Capítulo 7

Resultados de simulación del enlace

7.1. Introducción

En los capítulos anteriores se ha mostrado todo el flujo de diseño seguido para llegar hasta el layout del cicuito. Sin embargo, es necesario validar este proceso mediante la simulación a través de la herramienta *Spectre* de toda la circuitería propuesta. Como se dispone tanto de circuitos analógicos como digitales, es necesaria una simulación eléctrica para comprobar el correcto funcionamiento, por lo que el sistema ha sido analizado a nivel de transistor optimizando al máximo su respuesta. Esto también obliga a realizar análisis transitorios para las simulaciones: se quiere conocer la evolución temporal de las señales con respecto al tiempo y así poder estudiar las formas de onda y sus características temporales.

En los apartados que siguen a esta introducción se realizará un análisis exhaustivo del serializador y del deserializador, así de como de su interacción con los drivers y su respuesta temporal. Se podrán comprobar todas las consideraciones de diseño realizadas en el capítulo 5 y se destacarán los aspectos más relevantes y conflictivos del funcionamiento del circuito. Por otro lado, se hará énfasis en alguna consideración adicional sobre la realización de las simulaciones, como por ejemplo cómo se ha modelado el camino de señal existente entre el driver y el receptor LVDS ó cómo se han tenido en cuenta los errores que introduce el proceso de fabricación en la verificación del diseño. Se ha procurado que el modelo de simulación construido represente de la manera más fiel posible al comportamiento real del circuito. Esto es especialmente necesario cuando se usan interfaces de alta velocidad, donde cualquier detalle que se deje en el aire puede ser un problema a la hora de la implementación práctica.

Todas las figuras que se presentan, a pesar de estar divididas por bloques, se corresponden con una simulación global de todo el enlace. Es decir, se simula todo el enlace para luego representar únicamente la evolución temporal de las señales que nos interesen en cada momento. Se ha hecho así para que los resultados que aquí se presenten sean lo más representativos posible y se pueda comprobar cómo todos los bloques interactúan entre sí sin causar una degradación en el comportamiento de los mismos. No obstante, durante el proceso de diseño cada uno de los bloques fue estudiado y simulado individualmente para poder hacer así un estudio pormenorizado.

7.2. Simulación del transmisor

El objetivo de este bloque es el de recibir una palabra digital de entrada en formato paralelo y generar a la salida una señal LVDS que contenga esa información en un formato serie. Para ello, se necesitaban tres bloques que realizaban las funciones básicas: serializar, codificar en Manchester y transmitir en LVDS. Los bloques que se encargaban de estas tareas, así como las señales implicadas se describieron en el capítulo 5. También es importante comprobar cómo el transmisor presenta una interfaz compatible con el generador AER con el que tiene que interactuar, es decir, que la generación de la señal Ack es correcta en función de los cambios en Request.

La figura 7.1 nos muestra el funcionamiento del serializador, así como la generación de las señales de *Request* y *Ack*. En la simulación realizada se ha tomado como entrada una secuencia que alterna 1's y 0's y que se invierte entre un dato y otro: cada bit se invierte en transmisiones sucesivas.



Figura 7.1: Simulación del serializador

En la simulación realizada se observa cómo la señal *Request* es la que marca el inicio de la serialización de los datos y la que provoca el nivel bajo de la señal de asentimiento. Este estado se puede ver que se mantiene durante todo el proceso de transmisión, indicando que el receptor se encuentra ocupado. Se ha mostrado también la evolución de los datos serializados, donde podemos ver la secuencia de entrada que se introdujo de forma paralela, pero esta vez en formato serie. Los periodos de reposo entre dato y dato se dan con la línea a nivel alto, por lo que se confude con el preámbulo y no se observa bien en la simulación. Se verá con más claridad en la figura 7.2, donde se muestra el comportamiento del codificador Manchester.



