

ARQUITECTURA HARDWARE DE UN EMULADOR DE CANAL PLC EN FPGA

J.J. Sánchez-Martínez⁽¹⁾, L. Díez⁽¹⁾, J.A. Cortés⁽¹⁾, F. J. Cañete⁽¹⁾ and Luis M. Torres⁽²⁾
jjism@ic.uma.es.

⁽¹⁾ Dpto. de Ingeniería de Comunicaciones. Universidad de Málaga. Campus de Teatinos s/n 29071.

⁽²⁾ Design of Systems on Silicon (DS2), C/ Charles Robert Darwin 2, Parc Tecnològic, 4680 Paterna, Valencia.

Abstract- This paper presents the hardware architecture of a novel channel emulator for broadband Indoor Power Lines Communications (PLC) using a Field Programmable Gate Array (FPGA). The implemented model consists of a time-varying channel response and a wide catalogue of disturbances that includes impulsive noise components, narrowband interferences and background noise. The system allows the real-time emulation of channels in the frequency band up to 50MHz. The objective is to facilitate the design and testing of high speed power line communication systems. The description provided in this paper concentrates in the description of the elements that are responsible for controlling all the temporal events that give a dynamic and programmable behavior to the emulator.

I. INTRODUCCIÓN

En la última década se ha vivido un interés creciente por el uso de las redes de distribución eléctrica como medios de transmisión digital de banda ancha (PLC, Power Line Communications). Las líneas eléctricas son canales de transmisión con características fuertemente dependientes de la topología de la red, con gran selectividad en frecuencia y cambiantes con el tiempo. Sin embargo, debido al gran ancho de banda disponible constituyen una buena alternativa a las redes inalámbricas para la transmisión de datos, audio y vídeo, tanto en viviendas como en pequeñas oficinas, sin necesidad de obras ni cableado adicional. En la actualidad ya existen en el mercado módem comerciales que funcionan a 200Mbps y se están desarrollando nuevos estándares internacionales para potenciar la expansión de esta tecnología.

La emulación de un canal de comunicación es un método efectivo para la reproducción de sus propiedades en el laboratorio y bajo condiciones de tiempo real, con el propósito de desarrollar y probar los componentes de futuros sistemas de comunicación. La emulación de canal se usa también para evaluar y comparar las prestaciones de diferentes sistemas de comunicación bajo condiciones bien conocidas y reproducibles, que los convierten en dispositivos claves para el diseño y validación de estos sistemas. Son estos motivos los que le confieren al emulador de canal de la red eléctrica implementado una importancia destacada.

Existen en la literatura algunos emuladores de canal PLC asumiendo un modelo de canal invariante en el tiempo (LTI, Linear Time Invariant) [1], pero hasta la fecha este emulador constituye el primer emulador que considera un modelo de canal cíclico y síncrono con la frecuencia de red, que se ha demostrado más próximo a la realidad [2,3]. La funcionalidad del emulador ha sido expuesta en [4], siendo el objetivo de

este artículo profundizar en la descripción hardware de la implementación y en el control dinámico que permite al emulador gestionar de manera autónoma toda una secuencia de eventos temporales que describen su comportamiento.

El presente artículo posee la siguiente estructura. En la Sección II se describe la funcionalidad del emulador así como las decisiones de diseño finalmente adoptadas. En la Sección III se explica la arquitectura hardware del emulador prestando especial atención a los módulos encargados del control dinámico. La Sección IV muestra algunas particularidades de implementación que condicionan tanto la estructura interna de almacenamiento de datos del emulador como el procesamiento de señal que se efectúa a un nivel de trama. Finalmente, las conclusiones se exponen en la sección V.

II. FUNCIONALIDAD DEL EMULADOR

Para determinar los requerimientos que un emulador de canal PLC debe cumplir es necesario en primer lugar hacer una introducción a las características de los canales de la red eléctrica. Una descripción detallada de los mismos puede encontrarse en [2, 3], recogiendo aquí únicamente aquellos aspectos relacionados con la funcionalidad del emulador.

