

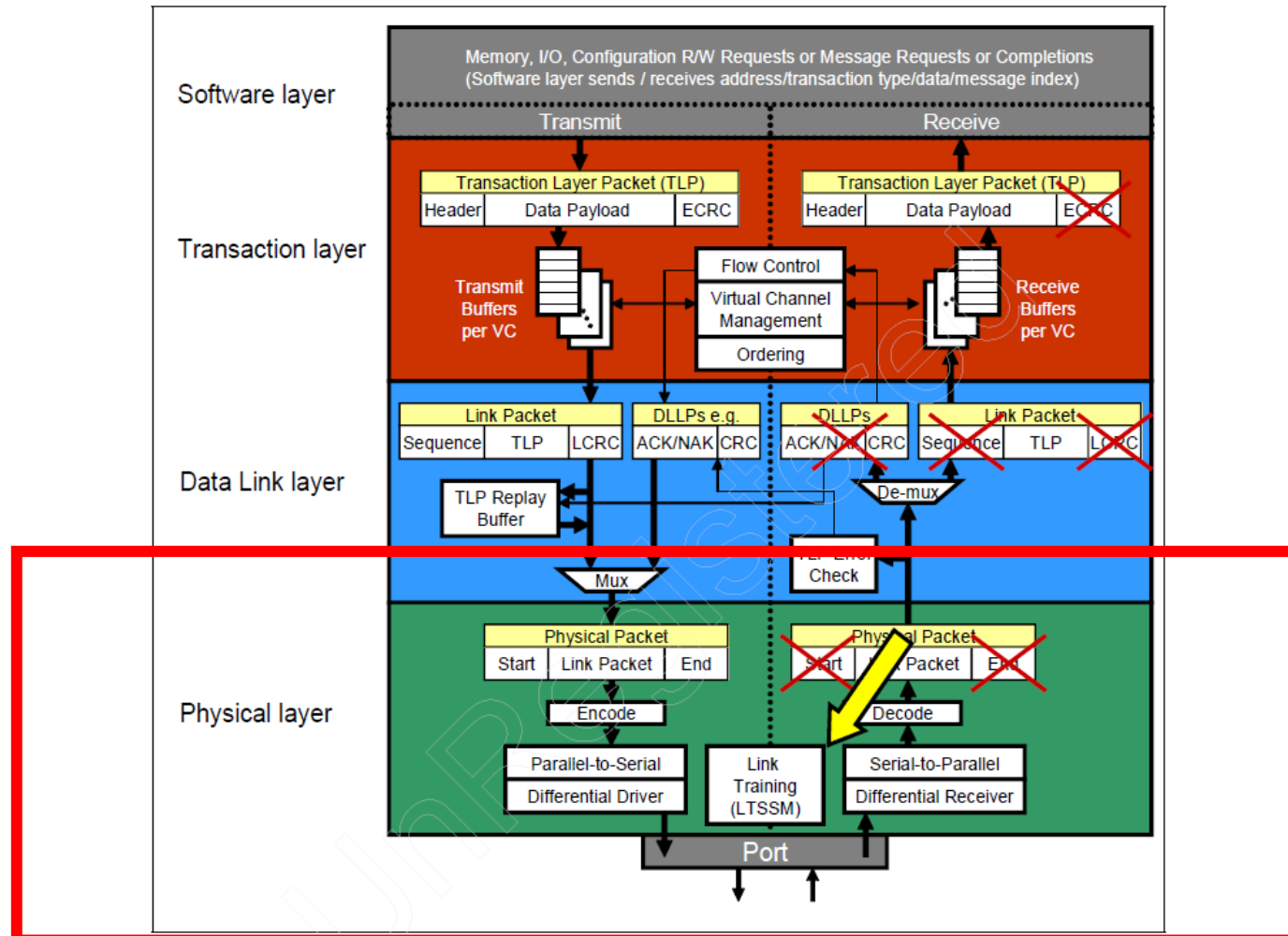
PHY PCIE

Prof. Jorge Soto

IE-0523 Circuitos Digitales II

Diagrama de capas PCIE

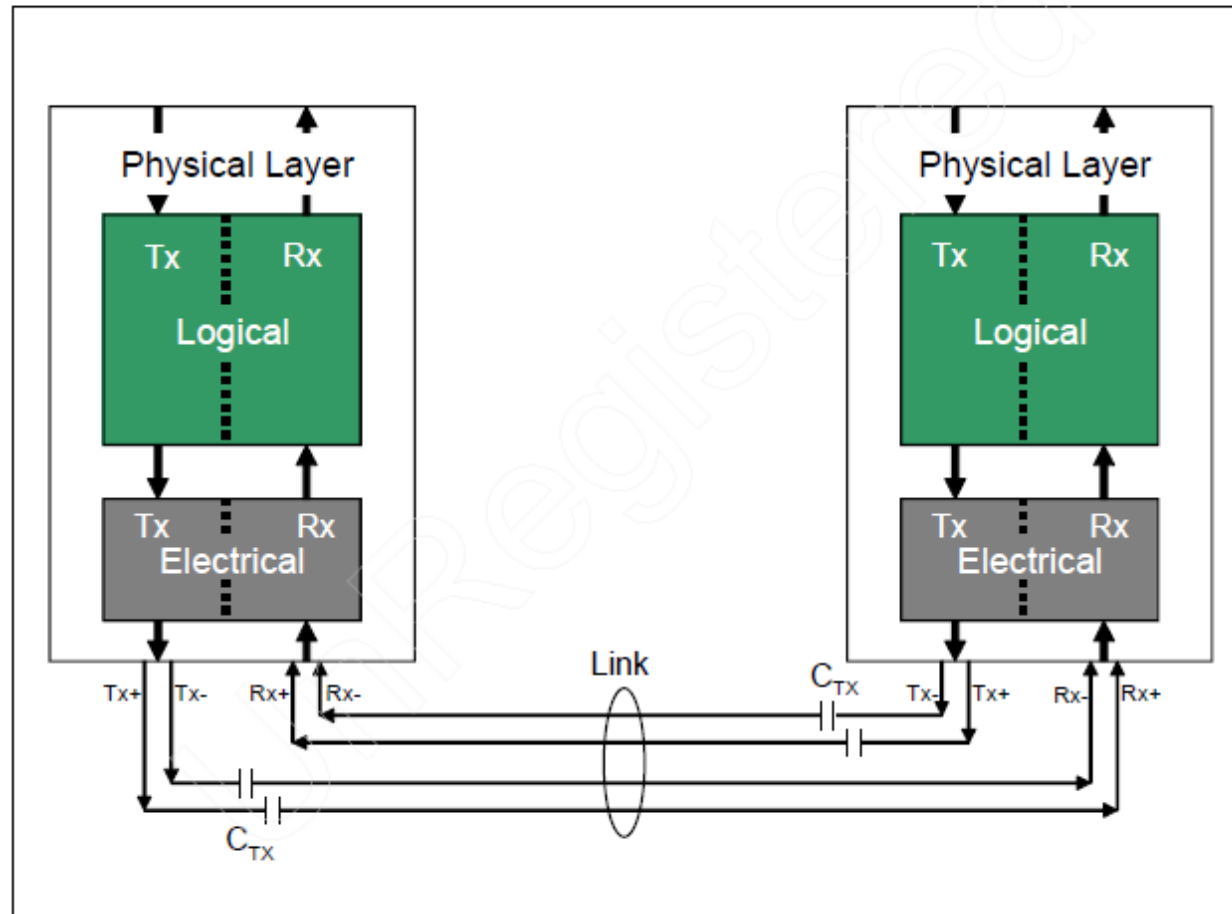
Figure 14-1: Link Training and Status State Machine Location



