

INSTITUTO TECNOLÓGICO DE COSTA RICA

ESCUELA DE INGENIERÍA ELECTRÓNICA



Diseño de recuperador de reloj y datos para un canal serial de alta velocidad en un entorno de simulación de señal mixta

Informe de Proyecto de Graduación para optar por el título de Ingeniero en Electrónica con el grado académico de Licenciatura

Allan Gerardo Carmona Cruz

Cartago, Costa Rica

6 de junio del 2018

Instituto Tecnológico de Costa Rica
Escuela de Ingeniería en Electrónica
Proyecto de Graduación
Tribunal Evaluador
Acta de aprobación

Proyecto de graduación defendido ante el presente Tribunal Evaluador como requisito para optar por el título de Ingeniero en Electrónica con el grado académico de Licenciatura, del Instituto Tecnológico de Costa Rica.

Estudiante: *Allan Gerardo Carmona Cruz*

Nombre del proyecto: *Diseño de recuperador de reloj y datos para un canal serial de alta velocidad en un entorno de simulación de señal mixta*

Miembros del Tribunal



Lic. Daniel Kohkemper Granados
Profesor Lector



M.Sc. Julio María Stradi Granados
Profesor Lector



M.Sc. Anibal Coto Cortés
Profesor Asesor

Los miembros de este Tribunal dan fe de que el presente trabajo de graduación ha sido aprobado y cumple con las normas establecidas por la Escuela de Ingeniería Electrónica.

Cartago, 6 de junio de 2018

Declaratoria de autenticidad

Declaro que el presente Proyecto de Graduación, Diseño de recuperador de reloj y datos para un canal serial de alta velocidad en un entorno de simulación de señal mixta ha sido realizado enteramente por mi persona, utilizando y aplicando literatura referente al tema e introduciendo conocimientos propios.

En los casos en que he utilizado bibliografía he procedido a indicar las fuentes mediante las respectivas citas bibliográficas. En consecuencia, asumo la responsabilidad total por el trabajo realizado y por el contenido del correspondiente informe final.

Allan Gerardo Carmona Cruz

Allan Gerardo Carmona Cruz

Cartago, 6 de junio de 2018

Cédula: 1-1568-0754

Resumen

La Escuela de Ingeniería en Electrónica del ITCR (Instituto Tecnológico de Costa Rica) está desarrollando un proyecto para la simulación de un canal serial de alta velocidad a nivel de comportamiento en un entorno de simulación de señal mixta. Una sección importante del mismo es el desarrollo de un recuperador de reloj y datos (CDR) entre el transmisor y el receptor que permita la sincronización en la comunicación. En este trabajo se verá el proceso de selección y diseño del CDR. Además se verificó el correcto funcionamiento por medio de simulaciones.

Palabras clave: CDR, Jitter, Offset de frecuencia, Detector de frecuencia y fase, Interpolador de fase, Diagrama de ojo.

Abstract

The ITCR is developing a project for the simulation of a high-speed serial link at behavioral level in a mixed signal simulation environment. An important section of it is the development of a clock and data recovery (CDR) between the transmitter and the receiver that allows synchronization in communication. In this work you will see the process of selection and design of the CDR. In addition, correct operation was verified through simulations.

Keywords: CDR, Jitter, Frequency offset, Frequency and phase detector, Phase interpolator, Eye diagram, Embedded clock.

A mi amada familia

Agradecimientos

Dedico este proyecto de graduación primeramente a mis padres que me inculcaron desde pequeño el deseo de estar en la universidad y obtener un grado académico, con el que pueda aportar a la sociedad. Además, les agradezco todos los sacrificios que hicieron para ayudarme a concretar mis estudios y que me han convertido en la persona que soy ahora.

Agradezco toda la ayuda y asesoría de los profesores que he tenido a lo largo de mi carrera, los cuales han aportado a mi formación profesional. Un agradecimiento especial al Dr. Renato Rímolo Donadio que depositó su confianza en mí, y creyó en mis capacidades para desarrollar el proyecto de graduación, además de guiarme en el desarrollo del mismo.

Por último a las personas que conocí en el tiempo que realicé mis estudios y que siempre estuvieron conmigo brindándome una ayuda desinteresada.

Y a todas las personas que de una u otra manera han sido parte de mi vida.

Allan Gerardo Carmona Cruz

Cartago, 18 de junio de 2018

ÍNDICE GENERAL

1. Introducción	1
1.1. Objetivos	2
1.1.1. Objetivo General	2
1.1.2. Objetivos Específicos	2
1.2. Organización del trabajo	2
2. Interfaces seriales de alta velocidad	3
2.1. Enlaces eléctricos de alta velocidad	4
2.2. Canal eléctrico	5
2.2.1. Atenuación	5
2.2.2. Dispersión	6
2.2.3. Interferencia intersímbolo (ISI)	6
2.2.4. Diafonía (<i>Crosstalk</i>)	6
2.2.5. Reflexiones	7
2.3. Ecualización	7
2.4. Temporización	9
2.5. Arquitecturas de interfaces de entrada y salida	9
2.5.1. Reloj común	11
2.5.2. Reloj hacia delante	11
2.5.3. Reloj embebido	12
2.6. Problemas en la temporización	12
2.6.1. Skew	12

2.6.2.	Offset de frecuencia	12
2.6.3.	Jitter	14
2.7.	Recuperador de reloj y Datos	16
3.	Arquitectura del CDR	17
3.1.	CDR basado en PLL	17
3.2.	CDR basado en DLL	20
3.3.	Combinación entre PLL y DLL	20
3.4.	CDR con sobre muestreo	20
3.5.	CDR con interpolador de fase	21
3.6.	Comparación entre arquitecturas	24
3.7.	Requerimientos del CDR del proyecto	24
3.8.	Requerimientos y limitaciones del entorno de simulación	25
3.9.	Selección de arquitectura y modificaciones	25
4.	Modelo de comportamiento del recuperador de reloj y datos	27
4.1.	Diseño y evaluación de los componentes de la arquitectura	27
4.1.1.	Detector de frecuencia y fase (PFD)	27
4.1.2.	Línea de retardos	32
4.1.3.	Multiplexor (Mux)	34
4.1.4.	Máquina de estados tipo Mealy (FSM)	34
4.1.5.	Activación de FSM	38
4.2.	Limitaciones de los componentes del CDR	39
5.	Evaluación y análisis del CDR	41

5.1. Prueba 1	41
5.2. Prueba 2	42
5.3. Prueba 3	44
5.4. Prueba 4	49
6. Conclusiones y recomendaciones	55
6.1. Conclusiones	55
6.2. Recomendaciones	55
7. Bibliografía	56

ÍNDICE DE FIGURAS

1.1. Canal serial de alta velocidad del proyecto.	1
2.1. Proyección de los anchos de banda por interfaz de entrada y salida [1]. . . .	4
2.2. Sistema de enlace eléctrico de alta velocidad [2].	5
2.3. Respuesta en frecuencia de diversos canales [2].	5
2.5. Interferencia entre símbolos [3].	6
2.4. Dispersión de una dato debida al canal [3].	6
2.6. Interferencia intersímbolo de cuatro datos consecutivos [4].	7
2.7. Diafonía entre dos conductores [5].	7
2.8. Reflexión en una línea de transmisión [5].	8
2.9. Ecualizadores en una canal serial de alta velocidad [2].	8
2.10. Punto de mayor estabilidad en un diagrama de ojo [6].	9
2.11. Arquitectura de reloj común [2].	11
2.12. Arquitectura reloj hacia delante [2].	12
2.13. Arquitectura de reloj embebido [2].	13
2.14. Skew entre dos señales de reloj [7].	13
2.15. Offset de frecuencia entre relojes [8].	14
2.16. Jitter presente en una señal [9].	14
2.17. Jitter aleatorio y determinista [9].	14
2.18. Modelo de doble dirac para el jitter total [5].	15
2.19. Esquema general de un recuperador de reloj y datos [10].	16
3.1. CDR basado en PLL sin referencia de reloj [10].	18
3.2. CDR basado en PLL con referencia [10].	19

3.3. CDR basado en PLL digital [10].	19
3.4. CDR basado en DLL con referencia [10].	20
3.5. Arquitectura combinada [10].	20
3.6. Arquitectura con sobremuestreo [10].	21
3.7. Interpolador de fase [11].	22
3.8. Implementación de arquitectura con interpolador de fase [2].	22
3.9. Variación de interpolador de fase [10].	23
3.10. Arquitectura con rotador de fase [10].	23
3.11. Arquitectura final.	26
4.1. Respuesta de un PFD a una entrada con diferentes frecuencias y fases [12].	27
4.2. Máquina de estados de un PFD [12].	28
4.3. Implementación de PFD [12].	28
4.4. Símbolo en ADS del PFD.	29
4.5. Circuito de prueba del PFD.	29
4.6. Salida del PFD con frecuencia y fase iguales.	30
4.7. Salida del PFD con frecuencia igual y fase diferente.	30
4.8. Salida del PFD con frecuencia igual y fase diferente 2.	30
4.9. Salida del PFD con frecuencia y fase diferente.	31
4.10. Salida del PFD con frecuencia y fase diferente 2.	31
4.11. Salida del PFD con frecuencia y fase diferente 3.	31
4.12. Línea de retardos.	33
4.13. Fases generadas en las líneas de retardo.	33
4.14. Línea de retardo de 180°.	34
4.15. Máquina de estados principal.	36

4.16. Circuito de activación para la FSM.	38
4.17. Detector de fase tipo Hogge [13].	39
4.18. Detector de fase tipo Alexander [13].	40
5.1. Señales en fase.	42
5.2. Presencia de skew 1.	42
5.3. Presencia de skew 2.	42
5.4. Presencia de skew 3.	43
5.5. Presencia de skew 4.	43
5.6. Presencia de skew 5.	43
5.7. Diferencia entre las señales.	44
5.8. Offset de frecuencia 1.	45
5.9. Offset de frecuencia 2.	46
5.10. Offset de frecuencia 3.	47
5.11. Offset de frecuencia 4.	47
5.12. Offset de frecuencia 5.	48
5.13. Offset de frecuencia 6.	48
5.14. Uso de PRBS 1.	49
5.15. Uso de PRBS 2.	49
5.16. Uso de PRBS 3.	50
5.17. Uso de PRBS 4.	50
5.18. Uso de PRBS 5.	51
5.19. Uso de PRBS 6.	51
5.20. Uso de PRBS 7.	51
5.21. Uso de PRBS 8.	52

5.22. Diagrama de ojo para offset de frecuencia de 2 kHz.	52
5.23. Diagrama de ojo para offset de frecuencia de 2 MHz.	53
5.24. Diagrama de ojo para offset de frecuencia de 20MHz.	53
5.25. Diagrama de ojo para offset de frecuencia de 2 kHz.	54

ÍNDICE DE TABLAS

2.1. Aplicaciones de enlaces de alta velocidad [14].	3
2.2. Grados de sincronía [6].	10
4.1. Tabla resumen de la entrada a FSM principal.	35
4.2. Tabla de verdad para la activación de la FSM	37
4.3. Tabla de verdad para la activación de la FSM	38

Lista de Abreviaciones

CDR	Clock and Data Recovery
CTLE	Continuous Time Linear Equalization
DFE	Decision Feedback Equalization
FSM	Finite State Machine
PLL	Phase Locked Loop
FIR	Finite Impulse Response
SoC	System on Chip
SoB	System on Board
RJ	Random Jitter
SJ	Sinusoidal Jitter
DJ	Deterministic Jitter
PJ	Periodic Jitter
PVT	Process Voltage Temperature
PI	Phase Interpolator
DLL	Delay Locked Loop
CP	Charge Pump
PFD	Phase and Frequency Detector
PD	Phase Detector
FD	Frequency Detector
VCO	Voltage Controlled Oscillator
ISI	Intersymbol Interference
VCDL	Voltage Control Delay Line
PRBS	Pseudo Random Bit Sequence

1 Introducción

La Escuela de Ingeniería en Electrónica del Instituto Tecnológico de Costa Rica (TEC), está desarrollando un proyecto para la simulación de un canal serial de alta velocidad a nivel de comportamiento. Las simulaciones generadas son de gran importancia ya que pueden validar el correcto funcionamiento de las interfaces de comunicación de alta velocidad que se diseñen en otros proyectos de la Escuela. Las partes que conforman este sistema se observan en la figura 1.1.

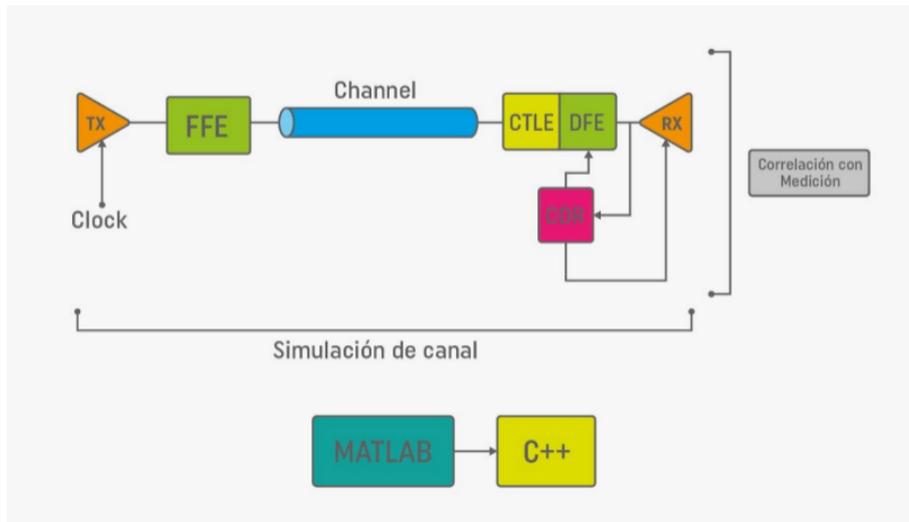


Figura 1.1: Canal serial de alta velocidad del proyecto.

En la actualidad, las comunicaciones cumplen un rol muy importante en la sociedad, por lo que se requiere una mayor velocidad de transmisión de los datos. Sin embargo, al aumentar la velocidad y la distancia, el uso de buses convencionales paralelos se vuelve poco práctico en términos de potencia, área y confiabilidad [15], y se cambió a canales seriales de comunicación. Al llegar a velocidades de transferencia en el orden de los Gb/s, el ancho de banda del canal se vuelve limitado debido a la dependencia entre la frecuencia y las pérdidas de señal del canal, por lo que se necesitaron nuevas técnicas para cumplir la velocidad de datos requerida, como el uso de ecualizadores lineales y no lineales en el transmisor y receptor [16].

Las interfaces seriales de alta velocidad utilizan dominios de reloj diferentes, de modo que no se encuentran en sincronía. Por otro lado, se ven afectadas por variaciones temporales deterministas y aleatorias por lo que es necesario un sistema de sincronización, para el entendimiento de los datos en el receptor. Para que una señal sea correctamente interpretada se debe muestrear el dato en el receptor donde presenta la mayor estabilidad como se detallará posteriormente.

El proyecto en cuestión tiene énfasis en el desarrollo de un CDR, como sincronizador entre el transmisor y el receptor en un entorno de simulación de señal mixta a nivel de comportamiento.

Posteriormente se validó el funcionamiento del CDR, de manera que se asegure muestrear en el lugar indicado ante las siguientes variaciones temporales:

1. Sesgo de reloj: Puede ocurrir cuando hay una diferencia en la distancia de la línea de transmisión de la fuente de reloj al bloque de muestreo del receptor por lo que la señal presenta una variación de fase. Una de las razones de esta alteración es la variabilidad en el proceso de manufactura del dispositivo.
2. *Offset* de frecuencia: Las fuentes de reloj como los osciladores por su naturaleza no se pueden fabricar exactamente iguales, por lo que siempre presentan una diferencia en frecuencia.
3. *Jitter*: Es la variación temporal de una señal del lugar ideal al esperado por su periodicidad.

El diseño debe mantener una tasa de transferencia de datos en el orden de los Giga bits por segundo, diferentes dominios de reloj, presencia de *jitter*, desfase y pérdidas en la señal. Además tiene que cumplir con los requerimientos del simulador, como la existencia de modelos de comportamiento para cada una de sus partes.

Es así como, para el diseño del CDR se investigó una arquitectura que pueda ser implementable y que cumpla con los aspectos y requerimientos anteriormente descritos.

1.1. *Objetivos*

1.1.1. *Objetivo General*

Diseñar un modelo de recuperador de reloj y datos para la sincronización en frecuencia y fase de un canal serial de alta velocidad.

1.1.2. *Objetivos Específicos*

1. Definir la arquitectura del recuperador de reloj y datos con base en las características de la comunicación serial de alta velocidad y con reloj embebido.
2. Implementar la etapa de un recuperador de reloj y datos en el entorno de simulación seleccionado.
3. Validar el modelo de simulación en términos de capacidad de compensar el sesgo de reloj y tolerancia a jitter.

1.2. *Organización del trabajo*

1. En el capítulo 2 se fundamenta teóricamente el funcionamiento y constitución de un canal serial de alta velocidad, además, la importancia y fundamentos de la sincronización de estos sistemas. Conceptos necesarios para comprender el trabajo, extraído de fuentes confiables y debidamente citadas.
2. En el capítulo 3 se define la arquitectura a utilizar para el CDR basándose en los requerimientos del proyecto y del entorno de simulación.
3. En el capítulo 4 se diseña e implementa el CDR definido en el capítulo 3, además se seleccionan los modelos de comportamiento de cada una de las partes que constituyen el CDR.
4. En el capítulo 5 se detallan y analizan las pruebas de funcionamiento del CDR.

2 Interfaces seriales de alta velocidad

En esta sección se ven los conceptos necesarios para comprender el funcionamiento general de los canales seriales de alta velocidad y su constitución. Además, los efectos negativos que degradan la integridad de la señal, la importancia de la sincronización en comunicaciones, los problemas y métodos para solucionar la temporización.

El interés en el diseño de interfaces seriales de alta velocidad se debe al incremento de computación, que se integra en secciones de silicio cada vez más pequeñas, lo que ha exigido un mayor ancho de banda en las interfaces de comunicación de los procesadores. En la tabla 2.1 se detallan protocolos de comunicación donde se utilizan enlaces seriales de alta velocidad.

Tabla 2.1: Aplicaciones de enlaces de alta velocidad [14].

Aplicaciones de canales seriales de alta velocidad			
Protocolo	Versión	Velocidad (Gbps)	
PCIe	3.0	7.8	
	4.0	16.0	
Infiniband	-	25.0	
Hyper-transport	2.0	6.0	
	3.0	10.0	
	3.1	13.0	
SATA Express	3.0	7.8	
USB	3.0	5.0	
	3.1	10.0	
	3.2	20.0	
HDMI	2.1	48.0	
100G Ethernet	-	100.0	
ThunderBolt	3.0	40.0	
DisplayPort	1.3	32.4	
HDMI	2.1	48.0	
PC4-25600 SDRAM	DDR4	-	204.8

Se proyecta que las interfaces de entrada y salida de los procesadores tenderán a incrementar su velocidad de transferencia pero no así el número de entradas y salidas debido a las limitaciones del empaquetado [2], como se observa en la figura 2.1, debido a lo cual deben tener la capacidad de manejar los anchos de banda requeridos.

