RECEPTOR DE INTERFAZ DE AUDIO DIGITAL SPDIF PARA FPGA

Matías Nogueira Paladini
Federico Cattivelli
Gustavo Bellora
Guillermo Langwagen
Universidad ORT Uruguay
matiasnogueira@adinet.com.uy
fcattiv@adinet.com.uy

ABSTRACT.

Las interfaces digitales de audio han sufrido un gran impulso en los últimos años y cada vez más se las encuentra en equipos de reproducción de audio ya sea profesionales o de consumo. Inicialmente las comunicaciones entre equipos de audio eran analógicas, en la actualidad se tiende a utilizar dichas interfaces en toda la cadena de audio para conservar la integridad de la señal durante la transmisión.

En este artículo proponemos un método sencillo y eficiente de sincronizarse con la interfaz digital de audio SPDIF y extraer las muestras de audio en formato paralelo para una posterior etapa de procesamiento. La interfaz funciona a más de 3 Mbps, y la detección de los bits se logra mediante un proceso de sobremuestreo a 25 MHz que evita el uso de un PLL o de un reloj sincronizado con la señal de entrada para la detección. El receptor se describe completamente en VHDL y se implementa en una FPGA Virtex de Xilinx, lo cual le otorga gran flexibilidad a la hora de su rediseño o modificación. La implementación utiliza un 15 % del área total de la FPGA de 100k compuertas.

1. INTRODUCCIÓN.

Con la introducción masiva del reproductor de discos compactos a comienzos de los años ochenta, se normalizan los sistemas de almacenamiento y distribución de audio digital. Como consecuencia directa surgió la necesidad de contar con interfaces de audio completamente digitales para no degradar la señal en la comunicación entre equipos.

El estándar de audio profesional denominado AES/EBU (*Audio Engineering Society / European Broadcasting Union*) a partir de las organizaciones que lo crearon en 1985, fue el primer estándar en definir una interfaz digital de audio profesional de dos canales ([1]). Este estándar se conoce más generalmente con el nombre de AES3 y se convirtió en un estándar mundial bajo el nombre IEC60958-4. Para el audio no profesional (o de consumidor) existe la interfaz SPDIF (*Sony Philips Digital Interface*), que también se convirtió en un estándar mundial bajo el nombre IEC60958-3. Ambas interfaces son muy similares entre sí, aunque difieren en la definición de la interfaz eléctrica y en el formato del canal auxiliar de datos.

En este trabajo se utiliza la interfaz SPDIF ya que es la que se encuentra comúnmente en equipos no profesionales, aunque el análisis se puede extender a la interfaz profesional cuidando la adaptación de la interfaz eléctrica.

2. EL ESTÁNDAR IEC 60958.

El estándar IEC60958 provee un protocolo para una transmisión serie digital de dos canales de audio muestreados periódicamente y cuantificados uniformemente para la transmisión de señales por medio de un par trenzado (profesional) o un cable coaxial (consumidor).

La interfaz soporta audio con frecuencias de muestreo de 32 kHz, 44,1 kHz o 48 kHz y muestras de 16, 20 o 24 bits, lo cual la hace muy versátil pues es fácilmente adaptable a diferentes ambientes. La velocidad de transmisión de bit es de 3,1 Mbps para una frecuencia de muestreo de 48 kHz. En el presente trabajo se asume que la frecuencia de muestreo es de 48 kHz, aunque el análisis se puede extender fácilmente a otras frecuencias.

La información se transmite en tramas cuya duración es la de un período de muestreo (1/48 kHz). Cada trama está formada por dos subtramas, correspondientes a los dos canales que es capaz de transmitir la interfaz (audio estereofónico). Cada subtrama se divide en 32 intervalos de tiempo o *time slots* de igual duración numerados del 0 al 31. Los primeros cuatro *time slots* (0 a 3) se denominan preámbulos, y permiten la sincronización de trama. Los *time slots* 4 a 27 llevan una muestra de audio (16, 20 o 24 bits), y los restantes cuatro *time slots* llevan los bits de validez (V), usuario (U), estado del canal (C) y paridad

(P) (ver figura 1). La paridad es par y la definición del resto de los bits se puede obtener en [2] y [3].