* PCI Express® System Architecture; Budruk, Anderson y Shanley; MindShare, Inc.; 2008

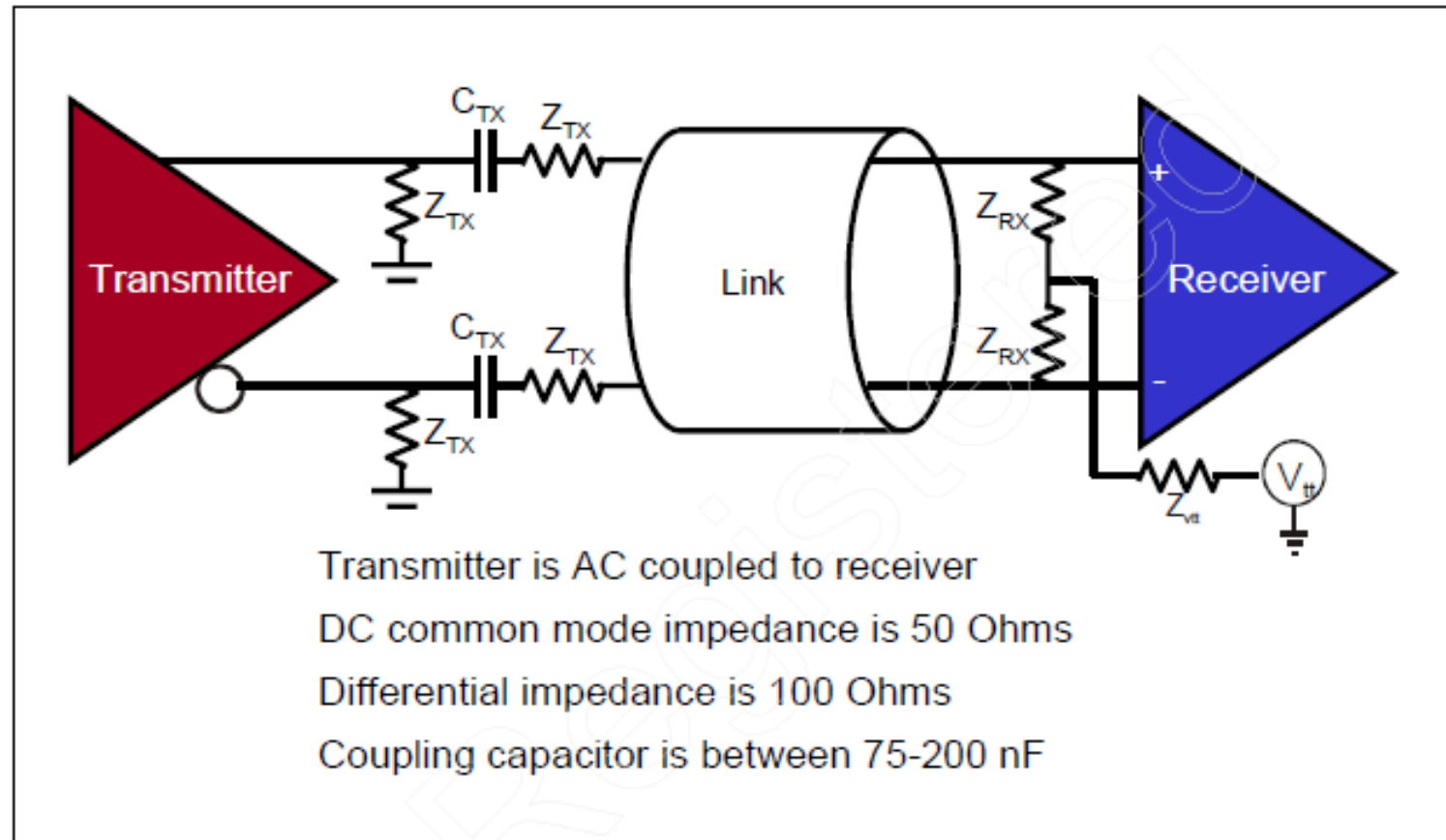
Enlace de dos capas físicas

Figure 11-2: Logical and Electrical Sub-Blocks of the Physical Layer



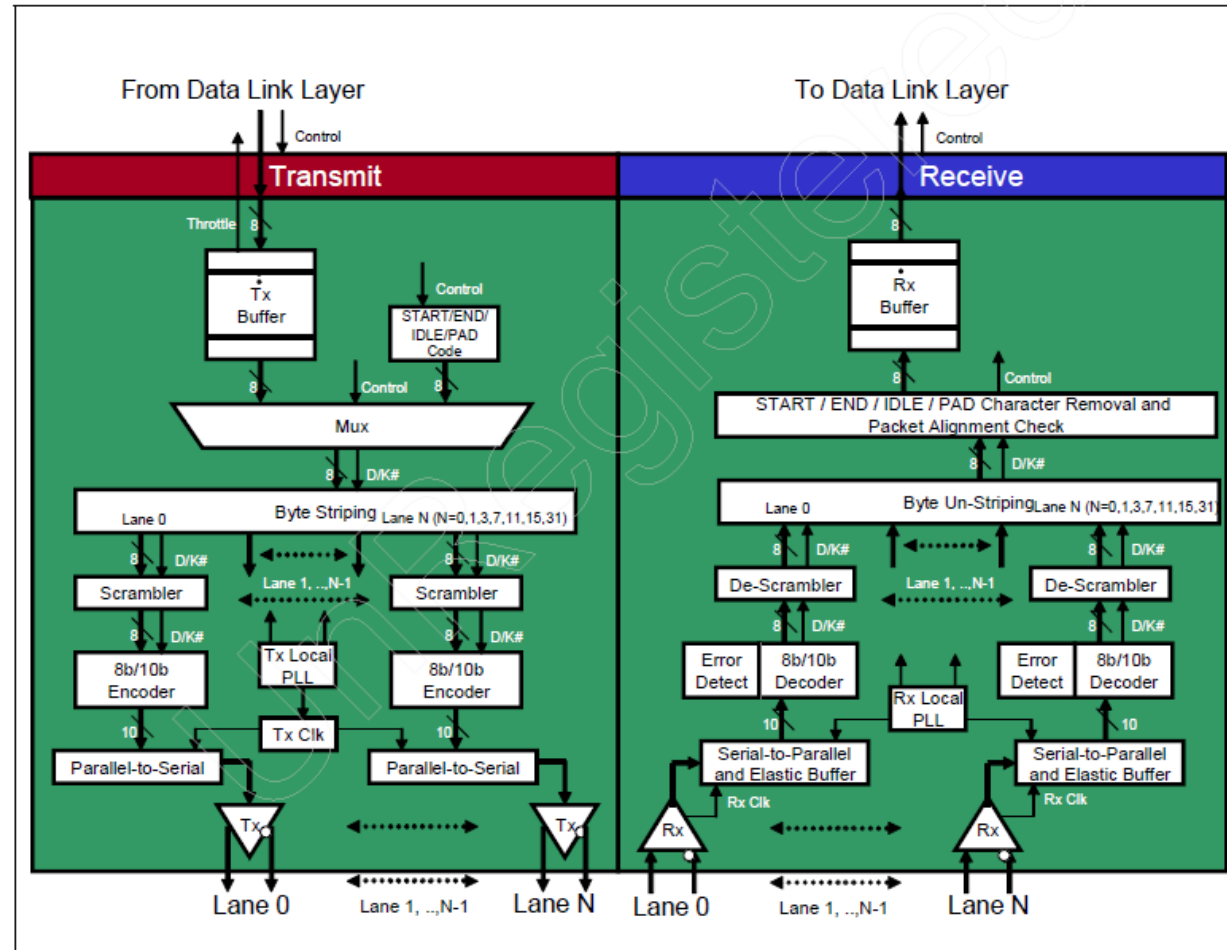
Capa física eléctrica

Figure 2-30: Electrical Physical Layer Showing Differential Transmitter and Receiver