A. Tipos de ruido incorporados

Los ruidos considerados en el emulador son:

-Ruido cicloestacionario. Modela el ruido de fondo. Tiene una distribución Gaussiana y está fuertemente coloreado con una densidad espectral de potencia (DEP) que decae con la frecuencia.

-Ruido impulsivo: Se consideran tres tipos de ruido impulsivo: síncrono, periódico y síncrono con la frecuencia de red; asíncrono, periódico pero no síncrono con la frecuencia de red y esporádico, aparece de manera aleatoria y no presenta ninguna periodicidad.

-Interferencia de banda estrecha: formada principalmente por sinusoides o señales moduladas procedentes de estaciones de radiodifusión, osciladores internos de dispositivos electrónicos del hogar, etc.

B. Comportamiento temporal

Los canales PLC presentan una variación temporal de dos tipos caracterizadas por su duración [3]:

-Variación a largo plazo: su comportamiento depende tanto de los dispositivos conectados a la red como de la propia estructura de la misma. Mientras esta configuración no cambie el comportamiento del canal permanece inalterado. En el emulador esta situación se denomina Estado de Red. La

transición entre Estados de Red se considera instantánea y se efectúa de manera abrupta.

-Variación a corto plazo: dentro de cada Estado de Red el comportamiento del canal se considera variante y dependiente del valor instantáneo de la tensión de red. El modelo desarrollado corresponde a un canal lineal y periódicamente variante (LPTV) [3] con una frecuencia de red de 50 y 60Hz, según el país.

La variación del canal se considera lo suficientemente lenta como para caracterizarla haciendo un muestreo de la misma en el ciclo de red. De esta forma el canal se puede modelar con un conjunto de respuestas LTI. El número de respuestas LTI y la separación entre las mismas dentro del ciclo de red dependerá de la forma de variación del canal. El ruido cicloestacionario se modela de forma similar, muestreando la DEP de ruido varias veces por periodo en instantes donde se asume aproximadamente estacionario.

El emulador incluye también la posibilidad de emular la variación que experimenta en su frecuencia la tensión de la red eléctrica (jitter).

C. Diseño de la evolución temporal

La variación cíclica del canal dentro del periodo de red va a estar definida por un conjunto de repuestas LTI que permanecerán inalteradas durante todo el Estado de Red. Para el ruido cicloestacionario se procede de igual forma, siendo caracterizado dentro del Estado de Red por diferentes respuestas LTI usadas para filtrar y colorear ruido blanco. El número de respuestas en ambos casos se ha limitado a ocho y el filtrado, ya sea del ruido blanco o de la señal de entrada, se efectúa en el dominio de la frecuencia.

Los ruidos impulsivos y la interferencia de banda estrecha se generan en el dominio del tiempo incorporando a la señal filtrada y con ruido coloreado dichas componentes [4]. La definición de configuración de estos ruidos permanece estable dentro de cada Estado de Red.

D. Descripción de la evolución temporal

Para desarrollar de manera flexible la evolución temporal del canal, el emulador ha sido diseñado como un autómatas que ejecuta un programa en el que se especifica de forma precisa dicha evolución. Este programa se genera utilizando una aplicación gráfica desarrollada en Java que permite la definición de un canal PLC usando para ello una base de datos con medidas reales llevadas a cabo en distintos escenarios (laboratorios y viviendas). La estructura de este programa se muestra en la Fig.1.

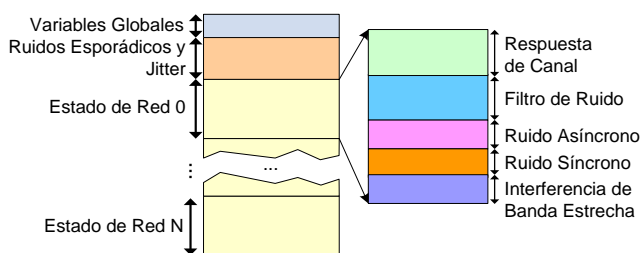


Fig. 1. Programa de configuración del emulador

Las variables globales permiten habilitar o deshabilitar los distintos módulos que conforman el emulador, definir la

duración de un periodo de red y el número de Estados de Red. Cada una de las restantes partes del fichero hace referencia a:

- Ruidos esporádicos: se almacena la forma de onda asociada a tres ruidos esporádicos y su descripción temporal, indicando el instante de aparición de los mismos a lo largo de todo el tiempo de emulación.

