como la plataforma Kefir [6], la necesidad de expandir y divulgar el uso de estas tecnologías a nivel local aún debe ser atendida.

Este trabajo busca proveer una solución de utilidad para estudiantes, instituciones, desarrolladores independientes y empresas que tengan la necesidad de iniciarse en este tipo de tecnologías, de tal manera que el costo, la complejidad, el uso de herramientas privativas y la falta de soporte no se transformen en obstáculos para emprender esta tarea. De esta manera, el trabajo puede dividirse en tres líneas fundamentales: la plataforma de hardware, el material didáctico compuesto por una Wiki y un repositorio de ejemplos y la red de colaboradores.

Finalmente, la plataforma se ha incorporado al proyecto Computadora Industrial Abierta Argentina (CIAA) [7] bajo el nombre de EDU-CIAA-FPGA [8].

II. PLATAFORMA DE HARDWARE

A. Características

De acuerdo a la problemática planteada en la introducción, los criterios generales para el diseño de hardware fueron: bajo costo, posibilidad de uso de herramientas libres, posibilidad de fabricación en el país en un entorno académico o industrial, facilidad para expandir sus funcionalidades y cumplimiento de normas de manufacturabilidad [9] para disminuir tiempos y costos. En base a lo anterior, se decidió utilizar una FPGA económica, con un encapsulado relativamente sencillo de soldar y que permita el uso de herramientas libres para su configuración. La opción seleccionada fue la familia ICE40 de Lattice Semiconductor. En particular, se eligió la ICE40HX4K-TQ144 [10], cuyo costo unitario es de aproximadamente 6 USD [11] y permite el uso de herramientas libres para su configuración. El diagrama en bloques del Hardware propuesto para la EDU-CIAA-FPGA [8, 12] se presenta en la Fig. 1, donde se pueden observar los componentes principales: FPGA, memoria flash y la interfaz FTDI que se utiliza para realizar el grabado de la memoria y para comunicar a la FPGA con la PC por medio de una interfaz serial virtual. La tabla 1 muestra las características de la FPGA utilizada y en la Fig. 2 se muestra la placa terminada.

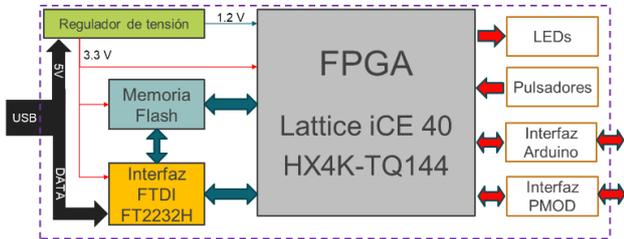


Fig. 1. Diagrama en bloques del hardware

TABLA I
CARACTERÍSTICAS DE LA FPGA LATTICE ICE40-HX4K

Número de Parte	<i>iCE40HX4K</i>
Celdas Lógicas (LUT + Flip-Flop)	3520
Bloques de Memoria RAM4K	20
Bits RAM RAM4K	80K
PLLs	2
Pines I/O Programables (Máximos)	95
Pines Diferenciales (Máximos)	12



Fig. 2. Placa de desarrollo EDU-CIAA-FPGA

B. Ejemplos de uso

Con el fin de verificar el funcionamiento del hardware para fines didácticos, se han realizado una serie de diseños, disponibles en el repositorio de ejemplos del proyecto [13]. La tabla 2 muestra un resumen de algunos de estos ejemplos, junto con la cantidad de LUTs utilizadas por cada uno de ellos como referencia. A modo ilustrativo, las figuras 3 y 4 muestran el diagrama en bloques y la señal transmitida por un generador de señales implementado en la FPGA, respectivamente.

TABLA II
LISTA DE EJEMPLOS DE DISEÑO Y SU CONSUMO DE LUTs

Diseño	LUTs utilizadas
Detector de trama de 8 bits	22 (0,63%)
UART TX	124 (3,52%)
Generador de señales + UART	365 (10,37%)
Muestreo de audio en 24 bits, filtrado y cuantización a 12 bits	881 (25,03%)
Periférico SPI Maestro	58 (1,65%)

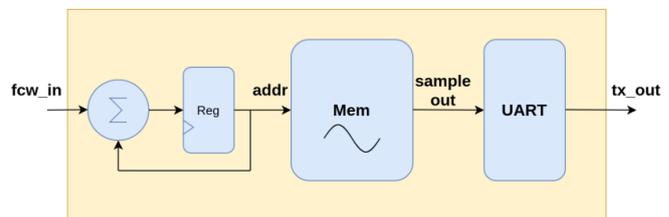


Fig. 3. Diagrama en bloques del generador de señales senoidales.

