

# FPGA Implementation of a Critical Railway Interlocking System

M. Menéndez, F. Larosa, R. Ghignone, N. Alvarez, and A. Lutenberg

**Abstract**—An interlocking is a railway system that automatically controls that changes in routes are safely managed, avoiding train crashes and derailments. This article presents an analysis of the state of the art technology used in several commercial interlocking equipment. Based on this analysis, design and implementation of an interlocking system architecture in FPGA technology is proposed. Appropriate techniques are applied to achieve the required level of security. The scalability of the design is also analyzed and conclusions are presented.

**Index Terms**—FPGA, Interlocking, Railway, Critical systems.

## I. INTRODUCCIÓN

LOS sistemas de enclavamiento tienen por función garantizar el accionamiento seguro de los dispositivos de señalamiento ferroviario asociados a las maniobras de cambio de vía [1], evitando el choque de trenes. En Argentina los sistemas de enclavamientos tienen entre 40 y 100 años de antigüedad, por lo que muchos han agotado su vida útil y deben ser reemplazados. La inversión a realizar en sistemas de enclavamientos supera los 1.000 millones de dólares y se trata en su totalidad de sistemas importados. Si se suma el mantenimiento posterior se duplica esa cifra en pocas décadas de uso [2], [3]. En este contexto Trenes Argentinos [4] requirió a nuestro grupo el diseño de sistemas electrónicos de enclavamientos.

Con el objetivo de diseñar sistemas electrónicos de enclavamientos se realizó una extensa búsqueda bibliográfica. Sin embargo, no se encontraron artículos que detallen el estado del arte en cuanto a las tecnologías que se utilizan en la industria, excepto por [1] que aborda un análisis superficial de los sistemas de enclavamientos de la India y dada la fecha de publicación la tecnología empleada resulta obsoleta hoy en día. Es por eso que se realizó en primer lugar un análisis de la bibliografía actual para determinar los puntos de partida con vistas a realizar una implementación propia.

Diversos artículos abordan parcialmente el problema a resolver. En [5] se analiza la problemática de definir un sistema de enclavamiento electrónico pero sin utilizar una FPGA dedicada. Un artículo reciente [6] abarca la modelización

de cada parte de la topología ferroviaria según el grado de interconexión entre las secciones, pero sin analizar la implementación de los sistemas en campo. Otros trabajos intentan automatizar parcialmente el modelo [7] o la verificación automática [8], pero sin generalizar a casos más complejos.

A partir del relevamiento realizado sobre una decena de empresas [9]-[19] se encontró que las mismas se localizan en su mayoría en Europa (73%), Norteamérica y parte de Asia. No hay empresas entre las relevadas que diseñen, fabriquen o comercialicen sistemas de enclavamiento en el resto de América, África u Oceanía. Las empresas analizadas recurren en todos los casos al uso de electrónica programable dedicada, ya sea con computadoras o microcontroladores específicos, haciendo uso de redundancia de software y hardware [20].

En la mayoría de los casos las empresas informan que los desarrollos son realizados en sus propios centros de I+D. Sin embargo, son muy pocas las empresas que informan el modelo que han adoptado, las herramientas utilizadas para su implementación y/o verificación automática. Es por eso que los detalles discutidos en el presente artículo cobran especial valor a la hora de actualizar y expandir la infraestructura ferroviaria en Latinoamérica, al abordar de forma integral el análisis de los requerimientos, exponer las decisiones de diseño y los detalles de la implementación de forma transparente y rigurosa.

La redundancia en sistemas críticos se define por medio de la abreviatura NooM, donde M representa a la cantidad de módulos de medición o decisión que posee el sistema y N la cantidad de dichos módulos que deben funcionar correctamente para que el sistema opere normalmente [21]. De la investigación realizada surge que el 66% de las empresas utiliza una redundancia 2oo2 o 2oo3 para alcanzar los niveles de seguridad requeridos [4]-[10],[13]-[19]. Sólo una pequeña porción de las mismas utiliza redundancias 1oo2 [11] o 2oo4 [12]. En consecuencia, se puede afirmar que una redundancia 2oo2 o 2oo3 es representativa de los sistemas analizados y puede utilizarse como esquema de partida para un diseño propio. En este trabajo no se aborda la implementación de la redundancia, sino que se propone una arquitectura automatizada en contraposición a los diseños de otras empresas que se realizan manualmente para cada caso en particular.

En la Sección II se presentarán los elementos básicos que componen la infraestructura ferroviaria y el enclavamiento. En la Sección III se abordarán los diferentes enfoques de implementación del sistema de enclavamiento. En la Sección IV se propondrá una estructura de capas para el sistema de enclavamiento. La Sección V profundizará en varios conceptos de diseño asumidos. El diseño es descrito en detalle en la

This work was partially supported by the UBA-PDE N° 26/2019.