- Jitter: indica la variación del periodo de red ciclo a ciclo.

- Estado de Red: estructura de datos que incluye toda la información referente a la variación cíclica tanto del canal como del filtro coloreador de ruido blanco para generar el ruido de fondo [4]. También incluye los ruidos asíncrono, síncrono e interferencia de banda estrecha (IBE) asociados con dicho Estado de Red, así como sus parámetros de configuración (instantes de aparición, períodos, etc.).

Este fichero de configuración se transfiere al emulador previamente a su puesta en marcha y desde ese momento queda configurado y opera de manera autónoma sin necesidad de estar conectado a ningún ordenador.

III. ARQUITECTURA HARDWARE

La implementación del emulador se ha realizado sobre una FPGA Xilinx Virtex4 XC4VVSX35 usando Verilog como lenguaje de descripción. La plataforma de desarrollo incluye dos memorias externas ZBT-SDRAM (133MHz, 512Kx32), dos convertidores analógico-digital (CAD) y dos digital-analógico (CDA) de 14 bits. De los convertidores sólo se utiliza un CAD y un CDA, siendo utilizada una de las memorias externas para el almacenamiento del fichero de configuración. Existen dos FPGA adicionales encargadas de la interacción entre el ordenador y el emulador, permitiendo la comunicación a través de USB y PCI.

A. Diseño modular

La Fig. 2 muestra un diagrama de bloques con los módulos principales de los que consta el emulador de canal. Se distinguen dos tipos fundamentales de elementos según la actividad que desarrollan, ya sea el procesado de la señal o la gestión de los eventos temporales. Cada bloque de procesado tiene asociado un Controlador encargado de actualizar las señales internas del mismo (ruidos, coeficientes de filtros, periodos, etc.) y de dictar las órdenes de actuación y de parada. Tras almacenar el fichero de configuración da comienzo la emulación, iniciándose un proceso de contienda en donde todos los bloques de control tratan de acceder a memoria externa para extraer la información referente a su estado de configuración. De esta forma no existe un maestro que gestiona el emulador en su conjunto sino que cada Controlador tiene el control de su actividad y es el encargado del correcto funcionamiento del bloque de procesado asociado.

B. Control dinámico de eventos

El emulador de canal PLC implementado contiene una lógica de control de eventos que permite emular el comportamiento variante del canal e incorporar los ruidos descritos en la Sección II de manera dinámica. Todos los Controladores pueden acceder a memoria externa, por lo que se ha implementado un protocolo CSMA/CD (Carrier Sense Multiple Access with Collision Detection) con prioridad. El programa cargado en memoria determina para cada