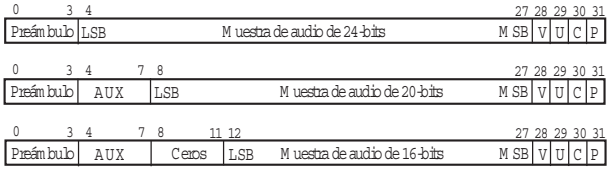

Fig. 1. Formato de Sub Trama

2-1. ***Código de línea.***

Los *time slots* 4 a 31 utilizan el código de línea *Biphase Mark* como se muestra en la figura 2. Cada bit a ser transmitido se representa mediante dos estados binarios consecutivos, cada uno de los cuales tiene una duración de un Intervalo Unidad (IU). Un valor lógico '0' se representa por dos estados consecutivos iguales, mientras que un valor lógico '1' se representa por estados diferentes. El primer estado de todo símbolo es siempre diferente al segundo estado del símbolo anterior. Este código tiene las ventajas de que no posee componente de continua y tiene alto contenido de información de reloj, lo que facilita su recuperación. Además es inmune a los cambios de polaridad.

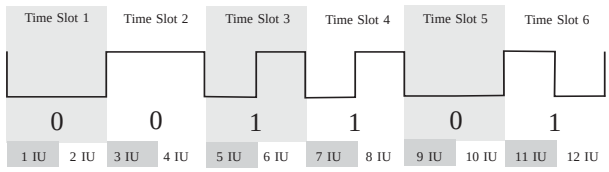

Fig. 2. Codificación Biphase Mark

2-2. ***Preámbulos.***

Los preámbulos son patrones que permiten identificar el inicio de una subtrama o de un bloque. Se colocan al inicio de cada subtrama (ver figura 3). Para evitar que se confundan con bits de información, estos patrones violan las reglas del código de línea dos veces durante su duración. Los preámbulos se transmiten al comienzo de cada subtrama y tienen una duración de 4 *time slots* (equivalente a 8 IU).

Existen tres tipos de preámbulos diferentes denominados X, Y y Z (ver figura 4). El preámbulo X indica que la subtrama siguiente pertenece al canal de audio izquierdo; el preámbulo Y indica una subtrama del canal derecho; el preámbulo Z indica una subtrama del canal izquierdo, ocurre una vez cada 192 tramas en lugar del preámbulo X indicando el inicio de un bloque de información adicional. El primer estado del preámbulo es siempre distinto que el segundo estado del símbolo anterior (el cual representa el bit de paridad de la subtrama anterior) por lo que pueden existir dos versiones de preámbulos X, Y y Z: los que se pueden ver en la figura 4 y sus equivalentes negados. Al igual que el código *Biphase Mark*, estos preámbulos no tienen componente de continua y proveen información de reloj.

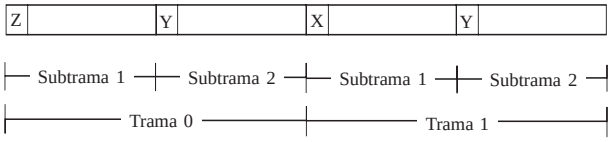

Fig. 3. Ubicación en la trama

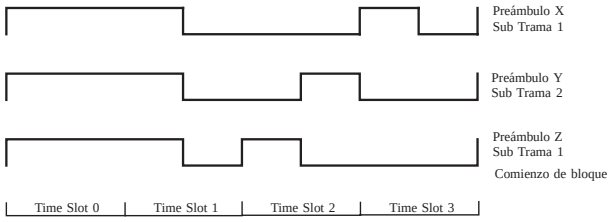

Fig. 4. Patrón de los preámbulos

2-3. ***Especificaciones eléctricas.***

La interfaz eléctrica se implementa mediante un cable coaxial de 75 Ohms con señales sin componente de continua y tensión pico a pico de 1 volt. Los estados binarios se representan mediante tensiones de (500 ± 200) mV y $(-500mV \pm 200)$ mV.

Existe una interfaz no considerada por el estándar que difiere de la SPDIF original en que maneja tensiones de 0 o 5 Voltios para los estados binarios, denominada SPDIF TTL. Hay una gran cantidad de equipos que utilizan señales del tipo SPDIF TTL y si bien muchos de los mismos lo utilizan solo internamente, hay otros como las tarjetas de sonido Sound Blaster Live! que lo hacen de manera externa. En este caso la interfaz fue diseñada para considerar tanto señales SPDIF TTL como señales SPDIF según indica el estándar.