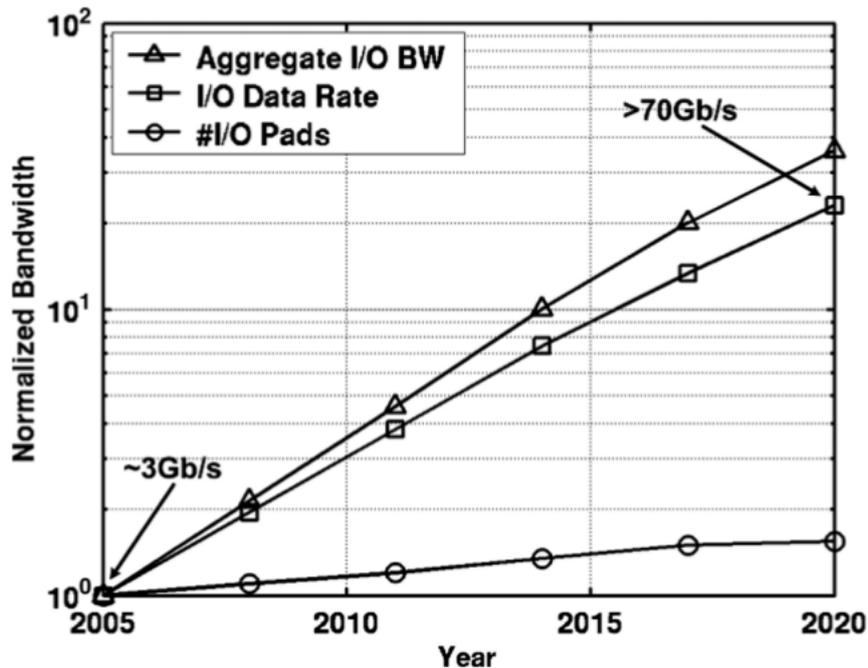


Figura 2.1: Proyección de los anchos de banda por interfaz de entrada y salida [1].

2.1. Enlaces eléctricos de alta velocidad

En la figura 2.2 se observa los componentes generales de un canal serial de alta velocidad los cuales son [2]:

- **Serializador y deserializador:** la limitada cantidad de entradas y salidas de un empaquetado obliga a serializar los datos de salida, además de las ventajas de utilizar un canal serial en términos de área, potencia y eficiencia [15].
- **Transmisor:** se encarga de poner la señal de salida en el canal con un apropiado valor de oscilación y de impedancia de salida de manera que no genere reflexiones indeseadas. Existen dos tipos de drivers utilizados que son en modo de corriente y modo de voltaje. El tipo de *driver* utilizado depende de la aplicación en específico. Otro factor importante es el uso de señal *single ended* o diferencial ya que permite el rechazo de ruido común [2].
- **Canal:** limita en gran medida el ancho de banda del sistema por sus pérdidas.
- **Receptor:** Se encarga de recibir la señal del canal, compararla con un umbral de tensión eléctrica y amplificarla a un valor *CMOS*.
- **Sistema de temporizado:** genera las bases de tiempo del sistema y sincroniza los datos en el receptor de manera que sean correctamente entendidos, lo que se muestra en la parte inferior de la figura 2.2. La base de tiempo en el transmisor se genera con un PLL (*phase locked loop*) y en el receptor se utiliza un recuperador de temporizado. Para esta sección existen diversos tipos de arquitecturas que serán explicadas con detalle en el capítulo 3.

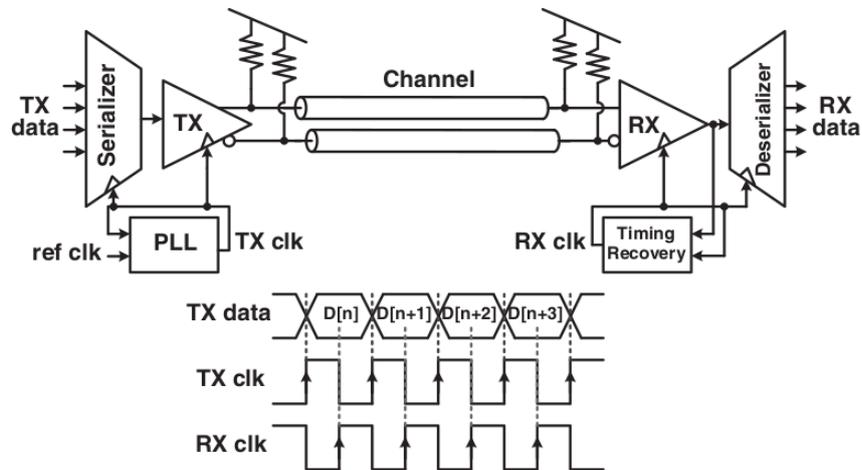


Figura 2.2: Sistema de enlace eléctrico de alta velocidad [2].

2.2. Canal eléctrico

Al compararse la longitud de onda de una señal con el tamaño de un canal, se modela este último como una línea de transmisión. Por lo tanto, el ancho de banda se ve limitado por las pérdidas de la línea, reflexiones por discontinuidades, interferencia y diafonía de canales paralelos.

2.2.1. Atenuación

La atenuación en alta frecuencia es debida a dos efectos, el primero es el efecto piel que aumenta la resistencia del cableado disminuyendo la profundidad por la que pasa la corriente en el conductor [6]. Segundo, las pérdidas del dieléctrico que son causadas por la resistencia de los dipolos magnéticos de mantenerse alineados a un campo eléctrico variante en el tiempo [5]. Ante los efectos mencionados se espera tener una mayor atenuación conforme aumenta la frecuencia, como se observa en la figura 2.3.

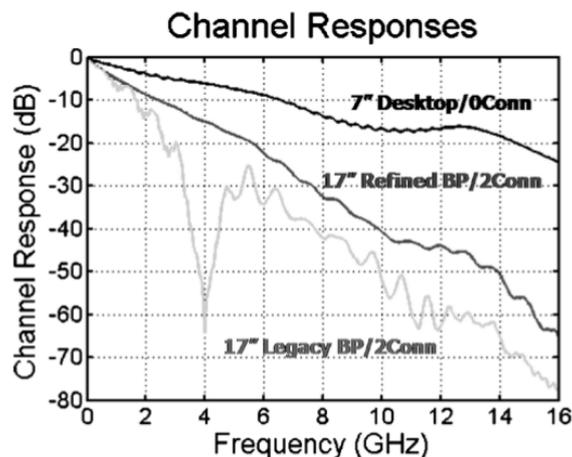


Figura 2.3: Respuesta en frecuencia de diversos canales [2].



Figura 2.5: Interferencia entre símbolos [3].

2.2.2. Dispersión

El canal tiende a filtrar las altas frecuencias de las señales por lo que modifica el contenido espectral de las señales, de manera que se genera una dispersión del dato, como se muestra en la figura 2.4. En el peor caso se puede deformar por completo la señal y no llegar al receptor.



Figura 2.4: Dispersión de una dato debida al canal [3].

2.2.3. Interferencia intersímbolo (ISI)

Otro efecto negativo en las comunicaciones de alta velocidad es dependiente de los datos y se presenta cuando se toma información errónea de una secuencia de bits como se muestra en la figura 2.5 donde la secuencia de *bits* "101" se ve como "111" en el receptor debido a la dispersión de los datos.

Para comprender su efecto se estudia la respuesta al impulso del canal, como se observa en la figura 2.6, donde se aprecia que el dato mantiene un valor residual por un tiempo corto, lo que podría afectar a los datos posteriores y anteriores. El *bit* de interés es el cursor y los causantes de la interferencia son los pre-cursores y post-cursores.

2.2.4. Diafonía (Crosstalk)

Cuando estructuras de transmisión vecinas se encuentran muy próximas, los campos eléctricos y magnéticos interactúan con los conductores adyacentes, por lo que se acopla la energía generando diafonía, como se muestra en la figura 2.7. Si las interconexiones son más pequeñas y rápidas, su efecto es mayor.

Al aumentar la diafonía se genera una degradación en la integridad de la señal, un aumento en el retardo y un acople del ruido entre los conductores [5].

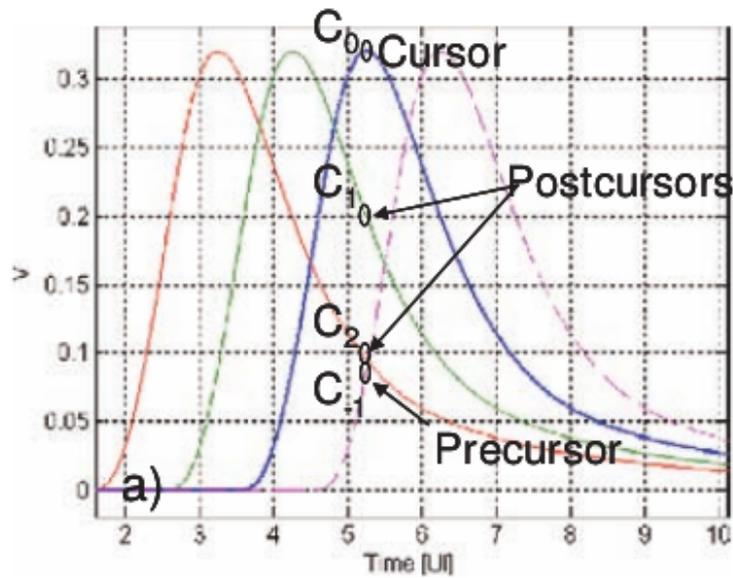


Figura 2.6: Interferencia intersímbolo de cuatro datos consecutivos [4].

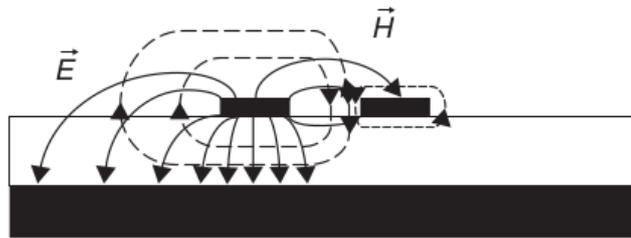


Figura 2.7: Diafonía entre dos conductores [5].

2.2.5. Reflexiones

Cuando una señal viaja por un canal y cambia la impedancia del medio o se encuentran discontinuidades en la línea, suceden dos efectos como se observa en la figura 2.8:

- Una porción de la señal se refleja hacia la fuente.
- Una porción de la señal es transmitida hacia la carga.

El resultado de la reflexión causa una atenuación en la señal y un retardo de la señal. Una de las mayores fuentes de discontinuidad son los *stubs* y desacoples en la línea [2].

2.3. Ecuación

Con el fin de extender la velocidad de envío máxima que permite el canal, se utilizan técnicas de ecuación para eliminar la interferencia intersímbolo. El ecuador es implementado como

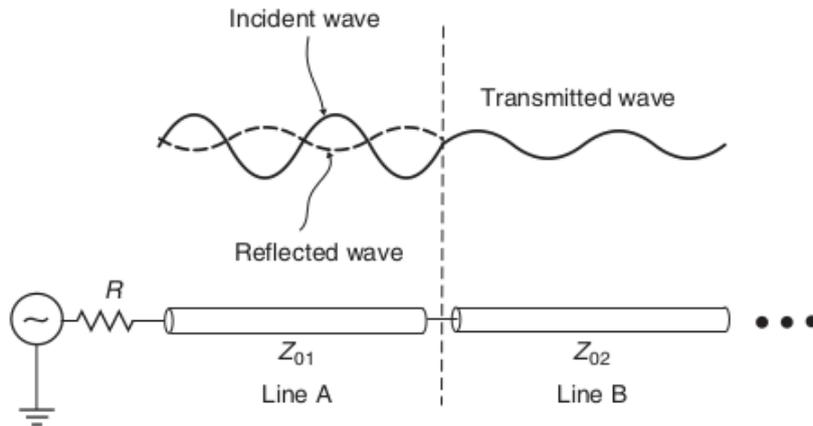


Figura 2.8: Reflexión en una línea de transmisión [5].

un filtro lineal (discreto o continuo) que busca aplanar la respuesta en frecuencia del canal o en el caso no lineal, busca eliminar directamente la interferencia intersímbolo según la secuencia de datos. No existe una única forma de hacerlo, ya que dependiendo de las especificaciones del sistema hay diversas combinaciones de ecualizadores en el transmisor y receptor [2] como se observa en la figura 2.9.

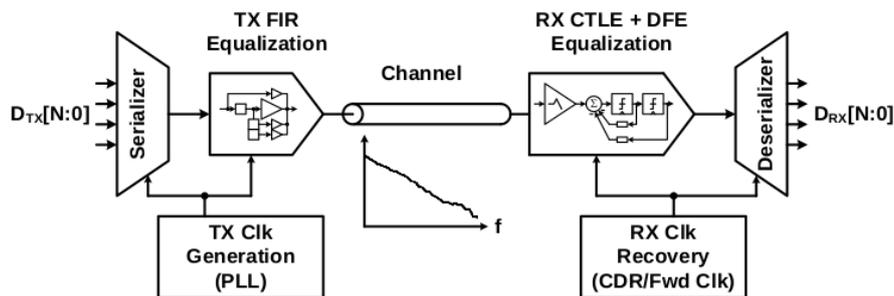


Figura 2.9: Ecualizadores en una canal serial de alta velocidad [2].

La ecualización en el transmisor se implementa con filtros FIR (*finite impulse response*), también conocidos como filtros de pre-énfasis y su único inconveniente es que amplifican el ruido y la diafonía. Posteriormente la ecualización en el receptor se implementa con ecualizadores lineales de tiempo continuo (CTLE) y funciona como un filtro paso altas pasivo o activo que compensa la respuesta del canal.

Por último en aplicaciones de alta velocidad se utilizan los filtros de decisión por realimentación (DFE), los cuales eliminan la ISI de la señal proveniente del canal, controlando el valor de los *taps* con respecto a la secuencia de datos.

2.4. Temporización

La temporización en un canal serial de alta velocidad presenta un reto en el diseño del sistema ya que se tiene que lograr una alta sincronización entre el transmisor y receptor de manera que los datos puedan ser muestreados en el punto de mayor estabilidad, el cual coincide con la mayor apertura vertical en el diagrama de ojo, como se puede ver en la figura 2.10.

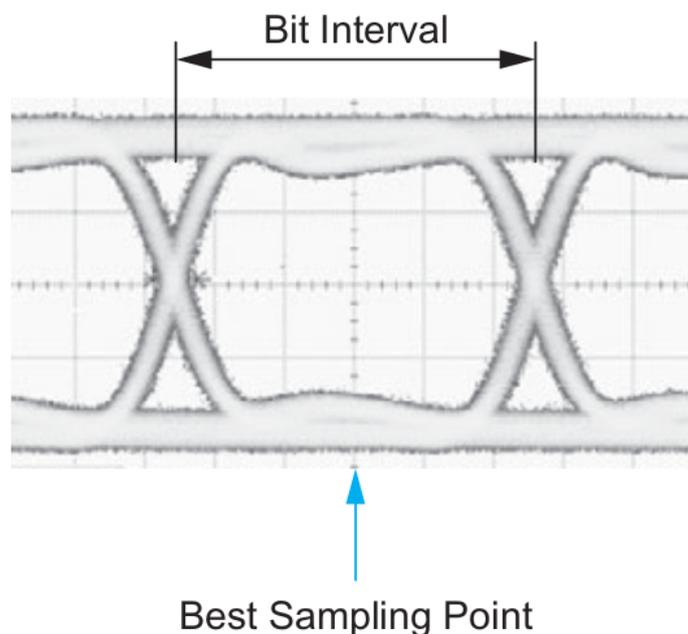


Figura 2.10: Punto de mayor estabilidad en un diagrama de ojo [6].

Al diseñar la temporización de un sistema de comunicación es relevante conocer el grado de sincronización entre los datos y el reloj del receptor basados en Δf y $\Delta \phi$ como se observa en la tabla 2.2.

2.5. Arquitecturas de interfaces de entrada y salida

Para las interfaces de entrada y salida de un procesador se utilizan tres tipos de arquitecturas para sincronizar la señal con el receptor, las cuales son:

1. Reloj Común.
2. Reloj hacia delante (*Forward Clock*).
3. Reloj embebido.

Normalmente todas utilizan un PLL para la señal de reloj, pero se diferencian en la forma que se recupera la señal de reloj en el receptor. Además, cada una se utiliza dependiendo de la aplicación y ancho de banda requerido.

Tabla 2.2: Grados de sincronía [6].

Aplicaciones de canales seriales de alta velocidad				
Clasificación	Periódico	$\Delta\phi$	Δf	Descripción
Síncrono	Sí	0	0	La señal tiene una frecuencia y fase igual al reloj. Se puede tomar la muestra sin problemas. Ejemplo: <i>flip flop</i> a <i>flip flop</i> dentro del chip.
Mesocrónico	Sí	Constante	0	La señal tiene la misma frecuencia pero se encuentra fuera de fase con el reloj y fase igual al reloj. Se necesita un retardo constante para tomar la muestra. Ejemplo: chip a chip con diferentes largos de línea.
Plesiocrónico	Sí	Varía despacio	Pequeño	La señal tiene una frecuencia muy cercana y diferencia de fase que aumenta despacio en el tiempo. Se necesita un retardo variable para tomar la muestra. Ejemplo: tarjeta a tarjeta con diferentes osciladores.
Periódico	Sí	Varía rápido	Grande	La señal es periódica a una frecuencia arbitraria. Se puede predecir la periodicidad para generar un retardo adecuado para el muestreo. Ejemplo: tarjeta a tarjeta con diferentes frecuencias de reloj.
Asíncrono	No	No conocido	No conocido	La señal cambia en tiempos arbitrarios. Se necesita un sincronizador completo. Ejemplo: entrada de un pulsador.

2.5.1. Reloj común

Arquitectura muy utilizada para frecuencias menores a los 100 Mbps donde solo existe una única fuente de reloj, como se observa en la figura 2.11, la cual llega al transmisor y receptor con alta sincronía, lo que se logra con el diseño de interconexiones del mismo largo [2]. A pesar de su alta sincronía presenta los siguientes problemas:

- Dificultad para controlar el sesgo de reloj y la propagación de retardos.
- Necesita tener un alto control de los retardos ya que definen el periodo de reloj mínimo.
- Sensitivo a variaciones de fabricación en SoC (*System on Chip*) y SoB (*System on Board*).
- Muy utilizado para comunicaciones SoC debido a su límite de ancho de banda.

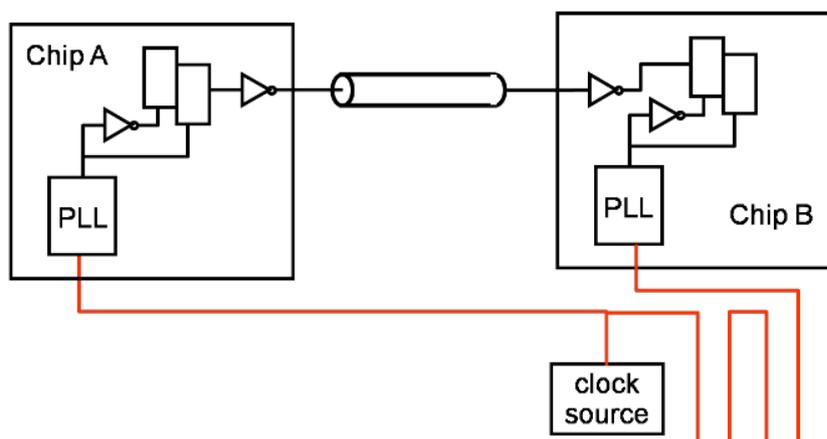


Figura 2.11: Arquitectura de reloj común [2].

2.5.2. Reloj hacia delante

En esta arquitectura el transmisor envía la señal de reloj en paralelo con los datos por lo que consume un puerto de entrada y salida, como se observa en la figura 2.12. Es utilizado para sincronías mesocrónicas, por lo que necesita un bloque de corrección de fase conocido como *deskew*. Este sistema es utilizado para conexiones entre procesador y memoria [2].