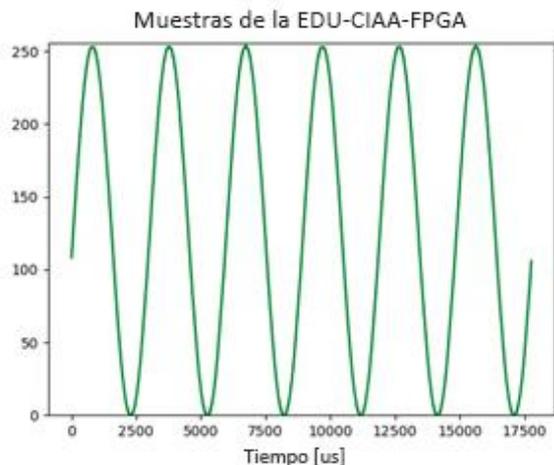


Fig. 4. Señal senoidal generada en la EDU-CIAA-FPGA y transmitida via UART a una PC

C. Producción

Dada la necesidad de abastecer al mercado local con una plataforma de desarrollo en FPGA, el objetivo de este trabajo no estaría del todo completo sin lograr una producción local de placas que puedan ser adquiridas por particulares, empresas o universidades. La liberación del hardware y el seguimiento de normas de manufacturabilidad fueron fundamentales en este sentido para garantizar que

empresas locales tengan la posibilidad de fabricar y comercializar las placas de desarrollo libremente.

Las reglas adoptadas para diseñar el hardware fueron tomadas de recomendaciones presentes en la norma IPC-2221A [9]. La tabla 3 muestra algunas de las determinaciones adoptadas en la etapa de diseño:

TABLA III
REGLAS DE DISEÑO RECOMENDADAS POR LA NORMA IPC-2221A

Propiedad	Valor
Distancia mínima entre componentes	0.2mm
Ancho mínimo de pistas	0.2mm
Distancia mínima entre pistas	0.2mm
Distancia mínima entre pistas y plano de alimentación	0.25mm
Diámetro mínimo de vías (perforación)	0.4mm
Diámetro total de vías mínimo	0.8mm
Distancia mínima entre borde de la placa y cobre	0.5mm
Tamaño del PCB	60mm x 80mm

En particular, el hardware, como todas las placas del proyecto CIAA ha sido publicado utilizando una licencia BSD de tres cláusulas modificada [14]. Este tipo de licencia permite la utilización del hardware como referencia para nuevos diseños sin ningún tipo de restricción. La Fig. 5 muestra el primer lote de producción con los componentes montados, listos para ser soldados en horno. En la Fig. 6 pueden verse dos paneles completos, con cuatro placas por panel.



Fig. 5. Primer lote de producción preparado para soldar en horno

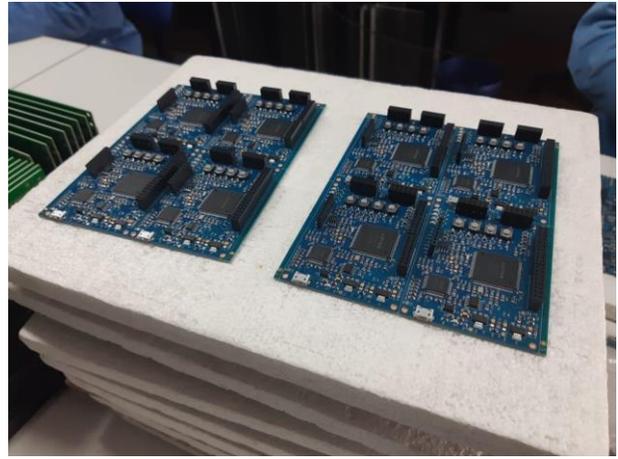


Fig. 6. Dos paneles de placas EDU-CIAA-FPGA terminados

A. Recepción del público

El 22 de septiembre de 2020 el proyecto fue presentado en sociedad por medio de la plataforma de streaming YouTube, asistiendo alrededor de 200 personas en promedio. Por medio de las estadísticas recolectadas se determinó que asistieron personas de 20 de las 24 secciones administrativas (provincias más la Capital Federal) de la Argentina, así como de algunos países de Sudamérica. Además, aproximadamente el 50% de los inscriptos se autodefinieron como personal de universidades y el resto como pertenecientes al sector privado y particulares. Estas cifras son alentadoras y permiten suponer que la convocatoria del proyecto es amplia y genera interés en un público diverso tanto geográfica como profesionalmente.