3. ACONDICIONAMIENTO DE SEÑAL.

El acondicionamiento de señal permite adaptar la impedancia de la línea de transmisión y reconstruir la señal deteriorada. La correcta adaptación de impedancia es necesaria para evitar que existan reflexiones capaces de producir ecos y distorsionar la señal portadora de la información.

Existen diversos métodos para adaptar impedancias (ver [6], [5], [7] y [4]). En este caso se decició utilizar la terminación de alterna mostrada en la figura 5.

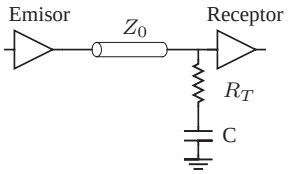

Fig. 5. Terminación AC

Este método es especialmente adecuado para señales balanceadas en continua ([4]). Los valores de los componentes se eligen de modo que la constante de tiempo $R_T C$ sea mucho mayor al tiempo de símbolo de la señal transmitida, con $R_T = Z_0$ y asumiendo Z_0 real. Lo que es lo mismo, se elige la reactancia del capacitor X_C mucho menor que la impedancia de línea a la frecuencia de $f = 1/T_s$ (T_s es el tiempo de duración de un símbolo, o un *time slot*). Esto permite que el capacitor se comporte como un circuito cerrado a altas frecuencias, con lo que la línea queda correctamente terminada. A bajas frecuencias y en continua, el capacitor se comporta como un circuito abierto, y la terminación presenta una alta impedancia dada por la entrada de la compuerta lógica, evitando un consumo excesivo de potencia.

Luego de la adaptación de impedancia, es necesario reconstruir la señal, la cual puede estar deteriorada por errores en la adaptación, degradación de la señal en la línea de transmisión y sobretiros excesivos tanto positivos como negativos que son capaces de dañar la entrada del receptor. Dicha reconstrucción debe hacer que los flancos sean lo más rápidos posible y las tensiones de los valores lógicos sean lo más estables posible.

Se probaron varios diseños compuestos de compuertas NOT concatenadas, pero estos producían grandes variaciones en la señal, haciendo que los tiempos altos y bajos no fueran iguales a sus respectivos en la señal de entrada. De esta manera cualquier cálculo realizado de ahí en adelante dependería estrictamente de esta modificación impuesta a la señal.

Entonces, se diseñó un sistema compuesto por un pequeño transistor en modo amplificador como se muestra en la figura 6. Esta solución conserva los tiempos alto y bajo, no retarda la señal y los tiempos de subida y bajada entran dentro del rango de la recomendación. Se eligió utilizar una entrada coaxial para alimentar el sistema.

La señal correctamente adaptada y regenerada, se aplica a la FPGA que extrae los datos de la interfaz SPDIF. Se utilizó una FPGA XCV-100 de Xilinx, la cual maneja tensiones LVTTL (tensiones binarias no balanceadas con valores de 0 y 3,3 voltios) entre otras y soporta tensiones TTL5V de entrada pero no las genera en sus salidas. Se decidió utilizar LVTTL como tensión de entrada a la

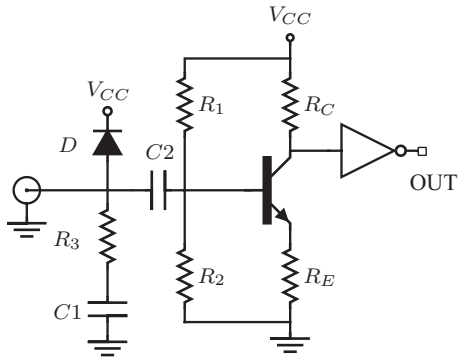

Fig. 6. Circuito eléctrico de la Interfaz de adaptación de señal

FPGA y por lo tanto fue necesario adaptar los 5V de la interfaz SPDIF TTL. Para ello se ubicó un diodo rápido entre la señal de entrada y la tensión de alimentación, $V_{CC} = 3,3V$, el cual entra en conducción cuando la entrada supera $V_{CC} + V_\gamma$ y limita la tensión a ese valor.

4. SINCRONIZACIÓN Y DECODIFICACIÓN DE LA TRAMA SPDIF.

Se desarrolló una entidad denominada “Sincronizador” como se muestra en la figura 7, la cual fue descrita en lenguaje VHDL e implementada en una FPGA del tipo XCV-100. Dicha entidad recibe como entradas la señal SPDIF, un reloj de 25 MHz y una señal de ‘Reset’. Como salidas posee las señales ‘P’, ‘U’, ‘C’, ‘V’ explicadas en la sección 2, la señal ‘NumCanal’ que indica qué canal está disponible para ser leído en la salida y dos buses de 24 bits que entregan las muestras decodificadas de los canales derecho e izquierdo.