Figura 7.2: Simulación del codificador Manchester

En la figura 7.2 se han representado como entradas el reloj, los datos serializados y Ack y se muestra el resultado de la codificación Manchester. Se observa cómo todos los datos comienzan con el preámbulo de dos bits a 1 (esta vez si es fácil observarlo por los continuos flancos de la señal Manchester) y a continuación se comienzan a codificar los 16 bits de la dirección AER. Se comprueba también como la señal Ack controla la habilitación de la codificación a la salida: cuando se encuentra a nivel bajo tenemos los datos a la salida, mientras que cuando no está habilitada se tiene un valor de reposo. Por último, cabe comentar cómo, gracias a la cadena de retraso con la que se modificó el transmisor Manchester clásico, no aparecen *glitches* en la señal generada.

La figura 7.3 presenta un detalle de la figura 7.2 con el objeto de confirmar que la codificación Manchester se realiza conforme a lo expuesto en el resto del documento y que se confirman las hipótesis de diseño que se hicieron en el capítulo 5. En dicha figura se puede observar, aquí con mayor detalle, cómo es la codificación Manchester y cómo los datos de salida se corresponden completamente con los de entrada. Será ésta la señal raíl a raíl (de 0 a 1V) la que alimente la operación del driver LVDS y que el receptor tenga que decodificar.



Figura 7.3: Detalle de la operación del codificador Manchester

En la figura 7.4 se representa la señal diferencial de salida del driver. Como todo el procesado que se realiza con estas señales se implementa a través de circuitos diferenciales no nos interesan tanto las señales unipolares, como las versiones diferencial y de modo común. Es por ello que en la figura se han representado ambas. Por supuesto, las simulaciones se han hecho con el enlace cargado con una resistencia de 100Ω de forma que se tenga adaptación de impedancias y la corriente nominal del driver dé lugar a unas amplitudes diferenciales adecuadas al estándar LVDS. De hecho, se tiene una amplitud del modo diferencial del orden de 350mV, más que suficiente para que el enlace opere dentro de los rangos adecuados de funcionamiento. En los periodos de no actividad del enlace, se observa también cómo el dirver es capaz de estabilizarse a un valor constante en cuanto a la amplitud del modo diferencial. Esto es esencial para que no se produzcan recepciones espúreas por disparos no deseados del receptor en los periodos de silencio.

Por otro lado, se observa cómo el modo diferencial se ajusta a un valor cercano a 1.2 V, tal y como indica el estándar LVDS. No obstante, se observa cómo este modo común presenta una cierta variación entorno a ese valor nominal. Por un lado, las conmutaciones en la señal de salida provocan picos en la señal de modo común, debido a los acoplamientos y al funcionamiento en conmutación de los transistores del driver. Estas son las variaciones rápidas que se observan, pero también parecen existir componentes de variación de más baja frecuencia. Éstas se pueden deber a pequeñas desadaptaciones y reflexiones que se puedan producir en el enlace y que se acoplan a la señal en forma de un ruido de modo común. Sin embargo, esto no debe preocupar en exceso pues los receptores LVDS deben tener un rango de rechazo al modo común suficiente para ser insensibles ante estas variaciones.



Figura 7.4: Señal LVDS a la salida del driver

7.3. Modelo para el medio de transmisión

LVDS es una interfaz para la comunicación de alta velocidad entre chips que pueden estar en la misma PCB o bien en lugares más distantes. En cualquier caso, la señal tiene que atravesar un medio de transmisión que con total seguridad tendrá características dispersivas, ya sea un cable o una pista de una PCB. Además, las frecuencias que estamos manejando (del orden del GHz) son lo suficientemente grandes como para que tengamos que tener en cuenta los efectos de línea de transmisión que se dan en estos medios dispersivos. La aproximación de la teoría de circuitos no es válida pues no se cumple que las distancias típicas de los medios de transmisión sean mucho menores que la longitud de onda de la señal que transportan (a 1 GHz le corresponde una longitud de onda de 30 cm. Por tanto, para asegurar la robustez del enlace, es necesario modelar estos efectos en las simulaciones con el objeto que se correspondan en el mayor grado posible con el comportamiento real.