M. Menéndez is with CONICET-GICSAFe and UBA (email: menendez91@live.com.ar).

A. Lutenberg is with CONICET-GICSAFe and UBA (email: lse@fi.uba.ar).

N. Alvarez is with CONICET-GICSAFe, UNSAM and UBA (email: nalvarez@gmail.com).

F. Larosa is with CONICET-GICSAFe and UTN (email: facundolarosa@gmail.com).

R. Ghignone is with CONICET-GICSAFe and UTN (email: rghignone@frh.utn.edu.ar).

Corresponding author: M. Menéndez

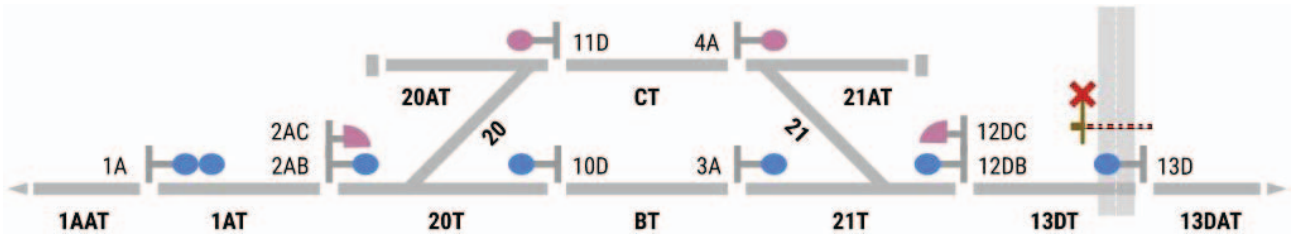


Fig. 1. Diagrama de topología de un bypass ferroviario.

Sección VI, continuando en la Sección VII con las pruebas realizadas, para finalizar en la Sección VIII con las conclusiones del artículo

II. ELEMENTOS FERROVIARIOS

Un sistema de enclavamientos se compone de los siguientes elementos:

- Secciones de vías: son los distintos tramos en los que se separa la vía. Cada uno de esos tramos está provisto de un detector, denominado circuito de vía, que permite detectar si hay una formación ocupando la sección.
- Semáforos: se utilizan para indicar al conductor del tren si puede o no ingresar a la siguiente sección de vía y a qué velocidad. Presenta diferentes colores denominados aspectos. Por ejemplo, rojo para detener la marcha, amarillo para marcha con precaución y velocidad reducida, y verde para indicar avance con velocidad normal. En la Fig. 1 se ilustran semáforos de circulación directa (azul) para circular por la vía principal y semáforos de maniobras (rosa) para ingresar o egresar de la secciones de vía 20AT, CT y 21AT.
- Pasos a nivel: cada cruce peatonal o vehicular sobre el trazado ferroviario debe estar protegido por un laberinto con campanilla y una barrera, respectivamente. Así se garantiza que la formación circula por vías despejadas de peatones y vehículos. En la Fig. 1 se ilustra un paso a nivel con barrera automática sobre la sección de vía 13DT.
- Máquinas de cambios: se utilizan para cambiar la dirección de circulación en una intersección de vías. En la Fig. 1 los cambios permiten circular desde las secciones 20T y 21T hacia 20AT y 21AT, respectivamente.
- Ruta: es el recorrido que puede realizar la formación entre un semáforo y el siguiente. Puede estar compuesto de una o más secciones de vías.

Los elementos que intervienen en cada ruta y su estado para poder habilitar o no la ruta se describen en la denominada "tabla de enclavamientos" [22]. Se incluye solo a modo de ejemplo una tabla de enclavamientos simplificada (Tabla I).

Los sistemas de enclavamiento deben ser a prueba de peligros en caso de fallos (fail-safe). Es decir, si ocurre una falla se debe evolucionar a un estado tal que no presente peligro para la seguridad de las personas. Por ejemplo, si por causa de una falla se recibe la orden de que un semáforo debe estar verde y rojo a la vez, entonces el sistema debe decidir por el aspecto rojo.

TABLA I  
TABLA DE ENCLAVAMIENTOS (SIMPLIFICADA)

Ruta	Señal de entrada	Señal de salida	Señales opuestas	Circuito de vía en ruta	Cambios requeridos	Paso a nivel
1	1A	2AB 2AC	-	1AT	-	Alto
2	2AB	3A	10D	20T BT 20T 20AT	20(N)	Bajo
3	2AC	4A	11D 12DB	CT 21T 13DT	20(R)	Bajo
4	3A	-	12DC 13D	13DAT 21AT 21T	21(N)	Bajo
5	4A	-	12DB 12DC 13D	13DT 13DAT	21(R)	Bajo

Un sistema es vital si para todos sus modos de falla no existe ninguno que sea fatal.

III. ENFOQUES DE IMPLEMENTACIÓN

Si bien las empresas relevadas proveen información sobre la arquitectura general del sistema [9]-[19], no se mencionan detalles de implementación específicos en cuanto a la arquitectura del hardware ni los principios básicos de diseño adoptados. Se analizaron diversos artículos [23]-[26] donde se puede observar la existencia de dos enfoques principales para obtener la arquitectura: un enfoque geográfico y un enfoque funcional. Ambos enfoques han sido aplicados con éxito a la hora de desarrollar sistemas electrónicos de enclavamiento y se describen a continuación.

A. Enfoque Geográfico

El enfoque geográfico se basa en modelar los objetos reales de campo. Para cada semáforo, barrera, cambio de vía o circuito de vía existe un bloque que lo modela. La interconexión de los bloques obedece a la interconexión que tienen los objetos reales a los que representan. De esta forma el modelo depende de la topología de la red (layout) y no de la tabla de enclavamientos [25]-[26].

