Frequently Asked Questions About the CYP(V)15G0403DXB Device

The following are Frequently Asked Questions (FAQs) by customers who are evaluating CYP(V)15G0403DXB devices. The CYP(V)15G0403DXB is a member of Cypress's High-Speed Frequency Agile HOTLink II™ product family. Within the device, all four channels can simultaneously operate at different data rates and transmit different types of data. The only difference between the CY**P**15G0403DXB and the CY**V**15G0403DXB devices is that the latter satisfies SMPTE 259M and SMPTE 292M pathological test requirements per SMPTE EG 34-1999.

These cursory answers will serve as an introduction for each topic. Separate application notes cover some of these topics in more detail. All the information in this application note will apply to the CYP(V)15G0403DXB device. The letter "x" following a signal name (e.g., TXRATEx) represents a particular, but arbitrary, channel (A, B, C, D) of the device.

1. What is the HOTLink II family of devices?

The HOTLink II family of devices is the second generation of the Cypress HOTLink® technology. This multichannel, frequency agile family of transceivers is capable of providing point-to-point or point-to-multipoint connectivity through high speed serial links at data rates from 195 Mbps to 1500 Mbps. Each transceiver channel has separate receive and transmit channels. The receive channel has a Clock and Data Recovery (CDR) PLL, deserializer, framer, optional 10B/8B decoder, and optional elasticity buffer. The transmit channel has a Clock Multiplier PLL, phase-align buffer, optional 8B/10B encoder, and serializer.

2. What are the different devices under the HOTLink II family? What are the major differences between each of them?

The devices in the HOTLink II family and the differences between them are shown in *Table 1*. The column "Number of Channels" indicates the number of transceiver channels available.

Secondary serial inputs and redundant outputs are present in some devices in the family. The secondary inputs are an additional set of serial input buffers that can be alternatively selected as a serial input to a particular receive channel. The redundant outputs are an additional set of serial outputs that can be used to output data from a particular transmit channel.

CYP(V)15G0403DXB has the unique capability of running each channel at an independent data rate. This device therefore needs four different reference clocks, one per channel.

Table 1. Major Differences Between Different Devices in HOTLink II Family

Device	# Channels	8B/10B Encoding and 10B/8B Decoding	Second Pair of Serial I/Os Per Channel	Independent Clocking for Each Channel	Package Type
CYP(V)15G0403DXB	4	Available	Available	Available	256-ball BGA
CYP(V)15G0401DXB	4	Available	Available	Not available	256-ball BGA
CYP(V)15G0402DXB	4	Not available	Not available	Not available	256-ball BGA
CYP(V)15G0201DXB	2	Available	Available	Not available	196-ball FBGA
CYP(V)15G0101DXB	1	Available	Available	Not available	100-ball BGA

3. What is the frequency range of operation for the CYP(V)15G0403DXB device? Is there any setting that needs to be changed for different operating frequencies? Are there any external loop filter components that need to be changed depending on the operating frequency?

The CYP(V)15G0403DXB device can operate at any serial data rate between 195 Mbps and 1500 Mbps. Hence, the corresponding character clock rate range is 19.5 MHz–150 MHz. The only setting that needs to change depending on the operating frequency and value of TXRATEx is SPDSELx. It must be LOW for serial data rates from 195 Mbps to 400 Mbps, MID (open) for 400 Mbps to 800 Mbps, and HIGH for 800 Mbps to 1500 Mbps. Note that the REFCLKx frequency will be either divide-by-10 or divide-by-20 of the serial data rate depending on the setting of TXRATEx. Refer to *Table 2* for more details.

Table 2. SPDSELx and TXRATEx Settings Depending on Serial Signaling Rate and REFCLKx Frequency

SPDSELx	TXRATEx	REFCLKx Frequency (MHz)	TXCLKOx Output Frequency (MHz)	Serial Signaling Rate (Mbps)
LOW	1	Reserved	19.5–40	195–400
	0	19.5–40		
MID (Open)	1	20–40	40–80	400–800
	0	40–80		
HIGH	1	40–75	80–150	800–1500
	0	80–150		

The CYP(V)15G0403DXB device does not require any external loop filter components. Hence, the user does not need to worry about them.

4. What are the major differences between HOTLink (first-generation) and HOTLink II devices?

The major differences between first-generation HOTLink and second-generation HOTLink II devices are shown in *Table 3*.

Table 3. Differences Between First-Generation HOTLink and HOTLink II

Feature	HOTLink (First-Generation)	HOTLink II
Operating Range	50 Mbps to 400 Mbps	195 Mbps to 1500 Mbps
Integration	CY7C924ADX, CY7C954ADX, CY7C9689A are transceivers that have integrated transmitters and receivers in the same IC.	Transmit and receive channels integrated in the same device
Maximum # Transceiver Channels/Device	One per transmitter and receiver pair	Four transceiver channels present in CYP(V)15G0401DXB CYP(V)15G0402DXB, and CYP(V)15G0403DXB
Power Supply Voltage	5V	3.3V
Framing Character	K28.5	K28.5 or Comma
Framing Modes	Single Byte Framer and multi-byte framer available.	Selectable as Single Byte, Cypress-mode multi-byte or Alternate mode multi-byte
Internal Phase-Align buffer	Transmit FIFO present in CY7C924ADX, CY7C954ADX and CY7C9689A.	Phase-align buffer is present in each transmit channel to absorb the phase differences between REFCLKx and TXCLKx.
Receive Elasticity Buffer	Receive FIFO in CY7C924ADX, CY7C954ADX and CY7C9689A	Receive Elasticity Buffer present to allow flexible clocking of parallel receive data using a clock that is neither phase coherent nor frequency coherent with the recovered character rate clock.