La figura 7.5 muestra el escenario sobre el que vamos a trabajar, que se ha tomado como el más desfavorable: los chips en distintas PCB's y comunicados a través de un cable. Como medio

de transmisión se ha elegido un cable UTP cat5e puesto que es barato y al mismo tiempo permite la transmisión de datos a alta velocidad sobre cortas distancias. Se van a tener en cuenta también los efectos de línea de transmisión que introducen las pistas de la PCB y las desadaptaciones que pueden introducir los circuitos de protección contra ESD, los hilos del bonding o el empaquetado del chip.



Figura 7.5: Esquema del canal

La figura 7.6 ilustra el modelo eléctrico utilizado para hacer las simulaciones [1]. Los circuitos de protección contra ESD y el efecto del encapsulado se han modelado como capacidades parásitas, mientras que los efectos del bonding y de los conectores mediantes inductancias en serie con el camino de señal. Se han tomado valores típicos para las capacidades y las bobinas, pues un estudio exacto resultaría bastante complejo y tampoco aportaría información adicional: C=1 pF y L=7.5 nH. Las trazas de la PCB se han modelado como líneas de transmisión microstrip, mientras que para el cable de pares se ha utilizado un modelo de línea de transmisión diferencial que tiene en cuenta los acoplamientos, retrasos, atenuaciones,... que se producen en el medio de transmisión.



Figura 7.6: Modelo eléctrico del canal de transmisión

El modelo de la línea microstrip se generó con una utilidad que ofrece Cadence para la creación de modelos de medios de transmisión [50]: líneas microstrip, stripline, acopladas,... Además, tal y como se reflejó en el capítulo 3, las líneas se construyeron acopladas, de forma que se pueda conseguir la mayor inmunidad posible ante el ruido acoplado en modo común y disminuir al mínimo las radiaciones EMI de la interfaz. Otro reto de diseño de la línea fue conseguir la adaptación de impedancias diferencial para que la resistencia que se vea desde la salida del driver se parezca lo más posible a los 100Ω que recomienda el estándar. Para el diseño nos basamos en la recomendación AN-888 de *National Semiconductors*, donde se exponen los criterios de diseño de interfaces para la transmisión a alta velocidad. La figura 7.7 muestra el diseño que resultó.



Figura 7.7: Diseño de la línea microstrip acoplada

Las expresiones que se usaron para las impedancias de la línea microstrip [28] fueron:

$$Z_0 = \frac{60}{\sqrt{0,475\epsilon_r + 0.67}} ln\left(\frac{4h}{0,67(0,8w+t)}\right) = 75,73\Omega$$
$$Z_{diff} \approx 2Z_0\left(1 - 0.48e^{-0.96\frac{s}{h}}\right) = 102,72\Omega$$

Como se observa, al menos teóricamente, se conseguirá una buena adaptación de impedancias en lo que se refiere a las trazas de la PCB. No obstante, también es importante asegurar la adaptación con el cable UTP categoría 5 [32], que, según el estándar, tiene una impedancia diferencial de 100 Ω . No obstante, en el entorno de diseño no se disponía de un modelo concreto para el cable UTP, por lo que se aprovecharon los modelos de líneas acopladas existentes y se modificaron para adecuar su comportamiento al del cable UTP. La figura 7.8 [25] muestra cómo quedó el modelo con las modificaciones propuestas.



Figura 7.8: Modelo usado para el cable UTP

Una vez que se ha estudiado el modelado del canal de transmisión, hay que estudiar la propagación de las señales a través de él. Las figuras 7.9 y 7.10 muestran la señal transmitida en distintos puntos del medio de transmisión a través de la representación de las señales en modo común y en modo diferencial.





