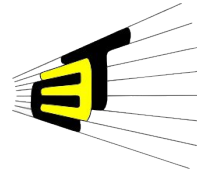


**DECODIFICADOR 132b/128b COMPATIBLE CON USB3.1
EN TECNOLOGÍA CMOS 180NM**

SERGIO ANDRÉS TOVAR DURÁN



UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y DE
TELECOMUNICACIONES
BUCARAMANGA
2017

DECODIFICADOR 132b/128b COMPATIBLE CON USB3.1
EN TECNOLOGÍA CMOS 180NM

SERGIO ANDRÉS TOVAR DURÁN

*Trabajo de grado para optar al título de
Ingeniero Electrónico*

Director
ELKIM FELIPE ROA FUENTES
PhD. en Microelectrónica

Codirector
CKRISTIAN RICARDO ESTEBAN DURÁN BLANCO
MSc. en Ingeniería Electrónica

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y DE
TELECOMUNICACIONES
BUCARAMANGA

2017

Índice general

INTRODUCCIÓN	11
1 CAPA FÍSICA DEL RECEPTOR USB 3.1	13
1.1 DECODIFICADOR 128B/132B	14
1.2 BÚFER ELÁSTICO	15
1.2.1 Tipos de búfer elástico	16
1.2.1.1 Búfer medio-lleno	16
1.2.1.2 Búfer de tipo control de flujo	16
2 IMPLEMENTACIÓN	18
2.1 DECODIFICADOR 128B/132B	18
2.1.1 Detección de error	20
2.2 BÚFER ELÁSTICO	21
2.2.1 Elección del tipo de Búfer Elástico	22
2.2.2 Profundidad del Búfer Elástico	22
2.2.3 Control del Búfer Elástico	23
2.2.4 Registro de desplazamiento PISO	24
2.2.5 Banco de registros	25
2.2.6 Detector de símbolos SKP	25
2.2.7 Contador ascendente/descendente	25
2.2.8 Máquina de estados	25
3 SÍNTESIS Y RESULTADOS	26
4 CONCLUSIONES	29
REFERENCIAS	30
BIBLIOGRAFÍA	32

Índice de figuras

Figura 1	Comparación de 4 generaciones USB.	12
Figura 2	Capa física del receptor UBS3.1	13
Figura 3	Circuito de recuperación de reloj	14
Figura 4	Búfer elástico de tipo medio-lleno	16
Figura 5	Búfer elástico de tipo control de flujo	17
Figura 6	Datapath del Decodificador 128b/132b.	19
Figura 7	Ubicación del encabezado y la carga útil de datos.	20
Figura 8	Máquina de estados finita del Decodificador 128b/132b.	20
Figura 9	Simulación comportamental del Decodificador.	21
Figura 10	Entradas y salidas del Búfer Elástico USB3.1	22
Figura 11	Nivelación del Búfer por medio de la eliminación de símbolos SKP	23
Figura 12	Nivelación del Búfer por medio de la inserción de símbolos SKP	24
Figura 13	Datapath del Búfer Elástico	24
Figura 14	Simulación comportamental del Búfer Elástico	25
Figura 15	Layout del Decodificador en tecnología CMOS 180nm.	27
Figura 16	Layout del Búfer Elástico en tecnología CMOS 180nm.	28

Índice de tablas

Tabla 1	Frecuencia de operación del Decodificador según el tamaño de los símbolos.	19
Tabla 2	Lógica de salida del detector de error en el encabezado.	21
Tabla 3	Área, consumo de potencia y frecuencia de operación del Decodificador en tecnología CMOS 180nm.	26
Tabla 4	Área, consumo de potencia y frecuencia de operación del Búfer Elástico en tecnología CMOS 180nm.	27

RESUMEN

Título: Decodificador 132b/128b compatible con USB3.1 en tecnología CMOS 180nm.¹

Autores:

Sergio Andrés Tovar Durán²

Palabras Clave: Decodificador; Transmisión; Frecuencia; Búfer; Datos, USB, Micro-electrónica.

DESCRIPCIÓN

El auge tecnológico actual, la capacidad y velocidad de cómputo de los dispositivos electrónicos y la incesante necesidad de transferir información dentro y fuera de estos, dieron lugar al desarrollo del estándar USB3.1. La generación 2 de USB, también denominada *SuperSpeedPlus*, ofrece un nuevo esquema de codificación llamado 128b/132b en contraste a la codificación 8b/10b usada por *SuperSpeed*. Adicionalmente, la arquitectura *SuperSpeedPlus* admite una fuente de reloj de referencia independiente en cada lado del enlace. Esta característica da lugar a una pequeña diferencia en la frecuencia de envío y recepción de datos, la cual debe ser manejada por un búfer elástico, capaz de almacenar y procesar los paquetes de datos provenientes del decodificador. Este proyecto presenta el diseño del Decodificador 128b/132b y el Búfer Elástico compatibles con la capa física del receptor USB3.1. El decodificador se utiliza para alinear los datos y detectar o corregir errores en el encabezado. El búfer elástico se encarga de nivelar la velocidad de transmisión y recepción de datos a cada lado del enlace. El decodificador y el búfer elástico se diseñan en lenguaje verilog HDL, se sintetizan y se realiza el enrute en tecnología CMOS 180nm, los resultados del diseño permiten una máxima frecuencia de 341MHz y 318MHz, respectivamente.

¹Trabajo de Grado

²Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones. Director: Elkim Felipe Roa Fuentes, PhD. en Microelectrónica. Co-Director: Ckristian Ricardo Esteban Durán Blanco, MSc. en Ingeniería Electrónica

ABSTRACT

Title: Compatible USB 3.1 132b/128b decoder in 180nm CMOS technology¹

Authors:

Sergio Andrés Tovar Durán²

Key Words: Decoder, Transmission, Frequency, Buffer, Data, USB, Microelectronics.

DESCRIPTION

The current technological boom, the capacity and computing speed of electronic devices, and the everlasting need to transfer information inside and outside them, gave rise to the development of the USB 3.1 standard. The USB 2nd generation, also known as SuperSpeedPlus, offers a new coding scheme called 128b/132b in contrast to the 8b/10b coding used by SuperSpeed. In addition, the SuperSpeedPlus architecture employs an independent clock source on each side of the link. This characteristic produces a small difference between the transmission and reception frequency, which must be handled by an elastic buffer that will be in charge of leveling out these frequencies and thus allow a proper data transmission. This book presents a 128b/132b decoder and an elastic buffer module design compatible with the physical layer USB 3.1. The decoder is used to align the data and detect or fix header errors. The elastic buffer deals with leveling out the data transmission and reception speed on each side of the link. The decoder and the elastic buffer design in Verilog HDL language and its synthesis will be shown. The place and route in 180nm CMOS technology is performed, the circuit design results allowing a maximum frequency of 341 MHz and 318 MHz, respectively.