5. Does the CYP(V)15G0403DXB device support serial data rates that are lower than 195 Mbps?

No, the CYP(V)15G0403DXB device has not been characterized and tested at data rates below 195 Mbps. The first-generation HOTLink devices (Eg. CY7B923, CY7B933) operate at data rates below 195 Mbps.

6. What standards are supported by the CYP(V)15G0403DXB device?

Fibre Channel (1x), Gigabit Ethernet (GbE), ESCON®, DVB-ASI, SMPTE 292M, SMPTE 259M (Levels C and D), custom backplanes and custom point-to-point links are all supported by the device. Each channel in the device can be configured for a different standard and/or data rate.

7. What is the relationship between TXCLKOx, the internal character rate clock and REFCLKx? What is the purpose of the signal called TXRATEx?

TXCLKOx is a buffered version of the internal character rate clock for the given channel. The internal character rate clock is the clock signal that has a divide-by-10 frequency of the internal bit rate clock for the associated channel. The internal bit rate clock is generated by the TXPLL of that channel to serialize the output data. It is 10× the REFCLKx frequency if TXRATEx = 0 and 20× the REFCLKx frequency if TXRATEx = 1. This allows the capability to use a lower (half-rate) frequency clock source as the reference clock for any channel, as shown in *Table 2*.

TXCLKOx can be used as the source clock for the upstream logic device that generates the parallel data input to the associated transmit channel, in order to enable synchronous data transfer.

8. What is the function of the phase-align buffer in the transmit path of the CYP(V)15G0403DXB device? Can the phase-align buffer be bypassed?

The function of the phase-align buffer is to absorb any phase difference between the transmit input clock for a given channel (TXCLKx) and the internal character rate clock for that channel. The buffer is bypassed whenever TXRATEx = 0 and TXCKSELx = 1. In other words, it is bypassed when full rate REFCLK is used to clock the parallel data into the device.

9. What should be done in order to ensure that phase differences between the input TXCLKx and the internal character rate clock are absorbed?

The phase-align buffer reset latch, $\overline{\text{PABRSTx}}$, should be rewritten with a ‘0’ after the device is configured and data/clock are presented at the inputs. For the exact sequence to initialize the device, refer to *Question 21.*

10. What are the different clocking modes for latching parallel data into the parallel input registers (TXDx[7:0] and TXCTx[1:0])?

Parallel data (TXDx[7:0] and TXCTx[1:0]) can be latched into the input registers of the CYP(V)15G0403DXB device using one of the following ways:

1. TXCLKx of the given channel (TXCKSELx = 0).

2. REFCLKx (TXCKSELx = 1).

Regardless of the clock being used to latch the parallel data into the device, the encoded data word is serialized using the bit rate clock generated by the TXPLL of the given channel. The bit rate clock for a given channel is either 10× the REFCLKx frequency or 20× the REFCLKx frequency, depending on the setting of TXRATEx. Also, the clock that is being used to latch the data into the input register of any channel should be synchronous in frequency with the REFCLKx of that channel. For example, if TXCLKx of a channel is used to latch parallel data into the corresponding input registers, TXCLKx should be synchronous with REFCLKx. All phase differences between REFCLKx and the clock that is being used to latch the parallel data will be absorbed by the phase-align buffer of the given channel. The phase-align buffer is initialized by the phase-align reset latch ($\overline{\text{PABRST}}$).

The various schemes by which REFCLK can be made synchronous with the clock selected to latch the parallel data into the input registers are shown in *Figure 1*, *Figure 2* and *Figure 3*. The schemes shown in *Figure 1* and *Figure 2* use TXCLKx in each channel to clock the data into each transmit channel input register. The TXCLKx in each channel can be made synchronous (0 ppm offset) with REFCLK as shown in *Figure 1* or *Figure 2*. In *Figure 1*, TXCLKO, a buffered version of the internal character rate clock, is fed as the clock source for the upstream device. This will ensure that TXCLKx is synchronous with the internal character rate clock. Any phase difference between TXCLKx and the internal character rate clock will be absorbed by the phase-align buffer, once it is initialized by providing a phase-align reset. In *Figure 2*, REFCLK and the clock input to the upstream device originate from the same source, which might be a global clock distribution system. In *Figure 3*, the clock output of the upstream device itself is used as REFCLK. Please refer to *Question 11* and *Question 12* for the potential issues that might be encountered when using this scheme.