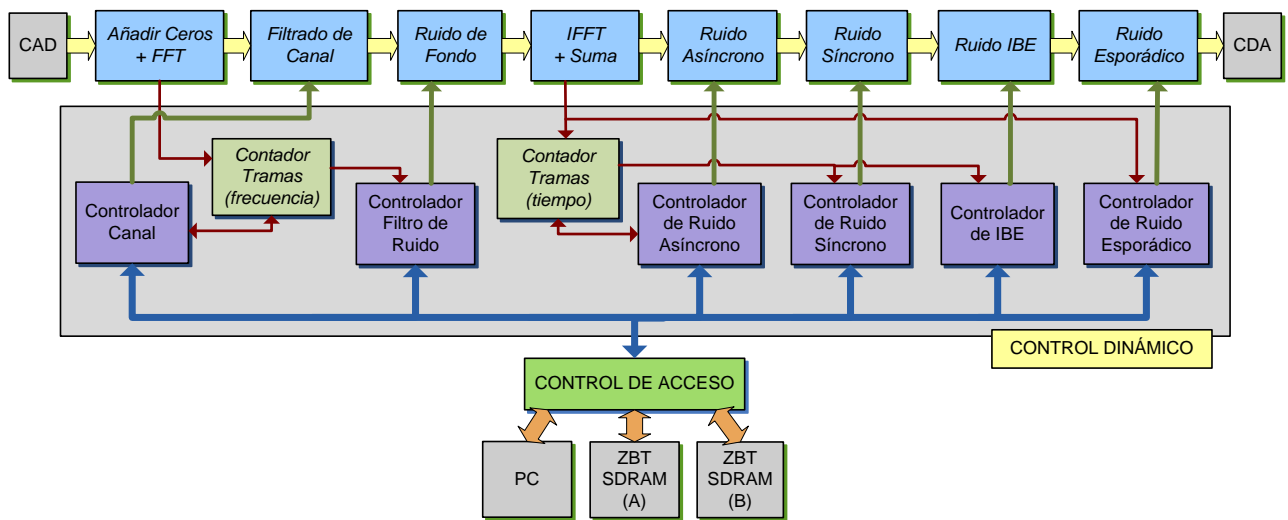


Fig. 2. Diagrama de bloques del emulador de canal PLC

Controlador el tiempo que debe permanecer en su estado actual y las acciones que debe realizar. Evidentemente, no todos los Controladores tienen una misma función y por tanto su complejidad está ligada a la propia complejidad del bloque de procesamiento que gestionan. Esto implica tanto una temporización distinta como la necesidad de traer mayor o menor cantidad de información de memoria externa. La Fig. 3 muestra el diagrama de flujo de las acciones que todos los controladores llevan a cabo para el acceso a memoria.

Cada Controlador se ha diseñado con una prioridad distinta que identifica el tiempo de espera tras detectar que el medio está ocupado o tras producirse una colisión cuando dos o más Controladores tratan de tomar el control del medio de forma simultánea. La transferencia de información de memoria externa a memoria interna se mantiene a lo largo de todo el tiempo de emulación, pues la situación es variante en cada uno de los Estados de Red. Incluso cuando sólo hay un Estado de Red, el comportamiento cíclico del canal hace que determinados módulos requieran actualizar su información de manera continua.

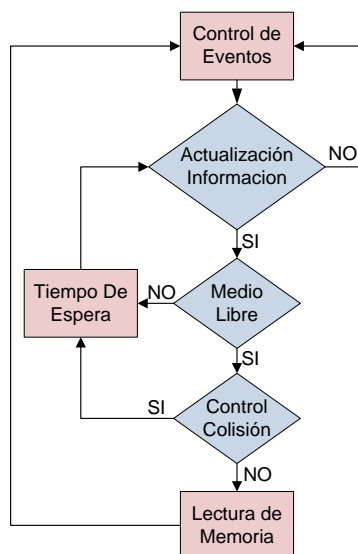


Fig. 3. Diagrama de flujo del protocolo de acceso a memoria externa.

IV. ESTRUCTURA INTERNA

En esta sección se describen dos aspectos importantes del emulador que dependen de su implementación: la secuencia temporal a nivel de trama y la organización de los parámetros de configuración a nivel interno.

A. Temporización a nivel de trama

La sincronización entre los distintos módulos que componen el emulador (Fig. 2) está condicionada por la implementación hardware del modelo de canal LPTV. El método de la convolución rápida (overlap-add) [5] se utiliza para filtrar la señal de entrada. Este método de filtrado implica fragmentar la señal en bloques de longitud determinada, añadir a cada bloque un número concreto de muestras nulas (dependiente de la longitud de la respuesta impulsiva de filtrado) y multiplicar por la DFT de la respuesta al impulso del canal (Fig. 4). Finalmente, en el dominio del tiempo, los bloques se solapan para reconstruir la señal. Este procedimiento de filtrado determina la resolución en el tiempo con la que opera el emulador, pues se genera una estructura de trama que condiciona el ancho de la ranura temporal.