La entidad “Sincronizador” se divide en tres bloques fundamentales. Un primer bloque denominado “Detector” encargado de la sincronización, es decir, se encarga de detectar el preámbulo de cada subtrama e informar su detección generando una señal de sincronismo. El segundo bloque, denominado “Bit Decoder”, se encarga de decodificar el código de línea utilizando como señal de inicio las señales de sincronismo que envía el primer módulo y produce como salida, los bits de la trama ya decodificados (‘Bit_Data’) y una nueva señal de reloj que indica cuando dichos bits son válidos (‘Bit_Ready’). También realiza el chequeo de paridad y lo informa en la salida P. El último bloque, está compuesto por los módulos “Serie Paralelo” y “Demultiplexor”, convierte el tren de bits a paralelo y utiliza el demultiplexor para entregar las muestras de audio a la salida del sistema.

*4-1. **Reloj del sistema.***

Normalmente para decodificar un tren de bits, se debe sincronizar con la señal portadora de la información y observar los valores lógicos en el medio del tiempo de

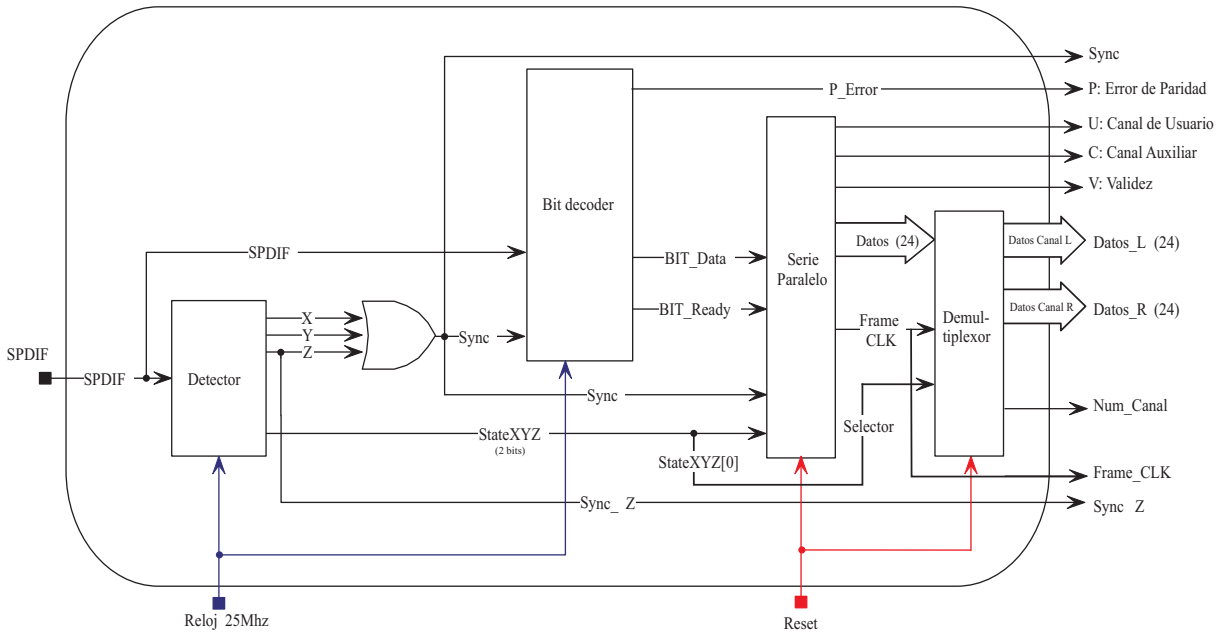

Fig. 7. Diagrama de bloques del sistema decodificador de la trama SPDIF

bit, ya que es más estable y se logra de esta manera minimizar el error de lectura. Luego se toma una decisión acorde a la medida. Utilizando esta técnica, para la interfaz SPDIF transmitiendo audio estereofónico de 48kHz, la mínima frecuencia de reloj necesaria para poder detectar todos los niveles posibles es de 6,144 MHz. Sin embargo, para muestrear a esta frecuencia es necesaria una perfecta sincronización entre las frecuencias del reloj y la interfaz, requiriendo de algún sistema basado en un PLL, por ejemplo.