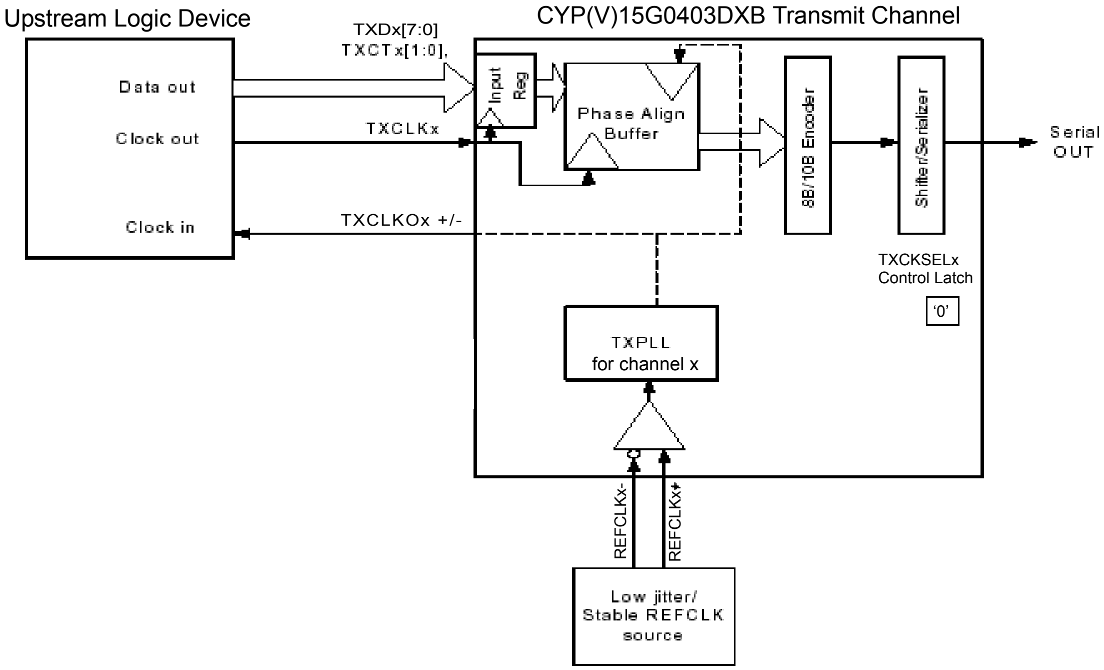


Figure 1. Clocking Scheme Using TXCLKx to Clock the Transmit Parallel Data into the CYP(V)15G0403DXB Transmit Channel

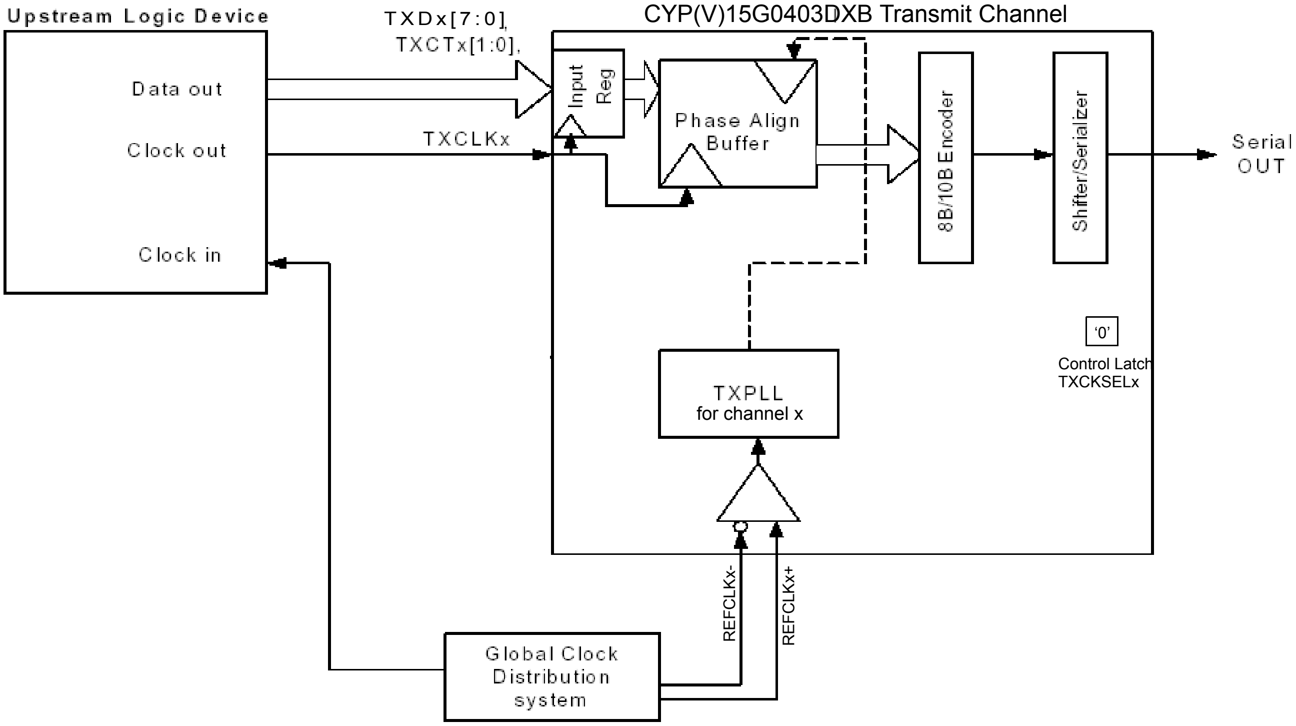


Figure 2. Variation of Scheme 1 (Without Using TXCLKOx as Input to an Upstream Device)