¹Bachelor Thesis

²Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones. Director: Elkim Felipe Roa Fuentes, PhD. en Microelectrónica. Co-Director: Ckristian Ricardo Esteban Durán Blanco, MSc. en Ingeniería Electrónica.

OBJETIVO GENERAL

Diseñar y verificar la operación del decodificador para el estándar USB 3.1 mediante simulación.

OBJETIVOS ESPECÍFICOS









- ❖ Describir en lenguaje verilog HDL un decodificador 132b/128b con entrada de 32 bits y verificar su operación mediante simulación.
- ❖ Describir en lenguaje verilog HDL un búfer elástico compatible con el decodificador y verificar su operación mediante simulación.
- ❖ Sintetizar y enrutar el decodificador y el búfer elástico en tecnología CMOS 180nm y verificar su operación.

INTRODUCCIÓN

El auge tecnológico actual, la capacidad y velocidad de cómputo de los dispositivos electrónicos y la incesante necesidad de transferir información dentro y fuera de estos, dieron lugar al desarrollo del estándar USB3.1. Éste protocolo de conexión, que permite enlazar diversos periféricos a un dispositivo electrónico, posee una arquitectura *source-synchronous timing* [1], donde tanto los datos como el reloj, se transmiten desde el controlador del dispositivo de origen. Esto significa que no existe un reloj en común, que el transmisor y el receptor puedan usar para sincronizar los datos salientes con los recibidos. En cambio, el dispositivo transmisor remite el reloj que usó para enviar los datos, y el receptor usará ese reloj reenviado para sincronizar los datos transmitidos. Debido a que los extremos de envío y recepción adoptan fuentes de reloj de referencia independientes, puede surgir una cierta diferencia de frecuencia entre el reloj recuperado y el reloj local. Con el fin de garantizar la correcta transmisión y sincronización de los datos, USB3.1 posee, en su capa física, un búfer elástico, capaz de nivelar los dominios de frecuencia que se encuentran a lado y lado de la transmisión.

El estándar USB3.1, también denominado *SuperSpeedPlus*, ofrece un nuevo esquema de codificación llamado 128b/132b en contraste a la codificación 8b/10b usada por *SuperSpeed*. La codificación se utiliza en los estándares de transmisión de datos en serie, para asegurar suficientes transiciones de datos, necesarias para la recuperación del reloj. Cada esquema de codificación conlleva una sobrecarga de protocolo, la cual relaciona el número de bits de codificación añadidos, y el número de bits de carga útil sin procesar. La sobrecarga de la codificación 132b/128b es de 4 bits añadidos por cada 128 bits de carga útil transmitidos. Esto es considerablemente más eficiente que la sobrecarga que ofrece el esquema de codificación 8b/10b utilizado anteriormente, que añadía 2 bits por cada 8 bits transmitidos. Este nuevo esquema de codificación simple y robusto, con una menor sobrecarga de bits, permite la transmisión de datos al doble de velocidad

Figura 1: Comparación de 4 generaciones USB.

LOGO	VERSION	SPEED NAME	MAX TRANSFER RATE IN BITS	DEFAULT POWER	CURRENT SYMBOL	TRANSFER TIME FOR SINGLE 1GB FILE
	USB 1.1	LOW SPEED FULL SPEED	1.5 Mbps 12 Mbps	5 Volt 500 mA		~22.2 Minutes
	USB 2.0	HIGH SPEED	480 Mbps	5 Volt 500 mA		~33.6 Seconds
	USB 3.0	SUPER SPEED	5 Gbps	5 Volt 900 mA		~2.8 Seconds
	USB 3.1	SUPER SPEED PLUS	10 Gbps	5 Volt 900 mA		~1.4 Seconds

de USB3.0, es decir, 10Gbps [4]. En la figura 1 se observa una comparación entre las generaciones USB desarrolladas hasta la fecha, destacando la velocidad de transmisión de cada una de ellas.

En este paper se presenta una descripción del comportamiento del decodificador 128b/132b y el búfer elástico pertenecientes al receptor compatible con el estándar USB3.1. Adicionalmente se describen estos circuitos en lenguaje verilog HDL y se sintetizan en tecnología CMOS 180nm, evidenciando el área utilizada, el consumo de potencia y el reporte de tiempo.

Capítulo 1

CAPA FÍSICA DEL RECEPTOR USB 3.1

Encargada de la transmisión y recepción de un flujo de bits desestructurados; la capa física es una de las 4 capas que componen el estándar USB3.1. Siempre en contacto con el medio físico de comunicación, describe las características eléctricas, mecánicas y funcionales de la interfaz y proporciona los datos a los niveles superiores. Terminan de componer el estándar, y en un nivel superior, la capa de enlace, capa de protocolo y la capa de host/dispositivo [4]. Para el estándar UBS3.1 la capa física se puede observar en la figura 2. En primera instancia la información es recibida en forma de bits en serie por el bloque de recepción diferencial y recuperación de reloj, el cual se observa en la figura 3 y es el encargado de procesar estos datos y extraer el reloj inmerso en ellos.

Figura 2: Capa física del receptor UBS3.1

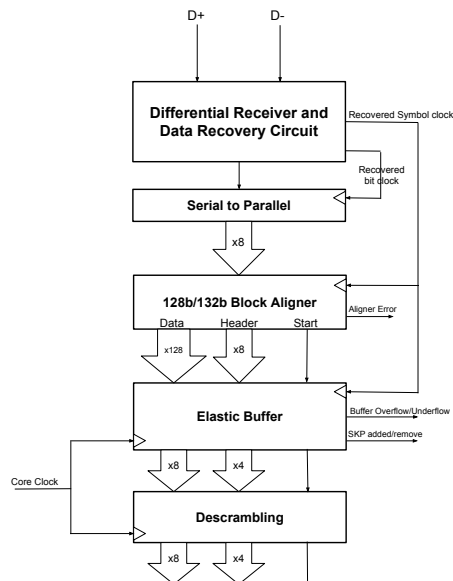
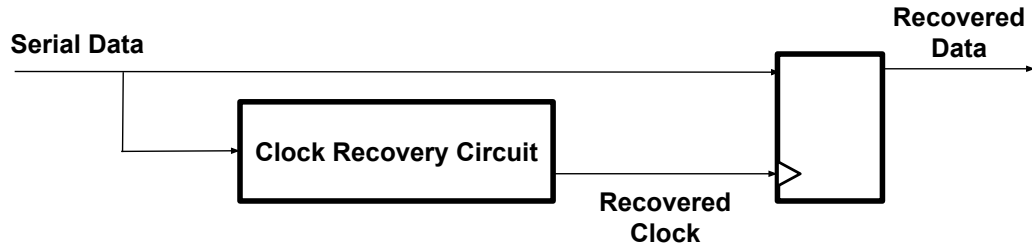


Figura 3: Circuito de recuperación de reloj



Posteriormente los datos son enviados a el bloque conversor serial/paralelo para ser empaquetados en forma de bytes (8 bits) y poder ser procesados por el decodificador 128b/132b el cual recibe un byte a la vez, y entrega 128 bits de datos (payload), 4 bits de encabezado y un bit de control, que funciona como *enable*. Acto seguido, los bits son procesados por un búfer elástico el cual regula la diferencia entre el reloj recuperado y el reloj del host para finalmente ser descrambleados por el *unscrambler* y seguir su camino hacia la capa de enlace. Este artículo se centra en dos de los bloques mencionados; el decodificador 128b/132b y el búfer elástico, los cuales se explican en detalle a continuación.