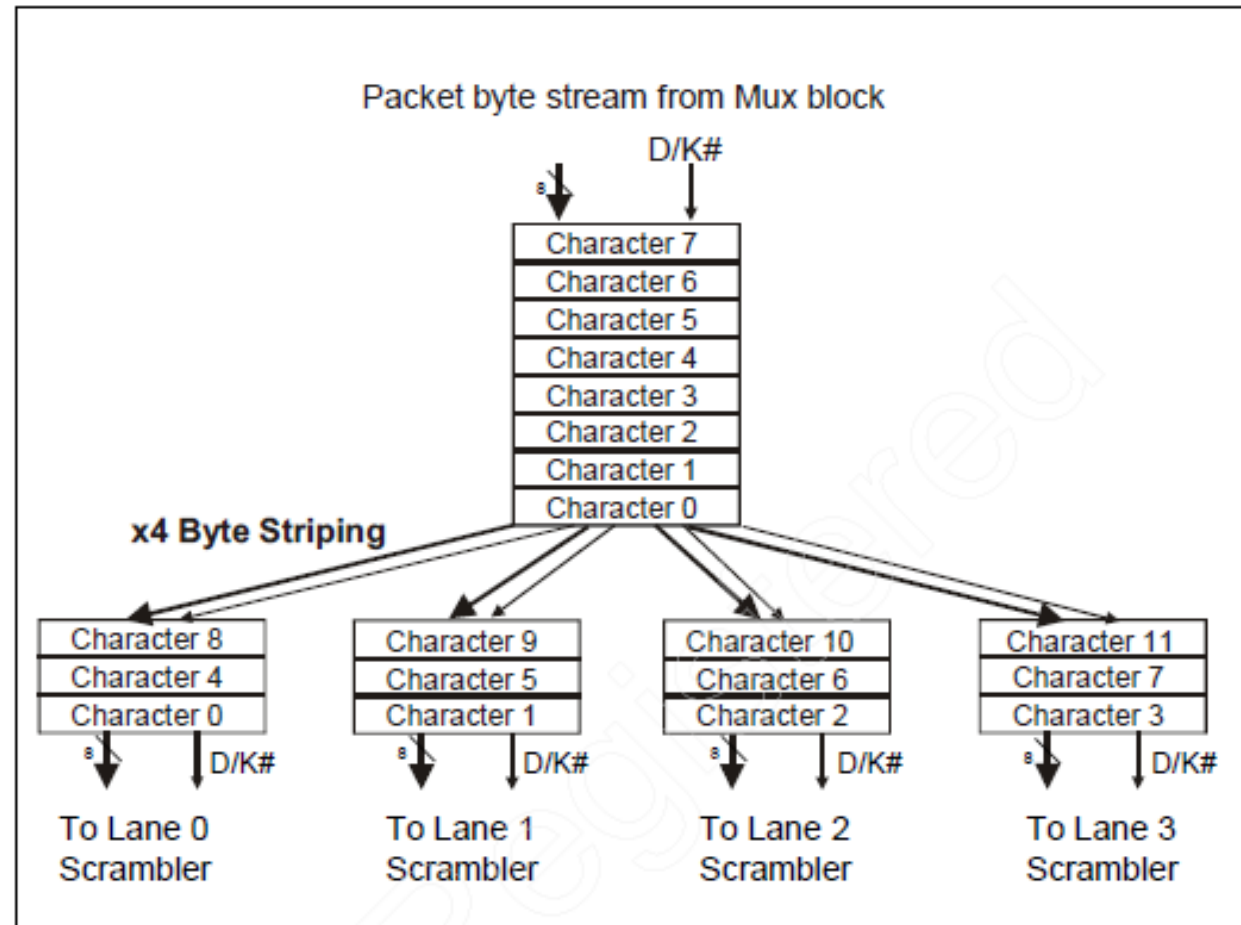
Detalles de la capa física en TX y RX

Figure 11-3: Physical Layer Details



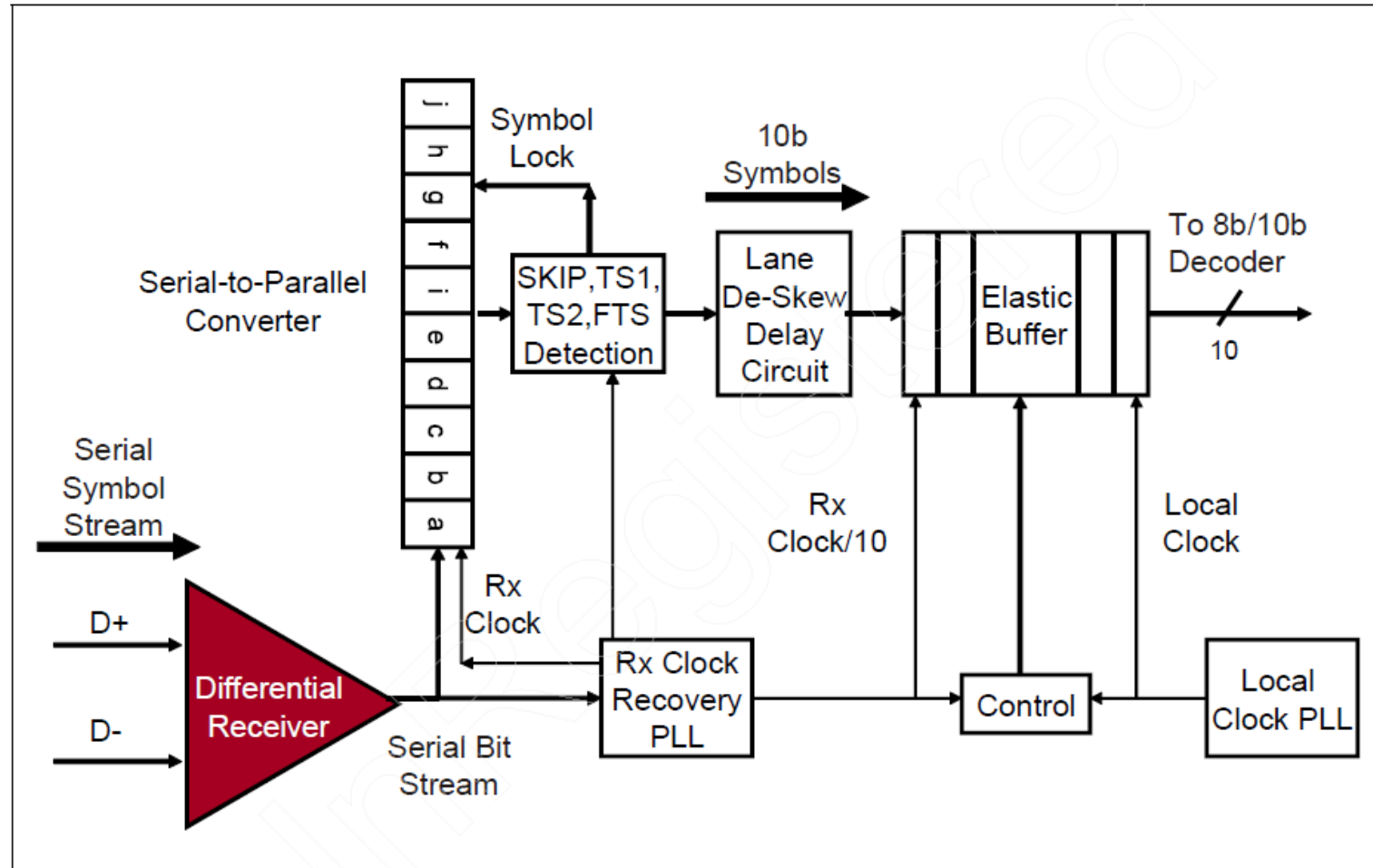
Byte Striping de datos hacia 4 lanes