Figura 7.10: Señal LVDS en distintos puntos del canal (modo diferencial)

7.4. Simulación del receptor

Las simulaciones del canal de transmisión han mostrado cómo se produce la propagación de la señal LVDS a través de él y cuáles son los efectos del canal. En última instancia se tiene la señal a la entrada del driver, que será la que se utilice para recuperar los datos. El receptor, para operar correctamente, necesita que la señal LVDS cumpla con las condiciones mínimas que exige el estándar. Por ejemplo, la amplitud del modo diferencial debe ser superior a 100 mV. La figura 7.11 muestra la señal LVDS a la entrada del receptor utilizado, puediéndose observar la evolución tanto del modo común como del modo diferencial.



Figura 7.11: Señal LVDS a la entrada del receptor

La figura 7.11 muestra cómo la amplitud del modo diferencial es más que suficiente para activar el disparo del comparador que actúa como receptor. Además, se observa que se conservan los flancos con los que la señal fue transmitida y se deja entrever la secuencia de bits transmitida (en este caso una alternancia de 1's y 0's más el preámbulo). El modo común no es todo lo estable que se podría esperar, pues se puede observar que varía rápidamente entorno a su valor nominal de 1.2 V. No obstante, no debe preocupar demasiado, pues se comprobará que el receptor tiene una relación de rechazo al modo común más que suficiente para eliminar este efecto.

Junto con las señales LVDS se ha representado en la figura 7.11 la señal raíl a raíl que da a su salida el driver. En ella, se comprueba que está toda la información de la señal transmitida y que la calidad de la misma (flancos de subida y de bajada, jitter,...) es adecuada como para ser decodificada y recuperada en el receptor implementado.

Uno de los elementos esenciales del receptor es la unidad de recuperación de reloj (CDR). Dicho bloque se había construido a través de un bucle de realimentación que generaba una señal de control para el ajuste de los retrasos que formaban parte del receptor Manchester. La figura 7.12 muestra la evolución de dicho bucle con el objeto de poner de manifiesto la estabilidad del mismo y cómo es capaz de fijar el valor de la señal de control a un valor constante tras una primera fase de entrenamiento.



Figura 7.12: Simulación del CDR (estabilidad del bucle)

En los primeros instantes de la representación de la figura 7.12 se observa el transitorio inicial que tiene lugar antes de la estabilización del bucle. Durante este tiempo el valor de los retrasos es totalmente inadecuado, por lo que se puede ver cómo el reloj recuperado no se corresponde con lo deseado. El DLL internamente detectará este desajute del retardo y, mediante el PFD, variará la tensión de control hasta que el retraso se estabilize a su valor nominal. Se puede comprobar también que la velocidad de convergencia del bucle es adecuada para la aplicación: con sólo dos datos transmitidos, el bucle ya se ha estabilizado. No obstante, habrá que tener en cuenta esto cuando se plantee el diseño del sistema AER: las interfaces de comunicación pueden tener un tiempo de latencia y las primeras direcciones que se pongan en el bus pueden ser incorrectas.

La figura 7.13 muestra un detalle de la figura anterior donde se observa la operación del circuito cuando se ha alcanzado la convergencia del bucle y el receptor funciona correctamente.



Figura 7.13: Simulación del CDR con el bucle estabilizado

Así mismo, en esa figura se puede observar con más precisión la forma del reloj recuperado y cómo la tensión de control permance constante durante todo el proceso de recepción de un dato. Se observa que los flancos de la señal de reloj interna generada son algo lentos, comparados, por ejemplo, con la señal Manchester de entrada (también presentada en la figura). Esto se debe a que dicha señal tiene que atacar al reloj de todos los biestables de un registro de desplazamiento de 16 bits, por lo que la capacidad de carga para la salida del biestable que genera esta señal es grande. No obstante, las simulaciones han demostrado que la señal de reloj recuperada es más que suficiente para la recuperación de los datos sin pérdida de información.

Por otro lado, se puede observar que en los periodos de silencio el reloj recuperado permanece a un valor fijo: no se detectan flancos en la señal de entrada y se suprime el proceso de sincronización. Es en ese momento cuando entran en juego los convertidores que memorizaban la señal de control, impidiendo que se pierda la sincronización. Se puede comprobar también que el valor que captura el convertidor es el correcto y que con 5 bits se tiene la precisión suficiente para representar la señal de control sin pérdida apreciable de información.