III. MATERIAL DIDÁCTICO

B. Wiki y documentación

Este aspecto del proyecto está compuesto por todo el material didáctico, bajo la forma de una plataforma colaborativa (wiki) [12] con herramientas y tutoriales, y un repositorio de ejemplos que complementan al hardware y permiten la realización de aportes por parte de la red de colaboradores.

La Wiki desarrolla conceptos fundamentales de diseño digital: circuitos combinatoriales, secuenciales, máquinas de estados finitos y memorias, así como también conceptos básicos sobre diseño en FPGA: sintaxis de lenguajes, flujos de trabajo de verificación y síntesis. Se presenta en la Wiki una formación inicial que provee las bases necesarias para iniciarse en el desarrollo para FPGA. Por último, se introduce la EDU-CIAA-FPGA y las herramientas propuestas para su uso y configuración.

En complemento con la wiki, se ofrece un repositorio de ejemplos orientados al uso de la EDU-CIAA-FPGA, cuyo objetivo es facilitar el acercamiento a estas tecnologías mediante una propuesta práctica y bien documentada. De esta forma, aquellas personas interesadas en el diseño digital en FPGA, tendrán a su disposición una base teórica sobre el tema, y además los recursos necesarios, tales como ejemplos, hardware e instalación de herramientas para el aprendizaje experimental.

C. Entorno de desarrollo

Uno de los principales factores que dificulta la entrada al diseño digital para FPGA es la profunda dispersión existente en herramientas y entornos de desarrollo, simulación e implementación. Por un lado, cada fabricante ofrece una *suite* de diseño específica para las plataformas que comercializa, que suelen tener una complejidad de uso y/o un costo relativamente elevado para las personas que están iniciando en la temática. Por el otro, hay varios proyectos de software libre orientados a reemplazar estas aplicaciones, pero requieren que el usuario final gestione la instalación y configuración manual de una gran cantidad de librerías y dependencias.

Para evitar las dificultades mencionadas, se creó un entorno de desarrollo basado en herramientas libres y modulares. Esta solución integra el editor de texto personalizable Atom [15] y la plataforma de desarrollo Docker [16]. La Fig. 7 muestra la arquitectura del entorno propuesto.

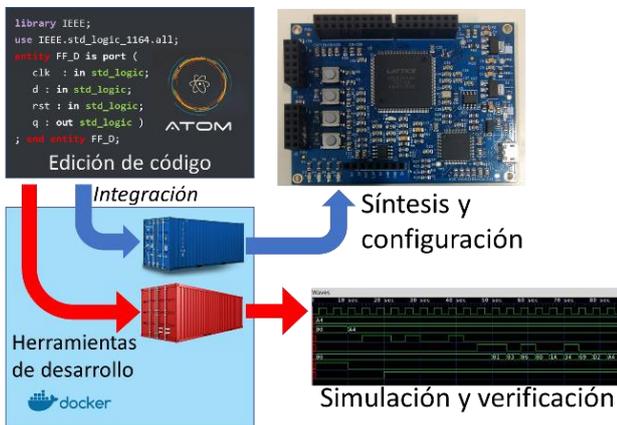


Fig. 7. Entorno de Desarrollo

Docker gestiona los contenedores modulares de código y librerías que forman las herramientas de simulación y configuración: Yosys [17]-[19], Arachne-pnr [20], NextPNR[21], Icestorm [22], GHDL [23], Icarus Verilog [24], Vunit [25] y GtkWave [26]. Al tratarse de módulos autocontenidos, requieren un mínimo esfuerzo de instalación.

Por otro lado, el editor Atom provee una interfaz gráfica y menús personalizados que permiten al usuario ejecutar las herramientas de desarrollo. También se pueden instalar *packages* adicionales para resaltar y completamiento de sintaxis en VHDL y Verilog.