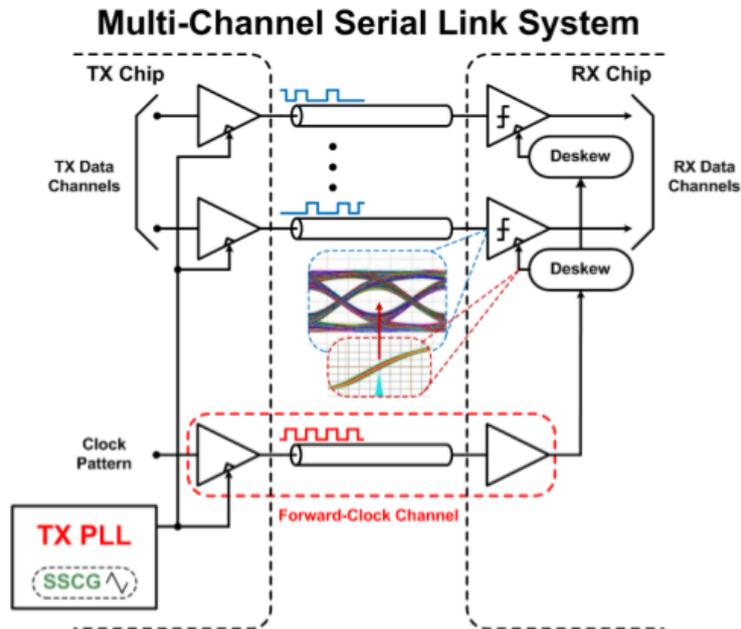


Figura 2.12: Arquitectura reloj hacia delante [2].

2.5.3. Reloj embebido

La última arquitectura posee una sincronía plesiocrónica por lo que se ocupa un recuperador de reloj y datos en el receptor como se observa en la figura 2.13. Dentro de las dificultades de su diseño se encuentra un aumento de área y la dificultad de implementar un CDR.

2.6. Problemas en la temporización

Dentro de las problemáticas para el diseño de la temporización se encuentran el *skew*, *jitter* y *offset* de frecuencia.

2.6.1. Skew

La diferencia de tiempo en la llegada del reloj entre dos registros adyacentes, como se observa en la figura 2.14 [7].

2.6.2. Offset de frecuencia

El *offset* de frecuencia que existe entre el transmisor y receptor debido a la naturaleza de los osciladores que no poseen exactamente la misma frecuencia, como se observa en la figura 2.15. La diferencia en frecuencia puede ser constante u oscilatoria.

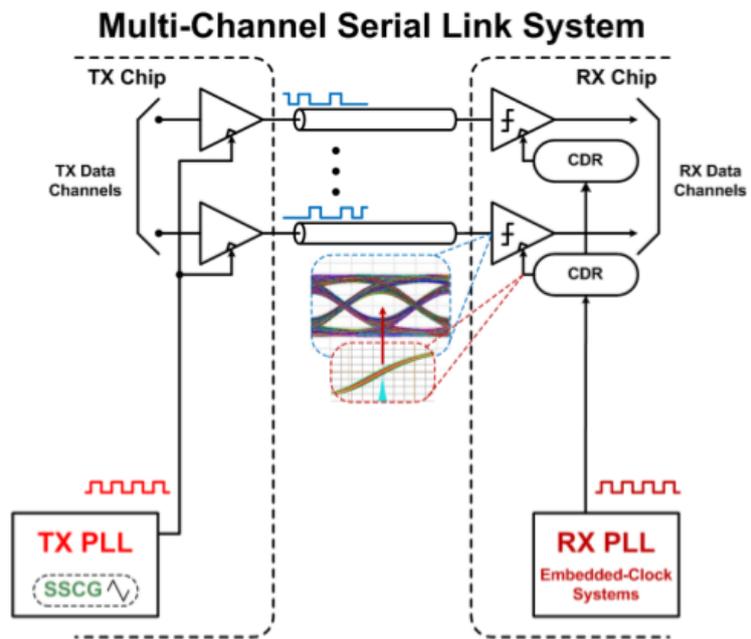


Figura 2.13: Arquitectura de reloj embebido [2].

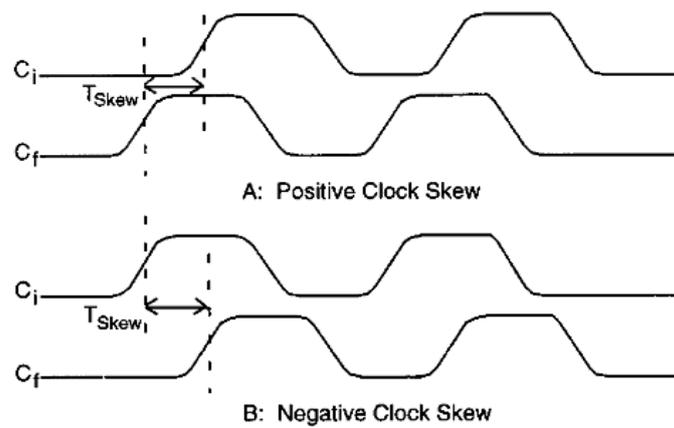


Figura 2.14: Skew entre dos señales de reloj [7].

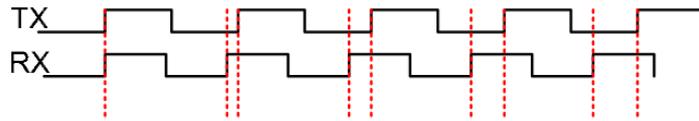


Figura 2.15: Offset de frecuencia entre relojes [8].

2.6.3. Jitter

Se conoce como *jitter* a la variación de los cruces por cero de una forma de onda de su posición ideal [17], como se observa en la figura 2.16.

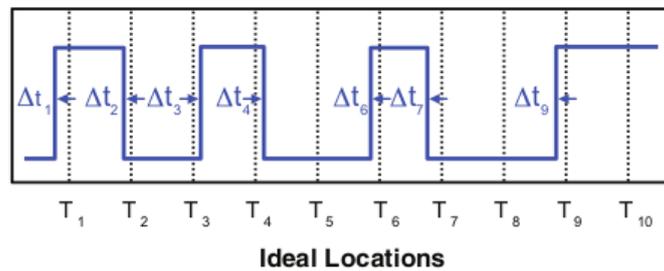


Figura 2.16: Jitter presente en una señal [9].

Se divide en dos grandes categorías conocidas como *jitter* determinista y aleatorio. El *jitter* determinista (DJ) define la trayectoria de cada transición en el diagrama de ojo, y el *jitter* aleatorio ensancha cada trayectoria, como se puede ver en la figura 2.17.

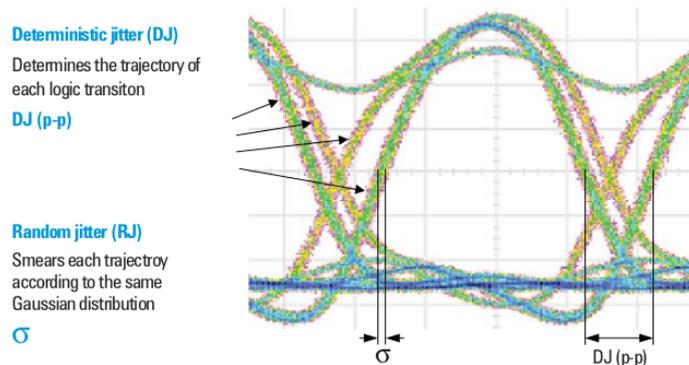


Figura 2.17: Jitter aleatorio y determinista [9].

Jitter aleatorio (RJ):

RJ es causado por la acumulación de un gran número de procesos en el cual cada uno aporta reducidas magnitudes como lo son el ruido térmico, variaciones pequeñas en el ancho y ruido en el disparo. La función de densidad de probabilidad corresponde a una distribución gaussiana y es

no ligada, por lo que hay poca probabilidad que cause una transición lógica en algún momento de manera arbitraria [9].

Jitter determinista (DJ):

Es causado por la pequeña cantidad de procesos que normalmente no son independientes ni despreciables, sino más bien tienen amplitudes grandes como lo son la interferencia intersímbolo, reflexiones y respuestas en frecuencia del canal. La importancia es que su función de densidad de probabilidad es ligada y tiene un valor pico a pico definido.

El *jitter* determinista se divide en:

- Jitter sinusoidal (SJ o PJ): se repite a una frecuencia fija.
- Jitter dependiente a los datos: es correlacionado con el patrón de datos enviado y se divide en tres categorías:
 1. Distorsión por el ciclo de trabajo.
 2. Interferencia intersímbolo.
 3. Jitter no correlacionado ni ligado (mayoritariamente proviene de la diafonía).

Modelo del doble Dirac

Este modelo aproxima el jitter total y permite el cálculo de las componentes aleatorias y deterministas. Se convolucionan las probabilidades de densidad de ambas componentes y se obtiene el *jitter* total como se observa en la figura 2.18.

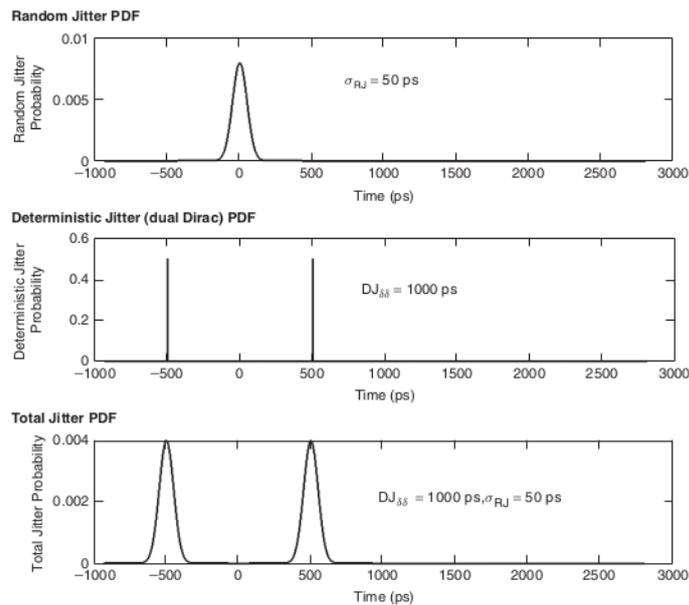


Figura 2.18: Modelo de doble dirac para el jitter total [5].

2.7. Recuperador de reloj y Datos

Un CDR extrae la secuencia de datos de una señal distorsionada y recupera la información de temporizado. Un diagrama general del CDR se observa en la figura 2.19.

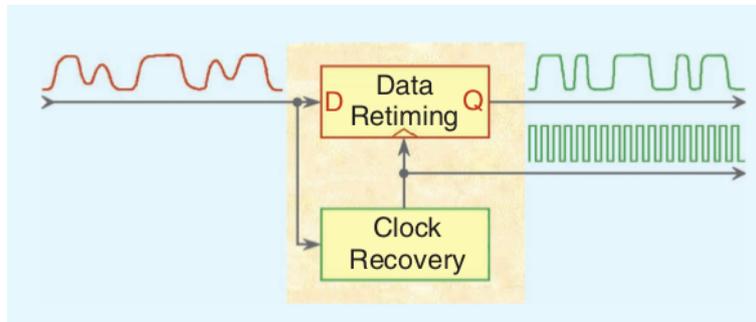


Figura 2.19: Esquema general de un recuperador de reloj y datos [10].

El diseño de un CDR no es único ya que el tipo de arquitectura depende de los siguientes aspectos:

- Tipo de comunicación: continua o ráfaga (*burst mode*).
- Grado de sincronía.
- Aplicación analógica o digital.
- Velocidad de transferencia.
- Dependencia PVT (*Process, voltage and temperature*).
- Rechazo de *jitter*.

3 Arquitectura del CDR

En esta sección se detalla el proceso de selección de la arquitectura a implementar en el simulador de señal mixta. Para realizar la selección se detallan las características, ventajas y desventajas de las principales arquitecturas utilizadas en la literatura. Además, se definen los requerimientos del sistema y las limitaciones del simulador.

Las arquitecturas de CDR se dividen en tres grandes secciones:

1. Topologías que utilizan seguimiento por realimentación como lo son *Phase locked loop* (PLL), *Delay locked loop* (DLL), *Phase interpolator* (PI).
2. Topologías basadas en sobre muestreo sin base de seguimiento por realimentación.
3. Topologías que utilizan el alineamiento de fase sin seguimiento de fase con realimentación incluyendo gated oscillators y arquitecturas con filtros pasabanda con alto factor de calidad (Q).

3.1. CDR basado en PLL

Esta arquitectura utiliza un PLL para recuperar los datos y la señal de temporización, se pueden clasificar según el uso de un reloj de referencia. Además, consiguen una velocidad de datos sintonizable y son fácilmente integrables en un diseño monolítico [10].

Sin reloj de referencia

En la figura 3.1 se observa que el CDR posee un lazo de sintonización de frecuencia y otro de fase. Para realizar la detección y comparación de fase y frecuencia se utilizan detectores que generan una señal proporcional a la diferencia entre la señal de datos y el reloj. Cuando el CDR inicia y se detecta una diferencia en frecuencia, el detector de frecuencia (FD) se activa produciendo una tensión de control a través del circuito *Charge Pump* (CP), la cual es acondicionada por el filtro de lazo desplazando la frecuencia de salida del oscilador controlado por tensión (VCO) a la tasa de datos de entrada. Una vez enganchada la frecuencia, se sincroniza la fase con el detector de fase (PD). Cada lazo representa un control de ajuste grueso y fino [10].

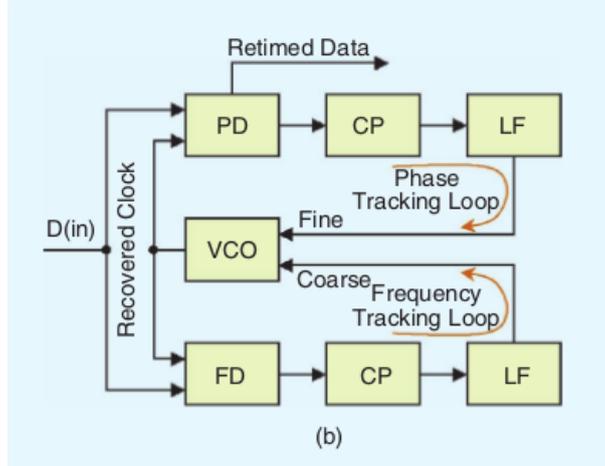


Figura 3.1: CDR basado en PLL sin referencia de reloj [10].

Existen dos problemas con esta configuración, primero el lazo de frecuencia y el de fase pueden interferir entre ellos, generando un rizado no deseado a la salida. Segundo, el FD se puede confundir momentáneamente si llegan bits aleatorios consecutivos e idénticos (CIDs) o si el canal corrompe el estado de los bits. Ante estos problemas se utiliza un ancho de banda más limitado para el FD que para el PD [10].

Con reloj de referencia

A diferencia del anterior, utiliza una señal de referencia para enganchar una frecuencia cercana a la de los datos de entrada como se observa en la figura 3.2. La separación del lazo de frecuencia con respecto a la entrada no afecta el seguimiento de la señal. Al tener una diferencia constante de fase entre dos señales

$$\phi_1(t) - \phi_2(t) = \text{constant}, \quad (1)$$

al derivar por ambos lados

$$\frac{d\phi_1(t)}{dt} = \frac{d\phi_2(t)}{dt}, \quad (2)$$

se obtiene que las frecuencias son iguales

$$\omega_1(t) = \omega_2(t). \quad (3)$$

Esto permite dar un seguimiento en frecuencia con solo seguir la fase.

3.2. CDR basado en DLL

Es similar a una configuración de PLL sin embargo el lazo de frecuencia da una referencia y no una señal de control como se observa en la figura 3.4. Además el lazo de fase utiliza una línea de *delay* controlada por voltaje (VCDL) para la sincronización de fase en lugar de un VCO. El reloj de referencia para el VCDL proviene de una multiplicación de frecuencia realizada en el PLL [10].

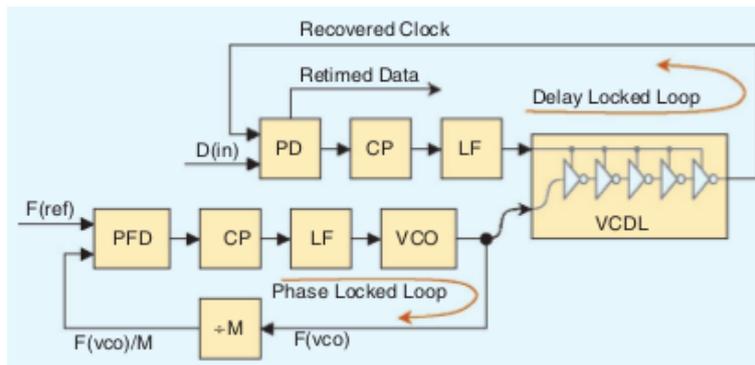


Figura 3.4: CDR basado en DLL con referencia [10].

Lo favorable de esta configuración es que no presenta una acumulación de *jitter*. Además, es un sistema más estable debido a la modificación directa de la fase y no así del VCO. El VCDL no introduce un polo en la función de transferencia del lazo y permite un bloqueo más rápido. Esta configuración es más utilizada para conexiones entre chips.

3.3. Combinación entre PLL y DLL

Se pueden combinar arquitecturas como la que se muestra en la figura 3.5, donde se utiliza un lazo de PLL y una línea de retardo controlada por tensión. Esta arquitectura presenta alta eliminación de *jitter* como resultado del ancho de banda limitado en el lazo de seguimiento de fase y un corto tiempo de adquisición en el lazo de frecuencia [10].

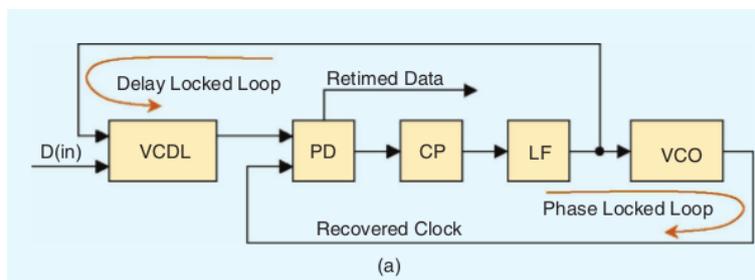


Figura 3.5: Arquitectura combinada [10].

3.4. CDR con sobre muestreo

Es una arquitectura poco utilizada que permite recuperar los datos sin ningún tipo de retardo. Como se ve en la figura 3.6 se utiliza un lazo de sintonización de frecuencia mediante un PLL y se

producen diversas fases en el oscilador. Las fases van a diversos registros que toman las muestras de manera que se realizan al menos tres muestras por bit recibido, además cuenta con un bloque de recuperación de datos que se encarga de detectar y seleccionar los datos de salida [10].

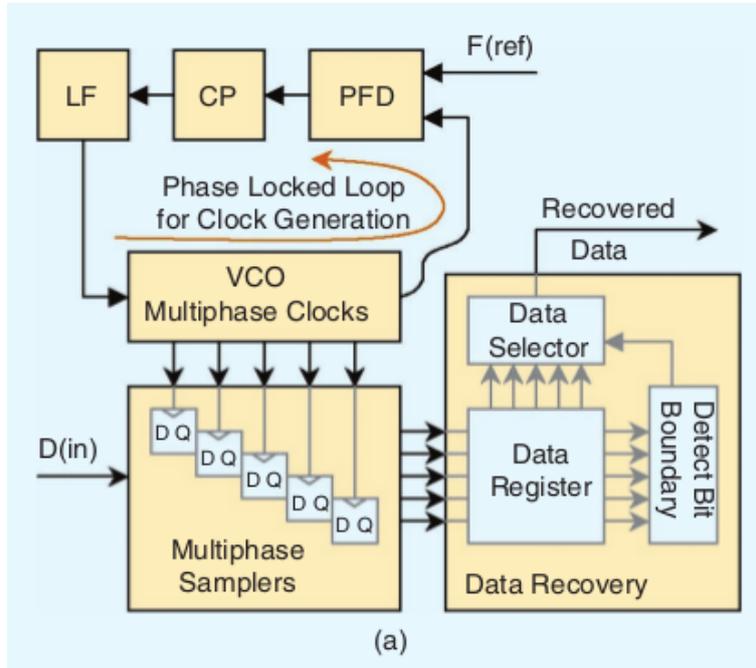


Figura 3.6: Arquitectura con sobremuestreo [10].