Para la recuperación del reloj, el CDR necesitaba un DLL que generara las tensiones de control para los retrasos. La figura 7.14 muestra la evolución de las señales implicadas en la generación de dicha referencia de tensión a lo largo del tiempo.

Durante el periodo de alineamiento del enlace, cuando la señal de control aún no se ha fijado, se puede comprobar que las señales *UP* y *DOWN* tienen un ancho de pulso distinto. Esto indica



Figura 7.14: Funcionamiento del DLL

que los retrasos no están correctamente fijados, por lo que, mientras esto ocurre, la señal que controla los retrasos varía significativamente. Cuando el bucle se estabiliza, simplemente las dos señales coinciden siempre a nivel alto y no se necesita modificar la tensión de control. El tiempo medido en el que ambas coinciden a nivel alto es de unos 150-200 ps, que se corresponde con los tiempos típicos que se tomaron en el diseño del LPF.

El funcionamiento de la señal *resetPD* también se corresponde con lo expuesto en el capítulo dedicado al diseño del circuito. Cada vez que finaliza la recepción de un dato, la tensión en el nudo asociado a *resetPD* comienza a aumentar, hasta llegar al umbral de activación de las llaves analógicas que controlan la fijación de la señal de control. En ese momento el convertidor digital-analógico pone en el nudo de control el último valor de tensión que leyó mientras el circuito funcionaba correctamente. La señal *resetPD* vuelve a valor bajo cada vez que comienza a recibirse un nuevo dato, dejando que el bucle evolucione libremente.

La figura 7.15 muestra un detalle de la figura anterior donde se puede observar el funcionamiento con más precisión. Se puede comprobar que el bucle está completamente estabilizado por que, según se ve, las señales UP y DOWN no difieren en nada, lo que indica que el reloj recuperado y el reloj de referencia desfasado 360° tienen sus flancos en las mismas posiciones temporales. Esto se puede comprobar directamente en la figura sin más que observar ambas señales. Se puede ver también un cierto rizado en la señal *resetPD* durante el periodo de transmisión de un dato. Éste se debe a que el circuito RC que controla esta señal evoluciona entre flanco y flanco de la señal de entrada. Sin embargo, como la constante de tiempo diseñada es muy superior a los tiempos típicos entre flancos, esta variación no es apreciable, ni mucho menos suficiente, como para producir un final de dato inesperado a mitad del periodo de recepción.



Figura 7.15: Detalle del funcionamiento del DLL

7.5. Corners de la tecnología

Todos los resultados anteriores se obtuvieron usando modelos típicos para los transistores que forman parte del diseño. Sin embargo, cuando se fabrique el circuito es de esperar que el comportamiento de los transistores no sea el que se modeló debido a las incertidumbres propias del proceso de fabricación. Por tanto, se podrán tener comportamientos inesperados sino se ha diseñado el sistema de una forma robusta. Para poder predecir posibles problemas y, de alguna forma, modelar esta incertidumbre, *STMicroelectronis* proporciona modelos de corners para todos sus transistores (incluyendo los de 2.5 y 3.3V). En estos modelos se recogen las situaciones extremas de la tecnología en cuanto a la fabricación del transistor, dando parámetros adecuados para contemplar todos los extremos posibles.

Normalmente, son cuatro las situaciones que se suelen modelar, que se han representado en la figura 7.16. Para cada transitor se modela una transistor típico, uno rápido y otro lento. Como hay transistores p y transistores n, esto da lugar a todas las combinaciones posibles que se nos puedan imaginar. Además, también suelen dar modelos para las capacidades y resistencias, tanto las integradas como las parásitas. De nuevo, aparecen los tres modelos anteriores. Como se ve las posibilidades crecen exponencialmente a medida que se incorporan nuevas consideraciones. No obstante, no siempre es necesario estudiar el circuito en todos los corners; basta con considerar las situaciones extremas donde nuesto sistema sea más sensible a los errores y crítico para el funcionamiento global.