Una alternativa al uso de un PLL es sobremuestrear el tren de datos digital, con lo cual la sincronización entre la interfaz y el reloj se vuelven innecesarias. Por ejemplo, se pueden tomar 4 muestras por Intervalo Unidad (IU), correspondiente a una frecuencia de muestreo de 24,576 MHz. En el presente trabajo se utilizó un reloj de 25 MHz para realizar el muestreo, ya que era el reloj disponible en la placa de desarrollo. Esto permitió tomar algo más de 4 muestras por IU. Sin embargo, las diferencias en el ciclo de trabajo de la señal reconstruida, y el jitter de los relojes de muestreo y de generación, pueden producir que los IU duren más o menos de 4 muestras de reloj, lo cual puede dificultar la detección.

Para corroborar que con 4 muestras por IU era suficiente para decodificar perfectamente la interfaz SPDIF, se realiza un estudio de la señal SPDIF recibida, muestreándola a 25 MHz y almacenando los resultados en memoria. Posteriormente se corre un ensayo acumulativo y se halla la cantidad de muestras consecutivas con igual nivel lógico, sin importar si son niveles altos o bajos. En la figura 8 se muestra el resultado obtenido graficado con escala logarítmica.

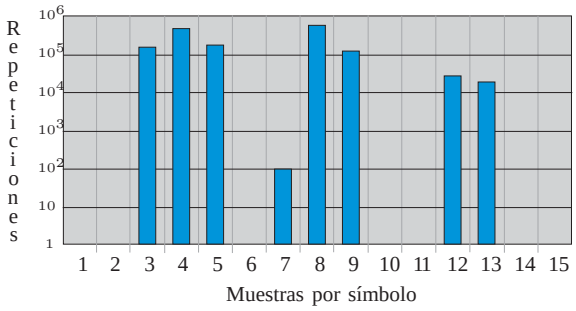

Fig. 8. Largo de los símbolos de la trama SPDIF

En el gráfico se distinguen claramente los tres tipos de símbolos. Los símbolos que duran entre 3 y 5 ciclos de reloj representan símbolos con duración 1 IU, correspondientes a un ‘1’ lógico. Los símbolos que duran entre 7 y 9 ciclos de reloj representan símbolos de 2 IU de duración, correspondientes a un ‘0’ lógico. El tercer grupo de símbolos son los correspondientes a las violaciones de código de línea por parte de los preámbulos. Los mismos tienen una duración entre 12 y 13 ciclos de reloj, correspondientes a 3 IU.

Lo más importante a destacar es que no existen símbolos

de 6 ciclos de reloj de duración. Esto permite concluir que el más largo de los símbolos de 1 IU (de 5 ciclos de reloj) es siempre menor que el más corto de los símbolos de 2 IU (de 7 ciclos de reloj), con lo cual estos dos símbolos son perfectamente distinguibles muestreando a 25 MHz. También indica que luego de 6 ciclos de reloj después del inicio de un *time slot*, estamos en el segundo IU del mismo. Vale destacar que si ocurriera solapamiento no se podría decodificar el código de línea con esta frecuencia de reloj, en cuyo caso habría que aumentarla.

El Ensayo se realizó aproximadamente 10^6 veces por cada tipo de símbolo, con lo cual se puede concluir que existe una probabilidad muy baja de error en los cálculos de tiempos recién descritos.

4-2. ***Detección del preámbulo.***

El módulo "Detector" es el encargado de la detección de los preámbulos de la interfaz SPDIF. El objetivo de este módulo es el de generar un pulso de sincronismo cada vez que se encuentra un preámbulo. Sus entradas son la señal SPDIF, una señal de reloj y sus salidas las señales de sincronismo 'X', 'Y', 'Z' y 'StateXYZ[2]'. Esta última indica el tipo de subtrama detectada y mantiene el valor durante su duración.

La detección se hace mediante un registro de desplazamiento de 30 bits compuesto por una cadena de flip flops tipo D, funcionando con un reloj de 25 MHz. Las salidas de los flip flops alimentan una compuerta AND (directamente o pasando a través de compuertas negadoras y exceptuando aquellas donde el estado es 'no importa') que colecciona los bits que se desean interrogar y genera un pulso de salida cuando encuentra el patrón deseado. Por ejemplo, en la figura 9 se muestra un circuito que busca la palabra '..10XX10..', donde 'X' representa un estado de 'no importa'. Note que el pulso generado en la detección dura como mínimo un ciclo de reloj.