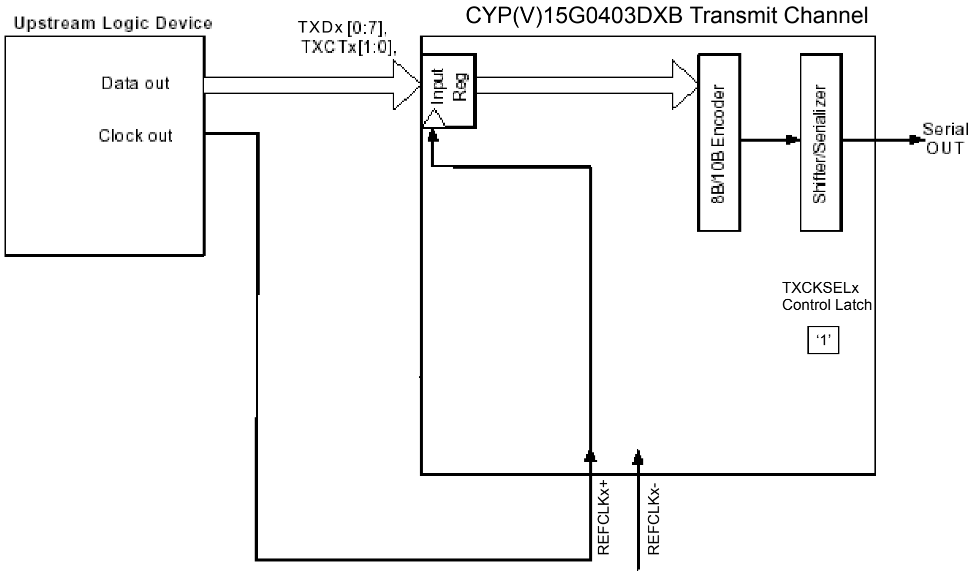


Figure 3. Clocking Scheme Using REFCLKx to Clock the Parallel Transmit Data into the CYP(V)15G0403DXB Transmit Channel

11. What are important issues when using REFCLKx to clock the parallel data into the input registers (TXCKSELx = 1)?

REFCLKx can be used to latch the parallel transmit data into the input register by setting the TXCKSELx latch to 1. The transmit parallel data for channel x should be aligned with REFCLKx so that the set up and hold time parameters for REFCLKx clocking (Data Sheet parameters t_{TREFDS} and t_{TREFDH}) are met.

12. What are the primary functions of the REFCLKx source? What are the requirements on the REFCLKx source? Can the REFCLKx source come from a logic device like an ASIC or FPGA?

The primary functions of the REFCLKx source are as follows:

1. It is the input source to the clock multiplying transmit PLL of the given channel. The output of the TXPLL is used as the bit rate clock to time the output serial data for that channel. Hence, the output jitter in the serial output data will be influenced by the jitter in REFCLKx of that channel.

2. It is used as range control for the Receive CDR PLL which determines the locking range of the incoming serial data by comparing the incoming data frequency with the internal character rate clock frequency. In order to achieve a lock, the receive CDR PLL incoming data frequency needs to be within ±1500 ppm of the internal character rate clock frequency. Though HOTLink II can tolerate ±1500 ppm, the REFCLKx frequency should adhere to the frequency stability (ppm) required by the application (varies depending on the standard).x

From the above mentioned points, it is clear that the REFCLKx source needs to be a stable and low jitter source with the required frequency stability. It is not recommended to use a signal from a logic device (e.g., a FPGA or an ASIC) as the REFCLKx source, unless it meets the jitter and stability requirements.

13. Can I use the receive clock, RXCLKx, from a receive channel as REFCLKx for the same channel?

No. This cannot be done, since REFCLKx is used in the range controller logic for HOTLink II receive CDR PLLs. The purpose of the range controller is to monitor the incoming data frequency relative to REFCLKx. When the incoming serial data rate is beyond the PPM tolerance of the HOTLink II receiver, the CDR will not lock to the incoming data. Therefore, the presence of a stable REFCLKx is necessary for the receive CDR PLL to work.

14. Can I use the receive clock, RXCLKx, from a receive channel as REFCLK for a different channel?

Yes. If there is constant incoming data to the channel and the data has nominal jitter, the receive clock could be used as the reference clock to another channel. For information on the jitter tolerance of the CYP(V)15G0403DXB device, refer to *Question 24* on page 8.

15. Define the data sheet parameter "Receive PLL lock to input data stream"?

The receive CDR PLL lock time is specified by the parameter called t_{RXLOCK} in the data sheet. This parameter is measured under two conditions.

The first condition is called “Receive PLL lock to input data stream.” This is defined as the maximum amount of time taken by the LFIx of a given receive channel CDR PLL to become inactive (HIGH) after the presence of valid data at the serial input buffers for that particular channel. This is measured by providing valid data to a particular channel and enabling the serial input buffers by activating the corresponding INSELx signal and measuring the time elapsed between INSELx going HIGH and LFIx going HIGH. This time varies with the frequency of operation of the channel.

The second condition is called “Receive PLL lock to input data stream—Cold Start.” This is defined as the maximum time between the RESET being deasserted and LFIx going high. In this test, valid data is presented at the serial input buffers even before RESET is deasserted and the receive channels are set to be enabled by default after deassertion of RESET.

Testing under both the above mentioned conditions yielded very similar results. The maximum RXPLL lock time under both conditions is specified in terms of unit interval (UI). The maximum number of UI taken for the RXPLL to lock under both the conditions was found to be 376,000 UI.