1.1. DECODIFICADOR 128B/132B

La codificación se utiliza en los estándares de transmisión de datos en serie para asegurar suficientes transiciones de datos, lo cual es esencial para la recepción de los mismos. El codificador 128b/132b provee las suficientes transiciones de datos y presenta sólo una sobrecarga del ancho de banda del 3%. En la codificación 8b/10b usada en la generación Superspeed cada símbolo estaba representado por 10 bits, es decir, los bytes de datos normales estaban representados por 10 bits causando una sobrecarga del 25% [4], mientras que la codificación SuperSpeedPlus tiene 128 bits que están representados por 132 bits por trama [4].

La capa física del emisor codifica los datos en bloques de 132 bits. Cada bloque está conformado por un encabezado de 4 bits y una carga útil de 128 bits. El encabezado de 4 bits es necesario para la identificación de la carga útil, siendo 1100b para datos y 0011b para los bloques de control [5]. Debido a que los datos son recibidos por el decodificador en bloques de 8 bits, como se observa en la figura 2, una de sus funciones es el alineamiento y organización de estos bytes para formar el bloque de 128 bits con

su respectivo encabezado, además de la señal de control que requiere el búfer elástico, la cual indica que los datos están listos para ser procesados por el búfer.

Para lograr el alineamiento y organización de los datos, el decodificador debe recibir y acumular 17 bytes (136 bits) provenientes del conversor serie/paralelo. El primer byte recibido contiene el encabezado del bloque y los primeros 4 bits de la carga útil, sean de datos o de control. Los 15 bytes siguientes serán puramente carga útil, y el último byte recibido contiene los últimos 4 bits de la carga útil, completando así los 128 bits requeridos, y el encabezado del siguiente bloque de datos. Es por esto que los siguientes 16 bytes entregados por el conversor serie/paralelo serán la carga útil del segundo bloque de datos.

La segunda función del decodificador corresponde a la detección de error en el encabezado. Las reglas de detección y corrección de error son las siguientes:

1. Los errores de un solo bit en el encabezado del bloque se informan a la capa de enlace por medio de la salida *aligner error*, y se corrigen [4].
2. Los errores de dos bits en el encabezado del bloque se informan a la capa de enlace por medio de la salida *aligner error*, y el encabezado no sufre modificaciones [4].

1.2. BÚFER ELÁSTICO

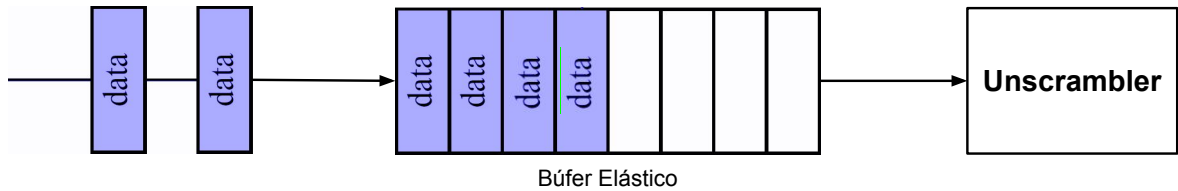
Los búfers elásticos (también conocidos amortiguadores de sincronización) son usados con el fin de garantizar la correcta sincronización de los datos al unir dos dominios de reloj diferentes. Un búfer, en términos más comunes, es un sistema FIFO (First-Input-First-Output) donde los datos se depositan a una velocidad determinada basada en un dominio de reloj, y se expulsan a una velocidad que proviene de un dominio de reloj diferente [2]. Debido a la total independencia de los relojes y las diferentes formas de generación de los mismos; nunca tendrán exactamente la misma frecuencia ni estarán perfectamente sincronizados. Esto posibilita el hecho de que ingresen más datos de los que se expulsan, causando un desbordamiento; o por el contrario, se soliciten más datos de los que estén ingresando, haciendo que el búfer se desocupe. Para evitar esta situación, un Buffer Elástico tiene la capacidad de almacenar y poner en cola cierta cantidad de datos, además de insertar o eliminar símbolos especiales durante interva-

los específicos, denominados *SKP ordered sets*, que permiten al búfer compensar las diferencias de reloj [1].

1.2.1 Tipos de búfer elástico Existen diferentes métodos de operación de un búfer elástico. El método **medio-lleno** y el método de **control de flujo** son los más comunes.

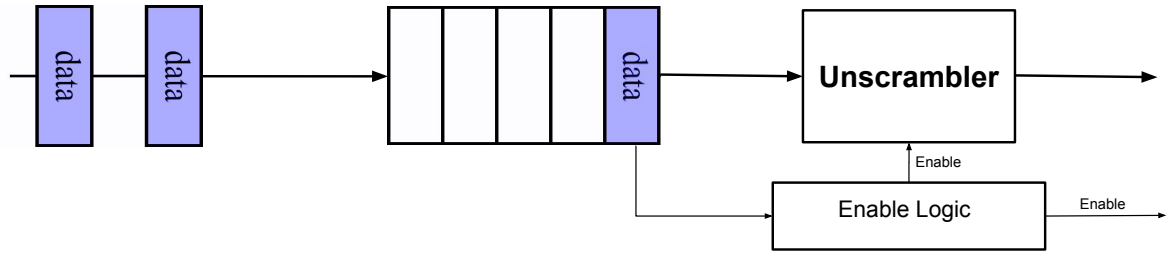
1.2.1.1 Búfer medio-lleno Esta implementación implica un búfer cuyo estado normal es medio lleno, con suficientes entradas antes y después de la entrada del medio para garantizar que las diferencias máximas del reloj puedan compensarse insertando o eliminando símbolos SKP de los conjuntos ordenados SKP. En la figura 4 se observa una ilustración del funcionamiento de este tipo de Búfer.