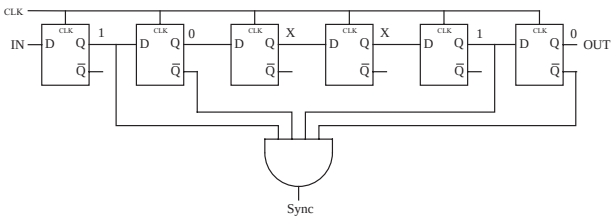

Fig. 9. Esquema de hardware

Para los preámbulos SPDIF, el patrón de comparación se elige de modo que para cada IU se descartan las dos muestras obtenidas al inicio y al final del mismo, y se evalúan las restantes dos muestras centrales. En la figura 10 se muestra un diagrama donde se especifican los patrones de búsqueda para un preámbulo del tipo 'X'.

En total se utilizan seis patrones simultáneamente con una misma cadena de flip flops para poder identificar los tres tipos de preámbulos (X, Y, Z) y sus respectivos negados.

4-3. ***Decodificación del código de línea.***

El módulo "Bit Decoder" es el encargado de decodificar cada *time slot* del código de línea.

La decodificación del código de línea se realiza muestreando la señal SPDIF a 25 MHz, con lo cual se obtienen cerca de 4 muestras por IU, como se describió anteriormente. Luego de un pulso de sincronismo generado por el detector de preámbulo, se inicia la búsqueda de un flanco en la entrada SPDIF, indicando el inicio del primer bit. Una vez encontrado un flanco, se almacena el valor actual de la entrada y se espera durante 6 ciclos de reloj, donde se vuelve a tomar el valor de la entrada. Si ambos valores difieren, se trata de un '1' lógico, mientras que si son iguales se trata de un '0' lógico. Luego comienza nuevamente la búsqueda de un flanco para detectar el inicio del próximo bit, y el proceso se repite para los 28 bits restantes.

La figura 11 muestra la decodificación de la secuencia '10'. Note que luego del sexto ciclo de muestreo se produce el dato válido en 'BIT_Data' y una señal de validez en 'BIT_Ready'. Estas dos señales alimentan posteriormente a la última parte del receptor, encargada de convertir los datos serie a paralelo y presentarlos en dos buses de 24 bits para su posterior lectura.

5. PRUEBAS DEL RECEPTOR.

La prueba de un sistema para asegurar su correcto funcionamiento es en general una tarea que requiere de mucho tiempo, en ocasiones mucho mayor que el tiempo de diseño. Las FPGAs poseen la ventaja de que cualquier señal interna puede ser asignada a un puerto de salida y ser así monitoreada.

Para depurar errores cometidos en las etapas iniciales del diseño se puede medir y estudiar la relación entre diferentes señales del sistema. Es así que muchos errores se eliminaron observando las señales de sincronismo generadas por las detecciones de preámbulo y de bit.

Para depurar errores cometidos en las etapas iniciales de diseño se procede a mirar en un osciloscopio las diferentes señales de sincronismo. El objetivo es encontrar señales internas que tengan una clara relación entre sí, ya sea en tiempo o en cantidad de eventos de la señal. Existe una relación estricta entre las parejas de señales 'Sync', 'SyncZ'; 'Sync', 'FrameCLK' y 'Sync', 'BIT_Ready'. Al medir los tiempos entre eventos de las señales 'Sync' y 'SynZ' se verifica el correcto funcionamiento del módulo "Detector", la señal 'Sync' indica cuando se ha detectado un preámbulo, los mismos deben suceder con un