Figure 11-8: x4 Byte Striping



Lógica RX

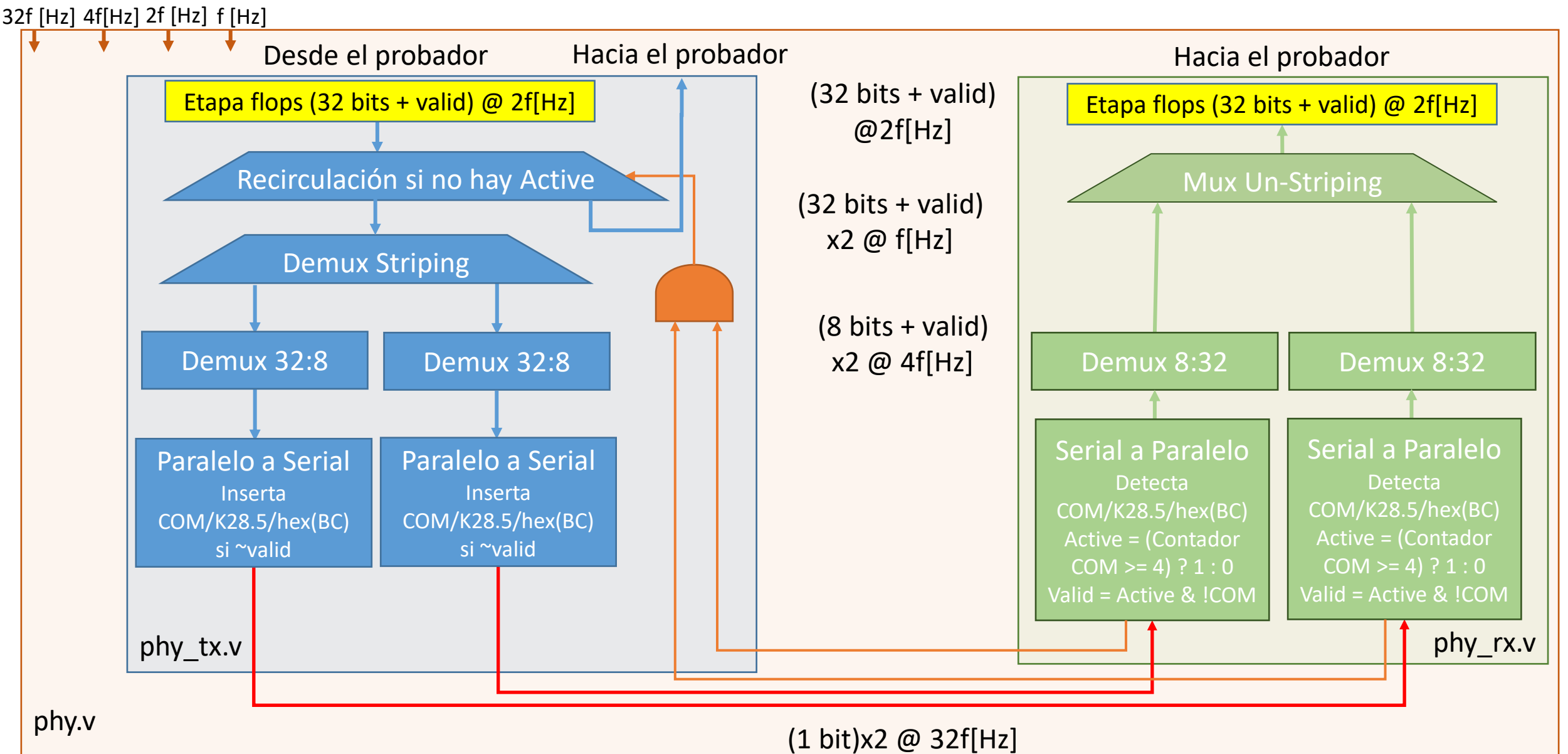
Figure 11-21: Receiver Logic's Front End Per Lane



Microarquitectura del proyecto #1