El flujo de desarrollo que se aplica en el enfoque geográfico se describe gráficamente en la Fig. 2. Partiendo del layout que describe la topología de la red se obtiene un grafo donde se modela cada circuito de vía con un nodo y cada conexión entre los mismos con una arista. Por medio de un algoritmo se determina que circuitos de vía son extremos de la red, cuales poseen cambios de vías y cuales deben tener semáforos y de qué tipo (para circulación directa o maniobra ante un cambio de vía). Los pasos a nivel se deben añadir manualmente.



Fig. 2. Diagrama de trabajo en el enfoque geográfico.

A partir de la lógica obtenida en base al grafo se genera una tabla de enclavamientos que enumera todas las rutas posibles.

**B. Enfoque Funcional**

El enfoque funcional sigue el flujo de trabajo descrito en la Fig. 3. Tiene como premisa el modelar las funciones que se quiere que tenga el sistema, partiendo de la interpretación que el diseñador del sistema realizó en la tabla de enclavamientos [25], [26].

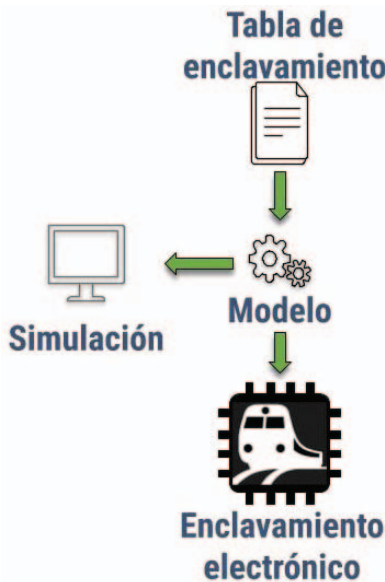


Fig. 3. Diagrama de trabajo en el enfoque funcional.

Las rutas se definen según las necesidades logísticas del operador ferroviario, por lo que puede definirse una cantidad de rutas menor a la que soporte la topología ferroviaria sobre la cual se opera.

Los dos enfoques propuestos, geográfico y funcional, poseen ventajas y desventajas en lo referente a su implementación. A modo de resumen de las ventajas y desventajas de los dos enfoques desarrollados se presenta la Tabla II.

TABLA II  
TOPOLOGÍAS FERROVIARIAS DE REFERENCIA

	Funcional	Geográfica
Tabla de enclavamiento	Necesaria desde el inicio	Generada como subproducto.
Comportamiento	Definido por la tabla de enclavamiento	Definido mediante el grafo que modela la red.
Modela	Rutas	Objetos reales(Semáforos, barreras,cambios).
Contempla	Solo las rutas que se necesitan	Todas las rutas soportadas por la red

En este trabajo se analiza la implementación del enfoque funcional sobre un kit de desarrollo FPGA. A modo de referencia se presenta en la Tabla III la cantidad de elementos aproximados que tienen distintas topologías y estaciones de referencia. Estaciones más complejas requieren una mayor cantidad de recursos de la FPGA y el objetivo de este artículo es evaluar las ventajas y desventajas de la implementación de diferentes arquitecturas y el correcto dimensionamiento de la plataforma a utilizar.

TABLA III  
TOPOLOGÍAS FERROVIARIAS DE REFERENCIA

Topología	Circuitos de vía	Rutas	Semáforos	Barreras	Cambios
Bypass (Fig. 1.)	10	8	8	0	2
Belgrano R	15	12	12	3	1
Temperley	40	40	40	0	10
Alejandro Korn	60	60	60	0	15

**IV. ESTRUCTURA PROPUESTA**

De acuerdo al análisis bibliográfico descrito en la Sección I se llegó a la conclusión de que una estructura de tres capas como la de la Fig. 4 es representativa de la mayoría de los casos estudiados.

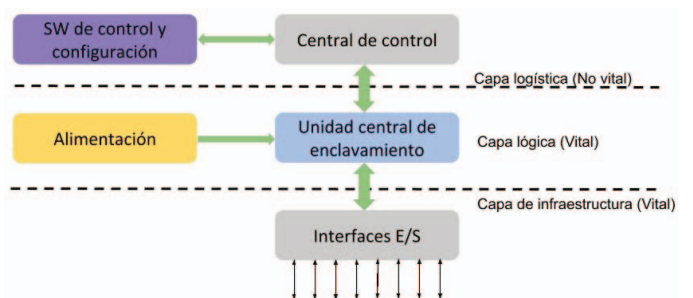


Fig. 4. Capas del sistema de enclavamiento.

En la Tabla IV se resume la descripción de las capas que conforman el sistema. A continuación se describen las diferentes capas que se ilustran en la Fig. 4.

TABLA IV  
CAPAS DEL SISTEMA DE ENCLAVAMIENTO

Capa	Tipo	Incluye	Descripción
Infraestructura	Vital	Módulo de adquisición	Lectura de las señales con redundancia para evitar valores erróneos.
Lógica	Vital	Alimentación y unidad central de enclavamiento	Procesamiento de las señales y enclavamiento de estados.
Logística	No vital	Interfaz gráfica y terminal de comandos	Presenta estado del sistema, ingreso de peticiones de ruta y cambios de modo

#### A. Capa de Infraestructura

La capa de infraestructura es la encargada de la lectura y escritura de las señales del campo, por ejemplo los aspectos de los semáforos, posición de los cambios de vías y barreras de pasos a nivel y la ocupación de los circuitos de vía.