Figura 7.16: Corners de la tecnología

De la misma forma, el diseño también se ha hecho suponiendo que las tensiones de alimentación utilizadas (de 1V y 2.5V) son constantes y estables en su valor ideal. Sin embargo, en la práctica estas tensiones vendrán dadas por circuitos que tienen errores y, además, estarán sometidas a un entorno hostil (acoplamientos electromagnéticos, ruido, acoplos con las señales LVDS,...). Por tanto, es muy posible que la tensión de alimentación pueda variar entorno a su valor nominal, cosa que también habrá que estudiar.

Por otro lado, se ha tomado como temperatura nominal de diseño 27°C, valor típico de laboratorio. No obstante, los circuitos operan en el mundo real donde la temperatura no está controlada y puede variar bastante con respecto a este valor. El problema es que muchos de los parámetros de los transistores (por ejemplo, la movilidad de los electrones y los huecos en el silicio) varían fuertemente con la temperatura. Por tanto, será también necesario ver qué ocurre y cómo se comporta el circuito cuando se varía ésta.

Para tener en cuenta las consideraciones anteriores, se hizo una análisis de corners sobre el circuito diseñado para comprobar que era lo suficientemente robusto como para operar en distintas circunstancias. La siguiente tabla nos muestra las situaciones que se tomaron como extremas en la tecnología y los parámetros que se variaron para hacer el análisis: modelos de los transistores, resistencias y condensadores, temperatura y tensión de alimentación. Los casos FF y SS son los extremos de la tecnología y los SF y FS se tomaron a partir de situaciones intermedias

Nombre del córner	Modelo	Resistencias	Capacidades	V_{dd}	Temp. (°C)
FF (Fast-Fast)	\mathbf{FF}	mínimas	mínimas	+5%	0
SS (Slow-Slow)	\mathbf{SS}	máximas	mínimas	-5%	80
FS (Fast-slow)	\mathbf{FS}	máxima	mínima	+5%	80
SF (Slow-Fast)	\mathbf{SF}	mínima	máxima	-5%	0

con el objetivo de cubrir una mayor cantidad de situaciones y comprobar si alguna combinación extraña de los parámetros generaba un comportamiento indeseado en nuestro circuito.

Teniendo en cuenta todos los criterios anteriores, se simuló el circuito en cada uno de los cuatro corners indicados anteriormente. El resultado de estas simulaciones fue satisfactorio, operando el enlace correctamente en todas ellas y recibiéndose los datos transmitidos sin error. El esquemático de simulación que se utilizó fue el mismo que para los resultados que se presentaron anteriormente, por lo que los resultados que se obtengan serán directamente comparables. En cada uno de los casos se analizó el comportamiento de todas las señales que se han comentado en los apartados anteriores para así detectar todos los pisbles errores que las simulaciones pongan de manifiesto. No obstante, se prestó especial atención a aquellas partes del enlace que son clave para el funcionamiento y más proclives a errores.

La primera gran duda que se nos plantea es si el bucle de realimentación que fija el valor de la tensión de control de las unidades de retrasos es capaz de converger a un valor de régimen permanente donde la sincronización sea posible. Al modificar los modelos de los transistores y la tempreratura, uno de los parámetros que sufre una mayor variabilidad es el retraso de las puertas lógicas. Así, los inversores tendrán, para un valor de la tensión de control dada, valores muy dispares según el modelo del transistor empleado. No obstante, esto ya fue previsto en el diseño y se diseñaron las unidades de retraso de forma que fueran lo más robustos posibles ante variaciones del proceso de fabricación y con un rango de programabilidad a través de la tensión de control lo más amplio posible.