Lógica de control y multiplexores para la transmisión (TX) y recepción (RX) de datos en el PHY.

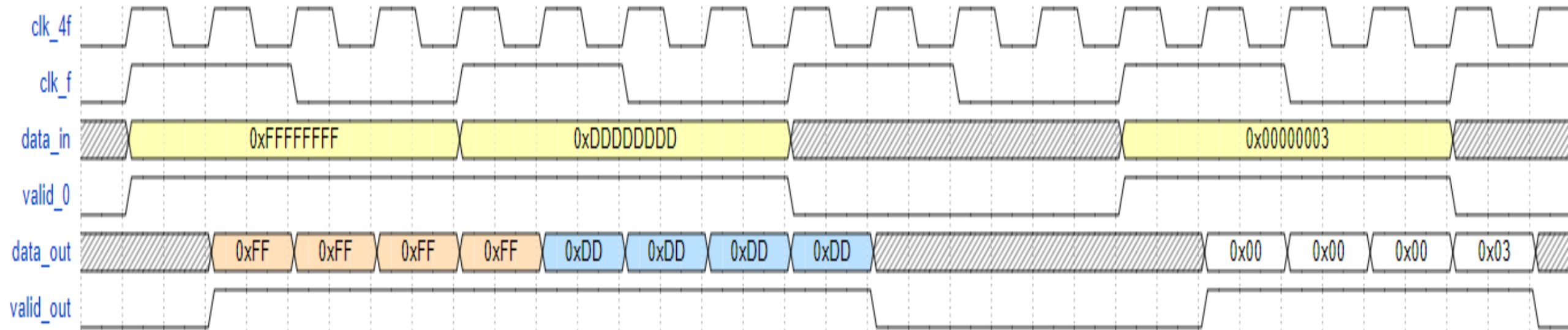
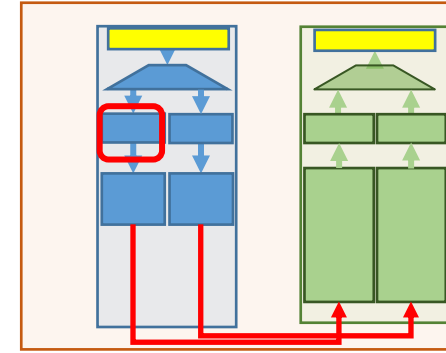
Detalles del PHY para el proyecto #1