3.5. CDR con interpolador de fase

La última topología utiliza un interpolador de fase (PI), en la que se define su señal de salida como [11] :

$$\sin(\omega t + \phi) = \sin(\omega t)\cos(\phi) + \cos(\omega t)\sin(\phi), \quad (4)$$

La señal de fase ϕ se genera mediante la amplitud de las señales de control $\sin(\phi)$ y $\cos(\phi)$, las cuales modulan los relojes en cuadratura de la entrada $\sin(\omega t)$ y $\cos(\omega t)$. Las señales de control se generan dentro del CDR [11]. Un ejemplo de un PI diferencial se muestra en la figura 3.7.

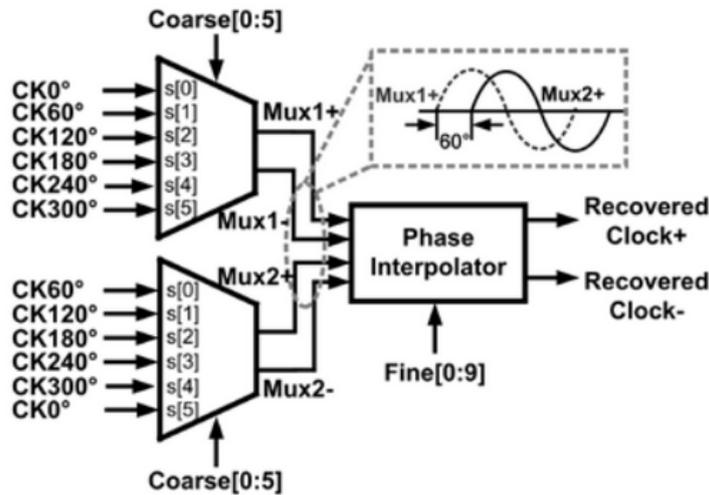


Figura 3.7: Interpolador de fase [11].

La arquitectura consta de un lazo de seguimiento de frecuencia implementado con un PLL y un lazo de seguimiento de fase con un PI, una máquina de estados y un detector de fase, como se puede ver en la figura 3.8. El detector de fase compara la señal de reloj proveniente del PLL con la del canal de comunicación, generando una señal de entrada para la máquina de estados. Posteriormente la FSM controla el interpolador de fase y las señales que ingresan a este hasta que se logre enganchar la señal.

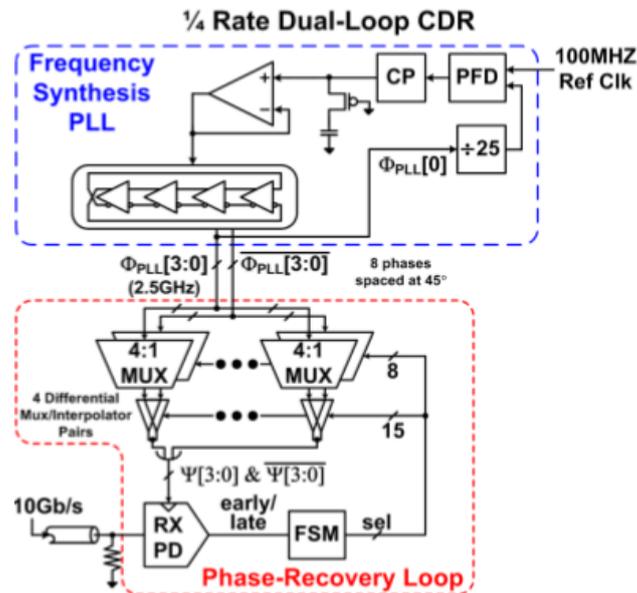


Figura 3.8: Implementación de arquitectura con interpolador de fase [2].

Una variación que permite mayor facilidad en el diseño es el uso de líneas de retardo colocadas

después del PLL y un selector de fase controlado por una máquina de estados (digital) o una señal analógica proveniente del comparador de fase, como se puede ver en la figura 3.9.

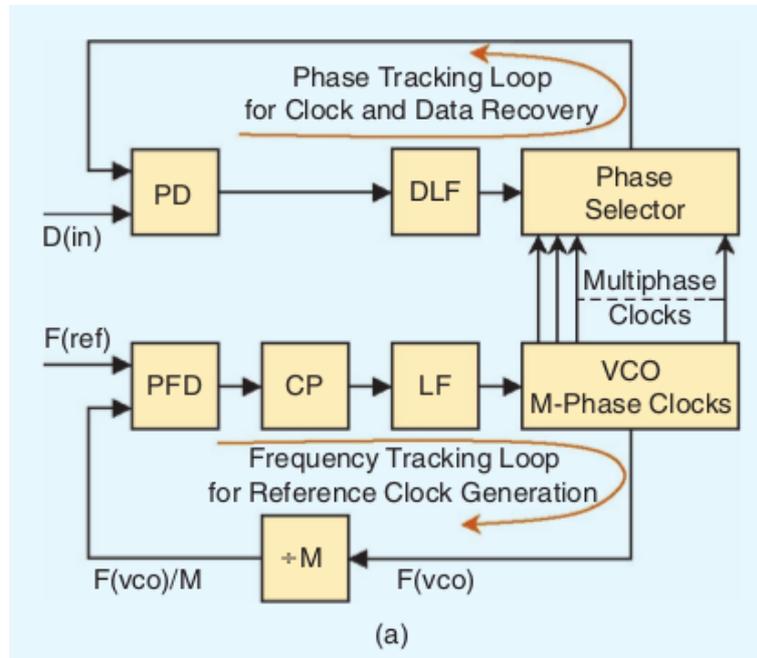


Figura 3.9: Variación de interpolador de fase [10].

Por último se tiene una arquitectura de la figura 3.10, parecida al interpolador de fase pero en lugar de separar los lazos de frecuencia y fase, se abre el lazo del PLL y se realimenta con la señal del interpolador. Su diseño es más complicado ya que se abre el lazo del PLL.

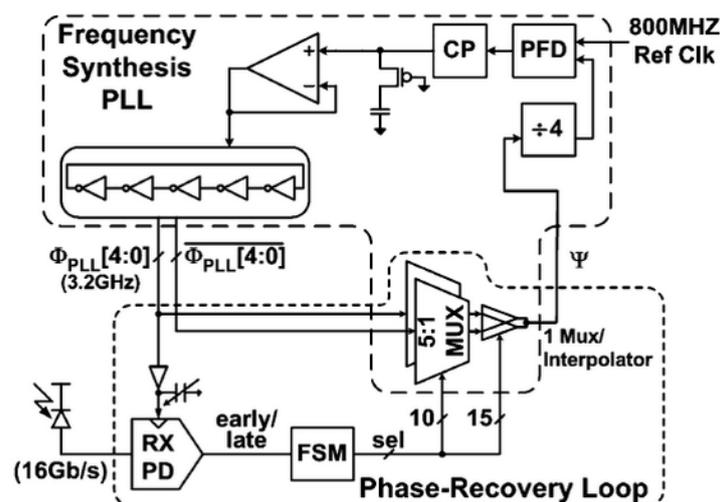


Figura 3.10: Arquitectura con rotador de fase [10].

3.6. Comparación entre arquitecturas

Se pueden categorizar las aplicaciones de los CDR según su tipo de transmisión en modo de ráfaga o continuo y se caracterizan por [10] :

Modo ráfaga

- Se utiliza en comunicaciones de punto a multi punto.
- Al transmitir en periodos fuera de sincronía, posee tiempos de silencio.
- Ocupa una adquisición rápida debido a la corta latencia de la comunicación.
- Utilizan arquitecturas sin realimentación o sobremuestreo.

Modo continuo

- Se utiliza en comunicaciones de punto a punto.
- No posee tiempos de silencio.
- Periodos largos de adquisición.
- Más sensitivos al *jitter*.
- Deben soportar CIDs.
- Arquitecturas más utilizadas son PLL, DLL, combinación de ambas, o interpolador de fase.
- Sufre de sobre impulsos debido a los PLL.

3.7. Requerimientos del CDR del proyecto

Con el fin de unificar todas las secciones del canal serial de alta velocidad del proyecto, se debe cumplir con los siguientes requerimientos:

- Velocidad de envío de datos en el orden de los Gb/s.
- Modo de comunicación continuo.
- Presencia de *skew*.
- Presencia de *jitter*.
- Presencia de *offset* en frecuencia.
- Se busca que los parámetros de la simulación sean modificables.

3.8. *Requerimientos y limitaciones del entorno de simulación*

Para el diseño del modelo de comportamiento del CDR se utilizó la herramienta de software *Advanced Design System* (ADS). Permite realizar simulaciones de diseños electrónicos en alta frecuencia, RF y microondas. Además trabaja en diferentes dominios como el tiempo, frecuencia y simulaciones de tipo Ptolemy [18].

La simulación tipo Ptolemy en ADS brinda las herramientas de simulación necesarias para la realización del proyecto como lo son:

- Diseñar y evaluar sistemas de comunicación modernos.
- Utiliza algoritmos de DSP en comunicaciones.
- Permite simulaciones en sincronía por flujos de datos.
- Posee modelos de comportamiento de al menos 500 dispositivos.
- Su interfaz es fácil de usar.
- Tiene una biblioteca específica de diseño de PLL.
- Permite el diseño de máquinas de estado de tipo *mealy*.

El simulador presenta las siguientes limitaciones:

- Los modelos de comportamiento no permiten un cambio en su diseño.
- No tiene modelos de comportamiento para interpoladores de fase.
- Para diseños grandes resulta incómodo utilizar esquemáticos.

3.9. *Selección de arquitectura y modificaciones*

De acuerdo a los requerimientos del proyecto y las limitaciones del simulador, se seleccionaron las siguientes opciones:

- Interpolador de fase.
- Interpolador de fase modificado con línea de retardo.
- Rotador de fase.

La arquitectura final seleccionada es: **Interpolador de fase modificado con línea de retardo**. Esta arquitectura no necesita el modelo de comportamiento del interpolador de fase y presenta un menor grado de dificultad que un rotador de fase. La arquitectura final se observa en la figura 3.11.

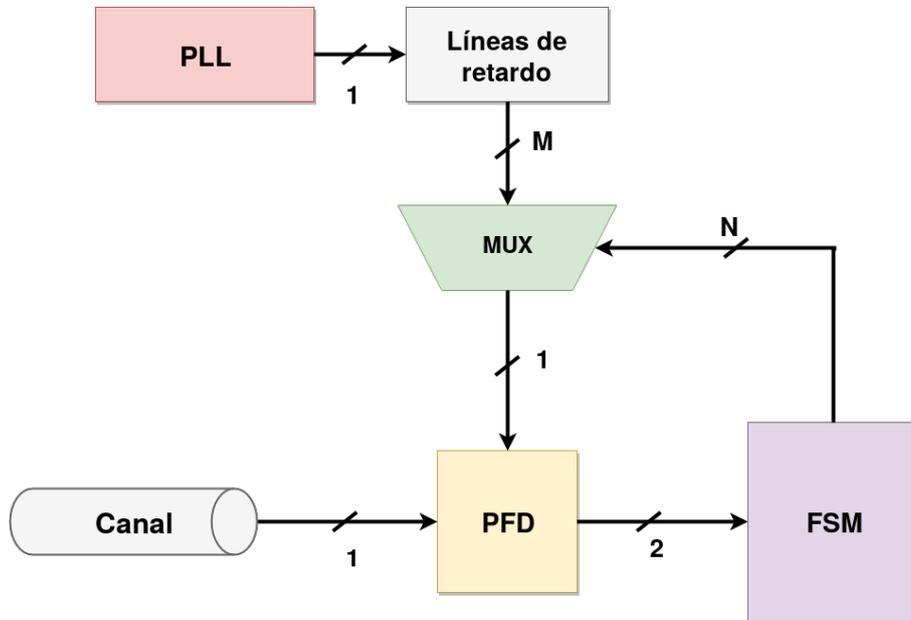


Figura 3.11: Arquitectura final.

El funcionamiento consiste en la comparación de la fase y frecuencia de los datos provenientes del canal en el PFD con respecto al reloj generado en el PLL local. EL PFD genera dos señales que corresponden a un adelanto o atraso de la señal de reloj en el receptor con respecto a los datos del canal. Posteriormente la máquina de estados genera la señal de control para que la fase más cercana salga por el multiplexor y se logre enganchar el sistema.

4 Modelo de comportamiento del recuperador de reloj y datos

En esta sección se desarrolla la implementación y diseño del recuperador de reloj y datos seleccionado en el capítulo 3, viéndose cada uno de los elementos que lo componen y su correcto funcionamiento. Los modelos de comportamiento y pruebas de cada componentes se realizan en el entorno de simulación *Advanced Design System* (ADS).

4.1. Diseño y evaluación de los componentes de la arquitectura

4.1.1. Detector de frecuencia y fase (PFD)

En la figura 4.1 se observa el comparador de frecuencia y fase, con entradas A y B que producen las salidas Q_A y Q_B . Su funcionamiento es el siguiente [12]:

1. Un flanco positivo en la entrada A provoca un flanco positivo en Q_A (Si, Q_A estaba en bajo).
2. Un flanco positivo en la entrada B reinicia Q_A (Si, Q_A estaba en alto).
3. Si $\omega_A > \omega_B$, solo Q_A produce pulsos y Q_B se mantiene en cero.
4. Si $\omega_A < \omega_B$, solo Q_B produce pulsos y Q_A se mantiene en cero.
5. Si $\omega_A = \omega_B$, se generan pulsos en Q_A y Q_B con un ancho igual a la diferencia de fase entre A y B.
6. El valor promedio de $Q_A - Q_B$ representa diferencia de frecuencia y fase.

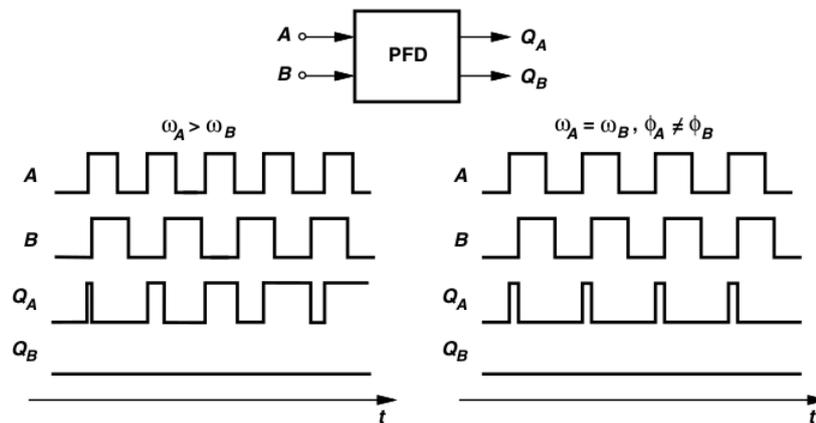


Figura 4.1: Respuesta de un PFD a una entrada con diferentes frecuencias y fases [12].

Lo antes descrito se resume en la máquina de estados activada por flancos de la figura 4.2.

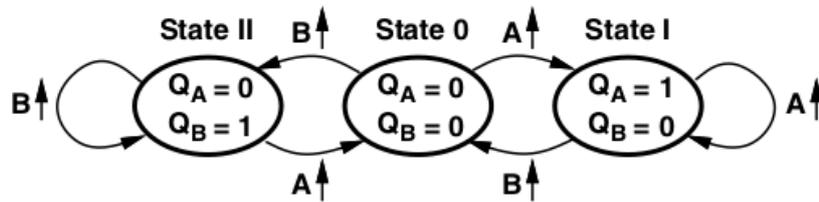


Figura 4.2: Máquina de estados de un PFD [12].

El modelo de comportamiento utilizado en ADS se basa en la implementación de la figura 4.3, consiste en una lógica de dos *flip flop* (FF). Las señales A y B actúan como los relojes y la compuerta AND reinicia los FF si $Q_A=Q_B=1$. Se genera un pulso en Q_B debido al retardo de la compuerta AND.

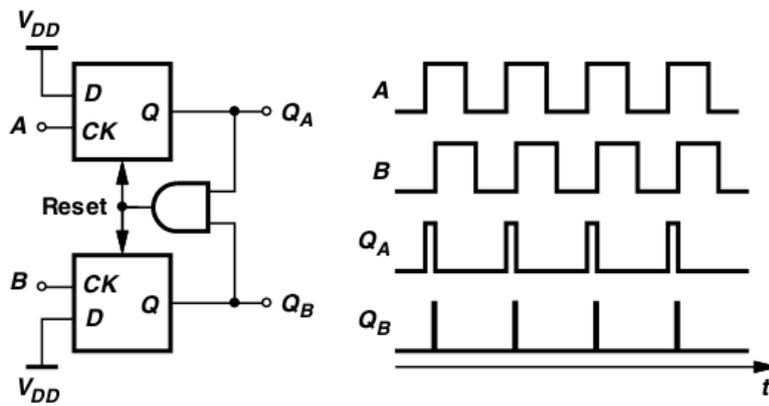


Figura 4.3: Implementación de PFD [12].

Prueba de comportamiento del PFD

El PFD que se observa en la figura 4.4 se implementó en un símbolo de ADS, lo que permite minimizar el tamaño del esquemático. Para observar su comportamiento se realizó una cosimulación entre *ptolemy* y una simulación de transitorio. El circuito de prueba consta de dos señales generadas en las entradas de un PFD y un sumidero de tiempo que permite analizar cada una de las salidas como se observa en la figura 4.5.

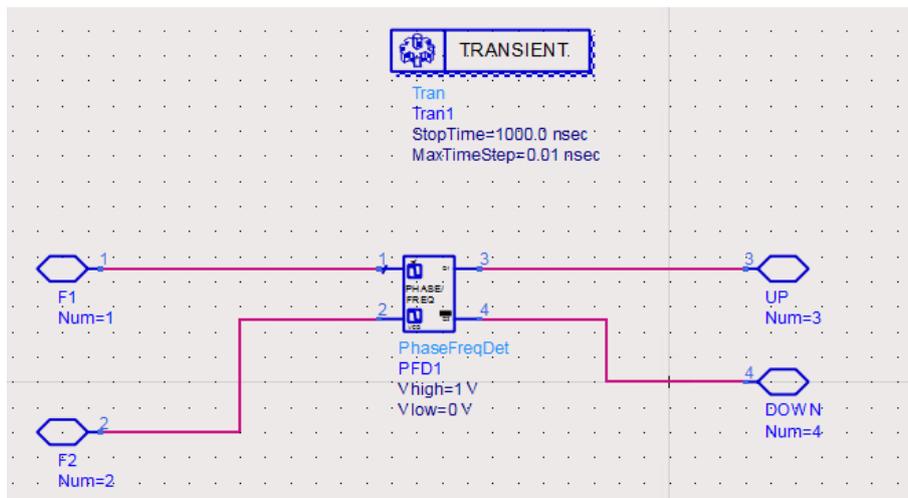


Figura 4.4: Símbolo en ADS del PFD.

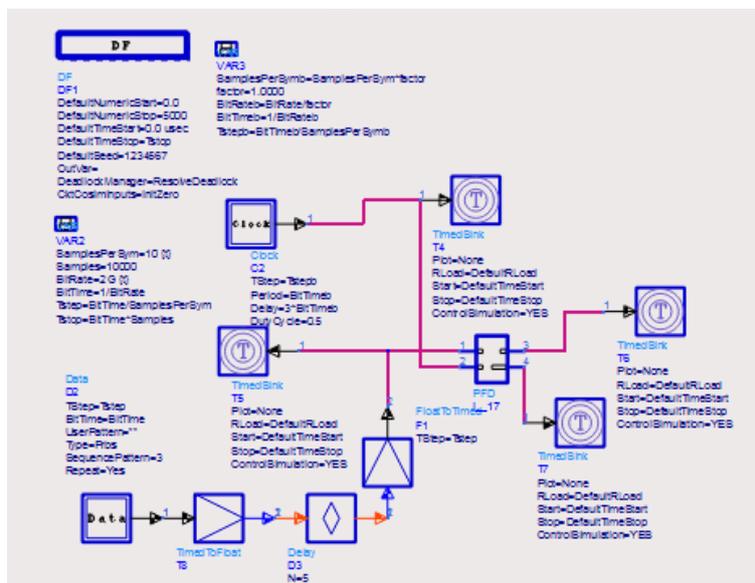


Figura 4.5: Circuito de prueba del PFD.