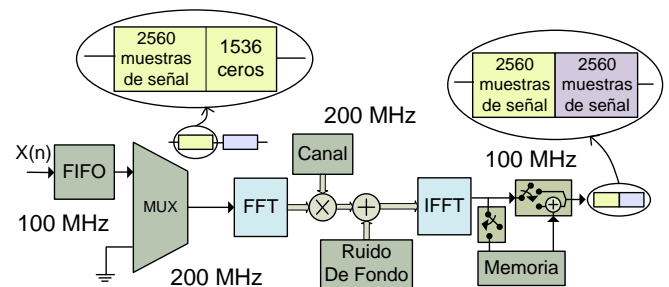


Fig. 4. Diagrama de bloques simplificado del filtrado en frecuencia.

El tamaño de trama divide el ciclo de red en un número entero de bloques (782 y 652 para 50 y 60Hz, respectivamente). La transformada de Fourier se aplica a bloques de 4096 muestras, donde se ha fijado una respuesta impulsiva máxima de 1536 muestras (15.36 μ s a 100MHz). La longitud de la respuesta impulsiva ha sido seleccionada según los resultados obtenidos en [6]. Este método de filtrado

permite un uso más eficiente de recursos que la convolución en el tiempo, pero por el contrario, el tener que introducir muestras nulas conlleva elevar la frecuencia de trabajo en el dominio de la frecuencia, lo que plantea una dificultad doble.

Por un lado, disminuir el periodo de reloj en frecuencia aumenta la complejidad de diseño. Conforme se reduce el periodo de reloj crece el número de caminos críticos donde el retardo por la propia propagación de la señal hace difícil el cumplimiento de tiempos. Esto normalmente se solventa mediante la introducción de registros intermedios que reducen la longitud de dichos caminos críticos, lo que a su vez se traduce de manera directa en el uso de más recursos y, en determinadas circunstancias, en un incremento de la latencia. Por otro lado, al incrementar la frecuencia de reloj se genera una estructura de trama con una separación entre las mismas (ΔT) proporcional a los periodos de reloj usados en tiempo (T_t) y en frecuencia (T_f) según la ecuación

$$\Delta T = 2560 T_t - T_f - 1536 \cdot T_f \quad (1)$$

Para facilitar la sincronización entre relojes se ha elegido una relación entera entre los mismos, usando 100MHz en el tiempo y 200MHz en frecuencia. Así pues, la sincronización en frecuencia se establece entre tramas de 4096 separadas entre sí $5.12\mu s$ y a 200MHz, mientras que en tiempo todo discurre de manera continua con tramas de 2560 muestras y a 100MHz. Esta distinción entre ambos dominios implica la necesidad de emplear dos temporizadores, uno que opera en frecuencia y otro en tiempo, que controlan la evolución temporal dentro de los Estados de Red.

B. Arquitectura de doble y triple memoria

El comportamiento dinámico del emulador requiere de unas prestaciones que permitan modificar el estado de configuración de los distintos elementos con una resolución temporal de una trama dentro del Estado de Red. Este requisito hace que cada módulo de procesado se haya provisto de una arquitectura con dos memorias de doble buffer. Mientras una de estas memorias almacena los datos que son utilizados en el estado actual, el módulo de control asociado puede escribir en la otra aquellos parámetros de configuración que serán usados en el siguiente estado. Dicha arquitectura es común a todos los bloques excepto para los de filtrado, tanto de la señal de entrada como del ruido blanco (generación del ruido de fondo) [4]. En ambos casos, dado que se utiliza una interpolación lineal dentro del Estado de Red [4] (Fig. 5), ha sido necesaria la inclusión de tres memorias de doble buffer.

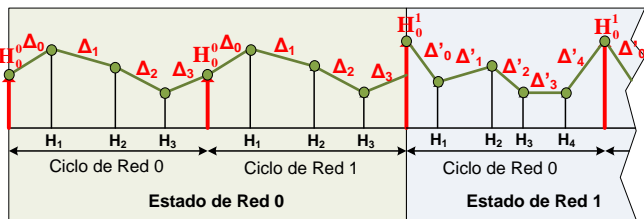


Fig. 5. Variación cíclica del canal y del filtro coloreador dentro de los Estados de Red.