Figura 4: Búfer elástico de tipo medio-lleno



1.2.1.2 Búfer de tipo control de flujo La implementación de un búfer elástico utilizando el método de control de flujo, difiere de una implementación basada en el método medio-lleno, esto se debe a que no utiliza el mismo mecanismo para compensar el caso en el que el dominio de reloj local es más rápido que el dominio de reloj recuperado. En otras palabras, el método de control de flujo no usa símbolos SKP para compensar el adelanto en la frecuencia del receptor. Sin embargo, los símbolos SKP siguen siendo necesarios al compensar el caso en que el reloj recuperado es más rápido que el dominio de reloj local [1]. Para lograr la compensación en la cual no se usan símbolos SKP, el método de control de flujo simplemente apaga los relojes suministrados a los bloques lógicos aguas arriba del búfer elástico, cuando la entrada inferior del búfer no es válida. En la figura 5 se observa una ilustración del funcionamiento de este tipo de Búfer.

Figura 5: Búfer elástico de tipo control de flujo



Los tamaños de estos búferes dependen de tres factores:

1. El intervalo en el que el transmisor planifica los conjuntos ordenados SKP a enviar.
2. El tamaño máximo de carga útil admitido por el dispositivo.
3. El máximo delta de frecuencia permitido por el estándar.

Cada una de estas variables se usa para determinar la cantidad máxima de tiempo que debe soportar el búfer elástico sin recibir ningún símbolo SKP.

Capítulo 2

IMPLEMENTACIÓN

Siguiendo el estándar USB3.1 son propuestos los módulos del decodificador 128b/132b y el búfer elástico compatibles con ésta generación. Los módulos son descritos en lenguaje Verilog HDL y sintetizados en una tecnología CMOS 180nm. A continuación se muestra su arquitectura, datapath y modo de funcionamiento.

2.1. DECODIFICADOR 128B/132B

Uno de los mayores inconvenientes a la hora de diseñar esquemas de comunicación serial, es la frecuencia de operación que éstos módulos deben implementar. USB3.1 permite la transferencia de datos a una velocidad de 10Gbps. Debido a que el decodificador procesa éstos bits a un nivel de agrupación mayor (mínimo en forma de bytes), la frecuencia real del reloj que controla éste módulo será:

$$\frac{10^9}{8} = 1,25[GHz]$$

Al momento de realizar el diseño del decodificador, uno de los principales aspectos a tener en cuenta es la tecnología CMOS en la cual se implementará, debido a la restricción en la velocidad de operación que conlleva dicha tecnología. Por este motivo es necesario rediseñar la rquitectura del decodificador, permitiendo que procese más información a una velocidad menor, manteniendo la compatibilidad USB3.1. En la tabla 1 se observa la relación entre el tamaño de los símbolos procesados por el decodificador y la frecuencia a la que éste módulo debe trabajar para mantener la compatibilidad. El datapath del decodificador 128b/132b diseñado se muestra en la figura 6; se eligió un tamaño de símbolo de 32 bits permitiendo al decodificador trabajar a una frecuencia

de 312.5MHz, admitida en CMOS 180nm.

Tabla 1: Frecuencia de operación del Decodificador según el tamaño de los símbolos.

Agrupación de bits	Frecuencia de operación
8	1.25GHz
16	625MHz
32	312.5MHz
64	156.25MHz

En la figura 6 también se observan la máquina de estados finita, el contador, y el banco de registros que componen el decodificador. Se observan cinco registros de 32 bits debido a que el decodificador debe entregar un total de 132 bits en su salida, 4 bits de encabezado y 128 bits de carga útil, por lo cual deben almacenarse mínimo cinco paquetes de 32 bits para poder entregar la primera trama decodificada. El tamaño del banco de registros permite almacenar 160 bits antes de entregar la primera trama, esto indica que además de la trama de datos decodificada que entrega éste módulo, fueron recibidos 28 bits pertenecientes a la segunda trama (los cuatro primeros bits de encabezado, y 24 bits de carga útil) por lo que la máquina de estados en conjunto con el contador deben seleccionar la ubicación del encabezado y la carga útil para cada trama de datos siguiente. Por este motivo, la máquina de estados que se observa en la figura 8, está compuesta de 8 estados (sin contar el estado WAIT), ya que el encabezado de cada trama decodificada se encontrará en 8 posiciones diferentes a lo largo del paquete de 32 bits que lo contenga, como se observa en la figura 7. Del mismo modo se ubicará la carga útil asociada a ese encabezado.

Figura 6: Datapath del Decodificador 128b/132b.

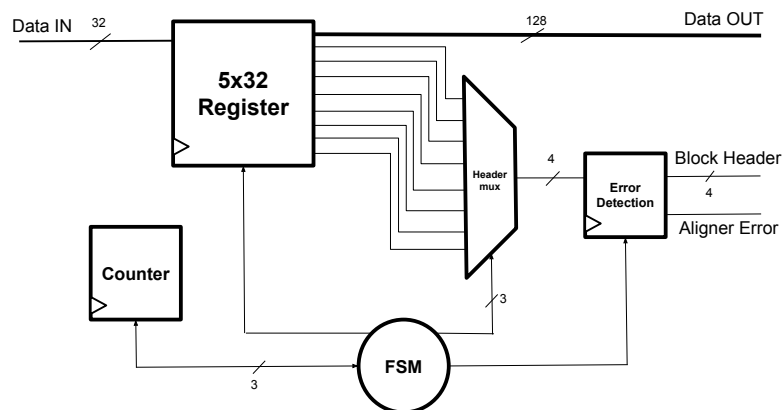
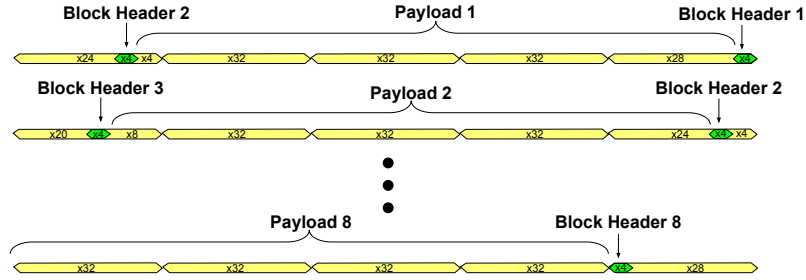


Figura 7: Ubicación del encabezado y la carga útil de datos.



2.1.1 Detección de error Una vez la máquina de estados seleccione el encabezado de la trama de datos a decodificar, éste pasará a través del bloque de detección y corrección de errores, el cual cumple la función de informar a la capa de enlace por medio de la salida *aligner error*, y corregir el encabezado si se presenta un error de un bit. Los errores de dos bits en el encabezado del bloque también son informados a la capa de enlace por medio de la misma salida, pero en este caso el encabezado no sufre modificaciones.

