MPC – SP Synchronization Procedure

Petersburg Nuclear Physics Institute / University of Florida
Revision 1.1
October 10, 2001

Introduction

The distance between on-chamber electronics (FEBs, TMBs, MPC) and the counting room (SP) is about 100 m. The MPC uses optical links to transmit 32 bits of data per link every 25 ns. Each MPC contains parallel to serial converters and optical transmitter modules, while each SP contains optical receiver modules and serial to parallel converters. Texas Instrument's TLK2501 1.6 to 2.5 Gbps Transceiver [1] serves as a serializer/deserializer circuit and Finisar's FTRJ-8519-1-2.5 2-Gigabit Short-Wavelength 2x5 Pin Small Form Factor (SFF) Transceiver [2] serves as an optical transmitter/receiver at both ends of the link. To transmit and receive 32 bits of data @ 40 MHz, while the TLK2501 is basically a 16-bit device, the data are multiplexed at the transmitter end @ 80 MHz and demultiplexed at the receiver end. Demultiplexers are implemented in Front FPGAs. For correct operation the SP should lock its deserializers to the link frequency, and its demultiplexers to the first frame of the event, Table 1. This note describes the initialization and synchronization procedure for the TLK2501 transmitters and FPGA demultiplexers to make this happen.

Table 1. Transmit Data Format

		15	14	13	12	11	08	07	06	00
Frame	1	Quality			L/R	CLCT Pattern #		CLCT Pattern ID		
	2	BC1	BC0	SE	VP	CSC ID		AM	Wire Group ID	

The TLK2501 Details

The transmitter latches 16-bit parallel data at a rate based on the supplied reference clock (GTX_CLK). The 16-bit parallel data is internally encoded into 20 bits using an 8-bit/10-bit (8B/10B) encoding format. The resulting 20-bit word is then transmitted differentially at 20 times the reference clock (GTX_CLK) rate. The receiver section performs the serial-to-parallel conversion on the input data, synchronizing the resulting 20-bit wide parallel data to the extracted reference clock (RX_CLK). It then decodes the 20-bit wide data using 8-bit/10-bit decoding format resulting in 16 bits of parallel data at the receive data terminals (RXD0-15). The outcome is an effective data payload of 1.28 Gbps to 2.0 Gbps (16-bit data times the GTX_CLK frequency).

The encoding is dependant upon two additional input signals, the TX_EN and TX_ER. When the TX_EN is HIGH and the TX_ER is LOW then the data bits TXD[0:15] are encoded and transmitted normally. When the TX_EN is LOW, and TX_ER is HIGH, then the encoder generates a carrier extend consisting of two K23.7 (F7F7) codes. Each

valid Transmission Character of the 8B/10B Transmission Code is given a name using the following convention: cxx.y [3], where c is used to show whether the Transmission Character is a Data Character (c is set to D) or an Extended Character (c is set to K). If the TX_EN and the TX_ER are both HIGH, then the encoder generates a K30.7 (FEFE) code. Table 2 provides the transmit data control decoding. Since the data is transmitted in 20-bit serial words, K codes indicating carrier extend and transmit error propagation are transmitted as two 10-bit K-codes. The encoder inserts the IDLE character set when no payload data is available to be sent. IDLE consists of a K28.5 (BC) code and either a D5.6 (C5) or a D16.2 (50) character. IEEE802.3z defines the K28.5 character as a pattern consisting of 0011111010 (a negative number beginning disparity) with the 7 MSBs (0011111) referred to as the comma character. Since data is latched into the TLK2501 16 bits at a time, this in turn is converted into two 10-bit codes that are transmitted sequentially. This means IDLE consists of two 10-bit codes, 20 bits wide that are transmitted during a single GTX_CLK cycle.

Table 2. Transmit Data Controls

TX_EN	TX_ER	ENCODED 20-BIT DATA
Low	Low	Idle (< K28.5, D5.6>, < K28.5, D16.2>)
Low	High	Carrier extend (K23.7, K23.7)
High	Low	Normal data character (DX.Y)
High	High	Transmit error propagation (K30.7, K30.7)

When parallel data is clocked into a parallel to serial converter, the byte boundary that was associated with the parallel data is now lost in the serialization of the data. When the serial data is received and converted to parallel format again, a way is needed to recognize the byte boundary. Generally this is accomplished through the use of a synchronization pattern. This is generally a unique pattern of 1s and 0s that either cannot occur as part of valid data or is a pattern that repeats at defined intervals. 8-bit/10-bit encoding contains a character called the comma (b0011111 or b1100000), which is used by the comma detect circuit on the TLK2501 to align the received serial data back to its original byte boundary. The decoder detects the K28.5 comma, generating a synchronization signal aligning the data to their 10-bit boundaries for decoding. It then converts the data back into 8-bit data, removing the control words. The output from the two decoders is latched into the 16-bit register synchronized to the recovered parallel data clock (RX_CLK) and output valid on the rising edge of the RX_CLK. Two output signals, RX_DV/LOS and RX_ER, are generated along with the decoded 16-bit data output on the RXD[0:15] terminals. The output status signals are asserted as shown in Table 3.