#### B. Capa Lógica

Esta capa se ocupa del enclavamiento propiamente dicho, analizando las señales de entrada, los estados actuales y las órdenes pedidas por el operador, obteniendo una salida segura y compatible con lograr la mayor densidad de trenes en el sistema sin conflictos entre ellos ni con la infraestructura, garantizando niveles de seguridad adecuados. El presente artículo se centra en la implementación del sistema en esta capa en particular.

Las empresas relevadas utilizan casi en su totalidad microprocesadores para implementar la capa lógica de sus enclavamientos. En este trabajo se explora la alternativa de utilizar tecnología FPGA ya que por la naturaleza concurrente del sistema bajo estudio se espera obtener mayor flexibilidad en la etapa de implementación para lograr el nivel de redundancia requerido [21],[27], [28].

#### C. Capa Logística

La capa logística tiene como función enviar a la capa lógica los comandos que solicitan los operadores ferroviarios. Entre los posibles comandos que puede enviar el operador se pueden nombrar como ejemplos: la solicitud o cancelación de rutas, el accionar un semáforo en particular o conocer el estado global del señalamiento.

### V. CONCEPTO DE DISEÑO

Para la implementación del diseño se consideraron dos posibles alternativas. Por un lado, la posibilidad de resolver el problema utilizando bloques diseñados específicamente para cada sistema de enclavamientos. Por otro lado, desarrollar una arquitectura que haga uso de bloques simples y genéricos, aplicables a distintos sistemas de enclavamientos.

Resulta de interés plantear un paralelismo entre los dos cursos de solución y los diferentes enfoques de arquitecturas de microprocesadores: RISC (Reduced Instruction Set Computer) y CISC (Complex Instruction Set Computer) [30], [31].

El uso de bloques diseñados específicamente para una topología de enclavamiento determinada puede equipararse a un procesador CISC que implementa instrucciones complejas optimizadas para una aplicación dada. Esto brinda al usuario de la arquitectura una solución óptima, pero a la vez restringe la genericidad del diseño y representa un esfuerzo de desarrollo que podría no aprovecharse al variar las especificaciones de la aplicación.

En contraposición, el uso extensivo de bloques simples y genéricos puede equipararse a un procesador RISC que implementa instrucciones sencillas que apuntan a resolver un amplio campo de problemas, pero que genera una mayor complejidad al momento de combinarlas para resolver una aplicación determinada.

A priori, cualquiera de los dos enfoques resulta válido. Sin embargo, se prefirió utilizar el enfoque de bloques simples y genéricos, dada la gran diversidad de sistemas de enclavamiento que es necesario implementar (ver por ejemplo la Tabla III).

El uso de bloques simples y genéricos resulta en apariencia más eficiente ya que se requiere menor tiempo de diseño y ensayos, facilitando además la reusabilidad de los módulos desarrollados en sistemas de enclavamiento de complejidad y topologías diversas.

Por otra parte, de forma análoga a las arquitectura RISC, la dificultad al utilizar bloques simples y genéricos pasa por realizar una adecuada combinación e implementación de los bloques, ya que se pretende reutilizar estos bloques para resolver una gran variedad de problemas.

En este trabajo, la generación de la arquitectura completa se realizó por medio de un generador automático de código desarrollado en lenguaje Python. El generador automático desarrollado recibe como entrada la tabla de enclavamientos y produce el código en VHDL que combina los bloques simples y genéricos necesarios para implementar el sistema de enclavamientos. Este concepto se describe en detalle en la sección VI. Diseño.

### VI. DISEÑO

Por razones de disponibilidad se utilizó el kit de desarrollo Arty Z7 (Fig. 5), el cual posee 17600 LUT's, 35200 FF's, 32 BUFG's y 100 IOB's [32]. Se lo utilizó como base para sintetizar el diseño y extraer conclusiones que permitan dimensionar los recursos lógicos necesarios para un desarrollo de estas características.

La metodología de trabajo siguió el modelo de desarrollo guiado por pruebas, del inglés, test-driven development, (TDD) [33]. Para implementar el desarrollo guiado por pruebas, cada módulo creado debe ser precedido por una serie de ensayos a cumplir que se corresponden uno a uno con los requerimientos del sistema. De esta forma se construye un sistema bajo el método de programación bottom-up para luego integrar los módulos funcionales progresivamente.

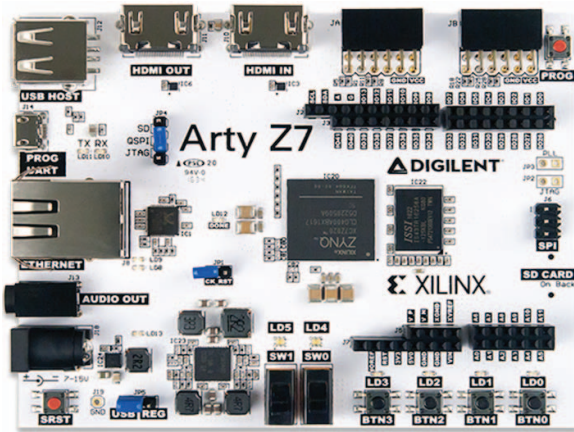


Fig. 5. Arty Z7 10 de Digilent.