16. What is framing? Why do I need framing?

Framing is the process of identifying the proper character boundaries after serial data has been deserialized to parallel data. When the serial data is recovered and retimed at the receive CDR PLL, there is no way to determine the location of character boundaries in the serial stream unless a framer is used. The framer looks for a particular sequence of bits (defined by the selected framing character and type of framer) in the recovered data and realigns the parallel data to reflect the correct character boundaries. After successful framing, the character boundaries of the receive data will match the character boundaries of the transmitted data. Therefore, framing is required whenever characters are transmitted from a source to a destination through a serial link.

17. What are the different framing modes available in the CYP(V)15G0403DXB device? How do the different modes of framing work in the CYP(V)15G0403DXB device?

The CYP(V)15G0403DXB device has three framing modes: 1. Low-latency single-byte framer; 2. Cypress-mode multi-byte framer; and 3. Alternate-mode multi-byte framing. These framers are selected by the RFMODEx[1:0] latches.

1. Low-latency framer: In this mode, the receiver frames to a single detection of the framing character (selected by FRAMCHARx). In this framing mode, on the reception of the framing character, the recovered character rate clock is stretched until it aligns with the recovered character boundaries. Please refer to *Question 19* on the constraints posed on the RFENx signal when this framing mode is selected.

2. Cypress-mode multi-byte framer: In this framing mode, the character boundaries are adjusted only if the selected framing character is detected at least twice within a span of 50 bits, with both instances on identical 10-bit character boundaries. In this mode, the recovered character rate clock does not have any phase changes during normal operation or framing.

3. Alternate-mode multi-byte framer: In this mode, the data stream must contain a minimum of four of the selected framing characters, received as consecutive characters, on identical 10-bit boundaries, before character framing is adjusted. In this mode, the recovered character rate clock does not have any phase changes during normal operation or framing.

18. When I bypass the 10B/8B decoder, can I perform framing? How do I perform framing when I am executing 10-bit wide parallel data transfer (unencoded) mode?

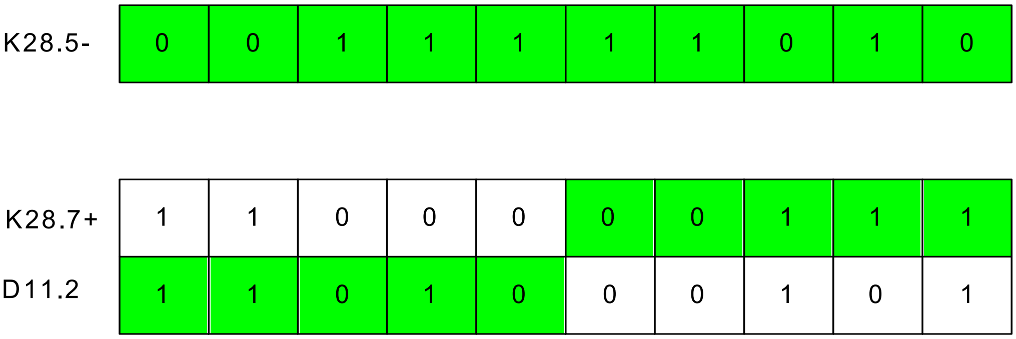
Yes. Framing can be performed when the 10B/8B encoder block is bypassed. This is possible since the framer is implemented before the 10-bit deserialized characters are passed to the 10B/8B decoder block. When the framer is enabled (RFENx = 1), it will frame to the 10-bit boundary that has the bit sequence identical to either positive or negative disparity versions of the framing character (as selected by FRAMCHARx). One or more framing characters may be required to establish framing depending on the type of framer selected by the RFMODEx[1:0] latches.

19. What should the value in the RFENx latch be? How does it differ based on what framer is selected?

Whenever RFENx = 1, the framer (as selected by RFMODEx[1:0]) is enabled and it is actively looking to align character boundaries to the bit pattern that matches the framing sequence.

If the low latency framer is selected, the receiver frames on a single reception of framing character. In this mode, if RFENx remains at 1, the framer can frame to K28.5 aliases created by certain combinations of two successive characters, or bit errors caused due to the serial transmission line which leads to unintentional K28.5 interpretation. The aliases will happen under two conditions: 1. a K28.7 is followed by a D11.x or D20.x; and 2. C0.7 is followed by a D11.x or D20.x. *Figure 4* shows an example of how a K28.5 alias is created when a K28.7 is followed by a D11.2. Therefore if the low latency framer is selected, RFENx should return to 0 after initial framing is performed or the application should make sure that K28.5 alias combinations are not transmitted.

When the other two framers are selected, RFENx can remain at 1 for the entire transmission, since they use multiple framing characters to frame to the data.



K28.5-	0	0	1	1	1	1	1	0	1	0
K28.7+	1	1	0	0	0	0	0	1	1	1
D11.2	1	1	0	1	0	0	0	1	0	1

Figure 4. Example of an Alias K28.5 Character Created When a K28.7 is Followed by a D11.2

20. What is the difference between the $\overline{\text{PABRSTx}}$ and $\overline{\text{RESET}}$ resets?

$\overline{\text{RESET}}$ is a global chip reset that initializes the entire device to a known state. This reset is necessary after power up. $\overline{\text{PABRSTx}}$ is a phase-align buffer reset latch that sets the phase-align buffer pointer for channel x to a maximum separation so that, once set, TXCLKx can skew up to +/-180 degrees with respect to REFCLKx. $\overline{\text{RESET}}$ is an external control signal, while $\overline{\text{PABRSTx}}$ is an internal control latch. $\overline{\text{PABRSTx}}$ is a self-clearing latch, eliminating the requirement of writing a ‘1’ to complete the initialization of the phase-align buffer. To perform a phase-alignment, the $\overline{\text{PABRSTx}}$ latch should be rewritten with a ‘0’.