La figura 7.17 muestra como los objetivos de diseño se han conseguido y el circuito opera correctamente en todas las condiciones simuladas. En ella se puede observar como se consigue que la tensión de control se estabilice a un valor concreto tras un cierto transitorio inicial cuya duración, en todos los casos, es similar y está perfectamente acotada. Se obtiene también un resultado esperado: la tensión de control en el córner FF es mayor que en el SS. Al ser en este último caso los transistores más lentos, los tiempos de retraso son menores y es más fácil alzanzar el valor nominal de retraso diseñado. Por otro lado, se ha representado junto a la señal de control, las formas de onda de los datos entregados por el receptor LVDS, así como el reloj recuperado a partir del CDR. Se comprueba como en todos los casos se obtiene resultados muy similares que en el caso típico: un reloj recuperado de 500MHz (una transición cada $T_b/2$ de la señal de datos de entrada) y transiciones lentas para el reloj recuperado y una secuencia de bits igual a la transmitida en los datos Manchester recuperados.



Figura 7.17: Estabilidad bucle control en los corners de la tecnología

Otro aspecto importante a tener en cuenta es la calidad de la señal LVDS que se genera y que se recibe a la entrada del receptor. El estándar establece que para todas las condiciones de operación (modelos de transistores, temperatura y tensión de alimentación) las señales diferenciales debe cumplir con las especificaciones tanto en el modo diferencial como en el modo común. La figura 7.18 nos muestra la señal de salida del driver LVDS en los corners FF y SS que se han considerado como los peores casos para la tecnología.



Figura 7.18: Salida del driver LVDS para los corners FF y SS

Dos son los parámetros que hay que controlar. Por un lado, la amplitud del modo diferencial debe ser suficiente como para que pueda ser intepretado por el receptor LVDS. En las dos situaciones representadas se puede comprobar como la amplitud del modo diferencial supera los 300 mV con holgura, lo cual es más que suficiente para la correcta operación del enlace. Por otro lado, se tiene que el modo común generado a la salida está entre 1.18-1.21 V, cerca del valor nominal que se pretendía de 1.2V. Además, las oscilaciones del modo común debida

a las conmutaciones y las pequeñas desadaptaciones de impedancias del enlace permanecen controladas en ambos casos, no suponiendo un problema especial para la operación del receptor. Aparte de la señal LVDS, se ha representado la salida del serializador, comprobándose en ambos casos que los datos se corresponden con los bits que se quieren transmitir y que la calidad de las señales (nivel alto, nivel bajo, tiempos de transición,...) es lo suficientemente elevada como para que el driver no tenga problemas en su operación.

Además de la señal generada por el driver, habrá que comprobar que la que llega al receptor tras propagarse en el canal es suficiente en todos los casos para que éste pueda trabajar en un régimen de funcionamiento correcto. La figura 7.19 muestra las formas de onda obtenidas.



Figura 7.19: Señal LVDS a la entrada del receptor para los corners FF y SS

El modo común que se recibe a la entrada es muy similar al que se transmite en ambos casos y, por supuesto, está dentro de los rangos de entrada que el fabricante da para los drivers LVDS que proporciona. El único inconveniente es el aumento de las oscilaciones del modo común debido a los efectos no ideales tantas veces comentados debidos a la propagación. No obtante, las variaciones relativas con respeto al valor medio no superan el 10% de la amplitud de la señal por lo que no debe preocupar en exceso. Por otro lado, en ambas gráficas se observa como el modo diferencial supera los 100mV que el estándar LVDS [26] especifica para el umbral de recepción, obteniéndose un valor superior a los 200mV en los dos casos. De la misma forma, se ha representado de forma adjunta los datos de salida que proporciona el driver que, como se puede comprobar, se corresponden de manera exacta con los transmitidos.

7.6. Comportamiento del receptor frente a jitter

El objetivo ahora es comprobar cómo afectan las variaciones en la frecuencia del reloj que genera los datos en la recepción de los mismos, es decir, la influencia del jitter. Cuando se presentó la arquitectura del receptor, se razonó que, por construcción, era muy robusta frente al jitter pues generaba los instantes de captura de los dats en función de los flancos de la señal recibida, no necesitando ninguna referencia temporal externa. El objetivo ahora es estudiar esta inmunidad mediante simulaciones y añadir una serie de efectos no ideales al canal para comprobar que la interfaz diseñada es lo suficientemente robusta como para operar en condiciones reales de funcionamiento.