Si bien el entorno de desarrollo recomendado por el proyecto es el descrito en esta sección, la Wiki del proyecto también incorpora documentación referida al uso de otras herramientas, tanto libres como propietarias. Por ejemplo, se puede utilizar la suite de desarrollo provista por *Lattice Semiconductors* [27] y compuesta por:

- *Aldec Active HDL* [28] para diseño y verificación de hardware digital en VHDL/Verilog.
- *iCEcube2* [29] para síntesis, Place and Route y análisis de la implementación física resultante.
- *Diamond Programmer* [30] para descarga del bitstream a la memoria Flash.

No obstante, el uso de las herramientas libres que se recomienda posee algunas ventajas respecto de utilizar las herramientas privativas. En primer lugar, se puede configurar la totalidad de los recursos físicos de la FPGA, duplicando la cantidad de celdas lógicas disponibles. Esto se debe a que la capacidad nominal de 3520 LUTs indicada en la Tabla I está impuesta artificialmente por el software de Lattice, pero el circuito integrado posee en realidad cerca de 8000 LUTs [31]. En segundo lugar, el uso de herramientas libres facilita el desarrollo y la integración de nuevas herramientas aportadas por la comunidad. Además, al no quedar el entorno cautivo de una determinada marca de dispositivos o proveedor de software, se mejora la portabilidad y reusabilidad de las herramientas. Otra ventaja es el menor peso y tiempo de instalación de las herramientas libres, sobre todo mediante el enfoque propuesto, basado en Atom y Docker. Por último, en torno a estas herramientas libres se ha conformado una amplia comunidad de aportantes, lo que facilita la colaboración relacionada con diseños, softcores, instalación de las herramientas, entre otras. La placa también ha tenido aceptación en la comunidad internacional de FPGA *Open Source*, siendo un ejemplo de ello su integración en el toolchain libre *Apio* [32] y en el entorno de desarrollo gráfico *iCEstudio* [33]. Si bien a la fecha de presentación de esta publicación no se lista en los sitios web oficiales, en la página del proyecto se presenta un tutorial detallado para la instalación y utilización, incluyendo varios casos de uso [34]. Esta herramienta permite crear diseños digitales de forma gráfica mediante la interconexión de bloques predefinidos, lo que facilita el acceso a la tecnología FPGA para interesados sin experiencia en lenguajes descripción de hardware. Esto puede aplicarse, por ejemplo, para introducir la temática en el nivel secundario o como introducción previa al aprendizaje de lenguajes de descripción de hardware. La Fig. 8 muestra un ejemplo de prueba para la EDU-CIAA-FPGA en Icestudio.



Fig. 8. Prueba de integración al software Icestudio

D. Repositorio de ejemplos

El repositorio de ejemplos del proyecto [13] está estructurado para ofrecer una progresión de proyectos de dificultad incremental. Cada ejemplo está estructurado como una carpeta donde se puede encontrar:

- a) Código fuente del ejemplo en HDL
- b) Scripts auxiliares para su prueba o implementación

c) Documentación explicando cómo se diseñó el ejemplo y el paso a paso para su simulación, verificación e implementación.

El repositorio abarca ejemplos de circuitos combinacionales o secuenciales sencillos y también diseños más avanzados de máquinas de estado, generadores de señales o interfaces de comunicación. Además, al tratarse de un repositorio abierto para la colaboración, se espera ampliarlo a través de nuevos ejemplos y proyectos aportados por terceros en función de su experiencia e interés. En ese sentido, se han elaborado lineamientos respecto a cómo deben elaborarse los ejemplos para mantener la homogeneidad del proyecto.

IV. RED DE COLABORADORES

En el espíritu del proyecto CIAA se plantea un enfoque abierto, colaborativo y descentralizado. Por ello, desde un principio en la especificación del diseño de la placa se encuestaron a profesionales relacionados con la temática en el ámbito de la docencia universitaria, la investigación y las empresas para que aporten propuestas concretas para delinear el diseño. En particular, participaron de estas consultas, profesionales las empresas Satellogic, Jotatec, de distintos ámbitos de la docencia e investigación universitaria (FIUBA, UNSAM, UTN FRBA, UTN FRBB, UTN FRSN, UNER, entre otras) y del sistema científico tecnológico nacional (INTI, CITEDEF, entre otros [35]).