Table 3. Receive Status Signals

RECEIVED 20-BIT DATA	RX_DV/LOS	RX_ER
Idle (< K28.5, D5.6>, < K28.5, D16.2>)	Low	Low
Carrier extend (K23.7, K23.7)	Low	High
Normal data character (DX.Y)	High	Low
Receive error propagation (K30.7, K30.7)	High	High

When the TLK2501 decodes normal data and outputs the data on RXD[0:15], RX_DV/LOS is HIGH and RX_ER is LOW. When the TLK2501 decodes a K23.7 code

(F7F7) indicating carrier extend, RX_DV/LOS is LOW and RX_ER is HIGH. If the decoded data is not a valid 8-bit/10-bit code, an error is reported by setting HIGH of both RX_DV/LOS and RX_ER. If the error was due to an error propagation code, the RXD bits outputs hex FEFE. If the error was due to an invalid pattern, the data output on RXD is undefined. When the TLK2501 decodes an IDLE code, both RX_DV/LOS and RX_ER are LOW and a K28.5 (BC) code followed by either a D5.6 (C5) or D16.2 (50) code are output on the RXD terminals.

The TLK2501 has a synchronization-state machine, which is responsible for handling link initialization and synchronization. Upon power up or reset, the state machine enters the acquisition (ACQ) state and searches for IDLE. Upon receiving three consecutive IDLEs or a carrier extend, the state machine enters the synchronization (SYNC) state. If, during the acquisition process, the state machine receives valid data or an error propagation code, it immediately transitions to the SYNC state. The SYNC state is the state for normal device transmission and reception.

The Demux Details

The demultiplexers are implemented in Front FPGAs. They are intended to restore the initial 32-bit @ 40MHz pattern structure from the received 16-bit @ 80MHz data stream. The frame content is specified in Table 1. Note that there is no “frame bit” available, which could have helped to distinguish between frames. A procedure is required to achieve frame synchronization in the SP.

Synchronization Procedure

The easiest way to synchronize TLK2501 receivers is to put corresponding TLK2501 transmitters into IDLE mode for at least three consecutive periods of GTX_CLK. A 2-step procedure below restores link synchronization between MPC and SP and simultaneously recovers the correct frame sequence in SP:

- Step 1.
  - The MPC receives a broadcast reset command and responds by setting its TLK2501 transmitter into IDLE mode (both TX_EN and TX_ER are LOW).
  - The TLK2501 receiver switches into IDLE mode after the signal arrives to the SP.
  - The broadcast command may be either a separate TTC command, any of the already specified reset commands, or any combination of the above => to be agreed upon.
  - Time position of the TTC command with respect to the orbit timing should be at least five bunch crossings before bunch crossing zero (BC0) to let enough time for TLK2501 receivers to synchronize.

- Step 2.
  - The MPC resumes normal operation (TX_EN goes HIGH and TX_ER stays LOW) just before transmitting the first frame of bunch crossing zero (BC0, FR0), see Figure 1.
  - The TLK2501 receiver resumes normal operation after the signal has reached the SP, see Figure 2.