La simulación consiste en dos pruebas:

- **Prueba 1:** Se ingresan dos señales de reloj con la misma frecuencia en la entrada del PFD. En la figura 4.6 no hay desfase entre las entradas, por otro lado en las figuras 4.7 y 4.8 existen desfases entre las entradas.
- **Prueba 2:** Se ingresa una señal de reloj en una entrada y otra de datos, con diferente frecuencia, en la segunda entrada del PFD. En la figura 4.9 se tiene un pequeño desfase entre las señales de entrada y en las figuras 4.10 y 4.11 se tienen diferentes fases en la entrada.

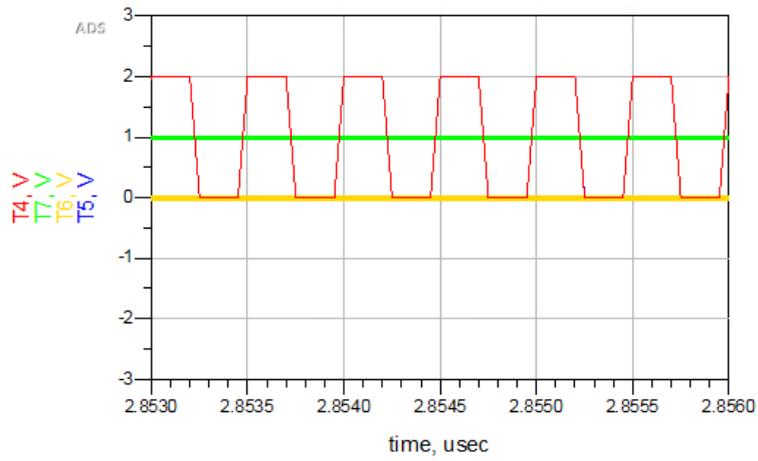


Figura 4.6: Salida del PFD con frecuencia y fase iguales.

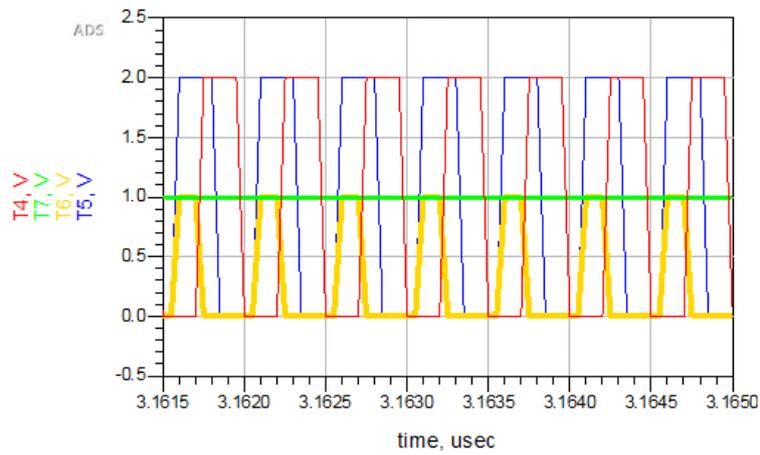


Figura 4.7: Salida del PFD con frecuencia igual y fase diferente.

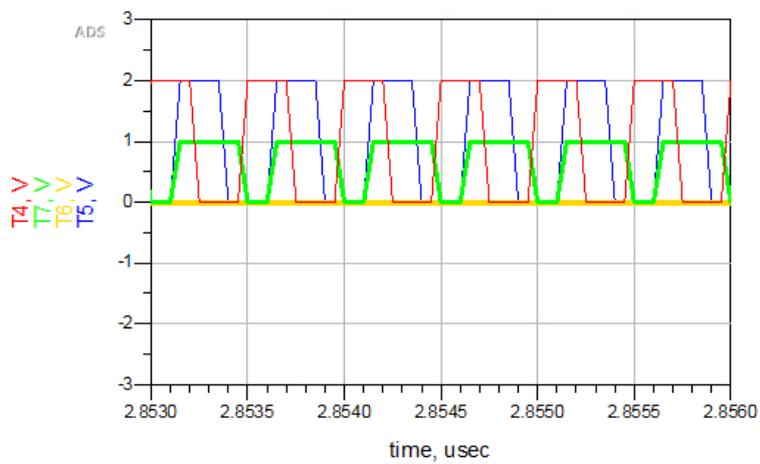


Figura 4.8: Salida del PFD con frecuencia igual y fase diferente 2.

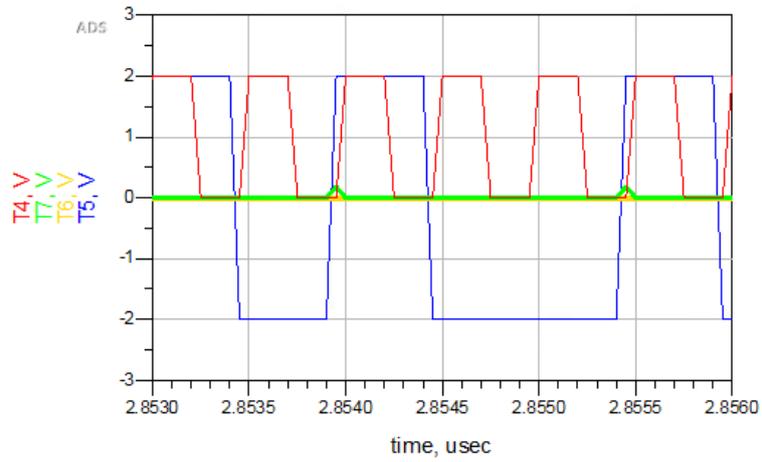


Figura 4.9: Salida del PFD con frecuencia y fase diferente.

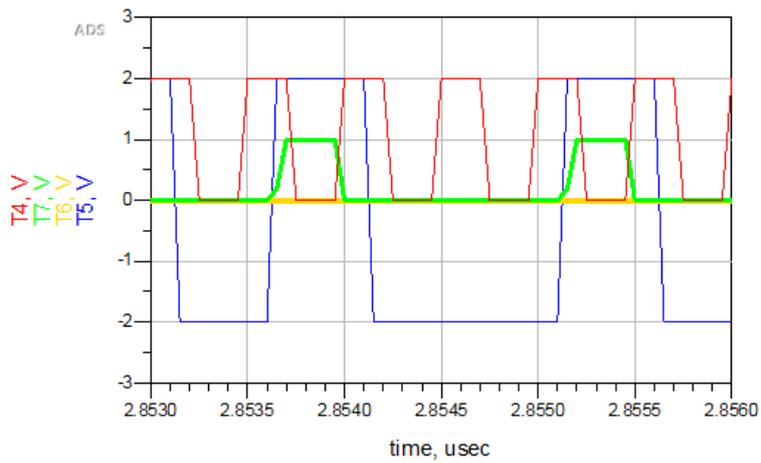


Figura 4.10: Salida del PFD con frecuencia y fase diferente 2.

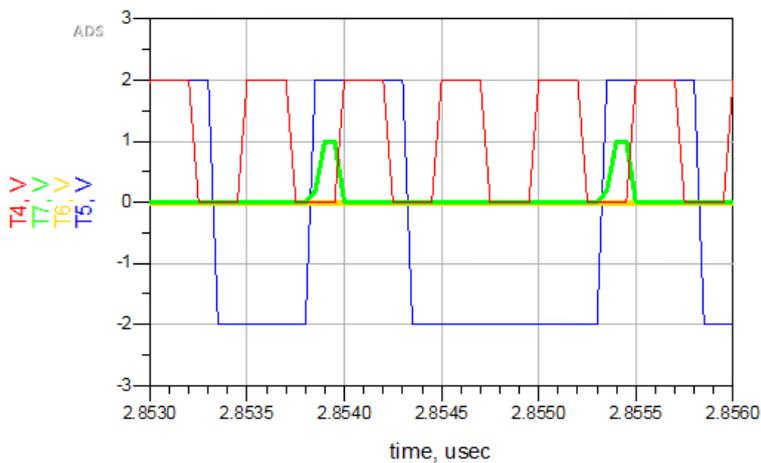


Figura 4.11: Salida del PFD con frecuencia y fase diferente 3.

Resultados obtenidos

De las dos pruebas realizadas se obtuvieron los siguientes resultados:

- **Prueba 1:** Al aplicar dos señales con la misma frecuencia, se obtuvo que al tener la misma fase no se genera un cambio en ninguna de las salidas. Al aplicar una diferencia de fase se generan cambios en las dos salidas según el adelanto o atraso entre las señales de entrada como era esperado.
- **Prueba 2:** Al aplicar dos señales de distinta frecuencia, se observa la diferencia en fase entre ambas señales. Con la diferencia de que es solo varía una de las salidas. Lo anterior limitará el funcionamiento del CDR, ya que no se puede detectar si la señal de reloj se encuentra en adelanto o atraso con respecto a los datos.

Ante los resultados obtenidos se tiene un correcto funcionamiento del modelo en ADS del PFD cuando existen frecuencia iguales o muy parecidas, pero no así para la aplicación del CDR donde se utiliza una señal de datos de la mitad de la frecuencia del reloj, lo que limita la detección en uno de los sentidos de adelanto u atraso.

Para el proyecto se trabajó con este modelo, sin embargo, se tiene que modificar posterior a la finalización del mismo, ya que no existe un modelo de comportamiento diferente en las bibliotecas de ADS, por lo que se tiene que diseñar desde los componentes internos que lo conforman.

4.1.2. Línea de retardos

Como se observa en la arquitectura de la figura 3.9, se deben generar diversas fases de la señal de reloj proveniente del oscilador que van a un selector de fase. Lo anterior se logra creando una línea fija de retardos para cada muestra, como se observa en la figura 4.12.

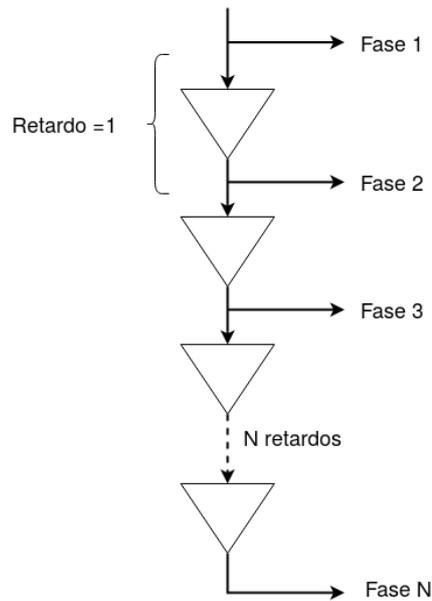


Figura 4.12: Línea de retardos.

Para implementar la línea de retardos se debe tomar en cuenta los siguientes puntos:

- Cada ciclo de reloj en la simulación contiene 10 muestras.
- La cantidad de grados necesarios para construir el CDR debe asegurar que el centro de algún diagrama de ojo quede dentro del interpolador por lo que se deben obtener las siguientes fases: 0° , 180° , 360° y 540° . La última fase es importante porque los elementos de retardo y el interpolador pueden no coincidir con precisión, ya que sus cargas pueden diferir ligeramente [19] lo que puede causar que el centro del ojo quede fuera del rango.
- Cada uno de los 180° se divide en 10 puntos correspondientes a 18° entre ellos como se observa en la figura 4.13.

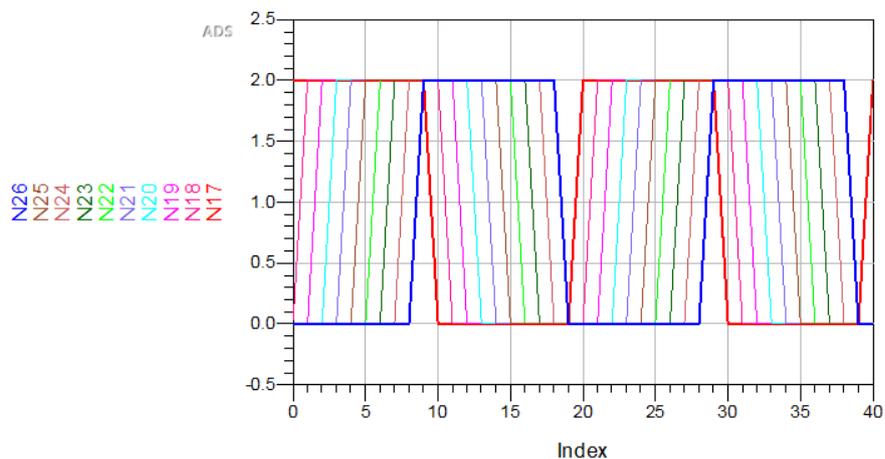


Figura 4.13: Fases generadas en las líneas de retardo.

- Se tiene un total de 40 fases constituidas por una línea de retardo de 4 salidas con una diferencia de fase de 180° que se observa en la figura 4.14, la que va a 4 líneas de retardo de 10 salidas con una diferencia de fase de 18° .

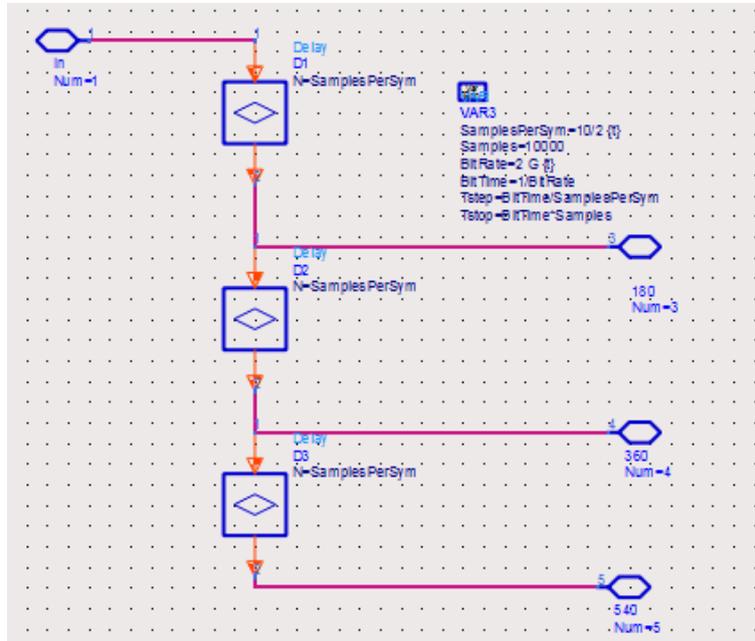


Figura 4.14: Línea de retardo de 180° .

4.1.3. Multiplexor (Mux)

El selector de fase se basa en un multiplexor en el que ingresan todas las señales de las líneas de retardo y sale únicamente la fase necesaria para sincronizar los datos en el receptor. El multiplexor cumple con las siguientes características:

- Le ingresan las 40 fases de las líneas de retardo.
- Es controlado por una máquina de estados.
- No genera retardos considerables en las señales de entrada.

La implementación se realizó con dos multiplexores en cascada, ya que no se puede ingresar las 40 fases en uno solo, por limitaciones del simulador. El primer MUX recibe 4 grupos de 10 fases y el segundo selecciona cual de las 10 se selecciona a la salida.

4.1.4. Máquina de estados tipo Mealy (FSM)

Se utiliza una máquina de estados de tipo Mealy por su dependencia de las entradas y el estado actual [20]. Su funcionamiento se describe en los siguientes puntos:

1. Las entradas son UP y DN.
2. La salida controla los dos multiplexores del selector de fase.
3. Las máquinas de estado no permiten separar el bus de salida en la simulación, por lo que se crea una **FSM principal** que lleva el estado actual (describe la fase actual) según las entradas del PFD y **otras tres FSM** decodifican las señales de los MUX según el estado presente en la entrada.
4. Se utilizan tres FSM para controlar cada uno de los MUX. Una implementación real solo toma partes de la salida de la FSM general hacia los iferentes MUX.
5. La máquina de estados va en orden desde el estado cero hasta el 39 (igual al número de fases), en orden ascendente o descendente dependiendo de la relación entre las fases.
6. La FSM principal únicamente importa el valor del estado actual y su salida no se utiliza.
7. De las tres FSM para los MUX solo se toma el valor de salida.

La entrada de la FSM se une en un solo bus y se resumen sus valores en la tabla 4.1. El diagrama de estados de la FSM principal se observa en la figura 4.15.

Tabla 4.1: Tabla resumen de la entrada a FSM principal.

Entrada	Significado
0	Clock adelantado
1	Indeterminado
2	En fase
3	Clock atrasado

El diagrama de estados de la FSM principal se observa en la figura 4.15, donde se presentan los 40 estados de manera resumida.

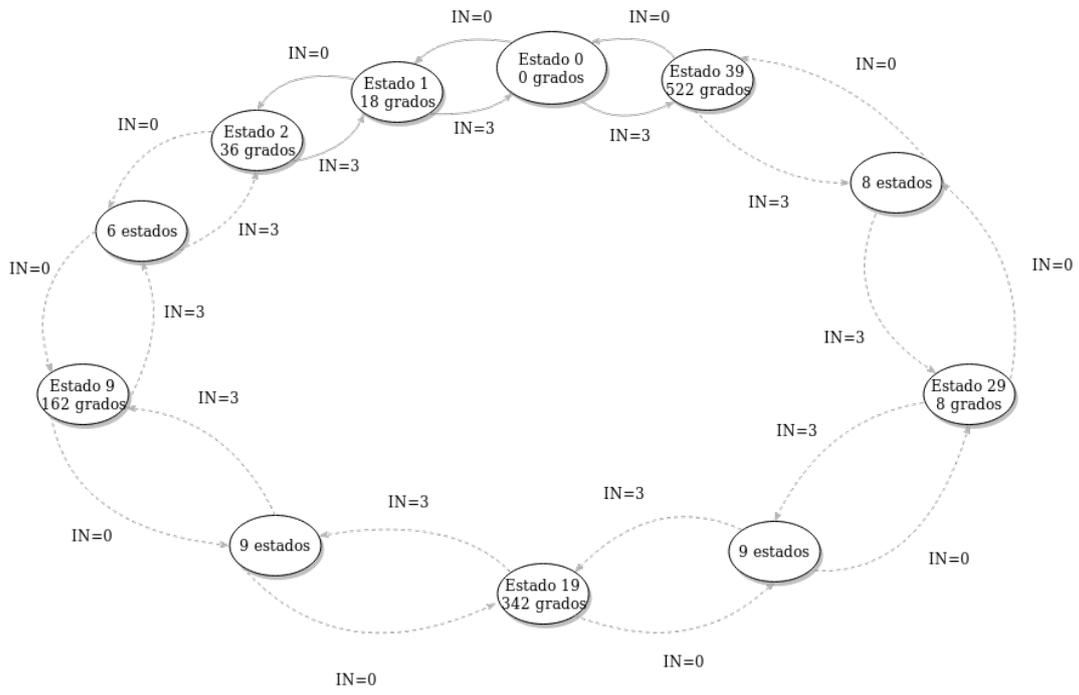


Figura 4.15: Máquina de estados principal.

Para las FSM restantes se debe seguir la lógica de la tabla 4.2.

Tabla 4.2: Tabla de verdad para la activación de la FSM

Estado	MUX 1 (binario)		MUX 2 (decimal)
	IN1	IN0	IN
0	0	0	0
1	0	0	1
2	0	0	2
3	0	0	3
4	0	0	4
5	0	0	5
6	0	0	6
7	0	0	7
8	0	0	8
9	0	0	9
10	0	1	0
11	0	1	1
12	0	1	2
13	0	1	3
14	0	1	4
15	0	1	5
16	0	1	6
17	0	1	7
18	0	1	8
19	0	1	9
20	1	0	0
21	1	0	1
22	1	0	2
23	1	0	3
24	1	0	4
25	1	0	5
26	1	0	6
27	1	0	7
28	1	0	8
29	1	0	9
30	1	1	0
31	1	1	1
32	1	1	2
33	1	1	3
34	1	1	4
35	1	1	5
36	1	1	6
37	1	1	7
38	1	1	8
39	1	1	9

4.1.5. Activación de FSM

La máquina de estados es activa por flancos positivos. Teniendo en cuenta que el cambio de la FSM debe dar tiempo a que la salida se estabilice en un valor de fase y se pueda evaluar de nuevo en el PFD.