V. CONCLUSIONES

Se logró realizar un proyecto educativo integral para el aprendizaje de tecnología FPGA en tres ejes: hardware, material didáctico y establecimiento de una red de colaboradores. Respecto al hardware, se concretó una producción inicial de 100 unidades, consiguiendo así la disponibilidad en el mercado local a un precio final de US\$ 85 por placa, el cual la coloca en una posición competitiva respecto a otras opciones comerciales [36]. Las herramientas que brinda el proyecto están orientadas a ofrecer posibilidades de uso para distintos niveles: estudiantes secundarios, terciarios, de carreras de grado, o profesionales. El uso de las herramientas que aquí se presentan no se restringe a un área en particular, sino que busca ser el nexo que permita a profesionales y estudiantes adentrarse en un campo que hoy en día, y en nuestra región, no se encuentra ampliamente difundido.

Actualmente, el proyecto EDU-CIAA-FPGA se ha publicado y se han recibido adhesiones de más de un centenar de profesionales del campo de los desarrolladores independientes, la universidad y empresas de diferentes rubros. Se han liberado los repositorios de ejemplos, hardware y material didáctico. El trabajo futuro se concentrará en ampliar el proyecto de manera horizontal, sumando colaboradores de distintos sectores con el objetivo de lograr un crecimiento en las diferentes áreas del proyecto, tales como: hardware, ejemplos, proyectos de aplicación, material didáctico, cursos de grado, entre otras.

VII. REFERENCIAS

- [1] Xilinx, San Jose, California, USA. *Introduction to FPGA Design with Vivado High-Level Synthesis (2019)*. Acceso: 26 Nov 2020 [En línea]. Disponible: https://www.xilinx.com/support/documentation/sw_manuals/ug998-vivado-intro-fpga-design-hls.pdf
- [2] Xilinx “Applications” Xilinx.com <https://www.xilinx.com/applications.html> (accedido 26 Nov 2020)
- [3] N. Hemsoth, T. Prickett Morgan. *FPGA Frontiers: New Applications in Reconfigurable Computing*, edición 2017. High Point, NC, USA: Next Platform Press, 2017
- [4] S. M. Trimberger, “Three Ages of FPGAs: A Retrospective on the First Thirty Years of FPGA Technology,” en “Proceedings of the IEEE”, vol. 103, no. 3, pp. 318-331, Marzo 2015, doi: 10.1109/JPROC.2015.2392104
- [5] Arrow “FPGA vs CPU vs GPU vs Microcontroller: How Do They Fit into the Processing Jigsaw Puzzle?” arrow.com <https://www.arrow.com/en/research-and-events/articles/fpga-vs-cpu-vs-gpu-vs-microcontroller> (accedido 26 Nov 2020)
- [6] S. Tropea. “FPGAs Libres: hardware, simulación y síntesis” presentado en el SASE, CABA, AR, 9 al 11 de Agosto 2017.
- [7] Computadora Industrial Abierta Argentina <http://www.proyecto-ciaa.com.ar/> (accedido 30 May 2020)
- [8] M. A. Heredia, F. S. Larosa, R. A. Ghignone, J. M. Cruz, N. Dassieu Blanchet, “Kit de FPGA para la enseñanza de lenguajes de descripción de hardware”, en *UEA, BA, AR, 25 al 26 de Septiembre 2019*.
- [9] IPC-2221A Task Group. *IPC-2221A Generic Standard on Printed Board Design*. Mayo, 2003.
- [10] “iCE40 LP/HX Family”, Lattice Semiconductor, Septiembre 2018. Disponible: http://www.latticesemi.com/-/media/LatticeSemi/Documents/DataSheets/iCE/FPGA-DS-02029-3-5-iCE40-LP-HX-Family-Data-Sheet.ashx?document_id=49312 (accedido 26 Nov 2020)
- [11] Mouser, “iCE40HX4K-TQ144”, Mouser.com <https://www.mouser.com/ProductDetail/Lattice/iCE40HX4K-TQ144> (accedido 26 Nov 2020)
- [12] Ramiro Ghignone, “FPGA para todos”, Gitlab.com Disponible: <https://gitlab.com/RamadrinanG/wiki---fpga-para-todos/-/wikis/FPGA-para-Todos> (accedido 30 May 2020)
- [13] Ramiro Ghignone et al, “Repositorio de ejemplos de la EDU CIAA FPGA”, Gitlab.com <https://gitlab.com/educiaafpga/ejemplos> (accedido 16 Oct 2020)
- [14] Open Source Initiative “The 3-Clause BSD License” opensource.org Disponible: <https://opensource.org/licenses/BSD-3-Clause> (accedido 16 Oct 2020)
- [15] Atom “Atom Flight Manual” atom.io Disponible: <https://flight-manual.atom.io/> (accedido 26 Nov 2020)
- [16] Docker Disponible: <https://www.docker.com/> (accedido 30 May 2020)
- [17] C. Wolf, J. Glaser. *Yosys, “A Free Verilog Synthesis Suite”*, en “Proceedings of Austrochip”, Linz, Austria, 10 de Octubre de 2013. Disponible: <http://www.clifford.at/yosys/files/yosys-austrochip2013.pdf> (accedido 26 Nov 2020)
- [18] C. Wolf. “Design and Implementation of the Yosys Open Synthesis Suite”. Bachelor Thesis, Vienna University of Technology, 2012
- [19] C. Wolf, “Yosys Open Synthesis Suite” Clifford.at Disponible: <http://www.clifford.at/yosys/> (accedido 18 Jul 2020)
- [20] C. Wolf et al, “Arachne-pnr” Github.com Disponible: <https://github.com/YosysHQ/arachne-pnr> (accedido 30 May 2020)
- [21] C. Wolf, “NextPNR: A portable FPGA Place and Route Tool” Github.com Disponible: <https://github.com/YosysHQ/nextpnr> (accedido 16 Oct 2020)
- [22] C. Brown. “Project IceStorm” Clifford.at Disponible: <http://www.clifford.at/icestorm/> (accedido 30 May 2020)
- [23] G. Tristan “GHDL” ghdl.free.fr Disponible: <http://ghdl.free.fr/> (accedido 30 May 2020)
- [24] W. Stephen “Icarus Verilog” Disponible: <http://iverilog.icarus.com/> (accedido 30 May 2020)
- [25] L. Asplund “VUnit” github.io Disponible: <https://vunit.github.io/> (accedido 30 May 2020)