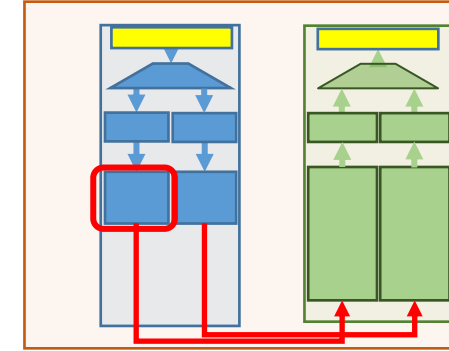
-
- Timing diagram for the 2-lane 10GbE SerDes receiver. The diagram shows the relationship between the clock (clk_2f), data input (data_in), and valid input signals (valid_in, valid_0, valid_1) and the corresponding lane data (lane_0, lane_1). The data is organized into 8-bit words. Lane 0 data is shown in yellow, and Lane 1 data is shown in orange. The diagram illustrates the sampling of data at the rising edge of the clock. The data_in signal is a 10GbE SerDes data stream, and the valid_in signal indicates when the data is valid. The lane_0 and lane_1 signals are the data received on each lane, and the valid_0 and valid_1 signals indicate when the data is valid on each lane.
- | Signal | Value |
|---------|--|
| data_in | 0xFF...FF, 0xEE...EE, 0xDD...DD, 0xCC...CC, 0x00...03, 0x00...04, 0xAA...AA, 0x99...99, 0x00...07, 0x00...08 |
| lane_0 | 0xFFFFFFFF, 0xDDDDDDDD, 0x00000003, 0xAAAAAAAA, 0x00000007 |
| lane_1 | 0xEEEEEEEE, 0CCCCCCCC, 0x00000004, 0x99999999, 0x00000008 |

Detalles del Demux 32:8

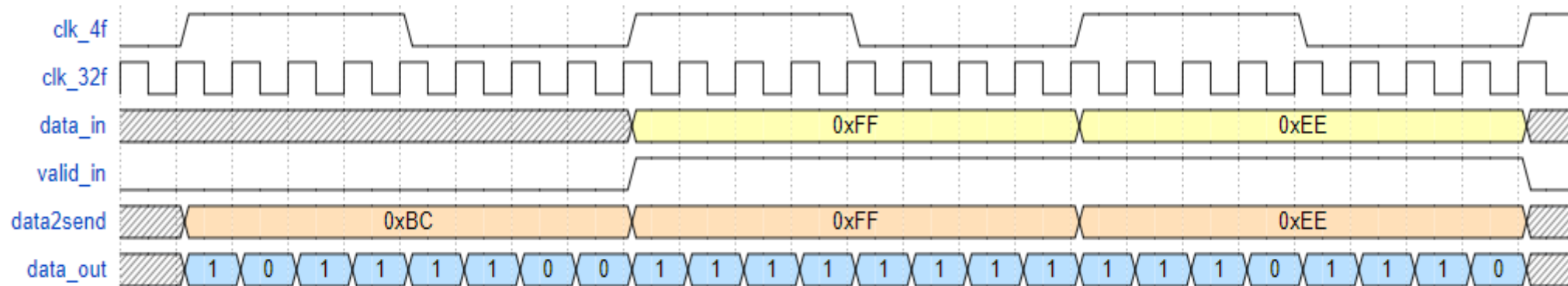
- Reduce tamaño del bus y aumenta la frecuencia 4 veces
- Prepara el bus para el paralelo a serial
- Asume que el bus completo de datos es válido
- Pueden haber retrasos en las señales de salida implementadas, siempre que sean consistentes para toda la lógica diseñada.



Detalles del paralelo-serial

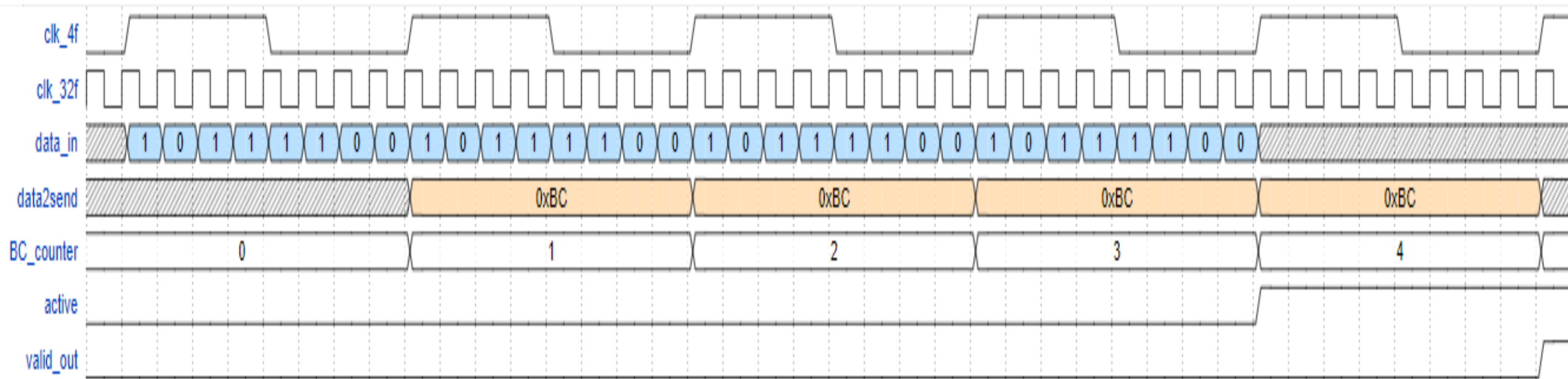
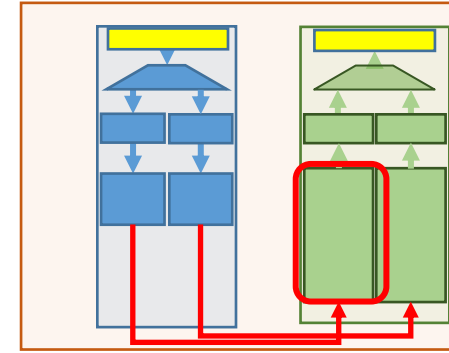


- Convierte el bus paralelo con reloj `clk_4f` a una señal serial con reloj `clk_32f`.
- Envía la señal paralela `hex(BC)` cuando `valid` está en bajo.
- No hay señal `valid` de salida, todos los datos diferentes de `hex(BC)` son válidos.
- Por simplificación del proyecto, no hay codificador 8b/10b y no se puede enviar `hex(BC)` como dato válidos.