Para activación de la FSM se utiliza una lógica con las salidas del comparador de frecuencia y fase que se observa en la tabla 4.3.

Tabla 4.3: Tabla de verdad para la activación de la FSM

Entradas		Salida
UP	DOWN	Y
0	0	1
0	1	1
1	0	0
1	1	1

De la tabla 4.3 anterior se obtiene la siguiente ecuación

$$Y = \overline{DN} \cdot \overline{UP} + \overline{DN} \cdot UP + DN \cdot UP, \quad (5)$$

$$Y = \overline{DN} + DN \cdot UP, \quad (6)$$

la cual, describe la lógica que se implementa en el esquemático de la figura 4.16.

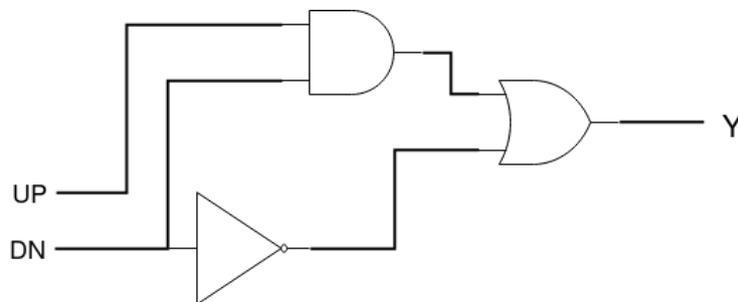


Figura 4.16: Circuito de activación para la FSM.

4.2. Limitaciones de los componentes del CDR

Al unir todas las partes del CDR descritas en este capítulo, se implementa la arquitectura de la figura 3.11. Del diseño de cada una de las partes se desprenden las siguientes limitaciones:

1. El uso de un PFD en lugar de un detector de fase ocasiona que el CDR solo pueda corregir en un sentido de la máquina de estados, ya que al no tener frecuencias iguales en las entradas, solo una de las salidas se activa.
2. Al utilizar muchas líneas de retardo para las fases el circuito puede llegar a ser de gran tamaño, por lo que es recomendable buscar un modelo para un interpolador de fase.
3. El uso una gran cantidad de fases sobrepasa la máxima cantidad de entradas de los MUX.
4. No se pueden modificar los modelos de comportamiento de ADS, lo que limita realizar un cambio en el diseño.

Dos propuestas para sustituir el PFD posterior a la finalización del proyecto son:

Detector de fase Hogge

Es un detector de fase lineal ya que proporciona información sobre el signo y la magnitud de la diferencia de fase [13]. Su implementación se puede ver en la figura 4.17.

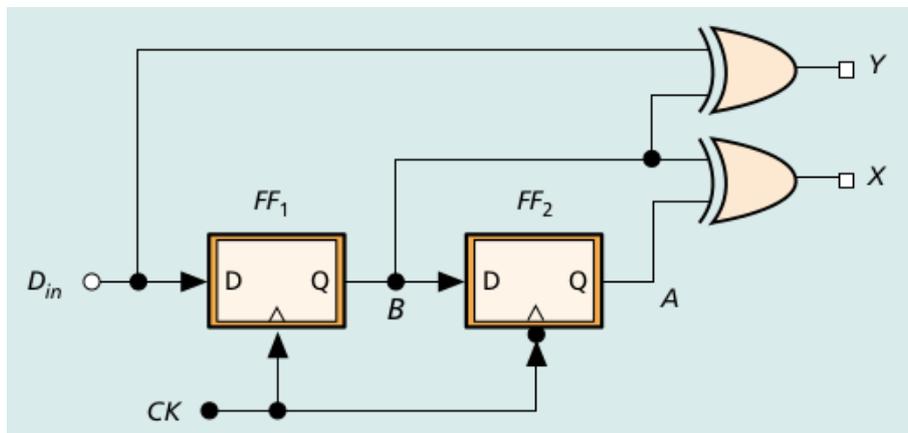


Figura 4.17: Detector de fase tipo Hogge [13].

Al ser lineal no es preferible para el diseño del CDR, sin embargo es una opción existente.

Detector de fase Alexander

A diferencia del anterior es considerado como un detector de fase no lineal, ya que únicamente brinda información sobre el signo. Su implementación es ocupa más área, como se puede observar en la figura 4.18.

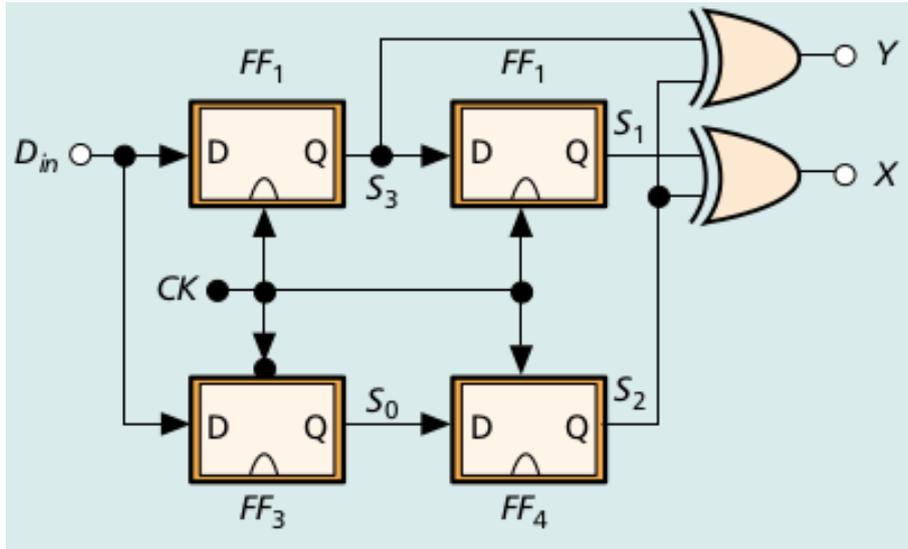


Figura 4.18: Detector de fase tipo Alexander [13].

Esta opción es la que mejor se adapta al CDR y debe ser examinada posterior a la finalización del proyecto.

5 Evaluación y análisis del CDR

Al tener el diseño completo en ADS del CDR, se procede a realizar algunas pruebas con el fin de validar su correcto funcionamiento. Se proponen las siguientes pruebas:

- **Prueba 1:** El reloj generado en el receptor y los datos provenientes del canal poseen la misma frecuencia y fase.
- **Prueba 2:** Se aplica la misma frecuencia pero diferente fase (*skew*).
- **Prueba 3:** Se aplican diferentes *offset* de frecuencia entre las señales.
- **Prueba 4:** Se aplican diferentes señales de PRBS a la entrada.

Para todas las pruebas se utilizó la siguiente configuración:

- Muestras por símbolo = 10.
- Número de muestras = 100 000.
- Tasa de envío de datos = 2 Gb/s. La evaluación permite cualquier tasa de envío, con la limitación que entre más velocidad, la simulación dura un mayor tiempo y la señal se ve más afectada cuando proviene del canal.
- Tiempo de bit = 0.5 ns.
- Tiempo de parada = 50 μ s.
- Factor: es la relación entre la frecuencia de los datos y el reloj.
- Las señales T4 y T5 corresponden al reloj en el receptor y la señal proveniente del canal respectivamente.
- Las señales T6 y T7 corresponden a las salidas del PFD DOWN y UP respectivamente.
- La señal T16 corresponde al estado actual en la FSM.

5.1. Prueba 1

Al colocar la misma frecuencia y fase, el CDR ya se encuentra enganchado. En la parte izquierda de la figura 5.1, se observa que el estado de la FSM no cambia ya que se encuentra enganchado y en la parte derecha las dos señales superpuestas, en fase. Para este caso el CDR se comporta correctamente según lo esperado.

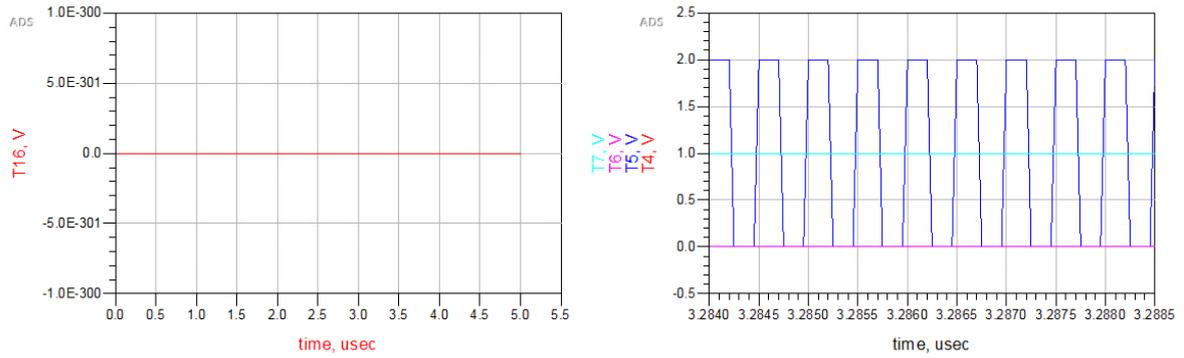


Figura 5.1: Señales en fase.

5.2. Prueba 2

Ahora al colocar un *skew* entre la señal de reloj y datos, el CDR dura un tiempo en enganchar. Entre mayor sea el *skew*, mayor tiempo tardará en enganchar, como se observa en las figuras 5.2, 5.3, 5.4. En estas tres figuras el reloj adelanta a la señal proveniente en el receptor.

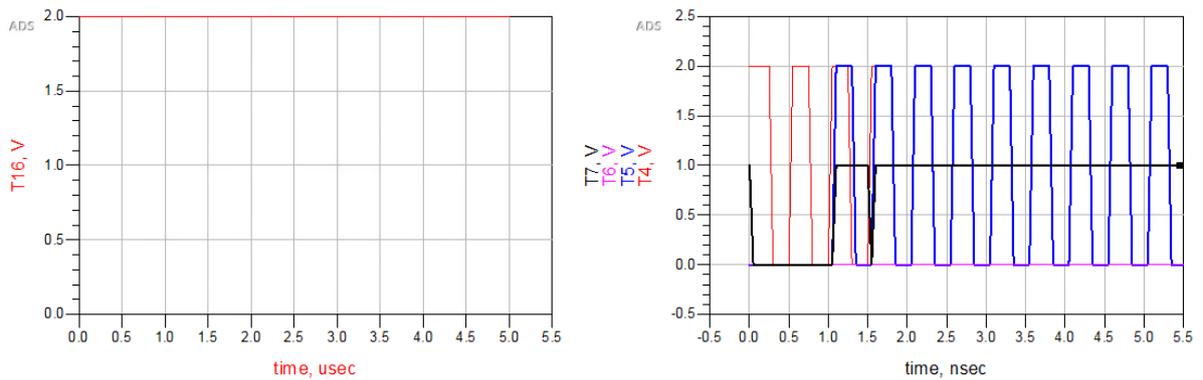


Figura 5.2: Presencia de skew 1.

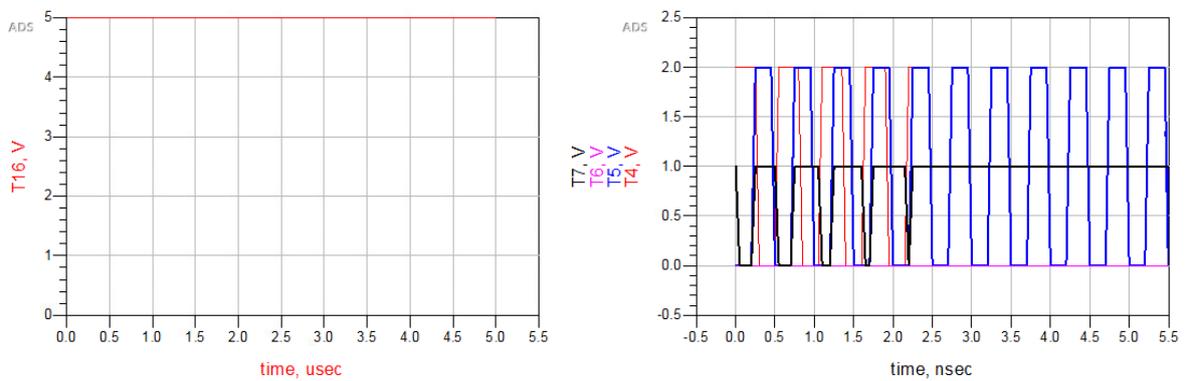


Figura 5.3: Presencia de skew 2.

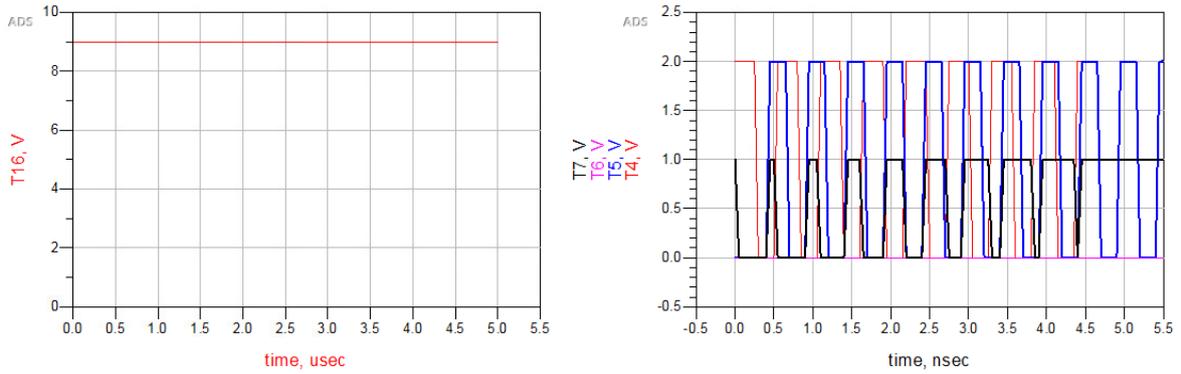


Figura 5.4: Presencia de skew 3.

En el caso de que la señal recibida quede entre dos fases del CDR, se engancha entre los dos estados que la contienen como se muestra en la parte izquierda de la figura 5.5.

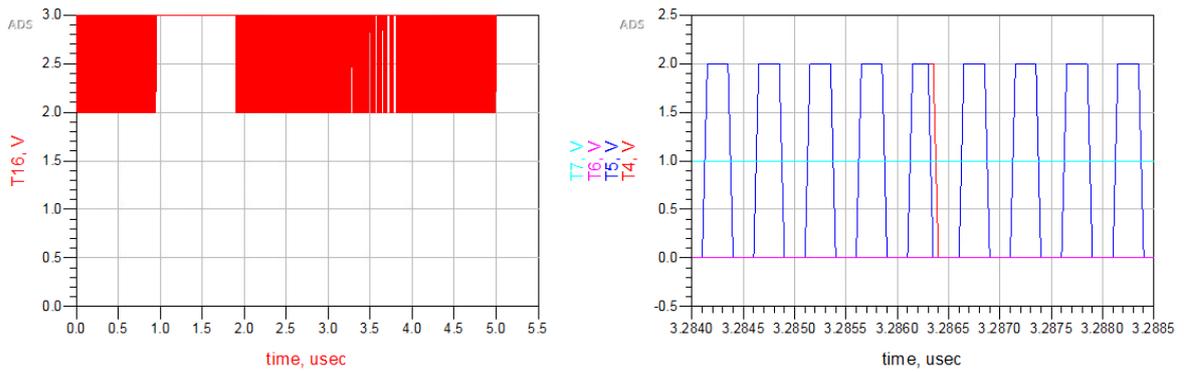


Figura 5.5: Presencia de skew 4.

Al colocar un *skew* con la señal de datos adelantada a la de reloj, el CDR funciona correctamente como se muestra en la figura 5.6. Además se generó la gráfica de la figura 5.7, que resulta de la diferencia entre las señales, la cual indica que las señales se sincronizan correctamente cuando el error tiende a cero.

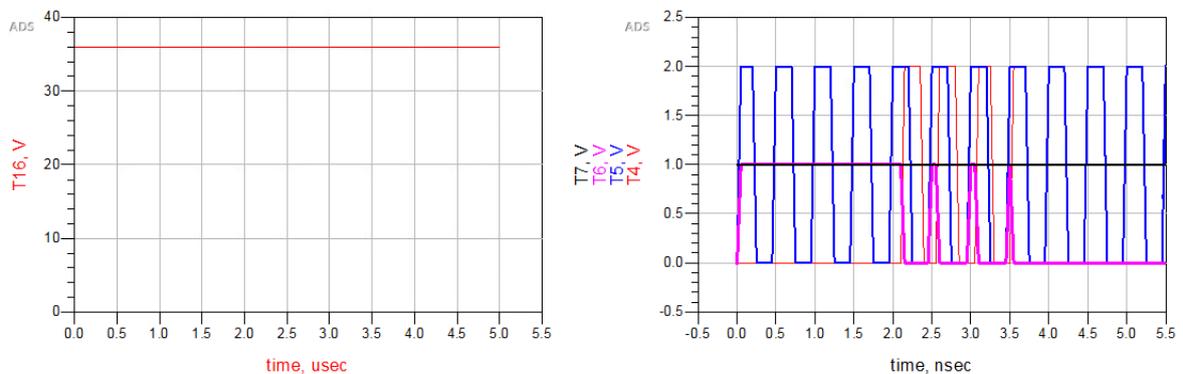


Figura 5.6: Presencia de skew 5.

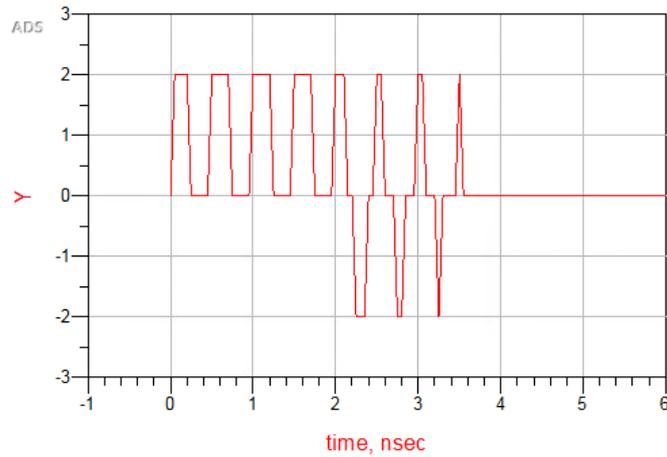


Figura 5.7: Diferencia entre las señales.

Por lo tanto el CDR presenta un correcto funcionamiento ante la presencia de skew positivo o negativo, si se utilizan señales de frecuencias iguales.

5.3. Prueba 3

La siguiente prueba ingresa un *offset* de frecuencia mediante el uso de la variable *factor* descrita al inicio de capítulo. Para la prueba se multiplica el factor por la frecuencia del reloj interno del receptor.

El primer *offset* corresponde a un factor de 0.99998 , que corresponde a una diferencia de frecuencia de 40 kHz menos para la señal de reloj. Se observa en la parte izquierda de la figura 5.8 que la diferencia en frecuencia causa un cambio en el estado actual de la FSM, debido a que el CDR sigue la excursión de la señal. Por otro lado, en la misma figura la recuperación de la señal (parte derecha) y la diferencia obtenida entre las señales (parte inferior), la cual coincide con los cambios de estado de la FSM.