- [26] GTKWave “*GTKWave 3.3 Wave Analyzer User’s Guide*”, 14 Nov 2020
 Disponible: <http://gtkwave.sourceforge.net/gtkwave.pdf> (accedido 26 Nov 2020)
- [27] Lattice, “*Lattice Semiconductor*”
 Disponible: <https://www.latticesemi.com> (accedido 16 Oct 2020)
- [28] Aldec Inc “*Active-HDL*” aldec.com
 Disponible: https://www.aldec.com/en/products/fpga_simulation/active-hdl (accedido 16 Oct 2020)
- [29] Lattice “*iCEcube2 Design Software*” latticesemi.com
 Disponible: <http://www.latticesemi.com/iCEcube2> (accedido 16 Oct 2020)
- [30] Lattice “*Programmer and Deployment Tool*” latticesemi.com
 Disponible: <http://www.latticesemi.com/programmer> (accedido 16 Oct 2020)
- [31] Ed Brindley, “Using the Programming FPGAs book with IceStorm and myStorm BlackIce”, Disponible: <https://maidavale.org/blog/using-simon-monks-programming-fpgas-book-with-icestorm/>
- [32] Juan Gonzalez-Gomez, Jesús Arroyo Torrens et al, “*Apio IDE*” Github.com
 Disponible: <https://github.com/FPGAwards/apio-ide> (accedido 16 Oct 2020)
- [33] Jesús Arroyo Torrens, Carlos Venegas Arrabé, Juan González Gómez “*Icestudio*” Icestudio.io
 Disponible: <https://icestudio.io> (accedido 16 Oct 2020)
- [34] Ramiro Ghignone et al, “*iCEstudio*”, Gitlab.com
<https://gitlab.com/RamadrianG/wiki---fpga-para-todos/-wikis/iCEstudio> (accedido 27 Nov 2020)
- [35] Facundo Larosa, Marín Heredia, Ramiro Ghignone, Federico Vazquez Saraullo, Hernán Mendes Gouveia, Martín Fernández, “*EDU-CIAA-FPGA*” proyecto-ciaa.com.ar
 Disponible: <http://www.proyecto-ciaa.com.ar/devwiki/doku.php?id=desarrollo:edu-fpga> (accedido 12 Jul 2020)
- [36] Xilinx “*Boards and Kits Targeting Cost-Optimized Design*” Xilinx.com
 Disponible: <https://www.xilinx.com/products/boards-and-kits/cost-optimized-design.html> (accedido 18 Jul 20)