Para realizar estas simulaciones, el primer objetivo era el de generar un reloj con jitter para generar unos datos Manchester que aplicar directamente al receptor. Para ello se creó un modelo AHDL que, mediante la generación de números aleatorios, fuera capaz de crear una señal cuadrada cuyo periodo fuera variable con el tiempo. A continuación, dicha señal de reloj, junto con un cierto patrón de bits, serán aplicados al codificador Manchester cuya salida atacará directamente al deserializador diseñado. El código AHDL utilizado se muestra a continuación:

```
module clk_jitter (clk_out,ref)(sigma,freq)
node [V,I] clk_out, ref;
parameter real sigma=0.016, freq=1e9;
{
    real tpo,clk,N;
    initial{
        clk=0.0;
        tpo=0.5/freq;}
    analog{
        V(clk_out,ref) <- $transition(clk*1,tpo-20p,10p,10p);
        if (V(clk_out,ref)>0.5)
            clk=0;
        else
            clk=1;
        N=$random();
    }
}
```

```
N=N+$random();
N=N+$random();
N=N+$random();
N=N+$random();
N=N+$random();
N=N+$random();
N=N+$random();
N=N+$random();
N=N+$random();
tpo=(N-5)*12/sqrt(10)*0.5*sigma/freq+0.5/freq;}
```

Con el código anterior se generó el reloj mostrado en la figura 7.20, junto con los datos Manchester también representados. En la figura se puede observar una pequeña variación de la frecuencia del reloj que se confirmó numéricamente calculando los instantes de tiempo de cruce por 0.5V (mitad del rango de señal).



Figura 7.20: Datos con jitter generados

Los datos anteriormente mostrados fueron aplicados a la entrada del receptor para estudiar su comportamiento. Como resultado de dicha simulación se obtuvieron las formas de onda representadas en la figura 7.21, donde se puede observar el proceso de generación del reloj recuperado y cómo éste se adapta al jitter de la señal de entrada. Esta adaptación se hace a través de la adecuación del flanco de reloj a los instantes donde se produce en la señal de entrada.

}

Así, se observa que el reloj recuperado tampoco sigue un patrón periódico, sino que reproduce el jitter de la entrada. También se puede ver como la señal de control se adapta a estos cambios modificando el retraso cuando es necesario a causa de la modificación de la frecuencia de reloj.



Figura 7.21: Proceso de recuperación del reloj con jitter

Una cosa interesante que han mostrado las simulaciones con jitter en la señal de reloj es la capacidad de adaptación del bucle de control de los retrasos a través de la adaptación de la señal de control a la frecuencia de reloj capturada. En ese sentido se muestra la figura 7.22 donde se muestra este proceso de adaptación a lo largo de la recepción de una serie de datos, así como un detalle para que se pueda observar claramente cómo se lleva a cabo.



Figura 7.22: Proceso de convergencia de la señal de control

La primera figura trata de ilustrar como, durante el proceso de recepción de cada dato, la señal de control sufre modificaciones en función del jitter que se esté recibiendo en cada bit que se desea interpretar. Para la simulación se ha usado un ancho de palabra bastante largo para que este proceso se vea más claro. Esta circunstancia no se observaba en las simulaciones anteriores porque el patrón de frecuencias de llegada era constante y, cuando se fijaba un valor de la tensión de control, éste era válido para todo el proceso de recepción. Estas simulaciones demuestran la versatilidad del bucle de realimentación que es capaz de adaptarse dinámicamente a los cambios, permitiendo a la vez muestrear la línea de datos en los instantes correctos asegurando la sincronización en todo momento. En el detalle de la figura 7.22 se puede observar más claramente la variación instantánea cada vez que se recibe un bit de la señal serie transmitida.