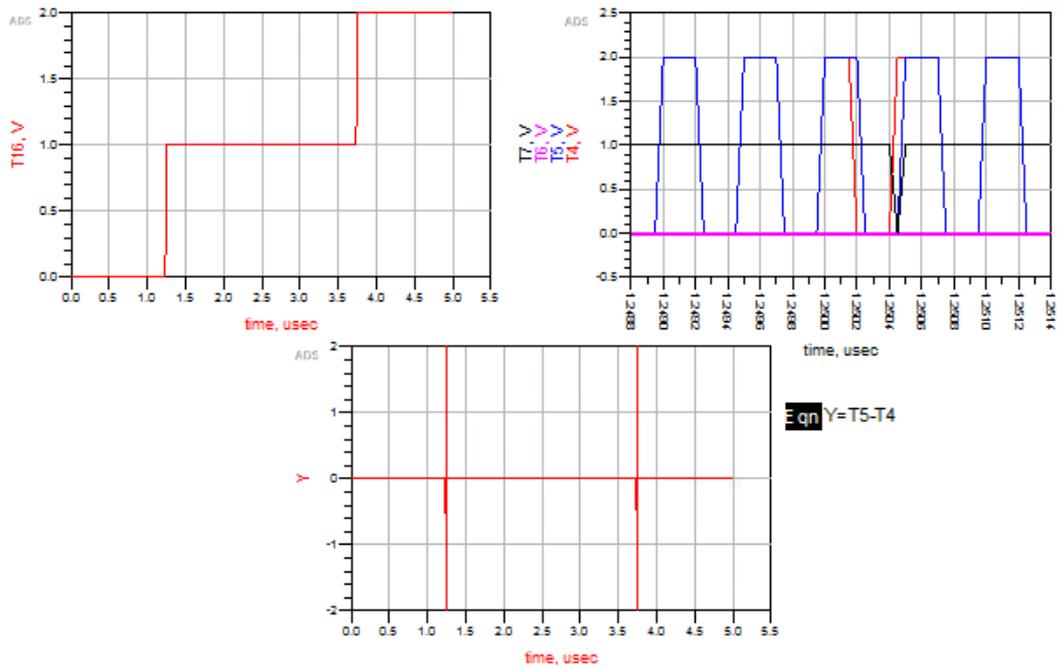


Figura 5.8: Offset de frecuencia 1.

Al aumentar la diferencia en frecuencia en un factor de 0.9999, corresponde a una diferencia de 200 kHz, lo que genera un mayor cambio en los estados de la FSM como se muestra en la parte izquierda de la figura 5.9, además, en la parte derecha de la figura se observa la correcta recuperación de la señal en un punto.

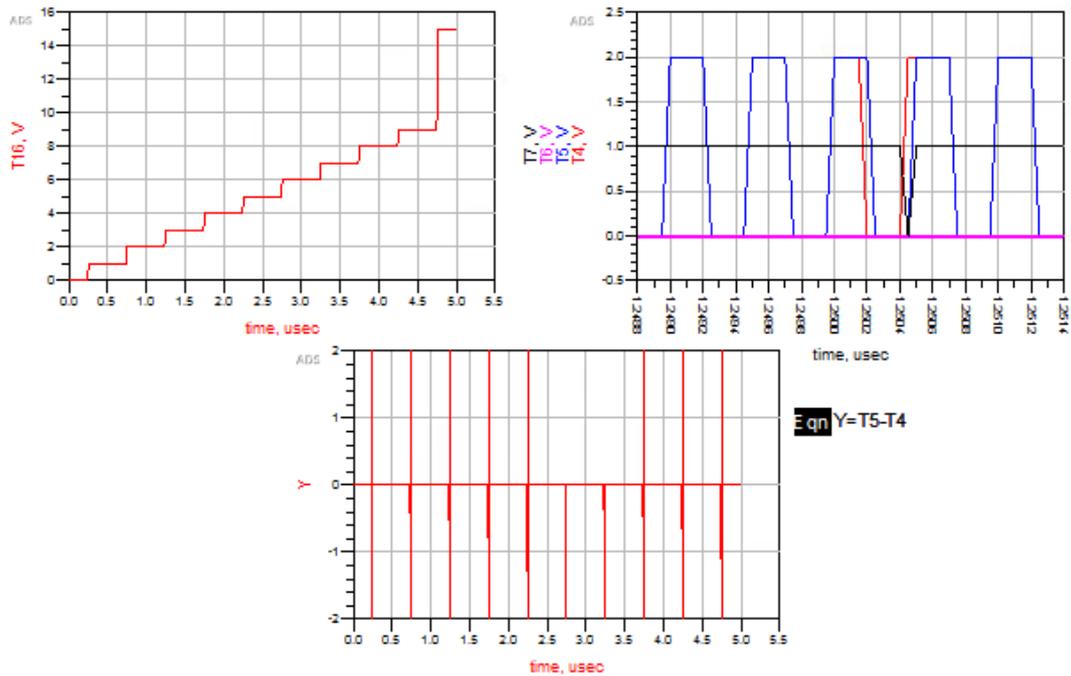


Figura 5.9: Offset de frecuencia 2.

Para variaciones de fase de 2 MHz se obtiene una correcta recuperación de la señal como se observa en la figura 5.10. En variaciones mayores a 20 MHz la recuperación es deficiente ya que la FSM cambia muy rápido de estado y no se logra tener el tiempo suficiente para recuperar la señal, como se muestra en la figura 5.10. No es recomendable la recuperación de la señal para diferencias elevadas de frecuencia como se observa en las figuras 5.11 y 5.13, ya que presentan variaciones en las que no le da tiempo a la FSM de enganchar la fase correctamente.

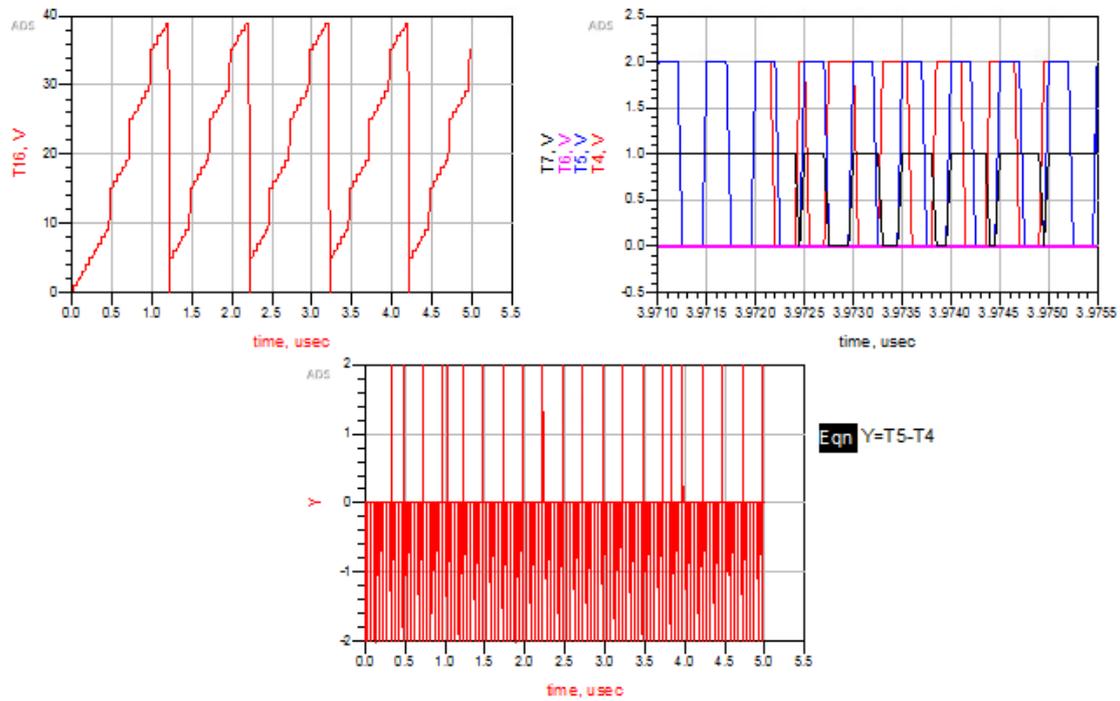


Figura 5.10: Offset de frecuencia 3.

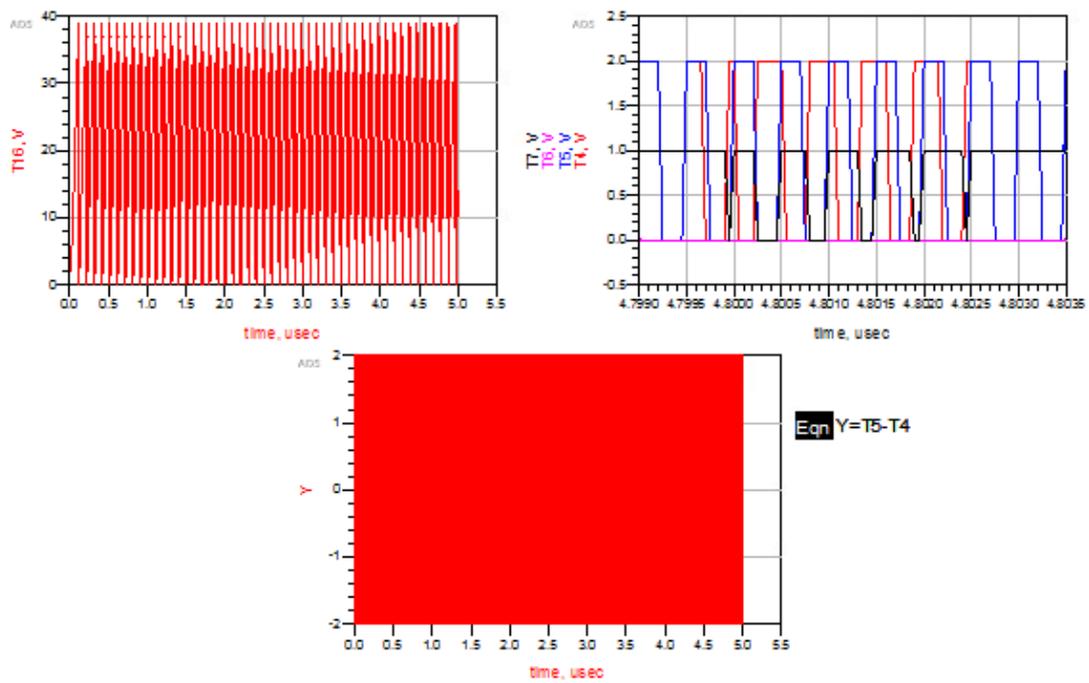


Figura 5.11: Offset de frecuencia 4.

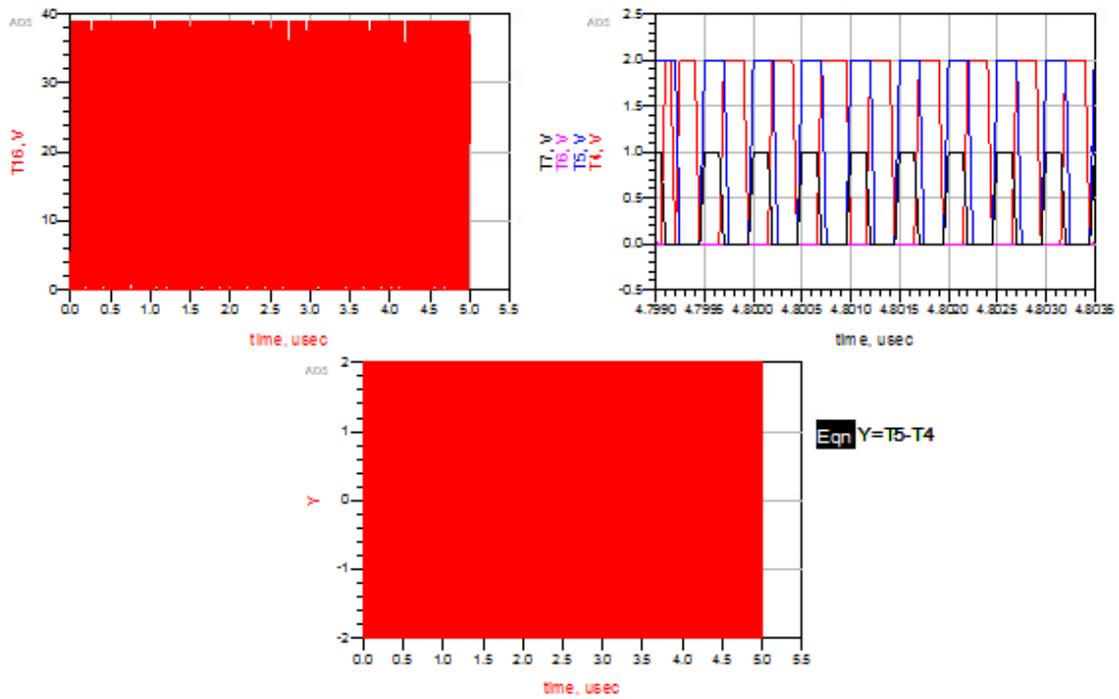


Figura 5.12: Offset de frecuencia 5.

Si la diferencia de frecuencia se da con factores mayores que 1, también se recupera la señal como se observa en la figura 5.13. Esto con la consideración de tener una pequeña variación entre las señales.

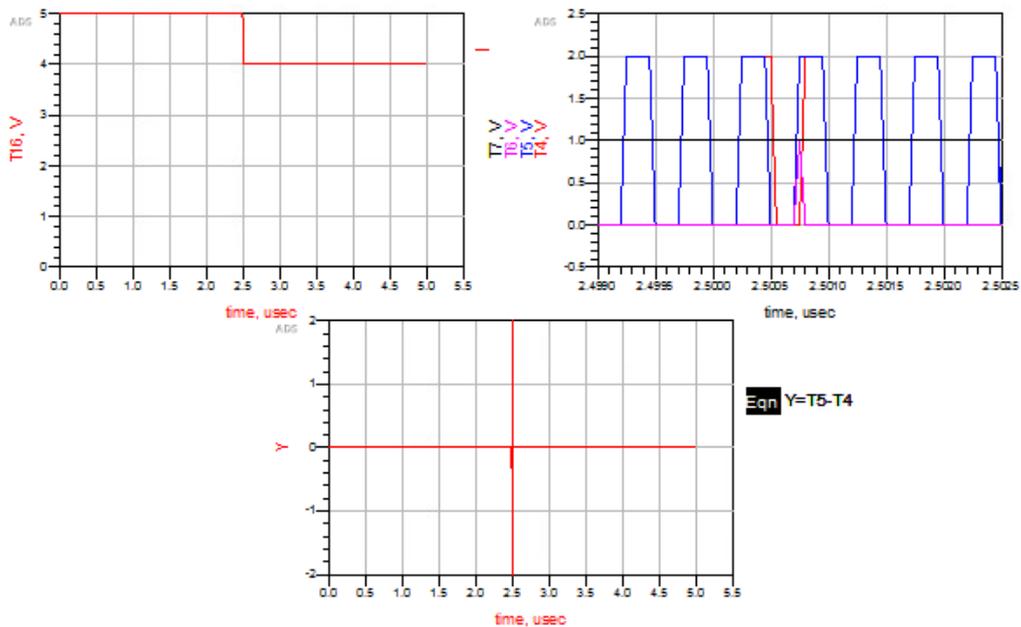


Figura 5.13: Offset de frecuencia 6.

5.4. Prueba 4

La última prueba consiste en el uso de un generador de secuencia binaria pseudo aleatoria (PRBS), en la entrada de datos del receptor. A diferencia de las pruebas anteriores, se trabaja con una señal más realista de los datos, tiene la mitad de la frecuencia de reloj y es una secuencia aleatoria de bits.

Al utilizar una diferencia de frecuencia de 20 kHz y un *skew* en el reloj como se observa en la figura 5.14, se recupera la señal correctamente. La FSM solo cambia un estado por la baja diferencia en frecuencia.

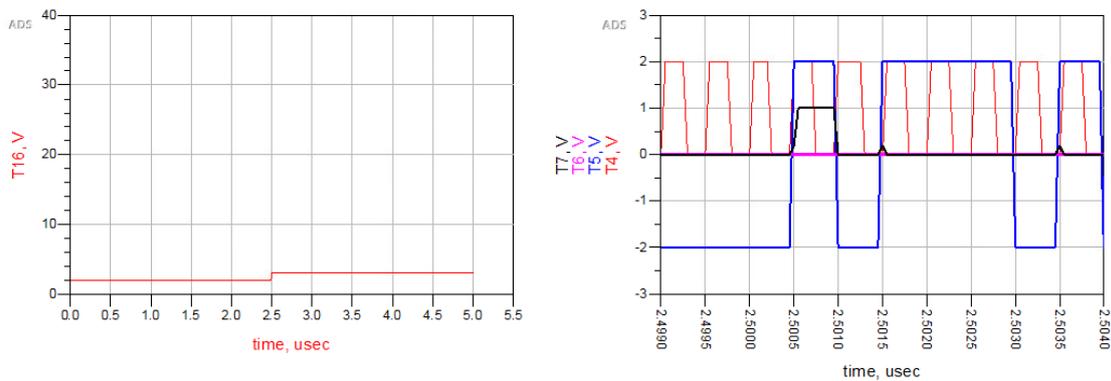


Figura 5.14: Uso de PRBS 1.

Conforme aumenta la diferencia de frecuencia a 200 kHz y 2 MHz, se logra recuperar la señal de reloj y datos, como se ve en las figuras 5.15 y 5.16.

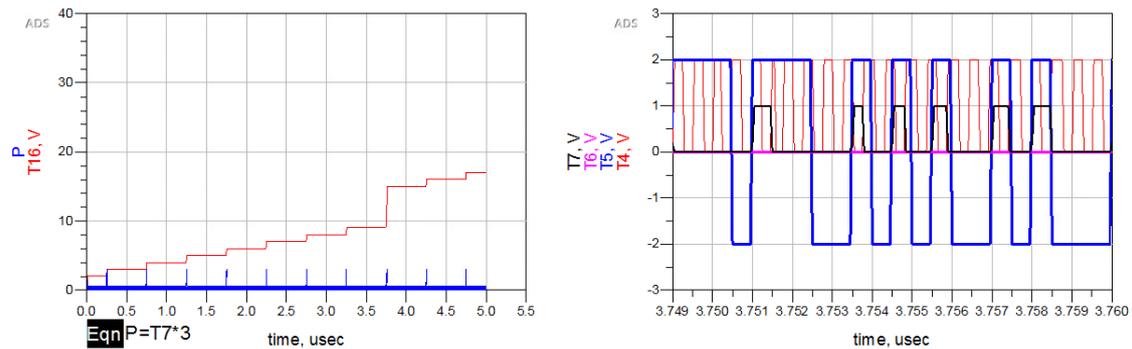


Figura 5.15: Uso de PRBS 2.

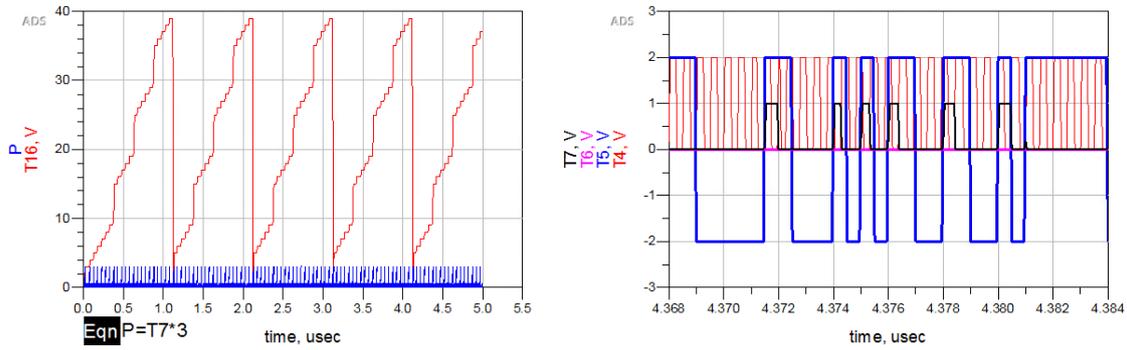


Figura 5.16: Uso de PRBS 3.