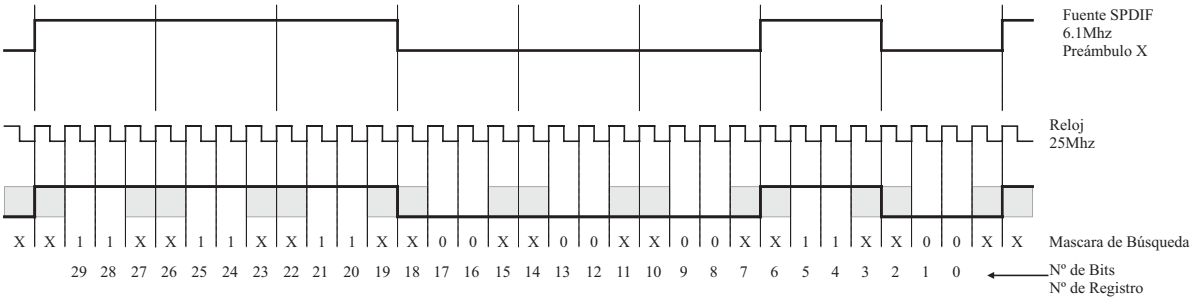

Fig. 10. Buscador de la palabra de alineamiento

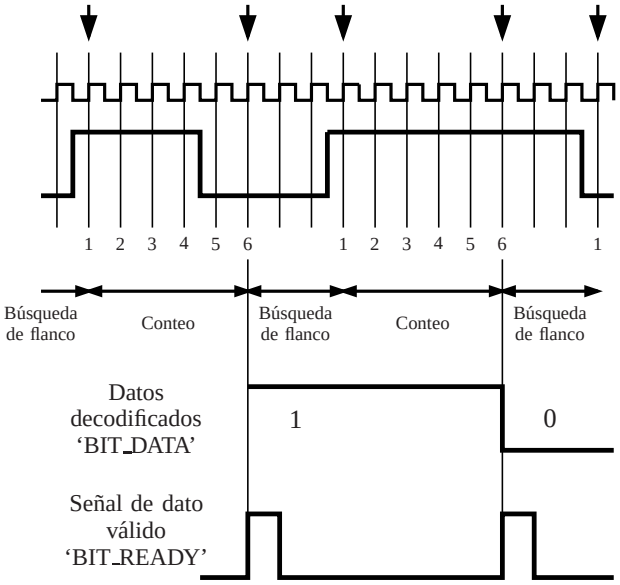

Fig. 11. Decodificación de código de línea

intervalo de 10.4μs y los preámbulos 'Z' deben suceder una vez cada 191 preámbulos generales. La señal 'Sync' y 'FarmeCLK' son muy parecidas, la primera indica la detección de un preámbulo y la segunda indica que se terminó de decodificar una trama completa y que la misma está pronta para ser leída. Es decir que por cada pulso de la señal 'Sync' existe un pulso de 'FrameCLK' que sucede un tiempo determinado después de las señal de sincronismo correspondiente a dicha trama. Este tiempo es la latencia del sistema decodificador de trama, dicho tiempo es la demora que agrega el módulo "Bit Decoder" y "Serie Paralelo" en tratar la señal. Vale destacar que este tiempo es fijo y no debe haber gran jitter en la señal de 'FrameCLK', lo cual indicaría una falla en la decodificación de trama. La pareja 'Sync' , 'BIT_Ready' entrega información sobre el funcionamiento del módulo "Bit Decoder", ya que por cada señal de sincronismo deben suceder 28 decodificaciones de bit, entonces por cada pulso de 'Sync' hay 28 pulsos de 'BIT_Ready'. Nuevamente los tiempos entre pulsos de la señal 'BIT_Ready' son constantes ya que el método de detección así lo indica.

De todas maneras al medir estos tiempos con un osciloscopio clásico no se puede asegurar que se estén detectando todos los preámbulos y no se esté perdiendo alguno. Surgen nuevas técnicas de depuración y prueba del sistema. Una opción, la cual prueba el sistema en su totalidad, es almacenar en memoria RAM, en formato paralelo, los datos ya convertidos y compararlos con las muestras de una señal de prueba que se envió por la interfaz SPDIF. De esta manera se puede leer la memoria RAM y analizar los datos recabados a nivel de trama, pudiendo mirar en pantalla la trama SPDIF completa tratando de ubicar zonas conocidas, tales como la señal de prueba utilizada (dicha tarea es difícil, a no ser que sea una señal muy sencilla y que no pase por ningún filtro antes de su transmisión) y el canal de estado que da información como ser el tipo de fuente de audio, la velocidad de trama, etc.. El decodificar este canal es una señal importante de que el receptor esta funcionando correctamente. Para leer la memoria RAM se utilizó el software *PC to SRAM Interface* de la Universidad de Queensland ([8]).

También se procedió, (por medio de trozos de software especialmente diseñado para el caso), a convertir los datos a formato "WAV" para ser analizados en 'MATLab' o 'SCILab' para detectar posibles errores y además para poder ser escuchados y evaluados por el oído humano.