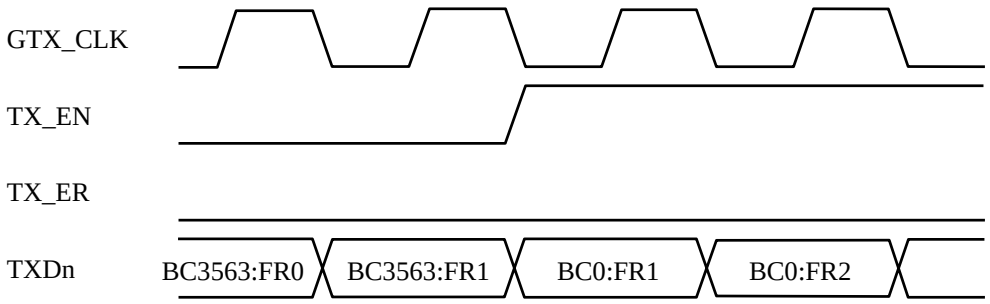

Figure 1. Transmit Side Timing Diagram

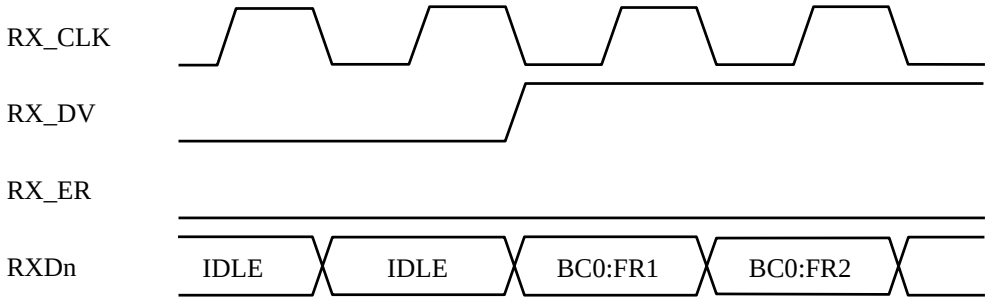

Figure 2. Receive Side Timing Diagram

Another nice feature of the above procedure is that it facilitates almost “automatic” alignment of MPC data at the SP inputs. Since reset command is a broadcast command, all MPCs get it simultaneously. All input SP links eventually switch to IDLE mode and then back to normal operation. By monitoring the RX_DV signals the SP easily aligns all input data to the latest one, see Figure 3.

The alignment FIFOs are implemented in Front FPGAs. The FIFOs are reset on the same broadcast command (reset not shown) and wait for valid data from each receiver. Starting from BC0 the data are written into alignment FIFOs synchronously with the individual receive clocks. After all the inputs start getting valid data (all RX_DV are HIGH), the AND of RX_DV is synchronized with the SP clock and enables the FIFO reads. All the data are coming out from the FIFOs aligned to the latest input.

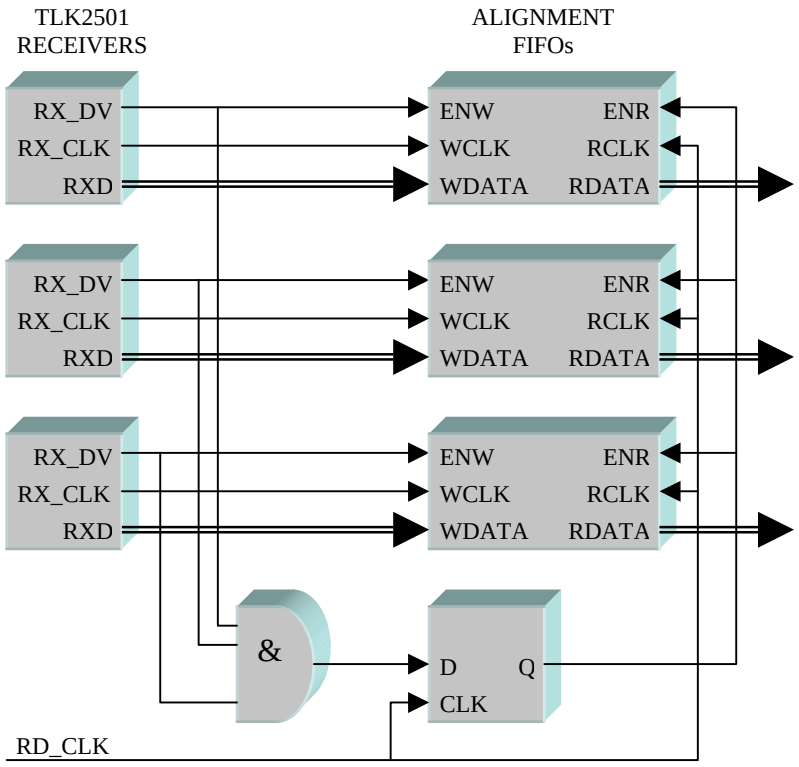

Figure 3. Simplified Alignment Schematics.

References

[1] TLK2501 1.6 to 2.5 GBPS Transceiver Datasheet available at:
http://red.pnpi.spb.ru/~uvarov/parts/TLK2501.pdf
[2] 2 Gigabit Short-Wavelength 2x5 Pin SFF Transceiver
FTRJ-8519-1-2.5 Product Specification available at:
http://red.pnpi.spb.ru/~uvarov/parts/FTRJ-8519-1-2dot5_Spec_Rev_D.pdf
[3] Fibre Channel Physical and Signaling Interface (FC-PH), Revision 4.3, page 63 (pdf-98), available at http://www.nowhere.net/~raster/FC/fcph_43.pdf

Revision History

Date	Version	Comment
October 3, 2001	1.0	Initial Release
October 10, 2001	1.1	Reference [3] has been added