Para ensayar el código en VHDL se deben implementar test-benchs que consisten en ensayos escritos también en VHDL, lo que eleva la complejidad de las pruebas y el tiempo de desarrollo. No obstante, utilizando la herramienta COCOTB [34] pueden disminuirse los tiempos de desarrollo y elevar el nivel de calidad de los ensayos. Esta herramienta permite desarrollar los ensayos en Python y aplicarlos por medio de diferentes compiladores. Dentro de la capa lógica se destaca la unidad central de enclavamiento por ser el núcleo del procesamiento y la sección del sistema que se implementa íntegramente en FPGA. Se decidió subdividir la capa lógica en tres módulos, a cada uno de los cuales se le asignó una función específica que se describe en las subsecciones VI.A, VI.B y VI.C.

#### A. Módulo Selector

El módulo selector es el encargado de recibir las señales de la capa de infraestructura y logística y asignar cada paquete de señales a la máquina de estados de cada objeto y ruta correspondiente.

En la Fig. 6 se ejemplifica el caso particular de la barrera 1 de una topología cualquiera. De todas las señales que se leen en campo tales como el estado de los circuitos de vía (CV), estado de los semáforos (SEM), estado de los pasos a nivel (PAN), estados de las máquinas de cambios (MDC) y las rutas requeridas (RUT), el módulo selector asigna al bloque “selector barrera 1” solo las señales que, según la tabla de enclavamientos, son necesarias para el funcionamiento de la barrera 1.

Luego esas señales asociadas a la barrera 1 se reparten según cuales son necesarias para cada ruta que requiera el uso de la barrera 1. En este ejemplo son las rutas 1, 2 y 3 representadas por las flechas roja, naranja y amarilla respectivamente.

El bloque selector se construye de forma análoga para todos los elementos que se leen del campo, siguiendo las indicaciones de la tabla de enclavamientos.

#### B. Módulo de Máquinas de Estado Finitos (MEFs)

El módulo de máquinas de estados finitos comprende todas las máquinas de estado necesarias para la operación del

enclavamiento. Una máquina de estados es un modelo de comportamiento de un sistema con entradas y salidas, en donde las salidas dependen tanto de las señales de entradas actuales como de su estado previo.

Siguiendo con el ejemplo de la Fig. 6, como hay tres rutas que quieren influir sobre el objeto barrera 1 (ruta 1, 2 y 3 en el ejemplo) y se quiere mantener la concurrencia del sistema, se replica tres veces la máquina de estados finitos de la barrera 1. La única diferencia entre estas tres máquinas de estados finitos es que reciben las entradas asociadas a diferentes ruta (rutas 1, 2 y 3 en el ejemplo).

#### C. Módulo Consolidador

El módulo consolidador tiene como función decidir cuál de las máquinas de estado impondrá su valor a la salida, aplicando el criterio de elegir el estado más seguro.

Siguiendo el ejemplo de la Fig. 6, las máquinas de estado de barrera 1 correspondientes a las rutas 1 y 2 piden que la barrera se encuentre en estado alto (flechas verdes) y la correspondiente a la ruta 3 pide que la barrera se encuentre en estado bajo (flecha roja). El bloque consolidador decide que el estado más restrictivo y por lo tanto más seguro como salida del sistema es el que sugiere la máquina de estados de la ruta 3 y procede a copiar su comportamiento y enviar la señal de bajar la barrera a la capa de infraestructura para operar sobre el actuador de la barrera.

### VII. IMPLEMENTACIÓN DEL MODELO FUNCIONAL

Se implementó en VHDL el sistema de enclavamiento utilizando un enfoque funcional, dejando como parámetros genéricos los siguientes elementos:

- Cantidad de circuitos de vía
- Cantidad de rutas
- Cantidad de semáforos
- Cantidad de barreras (pasos a nivel)
- Cantidad de cambios de vías

El código en VHDL se generó automáticamente con una librería desarrollada en Python por GICSAFe (Grupo de Investigación y Control para la Seguridad y Aplicaciones Ferroviarias).

Utilizando la herramienta de implementación y síntesis de código de VIVADO [35] se cuantificó el uso de los recursos del kit de desarrollo FPGA escogido. Variando los parámetros señalados con los valores indicados en la Tabla III se generaron diversas implementaciones de estaciones ferroviarias cuyos consumos de recursos se exponen en la Tabla V y son representativos de cualquier otra plataforma al haber incluido valores absolutos y no relativos al kit utilizado.

No obstante esta limitación, es importante mencionar que la topología ferroviaria denominada bypass (Fig. 1), que tiene 8 rutas (Tabla III), es la más utilizada en trayectos de larga distancia para realizar el cruce de trenes, o en estaciones ferroviarias urbanas pequeñas. En consecuencia no obstante la limitación encontrada la solución desarrollada permite resolver la mayor parte de los sistemas de enclavamientos que se necesitan en el sistema ferroviario.