21. What is the reset sequence for the CYP(V)15G0403DXB device? Can $\overline{\text{RESET}}$ be pulled HIGH permanently?

After device power-up, the CYP(V)15G0403DXB device should be reset using the following reset sequence:

1. Device power-up.

2. Pulse $\overline{\text{RESET}}$ low for at least the time specified in the data sheet (t_{RST}). This operation resets all four channels.

3. Wait at least 16 clock cycles before configuring the control latches to give time for the reset to propagate through the entire device. See *Question 30* for details about enabling and disabling channels.

4. Set the static receiver latch bank for the target channel. May be performed using a global operation, if the application permits it. [Optional step if the default settings match the desired configuration.]

5. Set the static transmitter latch bank for the target channel. May be performed using a global operation, if the application permits it. [Optional step if the default settings match the desired configuration.]

6. Set the dynamic bank of latches for the target channel. Enable the Receive PLLs and transmit channels. May be performed using a global operation, if the application permits it.

7. Reset the phase-align buffer for the target channel after the presence of TXCLKx and after the TXPLL has locked to the frequency of REFCLKx. This may be performed using a global operation if the application permits it. [Optional step if phase-align buffer is bypassed.]

22. What is the intrinsic output jitter generation of the CYP(V)15G0403DXB device?

The CYP(V)15G0403DXB device is frequency agile and supports multiple standards. Hence, the output jitter cannot be specified at all possible data rates. The device was characterized at the operating rates of the standards supported. Moreover, not all standards specify a Bit Error Rate (BER). Thus, only the deterministic jitter (DJ) and the RMS random jitter (σ) are alone specified in *Table 4*. The method for calculating the total peak-to-peak jitter from the values of DJ and random jitter (RJ) is given in *Question 23*.

All SMPTE standards supported by the CYP(V)15G0403DXB device do not specify a BER. For these standards, the total broadband jitter was measured using a communications signal analyzer (also known as digital sampling scope). The measurements were performed on the serial output by transmitting a BIST pattern. Note that the BIST pattern is very similar to a PRBS pattern. The measurement was run for 50,000 hits. The results of this measurement are shown in *Table 5*.

Table 4. Intrinsic Output Jitter for Standards That Specify BER

Standard	Data Rate (Mbps)	Bit Period (ps)	RMS Random Jitter σ (ps)	Deterministic Jitter (ps)
ESCON	200	5000	TBD	TBD
DVB-ASI	270	3703	TBD	TBD
Fibre Channel	1062.5	941	TBD	TBD
GbE	1250	800	TBD	TBD

Table 5. Intrinsic Output Jitter for Standards That Do Not Specify BER

Standard	Data rate (Mbps)	Bit Period (ps)	Total broadband P-P jitter (ps)
SMPTE 259M (Level C)	270	3703	TBD
SMPTE 259M (Level D)	360	2777	TBD
SMPTE 292M	1485	673	TBD

23. How can the total jitter be calculated using the RMS random jitter and deterministic jitter?

The value of Total Jitter (TJ) can be found from the values of RMS Random Jitter (RJ_{RMS}) and DJ using the following formula:

$TJ = N \times RJ_{RMS} + DJ$
where N is the scaling factor. The values of N depend on the BER spec for the given standard. The values of N for different values of BER are shown in *Table 6*.

Table 6. Scaling Factor N depending on BER

BER	N
10^{-6}	9.507
10^{-7}	10.399
10^{-8}	11.224
10^{-9}	11.996
10^{-10}	12.723
10^{-11}	13.412
10^{-12}	14.069
10^{-13}	14.698
10^{-14}	15.301
10^{-15}	15.883
10^{-16}	16.444

The values of BERs specified by certain standards are shown in *Table 7*.

Table 7. BER Spec for Different Standards

BER	N
DVB-ASI	10^{-13}
ESCON	10^{-15}
Fibre Channel	10^{-12}
Gigabit Ethernet	10^{-12}

24. What is the jitter tolerance of the CYP(V)15G0403DXB device?

The CYP(V)15G0403DXB device was characterized for jitter tolerance of ESCON, Fibre Channel and GbE. It passes the jitter tolerance requirements of all the standards with margin.

The GbE requirement for jitter tolerance is defined in the Gigabit Ethernet Standard (section 38.5 in IEEE 802.3z): the receiver should be capable of tolerating 0.749 UI of jitter, where at least 0.462 UI is Deterministic Jitter (DJ), while still meeting a BER of 10^{-12} at the GbE operating frequency (1.25 Gigabits/s). The CYP(V)15G0403DXB device passes this requirement with margin.

In the Fibre Channel standard, the receiver should be capable of tolerating 0.68 UI of jitter, where 0.37 UI is DJ, 0.21 UI is Random Jitter (RJ), and 0.1 UI is Sinusoidal Jitter (SJ), while still meeting a BER of 10^{-12} at the Fibre Channel operating frequency (1.0625 Gigabits/s). The mask for the sinusoidal component of the jitter tolerance is shown in *Figure 5* (source: FC-MJSQ document). Characterization was performed on HOTLink II devices to meet and exceed these requirements.