Para lograr la detección y corrección de errores se deben conocer las únicas posibilidades de encabezados que llegan al decodificador; esto es, 0011 para tramas de datos y 1100 para tramas de control. Así la identificación del tipo de error y corrección (cuando sea posible) se realizan con base en la tabla 2.

Figura 8: Máquina de estados finita del Decodificador 128b/132b.

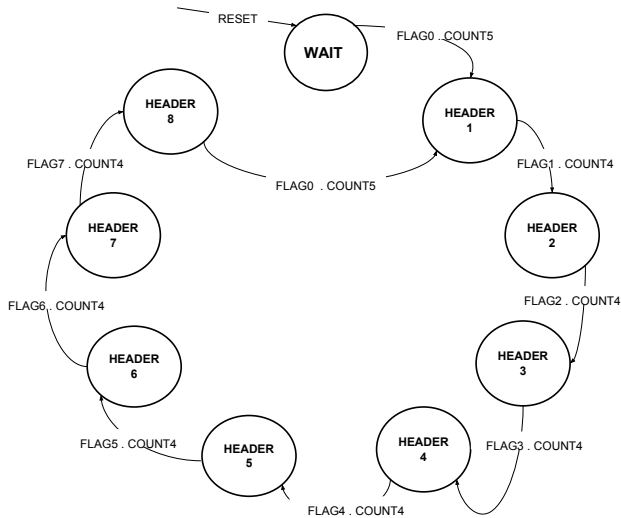


Figura 9: Simulación comportamental del Decodificador.



Tabla 2: Lógica de salida del detector de error en el encabezado.

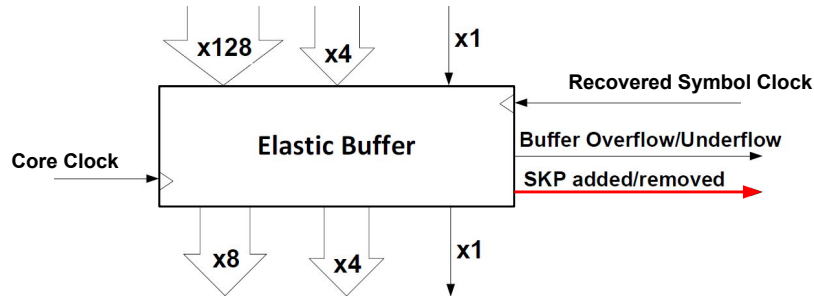
Block Header IN	Block Header OUT	Aligner Error	Bit Error
0000	0000	1	2 bits
0001	0011	1	1 bit
0010	0011	1	1 bit
0011	0011	0	No Error
0100	1100	1	1 bit
0101	1010	1	2 bits
0110	0110	1	2 bits
0111	0011	1	1 bit
1000	1100	1	1 bit
1001	1001	1	2 bits
1010	1010	1	2 bits
1011	0011	1	1 bit
1100	1100	0	No Error
1101	1100	1	1 bit
1110	1100	1	1 bit
1111	1111	1	2 bits

El funcionamiento comportamental del decodificador, en el cual se especifican sus entradas y salidas se muestra en la figura 9. Durante los primeros 5 ciclos de reloj, el decodificador recibe los primeros 5 paquetes de 32 bits y no hay ningún dato en su salida. En el sexto ciclo del reloj entrega la primera trama de datos decodificada, con su respectivo encabezado. De esta forma, continúa entregando los datos alineados cada cuatro ciclos de reloj.

2.2. BÚFER ELÁSTICO

La metodología de diseño para el búfer elástico compatible con el estándar USB3.1 se detalla a continuación.

Figura 10: Entradas y salidas del Búfer Elástico USB3.1



2.2.1 Elección del tipo de Búfer Elástico Con base en [4], y de acuerdo a la especificación de la capa física, que se observa detalladamente en la figura 10, donde se muestra la posibilidad de añadir símbolos SKP por el búfer para compensar el retardo en el reloj recuperado, se opta por el diseño del búfer elástico de tipo **medio-lleno**.

2.2.2 Profundidad del Búfer Elástico Para conocer la profundidad mínima que debe tener el búfer elástico es necesario tener en cuenta el peor caso de transmisión, el cual se produce al enviar el paquete de datos de tamaño máximo, durante el cual no es posible enviar símbolos SKP por parte del transmisor. Para el estándar USB3.1, el tamaño máximo de carga útil es de 1024 bytes [4]. Otra restricción importante hace referencia a la precisión de los relojes permitida por el estándar. Se requiere que la precisión de cada reloj de referencia esté dentro de ± 300 ppm. Esto proporciona una diferencia de frecuencia máxima entre los dos dispositivos del enlace de ± 600 ppm. Además, el SSC (*Spread-Spectrum Clocking*) crea un delta de frecuencia que tiene una diferencia máxima de 5000 ppm. La magnitud total del delta de frecuencia puede variar de -5300 a 300 ppm [4]. Con base en estos valores se calcula la profundidad mínima del búfer, como sigue:

$$Diferencia\ de\ frecuencia = \frac{|-5300 - (-300)|}{1000000} \quad (2.1)$$

$$Diferencia\ de\ frecuencia = 0,0056 \quad (2.2)$$

Un desplazamiento en la posición del búfer ocurre cada:

$$1\ desplazamiento = \frac{1}{0,0056} \text{ ciclos de reloj} \quad (2.3)$$

$$1 \text{ desplazamiento} = 179 \text{ ciclos de reloj} \quad (2.4)$$

Sabiendo que:

$$\text{Máxima carga útil} = 1024 \text{ bytes} \quad (2.5)$$

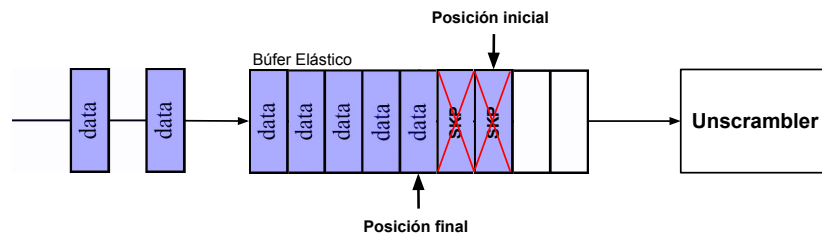
La cantidad de desplazamientos que ocurren mientras se transmite la máxima carga útil son:

$$\text{Cantidad de corrimientos} = \frac{1024}{179} = 6 \text{ bytes} \quad (2.6)$$

Por lo tanto el búfer debe tolerar desplazamientos hasta ± 6 ubicaciones durante la recepción de una trama. La profundidad mínima de este búfer es, por lo tanto, 13 bytes (un rango de ± 6 ubicaciones desde la ubicación actual).