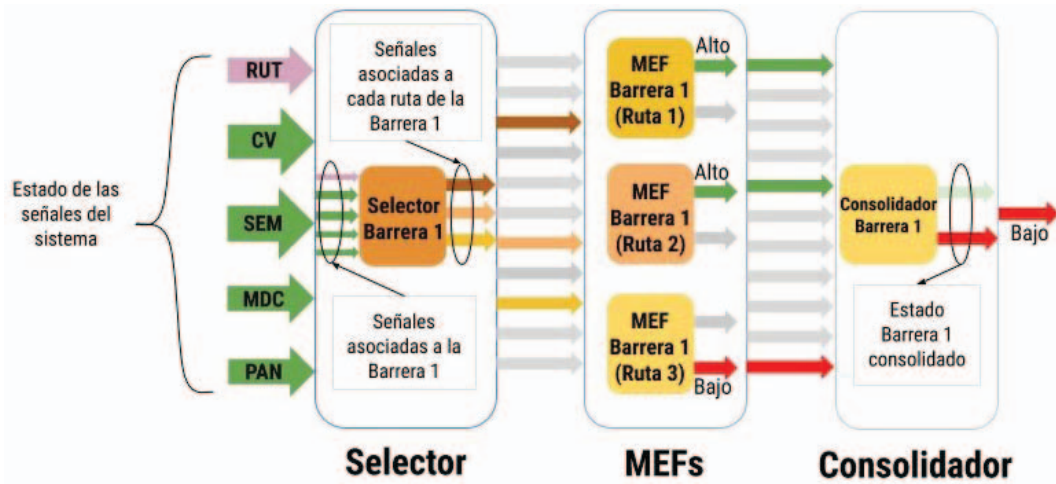


Fig. 6. Conexión de bloques en el enfoque funcional.

TABLA V  
RESULTADOS

Rutas	LUT's	FF's	IOB's	BUFG's
2	17	50	26	1
5	36	90	50	1
10	84	145	87	1
15	222	316	153	1
20	313	412	202	1
30	619	643	305	1
50	1540	905	465	1

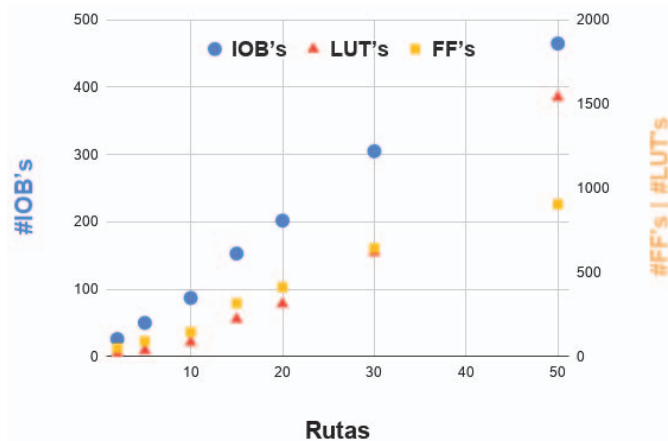


Fig. 7. Recursos en función de la cantidad de rutas.

VIII. CONCLUSIONES

La investigación realizada brindó información relevante para realizar el diseño de sistemas de enclavamiento. La técnica de diseño desarrollada permite estimar qué FPGA se podría utilizar en cada uno de los enclavamientos a implementar. De esta forma se puede concluir que una FPGA como la Arty Z7 10 es apropiada para una topología de bypass o estaciones rurales simples, que son las más utilizadas en la red ferroviaria. En tanto que se requieren mayores recursos o la adopción de

otra metodología de diseño para abordar la implementación de estaciones de mayor porte.

En el presente trabajo se hace énfasis en la automatización del proceso de síntesis, para poder abordar con mayor facilidad distintas locaciones, más allá de las particularidades de cada una de ellas. Este enfoque representa un cambio de paradigma respecto a la bibliografía, donde el proceso de diseño del sistema de enclavamiento se hace manualmente para cada topología específica.

Actualmente se está analizando el enfoque geográfico para determinar si por este medio es posible implementar sistemas de enclavamiento con un uso menor de recursos lógicos en la FPGA.

REFERENCIAS

- [1] Analysis of Railway Interlocking Systems (2019). [Online]. Disponible: [https://www.academia.edu/15034386/Analysis\\_of\\_Railway\\_Interlocking\\_Systems](https://www.academia.edu/15034386/Analysis_of_Railway_Interlocking_Systems)
- [2] I. G. Fiad, *Inversiones para la transformación en ferrocarriles*, 2019. [En línea]. Disponible en: [https://www.argentina.gob.ar/sites/default/files/20171018\\_presentacion\\_amcham.pdf](https://www.argentina.gob.ar/sites/default/files/20171018_presentacion_amcham.pdf)
- [3] G. Fiad, *Avances del plan en infraestructura ferroviaria y su impacto en el desarrollo del país*, 2019. [En línea]. Disponible en: [https://www.argentina.gob.ar/sites/default/files/20170724\\_presentacion\\_g\\_fiad.pdf](https://www.argentina.gob.ar/sites/default/files/20170724_presentacion_g_fiad.pdf)
- [4] Trenes Argentinos (2019). [Online]. Disponible: <https://www.argentina.gob.ar/transporte/trenes-argentinos>
- [5] Lutovac, T. (1998). Towards a universal computer interlocking system.
- [6] N. A. Zafar, S. A. Khan, A. Keijiro. (2012). *Towards the safety properties of moving block railway interlocking system*. International Journal of Innovative Computing, Information and Control.
- [7] Rehman, A., Latif, S. and Zafar, N. (2019). *Automata Based Railway Gate Control System at Level Crossing*. 2019 International Conference on Communication Technologies (ComTech).
- [8] Chadwick, S., James, P., Roggenbach, M. and Wetner, T. (2018). *Formal Methods for Industrial Interlocking Verification*. 2018 International Conference on Intelligent Rail Transportation (ICIRT).
- [9] Trackguard Westrace Mk II. (2014) Flexible safety processor.[Online]. Disponible: <https://www.mobility.siemens.com/mobility/global/SiteCollectionDocuments/en/rail-solutions/rail-automation/electronic-interlockings/trackguard-westrace-mk2-en.pdf>
- [10] Trackguard WESTLOCK. (2013). Cost-effective, highly available and reliable. [Online]. Disponible: <https://www.mobility.siemens.com/mobility/global/SiteCollectionDocuments/en/rail-solutions/rail-automation/electronic-interlockings/trackguard-westlock-en.pdf>