As defined in the SBCON/ESCON standard, the receiver should be capable of tolerating 0.72 UI of jitter for data pattern PRBS7, while still meeting a BER of 10^{-15} at the operating frequency of 200 MBaud. This translates to a receiver eye opening of 1.4 ns. Characterization was performed on the CYP(V)15G0403DXB device to meet and exceed these requirements.

For a detailed description of the jitter tolerance measurements obtained and methodology used, refer to the application note entitled *Jitter Performance of HOTLink II Devices*.

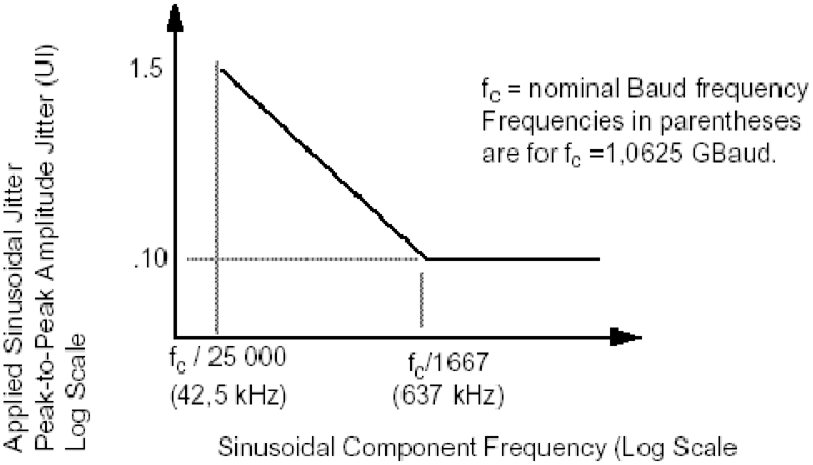


Figure 5. Mask for Sinusoidal Component of Fibre Channel Jitter Tolerance

25. How can jitter generation be measured for the CYP(V)15G0403DXB device?

In order to separate random jitter (RJ) from deterministic jitter (DJ), two different methodologies are adopted, one for RJ and one for DJ.

Random Jitter Measurement

A repeating pattern of K28.7 (0011111000) characters should be sent out at the serial outputs. A set-up to perform this test is shown in *Figure 6*. Using K28.7s will produce a clock like pattern at the outputs. For this pattern, the deterministic jitter will be a minimum. The Communications Signal Analyzer (CSA) should be set up to display the histogram of the zero crossing point. The measurement should be taken after the RMS value of the distribution converges and does not change significantly. The value that should be noted down is the RMS value of the distribution.

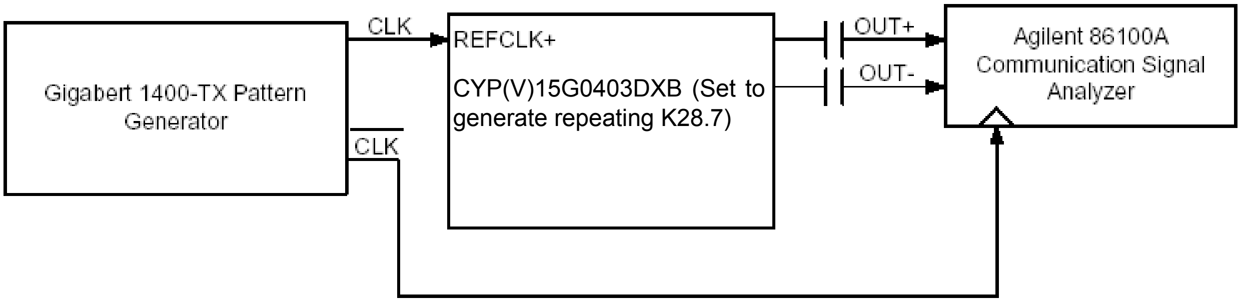

Gigabert 1400-TX Pattern Generator
CLK
$\overline{\text{CLK}}$
REFCLK+
CYP(V)15G0403DXB (Set to generate repeating K28.7)
OUT+
OUT-
Agilent 86100A Communication Signal Analyzer

Figure 6. Set-up to Measure Random Jitter (RJ)

Deterministic Jitter Measurement

The set up to measure deterministic jitter is shown in *Figure 7*. The CYP(V)15G0403DXB device is set up to transmit K28.5 characters with the correct disparity. This will produce a repeating pattern composed of a K28.5+ followed by a K28.5- (00111110101100000101). This pattern has the highest run length of zeros (“00000”) and ones (“11111”) possible in an 8B/10B encoded stream. It also has an isolated 1 (“010”), isolated 0 (101), isolated pair of ones (“0110”)and isolated pair of zeroes (“1001”) in the data stream. The output data stream is observed in a Communication Signal Analyzer like an Agilent 86100A. Tabulate the expected zero crossing times of each pattern edge in the data stream for the given data rate. Then, tabulate the discrepancy between the actual zero crossing times and the expected zero crossing times. The actual zero crossing times should be obtained after averaging at least a thousand samples. The difference between the maximum and minimum deviation time is the DJ.

An example of this calculation using the GbE data rate(1.25 GBaud) is shown in *Table 8*.