Vale destacar que hay que tener gran cuidado con las demoras intrínsecas a la hora de programar el receptor ya que las mismas hacen variar los cálculos y las medidas agregándole uno o pocos ciclos de reloj a la cuenta, pero en este tipo de decodificación puede hacer gran diferencia en la señal de salida. Tanto los ensayos realizados para elegir la frecuencia de reloj (ver 4-2) como las pruebas detalladas anteriormente, son todas muy importante a la hora de garantizar un correcto funcionamiento del sistema. Finalmente vale la pena mencionar que el sistema también se probó con éxito utilizando un reloj de 24,576 MHz (frecuencia múltiplo de la frecuencia de muestreo) en lugar

de 25 MHz.

6. CONCLUSIONES.

Las interfaces digitales de audio se están volviendo cada vez más populares. Entre otras razones porque conservan la integridad de la señal que transportan y por practicidad. La implementación de una interfaz simple, que pueda sincronizarse y decodificar una trama de este tipo sin la necesidad de tener un reloj sincronizado al transmisor de dicha señal, es muy importante, ya que se puede diseñar un sistema fácil de implementar, configurar y usar.

Una descripción en VHDL para implementar en una FPGA de un receptor de este tipo es muy flexible, ya que posibilita redefinir el diseño modificando unas pocas líneas de código para interfaces con diferentes frecuencias de muestreo, o diferente cantidad de bits por muestra de audio, etc.

Se logró desarrollar un método sencillo y eficiente de sincronizarse con la interfaz digital de audio SPDIF y extraer las muestras de audio de ambos canales en formato paralelo, así como los bits V, U, C, P de información adicional para una posterior etapa de procesamiento. La interfaz funciona a más de 3 Mbps y la detección de los bits se logra mediante un proceso de sobremuestreo a 25 MHz que evita el uso de un PLL o de un reloj sincronizado con la señal de entrada para la detección.

Las pruebas de cualquier sistema son parte muy importante del desarrollo del mismo, ya que definen la vida del mismo al garantizar o no su correcto funcionamiento. El receptor desarrollado permite la detección de la interfaz SPDIF sin el uso de un PLL y con un reloj no sincronizado con la señal de entrada, logrando una detección correcta de los datos enviados. El receptor fue descrito en VHDL e implementado en una FPGA Virtex XCV100 de Xilinx, ocupando un 15 % del total de dicha FPGA.

REFERENCIAS.

[1] BOHN, Dennis. 2001. *Interfacing AES3 and S/PDIF*. RaneNote 149 [online] [citado 18 Octubre 2004]. Disponible en Internet: <http://www.rane.com/note149.html>

[2] DUNN, Julian. 2001. *The AES3 and IEC60958 Digital Interface*. Technote TN-26 [online] [citado 16 Abril 2003]. Disponible en Internet: <http://www.audioprecision.com/ftp/publications/technotes/Tn-26.pdf>

[3] INTERNATIONAL ELECTROTECHNICAL COMMISSION. *Digital Audio Interface - Part 3: Consumer applications*. 2da ed. [NORMAS, ESTANDARES] Génova, IEC, Enero de 2003.

[4] JOHNSON, Howard W.; GRAHAM, Martin. 1993. *High Speed Digital Design. A Handbook of Black Magic*. Nueva Jersey: Prentice Hall PTR.

[5] MAGDALUYO, M. 1998. *Transmission lines and terminations with Philips Advanced Logic families*. Aplication Note AN246 [online] [citado 15 Mayo 2003]. Disponible en Internet: <http://www.semiconductors.philips.com/acrobat/applicationnotes/AN246.pdf>

[6] MOTOROLA INC. 1990. *Transmission line effects in PCB applications*. Application Note AN1051/D [online] [citado 5 Junio 2003]. Disponible en Internet: <http://e-www.motorola.com/files/microcontrollers/doc/app_note/AN1051.pdf>

[7] STEVENS, M. 1993. *Ground and Vcc Bounce of High-Speed Integrated Circuits*. Aplication Note AN223 [online] [citado 3 Mayo 2003]. Disponible en Internet: <http://www.philipslogic.com/support/appnotes/logic/pdf/an223.pdf>

[8] UNIVERSITY OF QUEENSLAND. 2001. *PC to SRAM Interface* [online] [citado 18 Octubre 2004]. Disponible en Internet: <http://www.itee.uq.edu.au/ peters/xsvboard/pctosram/pctosram.pdf>