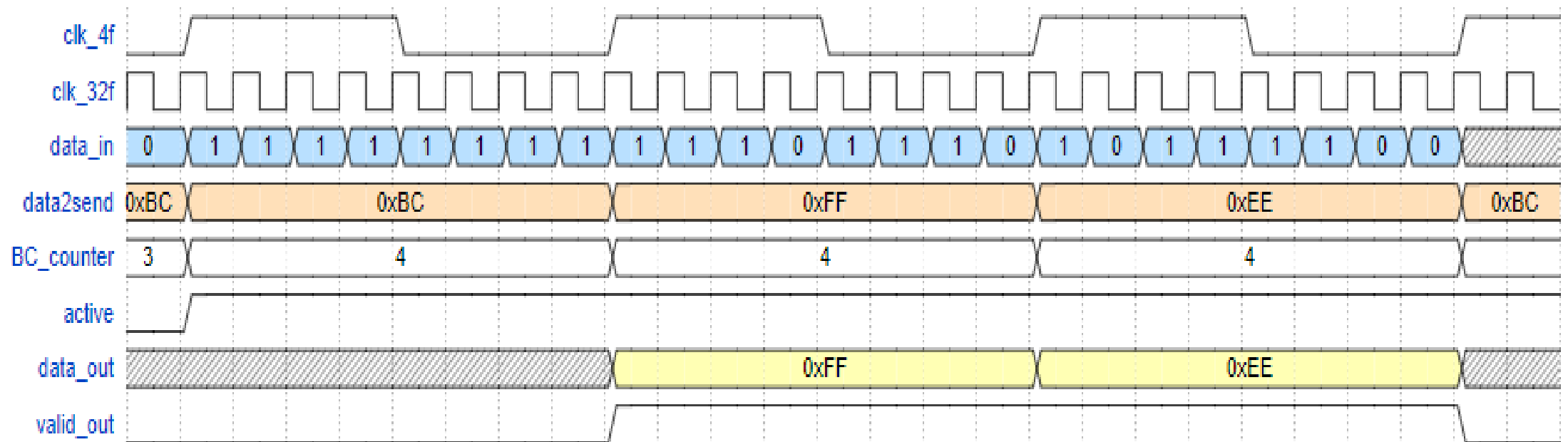
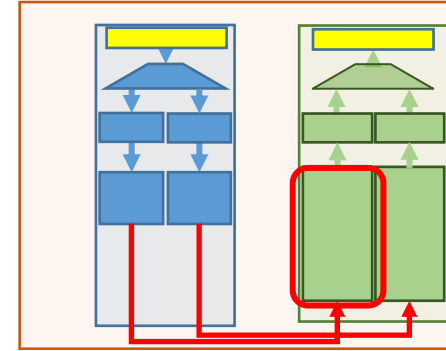
Detalles del serial-paralelo

- Se necesita enviar 4 hex(BC) para activar el receptor.
- Al enviar 4 hex(BC), se habilita la señal active y ésta se mantiene arriba.
- La señal valid_out estará arriba siempre que active esté arriba y no se tenga un hex(BC) en los datos.



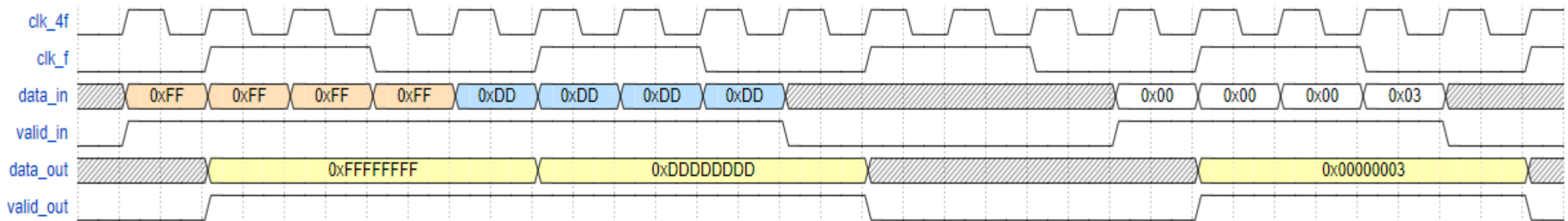
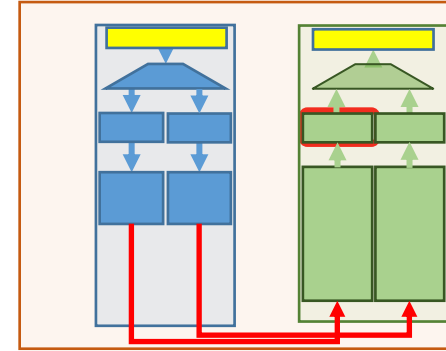
Detalles del serial-paralelo

- Envío de datos
- Si se recibe un hex(BC), se baja la señal de valid.
- Los datos estarán disponibles en la salida en el siguiente ciclo de clk_4f.



Detalles del Mux 8:32

- Aumenta el tamaño del bus y reduce la frecuencia 4 veces
- Prepara el bus para el byte striping
- Asume que el bus completo de datos de válido
- Pueden haber retrasos en las señales de salida implementadas, siempre que sean consistentes para toda la lógica diseñada.