Diferencias de frecuencia superiores a 2 MHz no permiten recuperar de manera correcta la señal de reloj y datos, ya que la FSM no cuenta con el tiempo para corregir la fase, como se observa en la figura 5.17.

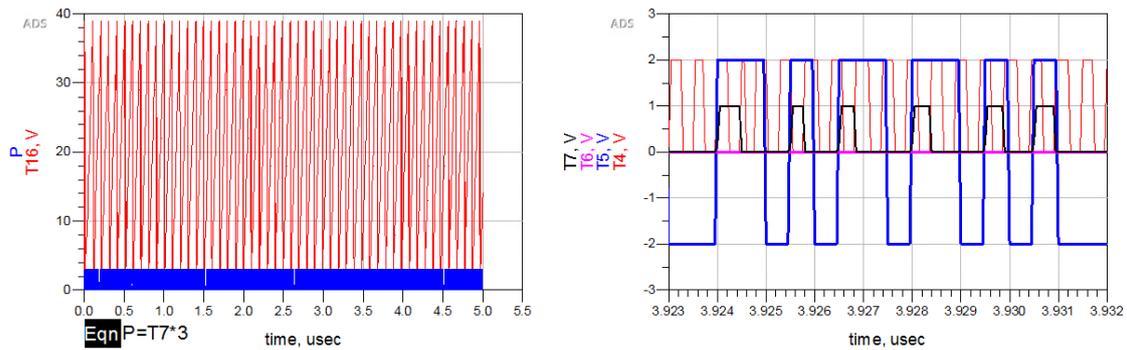


Figura 5.17: Uso de PRBS 4.

En el caso contrario en el que la variación de frecuencia es con un factor mayor a 1 se presenta el problema descrito anteriormente en el diseño del PFD, ya que la señal de PRBS al ser de menor frecuencia, provoca que el PFD solo cambie en una de sus salidas, por lo que el CDR generará cambios bruscos en los estados de la FSM, ya que no logra enganchar correctamente. Lo anterior se observa en las figuras 5.18, 5.19, 5.20 y 5.21, donde se aprecian los cambios bruscos en los estados de la FSM en la parte izquierda de las figuras.

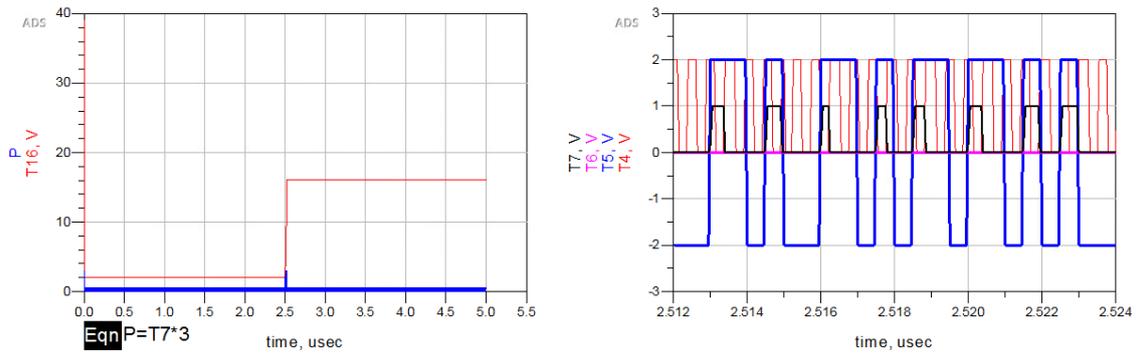


Figura 5.18: Uso de PRBS 5.

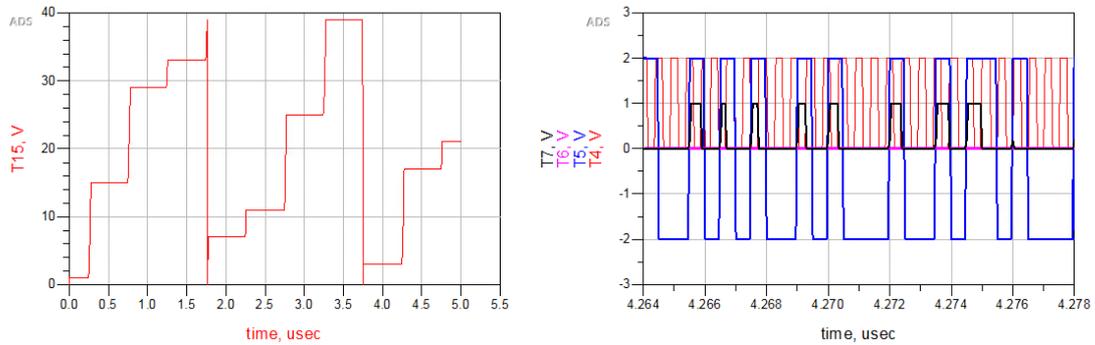


Figura 5.19: Uso de PRBS 6.

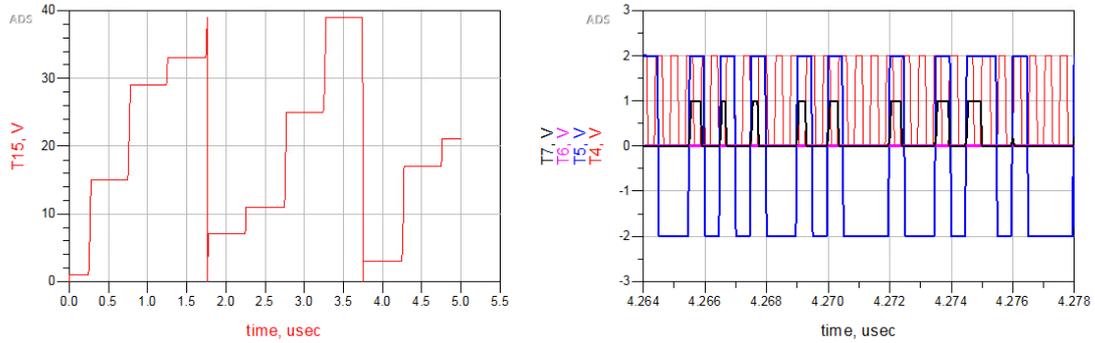


Figura 5.20: Uso de PRBS 7.

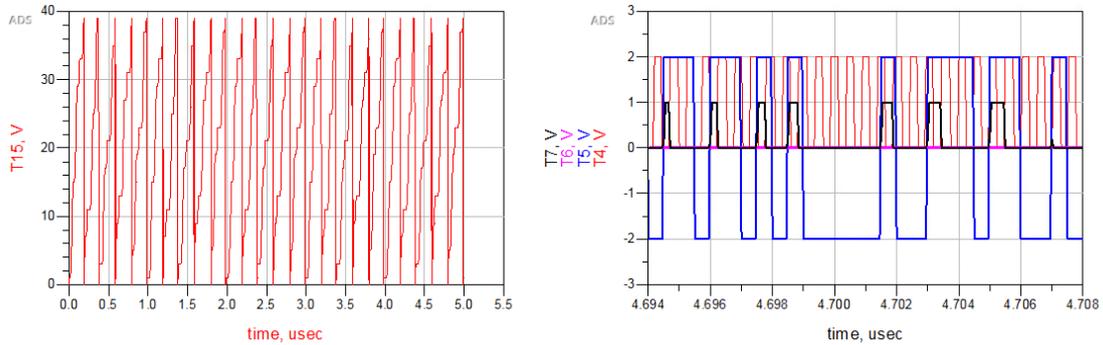


Figura 5.21: Uso de PRBS 8.

Otra manera de analizar esta limitación es mediante los diagrama de ojo con dos unidades de intervalo. La medición del diagrama se realiza en un *flip flop* tipo D a la salida, controlado por el reloj generado en el receptor. Al tener una diferencia de frecuencia de 2 kHz, 200 kHz y 20 MHz se obtienen los diagramas de ojo de las figuras 5.22, 5.23 y 5.24 respectivamente. Conforme aumenta la diferencia en frecuencia, el diagrama de ojo se "desplaza" a la izquierda lo que indica que el dato es recuperado a la salida, pero disminuye la apertura horizontal del ojo, por lo que se puede perder información al tener una diferencia considerable entre la frecuencia del receptor y los datos.

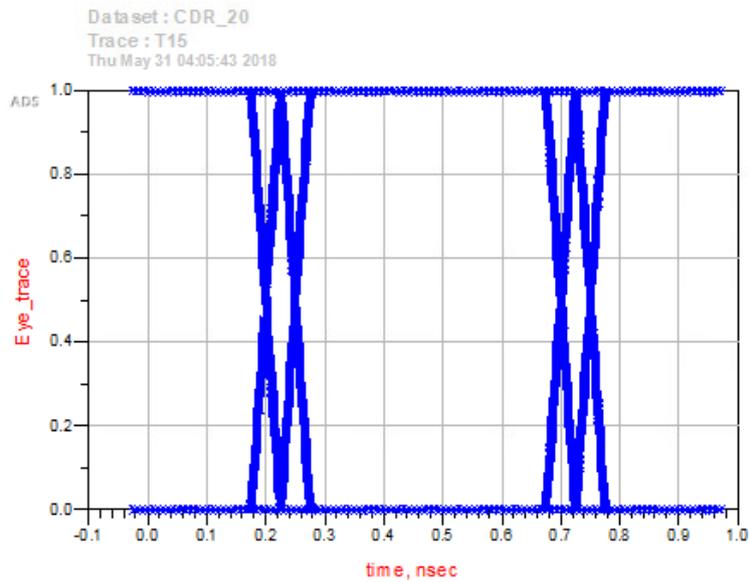


Figura 5.22: Diagrama de ojo para offset de frecuencia de 2 kHz.

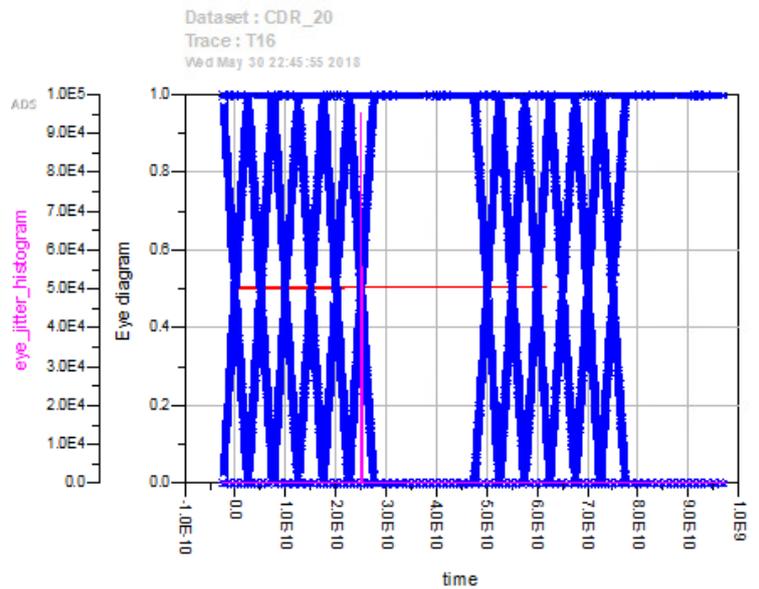


Figura 5.23: Diagrama de ojo para offset de frecuencia de 2 MHz.

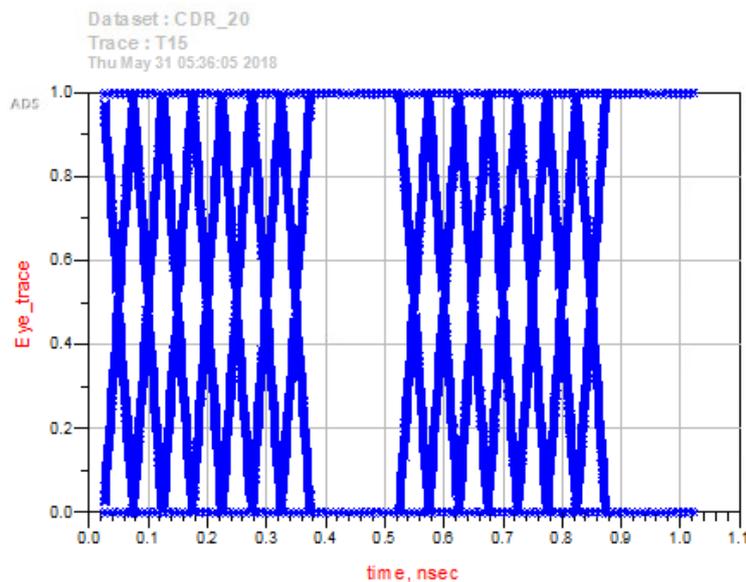


Figura 5.24: Diagrama de ojo para offset de frecuencia de 20MHz.

Por último, al obtener el diagrama de ojo para una diferencia de frecuencia con un factor mayor a uno, se obtiene el diagrama de ojo de la figura 5.25 que corresponde a una diferencia de frecuencia de 2 kHz. Como se observa el ojo se encuentra prácticamente cerrado ya que el CDR no logra enganchar en este caso por las limitaciones del PDF.

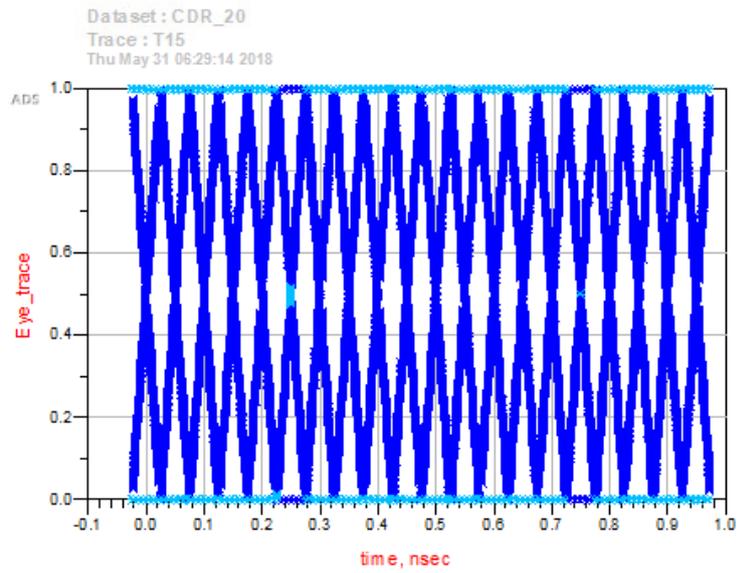


Figura 5.25: Diagrama de ojo para offset de frecuencia de 2 kHz.

6 Conclusiones y recomendaciones

6.1. Conclusiones

Se obtuvo un recuperador de reloj y datos para una arquitectura de reloj embebido a nivel de comportamiento en un entorno de simulación de señal mixta, garantizando la recuperación de una señal de datos en el orden de los Gb/s. La arquitectura de CDR por interpolador de fase funcionó correctamente de acuerdo a los resultados obtenidos.

Otro aspecto importante es la recuperación correcta de la señal ante los efectos del *skew* y *offset* de frecuencia, para el caso en el que el reloj generado en el receptor sea menor a la frecuencia de los datos provenientes del canal.

El uso del detector de fase y frecuencia propuesto no permite el correcto funcionamiento cuando la frecuencia de reloj en el receptor es mayor que la de los datos.

Es preferible diseñar y simular el CDR, con modelos de comportamiento propios ya que permite tener más control del diseño y de las simulaciones, por otro lado, se pueden reducir los tiempos de simulación si se implementa en herramientas de software matemáticas o lenguajes de programación de alto nivel.

6.2. Recomendaciones

- Generar un modelo de comportamiento para un interpolador de fase de manera que se ahorren las líneas de retardo.
- Utilizar otro tipo de comparador de fase y frecuencia que permita trabajar en frecuencias de reloj mayores que la de los datos provenientes del canal. Dentro de las propuestas se encuentra el detector Hogge y Alexander.
- Generar los modelos de comportamiento propios, ya que los de ADS no son modificables.
- Implementar un *Phased-locked loop* para generar una señal de reloj más realista.
- Implementar los modelos de comportamiento en una herramienta de software matemática o lenguaje de alto nivel para optimizar el diseño y tiempos de la simulación.

7 Bibliografía

- [1] S. Palermo, K. Emami-Neyestanak.A, and Horowitz.A, “A 90 nm cmos 16 gb/s transceiver for optical interconnects,” *IEEE journal of solid state circuits*, vol. 43, no. 5, pp. 1235–1246, May 2008. [Online]. Available: <https://ieeexplore.ieee.org/document/4494666/>
- [2] S.Palermo, *CMOS Nanoelectronics Analog and RF VLSI Circuits*. McGraw-Hill, 2011.
- [3] “Fiber optic basics,” Mar. 2018. [Online]. Available: <https://www.newport.com/t/fiber-optic-basics>
- [4] Bulzacchelli.J, Menolfi.C, and Beukema.T, “A 0.002-mm² 6.4-mw 10-gb/s full-rate direct dfe receiver with 59.6opening under 23.3-db channel loss at nyquist frequency,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 62, no. 12, pp. 3107 – 3117, Dec. 2014. [Online]. Available: <https://ieeexplore.ieee.org/document/6918554/>
- [5] S. H. Hall and L. Heck., *Advanced signal integrity for high speed digital designs*. Wiley, 2009.
- [6] Weste.N and Harris.D, *CMOS VLSI Design a Circuits and systems Circuits*. Addison-Wesley, 2011.
- [7] E. G. Friedman, “Clock distribution networks in synchronous digital integrated circuits,” *PROCEEDINGS OF THE IEEE*, vol. 89, no. 5, pp. 665–692, 1 2001.
- [8] H.-C. Lee, “An estimation approach to clock and data recovery,” Ph.D. dissertation, Stanford University, 11 2006.
- [9] K. Suk and X. Yuan, *High speed signaling: Jitter modeling, analysis and budgeting*. Prentice hall, 2012.
- [10] M. ta Hsieh and E.Sobelman, “Architectures for multi-gigabit wire-linked clock and data recovery,” *IEEE circuits and systems magazine*, vol. 1, no. 1, pp. 45–57, 1 2008.
- [11] C.Lidong, F.Spagna, and M.Phil, “A 90nm 1-4.25-gb/s multi data rate receiver for high speed serial links,” *Solid-State Circuits Conference*, vol. 14, no. 3, pp. 391–394, 1 2006.
- [12] B. Razavi, *RF microelectronics*. Prentice hall, 2011.
- [13] —, “Challenges in the design of high-speed clock and data recovery circuits,” *IEEE Communications Magazine*, vol. 1, no. 1, pp. 94–101, 8 2002.
- [14] “List of interface bit rates,” Mar. 2018. [Online]. Available: https://en.wikipedia.org/wiki/List_of_interface_bit_rates
- [15] Wei.L, Ji.J, and Liu.H, “A multi-rate serdes transceiver for ieee 1394b applications,” *Circuits and Systems (APCCAS)*, vol. 12, Jan. 2013. [Online]. Available: <https://ieeexplore.ieee.org/document/6419035/>
- [16] Bulzacchelli.J, Menolfi.C, and Beukema.T, “A 28-gb/s 4-tap ffe/15-tap dfe serial link transceiver in 32-nm soi cmos technology,” *IEEE Journal of Solid-State Circuits*, vol. 47, no. 12, pp. 3232 – 3248, Dec. 2012. [Online]. Available: <https://ieeexplore.ieee.org/document/6327618/>
- [17] B. Razavi, *Design or Integrated Circuits for Optical Communications*. Wiley, 2012.
- [18] *ADS Ptolemy Simulation*, 1st ed., Agilent, Santa Clara,USA, 2 2011.
- [19] “Phase interpolator,” May 2018. [Online]. Available: <http://iram.cs.berkeley.edu/serialio/cs254/interpolator/interp.html>
- [20] T. Ndjountche, *Digital Electronics : Finite-state Machines*. Wiley, 2016.