La interpolación lineal realizada atiende a la expresión

$$H_i(k) = H_{i-1}(k) + \Delta_i(k), \quad (2)$$

usando una de las tres memorias para el primer estado de canal o ruido (primera respuesta LTI) y las otras dos se van alternando para almacenar los distintos incrementos. Estos incrementos sumados al primer estado de canal o ruido permiten emular un canal lineal con variación cíclica. La Fig. 5 muestra un ejemplo con dos Estados de Red y para cada uno de ellos los correspondientes incrementos y la primera respuesta LTI, H_0^0 y H_0^1 respectivamente. Esta estructura es equivalente tanto para el filtrado de canal como de ruido blanco, variando únicamente las respuestas LTI y la disposición temporal de las mismas dentro del Estado de Red. La Fig.6 muestra un esquemático de los elementos que intervienen en el proceso de interpolación. Para reducir el tamaño de las memorias a la mitad se ha hecho uso de la propiedad de hermiticidad.

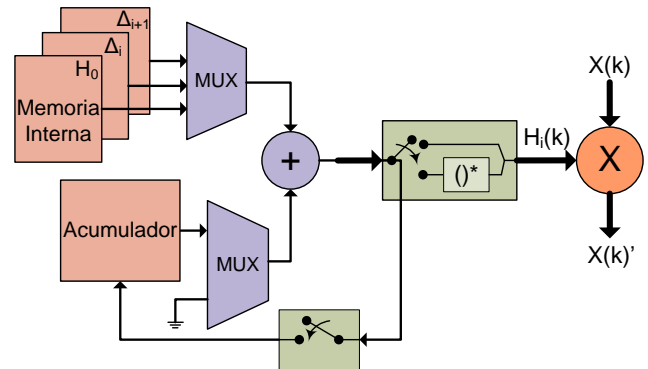


Fig. 6. Diagrama de bloques simplificado de los módulos de filtrado.

V. CONCLUSIONES

En este artículo se ha descrito la arquitectura hardware de un emulador de canal PLC atendiendo fundamentalmente a la parte de control dinámico de los distintos módulos que lo componen. El emulador desarrollado implementa un modelo de canal variante en el tiempo que recoge tanto las variaciones a largo plazo, debido a la conexión-desconexión de aparatos eléctricos, como la variación a corto plazo condicionada por el valor instantáneo de la tensión de red. Esta funcionalidad le otorga unas prestaciones que son de gran interés para el diseño y prueba de nuevas técnicas de transmisión, así como para la verificación de sistemas de transmisión completos.

REFERENCIAS

- [1] H. Philipps, "A hardware fading simulator for powerline communication channels," in *Internacional Symposium on Power-Line Communications and its Applications (ISPLC)*, 2001, pp.241-246.
- [2] F. Cañete, J. Cortés, L. Díez, and J. Entrambasaguas, "Modeling and evaluation of the indoor power line channel," *IEEE Communication Magazine*, vol. 41, pp. 41-47, Apr 2003.
- [3] F. Cañete, J. Cortés, L. Díez, and J. Entrambasaguas, "Analysis of the cyclic short-term variation of indoor power-line channels," *IEEE Journal on Selected Areas in Communication*, vol. 24, no. 7, pp. 1327-1338, July 2006.
- [4] F.J. Cañete, L. Díez, J. A. Cortés, J. J. Sánchez-Martínez and L. M. Torres, "Time-Varying Channel Emulator for Indoor Power Line Communications," in *Proceedings of the 51th Annual IEEE Global Telecommunications Conference (GLOBECOM)*, 2008, pp. 1-5.
- [6] A. Oppenheim and R. Schaffer, *Discrete-Time Signal Processing*, Prentice Hall, 1989.
- [5] J.A. Cortés, F.J. Cañete, L. Díez and J.T. Entrambasaguas, "Characterization of the cyclic short-time variation of indoor power-line channels response," in *Internacional Symposium on Power-Line Communications and its Applications (ISPLC)*, 2005, pp. 326-330.