DATA IN

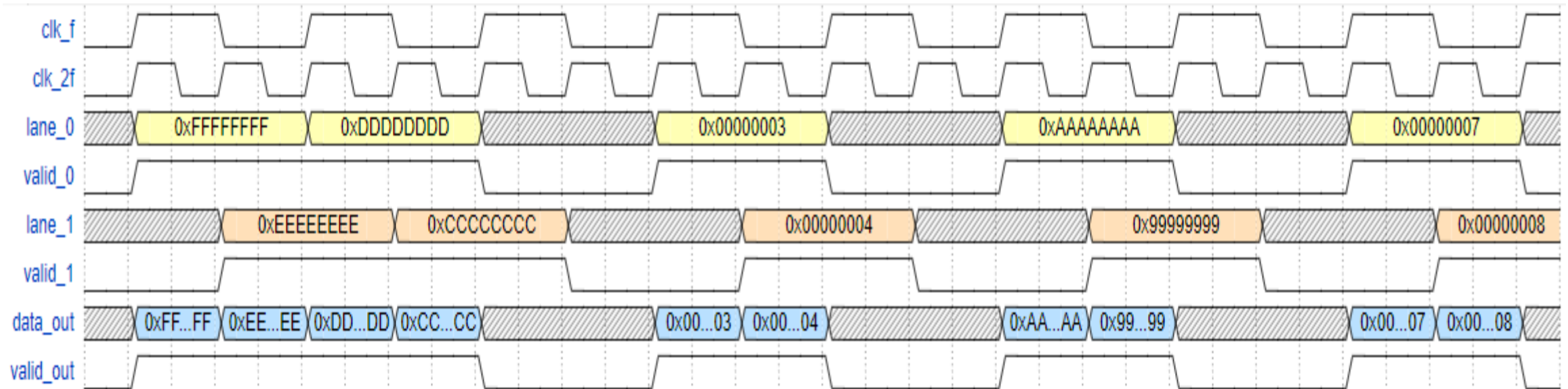
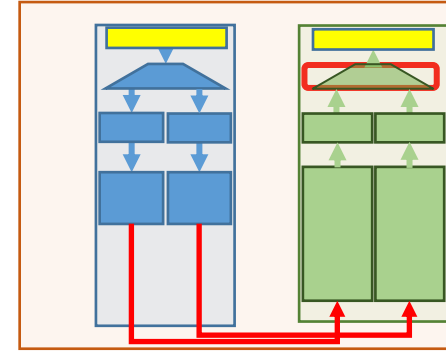
0xFF 0xFF 0xFF 0xFF 0xDD 0xDD 0xDD 0xDD

DATA OUT

0xFFXXXXXX 0xFFFFXXXX 0xFFFFFXX 0xFFFFFFFF 0xDDFFFFFF 0xDDDDFFFF 0xDDDDDDFF 0xDDDDDDDD
0XXXXXXXXX 0XXXXXXXXX 0XXXXXXXXX 0FFFFFFF 0FFFFFFF 0FFFFFFF 0FFFFFFF 0DDDDDDDD

Detalles del Mux un-stripping

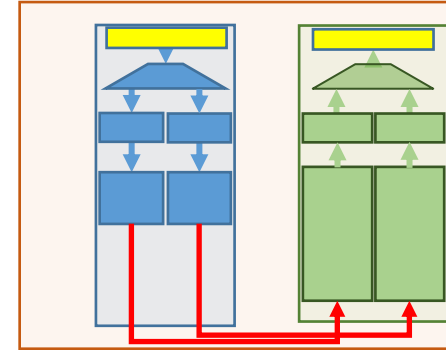
- Recopilación de datos de ambos carriles a uno único.
- Se toman siempre primero los datos del lane 0.
- Para no perder datos válidos, la frecuencia de salida de los datos debe ser el doble de la frecuencia de entrada.



Símbolos de control

Table 11-5: Control Character Encoding and Definition

Character Name	8b Name	10b (CRD-)	10b (CRD+)	Description
COM	K28.5 (BCh)	001111 1010	110000 0101	First character in any Ordered-Set. Detected by receiver and used to achieve symbol lock during TS1/TS2 Ordered-Set reception at receiver
PAD	K23.7 (F7h)	111010 1000	000101 0111	Packet Padding character
SKP	K28.0 (1Ch)	001111 0100	110000 1011	Used in SKIP Ordered-Set. This Ordered-Set is used for Clock Tolerance Compensation
STP	K27.7 (FBh)	110110 1000	001001 0111	Start of TLP character
SDP	K28.2 (5Ch)	001111 0101	110000 1010	Start of DLLP character
END	K29.7 (FDh)	101110 1000	010001 0111	End of Good Packet character
EDB	K30.7 (FEh)	011110 1000	100001 0111	Character used to mark the end of a 'nullified' TLP.



Símbolos de control

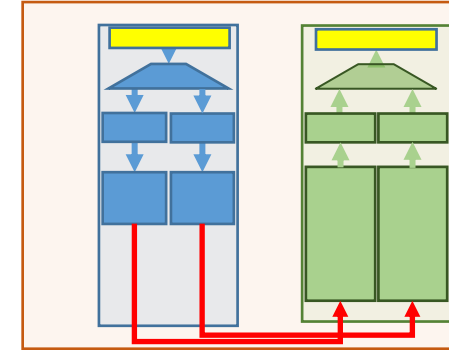


Table 11-5: Control Character Encoding and Definition

Character Name	8b Name	10b (CRD-)	10b (CRD+)	Description
FTS	K28.1 (3Ch)	001111 1001	110000 0110	Used in FTS Ordered-Set. This Ordered-Set used to exit from L0s low power state to L0
IDL	K28.3 (7Ch)	001111 0011	110000 1100	Used in Electrical Idle Ordered-Set. This Ordered-Set used to place Link in Electrical Idle state

Proyecto #1

- Ver detalles de evaluación y entregables en “Especificaciones Proyecto 1”.