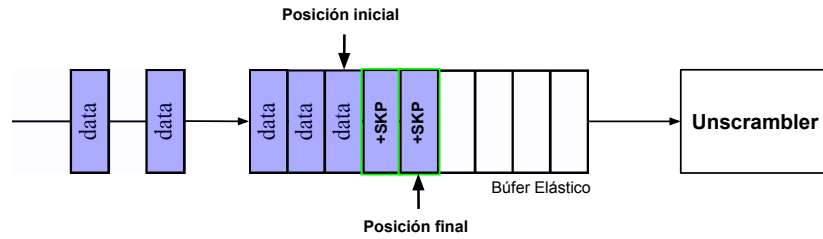
2.2.3 Control del Búfer Elástico Para lograr la correcta inserción o eliminación de los símbolos SKP, es necesario implementar un módulo que detecte el ingreso de estos símbolos, y reporte su disponibilidad de uso a la máquina de estados central. La cual, con base en la información obtenida de los contadores de frecuencia, quienes proveen el estado de llenado actual del búfer, decide si eliminar o insertar estos símbolos con el fin de retornar el búfer a su posición ideal. En las figuras 11 y 12 se observa el mecanismo de eliminación o inserción de los símbolos SKP, respectivamente.

Figura 11: Nivelación del Búfer por medio de la eliminación de símbolos SKP



El datapath del búfer se puede observar en la figura 13; se ha realizado un diseño con 3 símbolos de profundidad, de 32 bits cada uno, a cada lado del búfer, cumpliendo con la profundidad mínima posible, manteniendo la compatibilidad con el Decodificador 128b/132b diseñado y sobrepasando el requisito del tamaño especificado en [4], el cual requiere mínimo 22 símbolos de profundidad, de 8 bits cada uno. En la figura además se observan los bloques que lo conforman, tales como el registro de desplazamiento

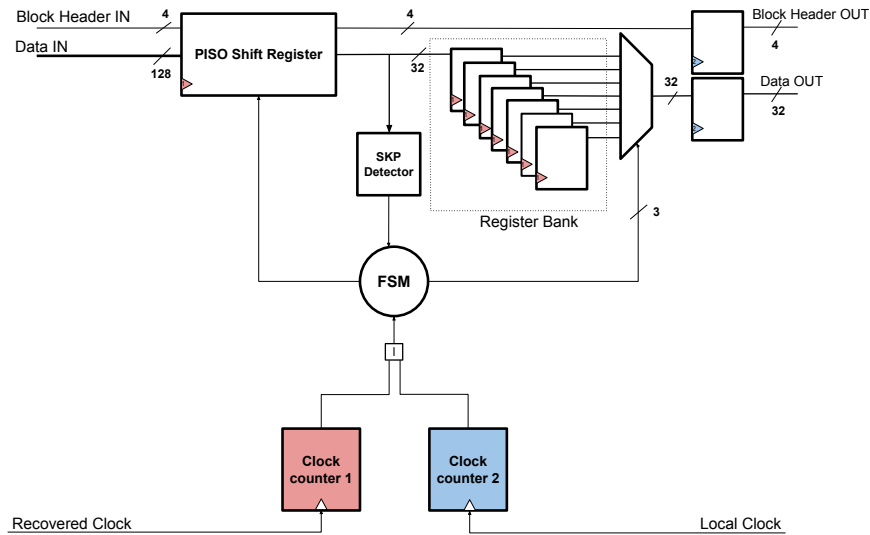
Figura 12: Nivelación del Búfer por medio de la inserción de símbolos SKP



PISO, el contador ascendente/descendente, el banco de registros del búfer, el detector de símbolos SKP y la máquina de estados finita.

Para entender el funcionamiento del búfer es necesario conocer la tarea que desempeña cada uno de los bloques que lo conforma.

Figura 13: Datapath del Búfer Elástico



2.2.4 Registro de desplazamiento PISO Es el encargado de recibir la trama de datos decodificada proveniente del decodificador 128b/132b, es decir, los 128 bits de carga útil con su respectivo encabezado y entregar paquetes más pequeños (32 bits) al banco de registros del búfer. En otras palabras, se encarga de dividir la carga útil en cuatro (4) partes y entregar una de esas cuatro partes a la vez, con el fin de serializar los datos provenientes del decodificador.

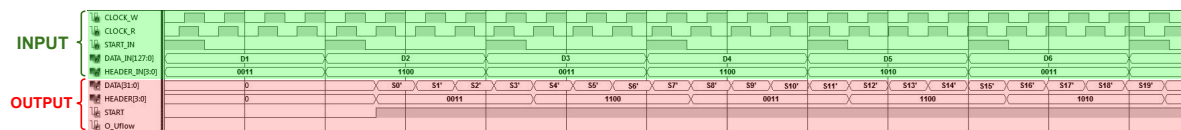
2.2.5 Banco de registros Recibe los paquetes de 32 bits provenientes del registro de desplazamiento PISO y desplaza cada uno de ellos serialmente hasta que son expulsados del búfer. El banco cuenta con 7 registros de profundidad e idealmente debe permanecer medio lleno, por lo que, si se produce un aumento de frecuencia del reloj recuperado con respecto al reloj local, se pueda compensar con un tiempo igual a 3 ciclos de reloj, que corresponden a los 3 registros vacíos del banco. De igual forma, si la frecuencia del reloj local es la que aumenta con respecto al reloj recuperado, produciendo que los datos salgan del búfer más rápido de lo que llegan, el banco se irá desocupando gradualmente (recordar que se encuentra medio lleno) permitiendo la compensación de las frecuencias.

2.2.6 Detector de símbolos SKP Su función es inspeccionar todos los datos que ingresan al búfer y reportar a la máquina de estados finita, el ingreso de símbolos SKP, para que puedan ser usados en la nivelación del estado del búfer.

2.2.7 Contador ascendente/descendente Está conformado por dos contadores de tipo ascendente, uno para cada domino de reloj, y su objetivo es llevar el registro en tiempo real de las velocidades de los relojes, calculando la diferencia entre ellos y seleccionando la posición del registro que indica el estado de lleno del búfer.

2.2.8 Máquina de estados Se encarga de controlar e integrar todos los bloques pertenecientes al búfer, así mismo indica cuando hay datos válidos en la salida del búfer y reinicia todo el sistema en caso de presentarse desborde o vaciado. El diagrama de tiempo del búfer elástico puede observarse en la figura 14.