- [11] Alstoms Smartlock equipment at the heart of modular signalling project in the UK (2011).[Online]. Disponible: <https://www.alstom.com/press-releases-news/2011/3/Alstom-Smartlock>
- [12] Computer based interlocking (2019).[Online]. Disponible: <http://sts.hitachirail.com/en/products-services/business-segments/computer-based-interlocking>
- [13] Systémy pro kolejovou dopravu (2019). Elektronické Stavedlo TYP ESA® 33[Online]. Disponible: <https://www.azd.cz/admin-data/storage/get/198->
- [14] Route Control Systems (2019). LockTrac 6111 ESTW L90 [Online]. Disponible: <https://www.thalesgroup.com/en/route-control-systems>
- [15] EBI LOCK (2014). Computer-Based Interlocking System [Online]. Disponible: [http://www.infrasig.net/media/88880/pub-00074\\_ebi\\_lock\\_bro\\_prof\\_print\\_english.pdf](http://www.infrasig.net/media/88880/pub-00074_ebi_lock_bro_prof_print_english.pdf)
- [16] Bombardier Transportation (2015). Rail Control Solutions (RCS) [Online]. Disponible: [http://www.irse.org/knowledge/publicdocuments/Bombardier%2001%20IRSE\\_BT\\_RCS\\_Introduction\\_%20November%202015%20final.pdf](http://www.irse.org/knowledge/publicdocuments/Bombardier%2001%20IRSE_BT_RCS_Introduction_%20November%202015%20final.pdf)
- [17] Railway Signaling Solutions (2019). Interlocking Equipment [Online]. Disponible: <https://www.kyosan.co.jp/english/product/signal03.html>
- [18] Kyosan history (2019). [Online]. Disponible: <https://www.kyosan.co.jp/english/company/history02.html>
- [19] Hima, Rail references (2019). COTS Rail Applications[Online]. Disponible: <https://www.hima.com/en/industries-solutions/cots-rail-references>
- [20] Shimamura, K., Yamaguchi, S., Kanekawa, N., Miyazaki, N., Yamada, H., Takahashi, Y., Hirotsu, T., Tomobe, K., Satoh, K., Hotta, T. and Fujita, R. (2002). A fail-safe microprocessor using dual synthesizable processor cores. AP-ASIC'99. *First IEEE Asia Pacific Conference on ASICs (Cat. No.99EX360)*.
- [21] R. Dobias and H. Kubatova, "FPGA based design of the railways interlocking equipments" *Euromicro Symposium on Digital System Design*, 2004. DSD 2004., 2004.
- [22] B. T. Celebi y O. T. Kaymakci, *Verifying the accuracy of interlocking tables for railway signalling systems using abstract state machines*, Journal of Modern Transportation, vol. 24, n.º 4, pp. 277-283, oct. 2016 [Online]. Disponible: <http://dx.doi.org/10.1007/s40534-016-0119-1>
- [23] M. A. N. Oz, I. Sener, O. T. Kaymakci, I. Ustoglu, y G. Cansever, *Topology Based Automatic Formal Model Generation for Point Automation Systems*, Information Technology And Control, vol. 44, n.º 1, mar. 2015 [Online]. Disponible: <http://dx.doi.org/10.5755/j01.itc.44.1.7382>
- [24] F. van Dijk, W. Fokkink, G. Kolk, P. van de Ven, y B. van Vlijmen, *EURIS, a Specification Method for Distributed Interlockings*, en Computer Safety, Reliability and Security, Springer Berlin Heidelberg, 1998, pp. 296-305 [Online]. Disponible: [http://dx.doi.org/10.1007/3-540-49646-7\\_23](http://dx.doi.org/10.1007/3-540-49646-7_23)
- [25] U. Yildirim, M. S. Durmuş, y M. T. Söylemez, *Automatic Interlocking Table Generation for Railway Stations using Symbolic Algebra*, IFAC Proceedings Volumes, vol. 45, n.º 24, pp. 171-176, sep. 2012 [Online]. Disponible: <http://dx.doi.org/10.3182/20120912-3-BG-2031.00035>
- [26] M. Banci y A. Fantechi, *Geographical Versus Functional Modelling by Statecharts of Interlocking Systems*, Electronic Notes in Theoretical Computer Science, vol. 133, pp. 3-19, may 2005 [Online]. Disponible: <http://dx.doi.org/10.1016/j.entcs.2004.08.055>
- [27] Altera Corporation (2013). 8 Reasons to Use FPGAs in IEC 61508 Functional Safety Applications. Altera.
- [28] Xilinx (2006). *Triple Module Redundancy Design Techniques for Virtex FPGAs*. Virtex Series. Xilinx
- [29] F. L. Kastensmidt, L. Sterpone, L. Carro, y M. S. Reorda, *On the Optimal Design of Triple Modular Redundancy Logic for SRAM-based FPGAs*, en Design, Automation and Test in Europe [Online]. Disponible: <http://dx.doi.org/10.1109/DATE.2005.229>
- [30] George, A. (1990). *An overview of RISC vs. CISC*. [1990] *Proceedings. The Twenty-Second Southeastern Symposium on System Theory*.
- [31] Jamil, T. (1995). RISC versus CISC. *IEEE Potentials*, 14(3), pp.13-16.
- [32] Reference.digilentinc.com. (2019). *Arty Z7 Reference Manual* [Reference.Digilentinc]. [online] Disponible: <https://reference.digilentinc.com/reference/programmable-logic/artzy-z7/reference-manual>
- [33] Test-driven development concepts, taxonomy, and future direction (2005). D. Janzen, H. Saiedian. [IEEE]
- [34] COroutine based COsimulation TestBench. (2014). COCOTB. [Online]. Disponible: <https://cocotb.readthedocs.io/en/latest/introduction.html>
- [35] Xilinx.com. (2019). *Vivado Design Suite*. [online] Disponible: <https://www.xilinx.com/products/design-tools/vivado.html>