Figura 14: Simulación comportamental del Búfer Elástico



Capítulo 3

SÍNTESIS Y RESULTADOS

El decodificador y el búfer elástico han sido sintetizados en una tecnología CMOS 180nm. La frecuencia del reloj de ambos bloques es de 312.5MHz y el jitter se configuró en 100ps siguiendo el estandar USB3.1. A continuación se muestran los resultados del peor caso de síntesis con el fin de conocer el área usada, consumo de potencia y la máxima frecuencia de operación según el *critical path* de cada uno de los bloques diseñados.

En la tabla 3 se muestra el reporte de área usada, consumo de potencia y máxima frecuencia de operación del decodificador.

Tabla 3: Área, consumo de potencia y frecuencia de operación del Decodificador en tecnología CMOS 180nm.

Modulo	Celdas	Área Total [μm]	Potencia Total [mW]	Margen de Frecuencia +[ps]	Máxima Frecuencia [MHz]
Decodificador	1066	23587	15.671	270	341

Se observa que el diseño sintetizado cumple con los requisitos de frecuencia para ésta tecnología, obteniendo un margen positivo de frecuencia de operación de 270ps, dando como resultado una máxima frecuencia de operación de 341MHz. De igual forma, en la tabla 4 se muestra el reporte de área usada, consumo de potencia y máxima frecuencia de operación del búfer elástico. Al igual que en el caso del decodificador, el diseño sintetizado cumple con los requisitos de frecuencia para ésta tecnología, obteniendo un margen positivo de frecuencia de operación de 60ps, dando como resultado una máxima frecuencia de operación de 318MHz para éste módulo. Finalmente en las figuras 15 y

16 se muestra el layout realizado para cada uno de los bloques.

Tabla 4: Área, consumo de potencia y frecuencia de operación del Búfer Elástico en tecnología CMOS 180nm.

Modulo	Celdas	Área Total [μm^2]	Potencia Total [mW]	Margen de Frecuencia +[ps]	Máxima Frecuencia [MHz]
Búfer Elástico	1279	33027	22.102	60	318

Figura 15: Layout del Decodificador en tecnología CMOS 180nm.

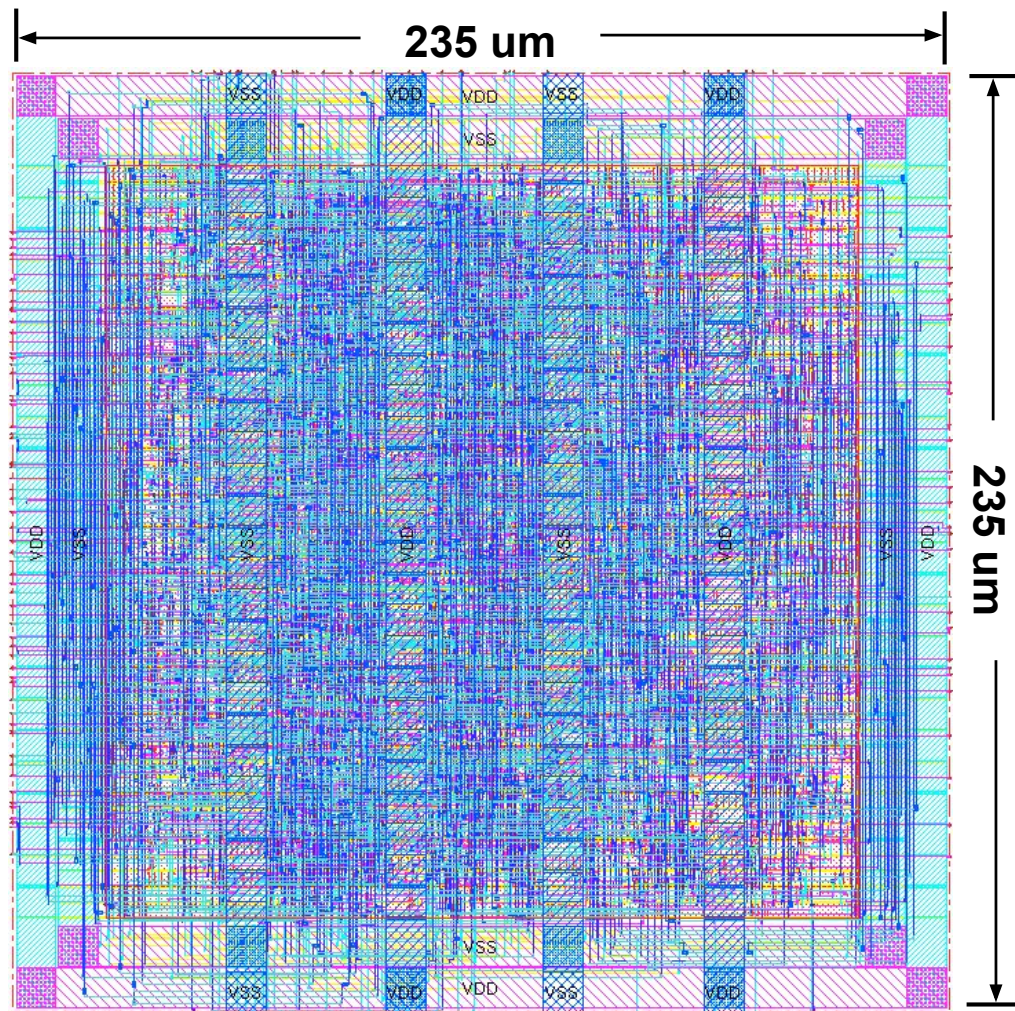
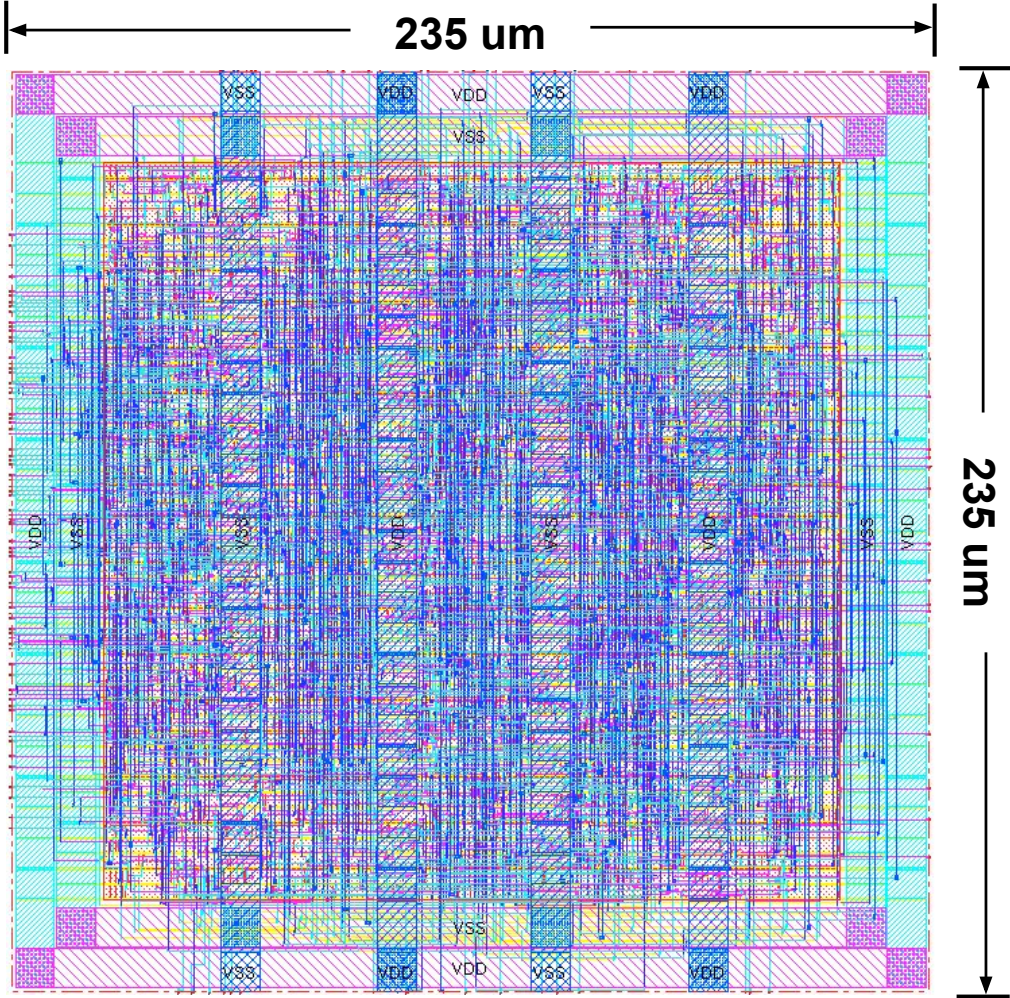