**Martín N. Menéndez** es ingeniero electrónico graduado de la UBA en 2017. En 2018 obtuvo el título de Especialista en Sistemas Embebidos otorgado por la misma institución, y actualmente se encuentra cursando la Maestría en la misma temática. Desde fines del 2017 se desempeña como miembro del grupo CONICET-GICSAFe, en el cual desarrolló proyectos de monitoreo de número de pasajeros y de implementación de sistemas de enclavamiento sobre FPGA.



**Facundo S. Larosa** es ingeniero electrónico, graduado en 2009 de la UTN-FRH En 2010 recibió una distinción de la Academia Nacional de Ingeniería por su promedio académico, y en 2017 obtuvo el título de Magister en Ingeniería en Sistemas Embebidos otorgado por la UBA. Actualmente se encuentra trabajando en su doctorado sobre el mismo campo de investigación. Hoy en día se desempeña como Profesor Adjunto con dedicación exclusiva en la UTN-FRH y en la Maestría en Sistemas Embebidos (MSE) dictada por la UBA. También es

Director del grupo ASE de la UTN-FRH y miembro del grupo CONICET-GICSAFe, donde trabaja en la investigación y desarrollo de sistemas críticos sobre FPGA.



**Ramiro A. Ghignone** es ingeniero electrónico egresado de la Universidad Tecnológica Nacional – Facultad Regional Haedo (UTN-FRH) a principios de 2019. Entre 2016 y 2018 participó como becario del grupo de I+D en Aplicaciones de Sistemas Embebidos (ASE) de esa facultad. Allí trabajó en diversos proyectos sobre sistemas de navegación satelital e inercial, y sobre optimización de sistemas embebidos en bajo nivel. Actualmente se desempeña como integrante del mismo grupo de investigación, donde posee una Dedicación Simple como Ayudante

de Trabajos Prácticos, y donde coordina el desarrollo del proyecto RAILIB. Desde fines de 2018 participa en el grupo CONICET-GICSAFe, donde se especializa en sistemas de enclavamiento ferroviario.



**Nicolás Álvarez** es ingeniero electrónico graduado de la UBA en 2003. Es especialista en el desarrollo e investigación de diferentes arquitecturas para procesamiento de señales sobre FPGA, dispositivo que se utiliza en diversos campos, como por ejemplo el prototipado de circuitos electrónicos, el procesamiento de video, el control de centrales nucleares y el manejo masivo de datos en los arreglos de servidores. Es jefe de Trabajos Prácticos con dedicación exclusiva en la Universidad de San Martín(UNSAM) y actualmente cursa la Maestría de

Seguridad Informática de la UBA.



**Ariel Lutenberg** es ingeniero electrónico graduado de la UBA en 2006. En 2009 obtuvo su diploma de Doctor en Ingeniería de la misma universidad, con mención de honor "Summa Cum Laude". En 2018 recibió el Gran Premio INNOVAR por el desarrollo de un sistema de monitoreo remoto de barreras ferroviarias automáticas, y desde 2019 forma parte del jurado del Premio INNOVAR. Por ese proyecto también recibió el primer premio del Concurso "Desafío Eureka". Actualmente es Profesor Adjunto con dedicación exclusiva en la FI-UBA, Investigador

Adjunto del CONICET, Director del Laboratorio de Sistemas Embebidos de la UBA y Director de la Carrera de Especialización en Sistemas Embebidos y la Maestría en Sistemas Embebidos de la misma Universidad. A su vez, es Presidente de la Asociación Civil para la Investigación, Promoción y Desarrollo de los Sistemas Electrónicos Embebidos.