Figura 16: Layout del Búfer Elástico en tecnología CMOS 180nm.



Capítulo 4

CONCLUSIONES

En este trabajo se ha realizado la implementación y verificación de dos bloques compatibles con la capa física del receptor USB3.1, el Decodificador 128b/132b y el Búfer Elástico, haciendo uso del lenguaje verilog HDL. El esquema RTL y la forma de onda del modelo de comportamiento simulado para estos bloques se ha obtenido de acuerdo con los métodos disponibles en la literatura. Además se realizó la síntesis de ambos bloques en una tecnología CMOS 180nm obteniéndose resultados satisfactorios.

El Decodificador y el Búfer Elástico soportan la frecuencia de reloj necesaria para la comunicación USB3.1 incluso en el peor caso de síntesis, en el cual podrían funcionar a una frecuencia de 341MHz y 318MHz respectivamente.

Como trabajo futuro es necesario el diseño e implementación del *unscrambler* compatible con los bloques realizados anteriormente, así como la integración y simulación en conjunto con el decodificador y el búfer, para posteriormente sintetizar y realizar layout de la capa física del USB3.1 completa.

REFERENCIAS

- [1] CHANG, H., PENG, T., KE, D., XU, T., AND MENG, J. Design and implementation of elastic buffer for universal serial bus 3.0. In *IEICE Electronics Express* (2016).
- [2] DAVID R. STAUFFER, JEANNE TRINKO MECHLER, M. S. K. D.-C. R. O. A. *High Speed Serdes Devices and Applications*, 1 ed. 2008.
- [3] GHAFFARI, F., AKOGLU, A., VASIC, B., AND DECLERCQ, D. Multi-mode low-latency software-defined error correction for data centers. In *2017 26th International Conference on Computer Communication and Networks* (July 2017), pp. 1–8.
- [4] HEWLETT-PACKARD COMPANY, INTEL CORPORATION, M. C. R. C. S.-E. T. I. Universal serial bus "3.1 specification", revision 1.0.
- [5] MANDAL, R. R., AND DANIEL, P. A study to implement superspeedplus usb 3.1 physical layer. In *Proceedings of ASAR-IJIEEE International Conference, Mysore India* (November 2016).
- [6] MICHELOGIANNAKIS, G., BALFOUR, J., AND DALLY, W. J. Elastic-buffer flow control for on-chip networks. In *2009 IEEE 15th International Symposium on High Performance Computer Architecture* (Feb 2009), pp. 151–162.
- [7] S MONIKA, G. S. Study and implementation of physical layer coding used in super speed usb. In *International Journal of Scientific Engineering and Research* (January 2014).
- [8] SEITANIDIS, I., PSARRAS, A., CHRYSANTHOU, K., NICOPOULOS, C., AND DIMITRAKOPOULOS, G. Elastistore: Flexible elastic buffering for virtual-channel-based networks on chip. *IEEE Transactions on Very Large Scale Integration Systems* 23, 12 (Dec 2015), 3015–3028.

- [9] SRIDHAR GANGADHARAN, S. C. A. *Constraining Designs for Synthesis and Timing Analysis: A Practical Guide to Synopsys Design Constraints*, 1 ed. Springer-Verlag New York, 2013.
- [10] V, G. T., AND D, M. Elastic buffer for virtual channels in heterogenous switching network on chip. In *2016 International Conference on Wireless Communications, Signal Processing and Networking* (March 2016), pp. 2045–2048.

BIBLIOGRAFÍA

GHAFFARI, Fakhreddine, *et al.* Multi-Mode Low-Latency Software-Defined Error Correction for Data Centers. 26th International Conference on Computer Communication and Networks. 2017.p.1–8.

HONG, Chang, *et al.* Design and implementation of elastic buffer for Universal Serial Bus 3.0. IEICE Electronics Express. 2016.p.105.

MANDAL, Rakesh y PHILEMON, Daniel. A study to implement Superspeedplus USB 3.1 Physical layer. Proceedings of ASARIJ IEEE International Conference,2016.

MICHELOGIANNAKIS, George, y DALLY, William. Elastic-buffer flow control for on-chip networks. IEEE 15th International Symposium on High Performance Computer Architecture,2009.p.151–162.

SEITANIDIS, Ioannis *et al.* ElastiStore: Flexible Elastic Buffering for Virtual-Channel-Based Networks on Chip. IEEE Transactions on Very Large Scale Integration Systems,2015.p.3015–3028.

SKARTEL, Monika y GAGHEL, Shruthi. Study and Implementation of Physical Layer Coding Used in Super Speed USB. International Journal of Scientific Engineering and Research,2014.p.52-55.

STAUFFER, David *et al.* High Speed Serdes Devices and Applications: Springer,2008. ISBN 978-0-387-79833-2.