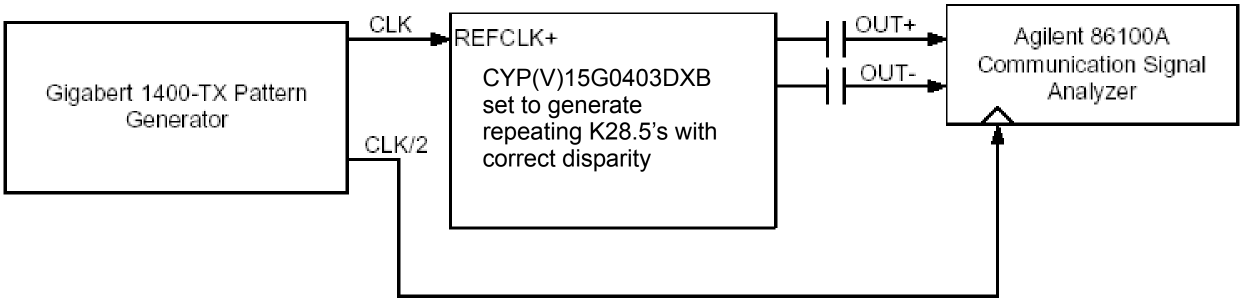

Gigabert 1400-TX Pattern Generator
CLK
CLK/2
REFCLK+
CYP(V)15G0403DXB set to generate repeating K28.5's with correct disparity
OUT+
OUT-
Agilent 86100A Communication Signal Analyzer

Figure 7. Set-up to Measure Deterministic Jitter

Table 8. Example Calculation of Deterministic Jitter

Actual Time Instant Observed in Scope Where First K28.5 Transition Occurred: $T_{0\ (ns)}$				**30**
K28.5 Transitions in bit positions N	**K28.5 transitions in time X=N x bit period (ns)**	**Theoretical K28.5 transitions $T_t = T_0 + X$ (ns)**	**Actual K28.5 Transitions T_a *(ns)***	**Deviation time $T_\delta = T_a – T_t$ (ps)**
0	0	30	30	0
5	4	34	34.006	6
6	4.8	34.8	34.810	10
7	5.6	35.6	35.595	–5
8	6.4	36.4	36.406	6
10	8	38	38.001	1
15	12	42	42.005	5
16	12.8	42.8	42.812	8
17	13.6	43.6	43.594	–6
18	14.4	44.4	44.405	5
20	16	46	46	0
Deterministic Jitter: DJ (ps)= $T_{\delta max}$-$T_{\delta min}$				= 10–(–6) =16 ps

26. What are the decoupling recommendations for the CYP(V)15G0403DXB device?

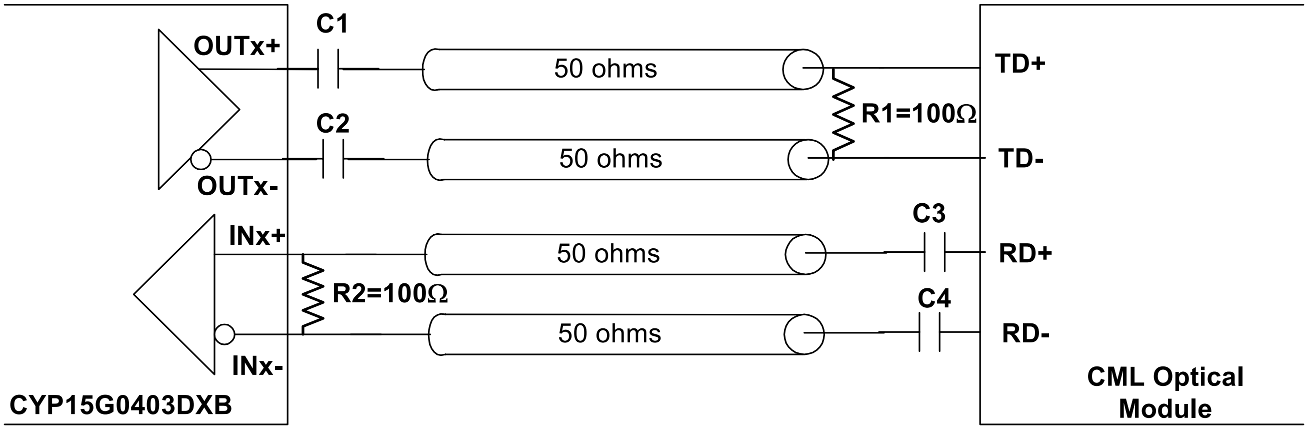
It is recommended that 0.1 μf capacitors be placed as close to the power pins as possible. Also, bulk capacitors within the range of 4.7 μf to 10 μf should be used for low frequency filtering. A rule of thumb is to use four to five 0.1 μf capacitors for each channel, plus one or two equal value bulk capacitors for each device. Refer to the application note *Using Decoupling Capacitors* and the Power Supply Decoupling chapter (5) in the *Perfect Timing Book* for more details.

27. How do I interface the CYP(V)15G0403DXB device with a CML optical module?

The commonly used interface circuit between the CYP(V)15G0403DXB device and a CML optical module is shown in *Figure 8*. Most of the current optical modules have internal termination and/or AC coupling. For these optical modules, therefore, C1, C2, C3, C4 and R1 might not be necessary. However, the termination resistor R2 = 100 ohms is a required component. Refer to the Optical Module data sheet for other special requirements in the interfacing. Also, make sure that the differential P–P swing of the optical module is within the acceptable range of the CYP(V)15G0403DXB input receivers, and vice versa. If the DC levels of the optical module are compatible with the DC levels of the HOTLink II serial I/O's, DC coupling can be performed.

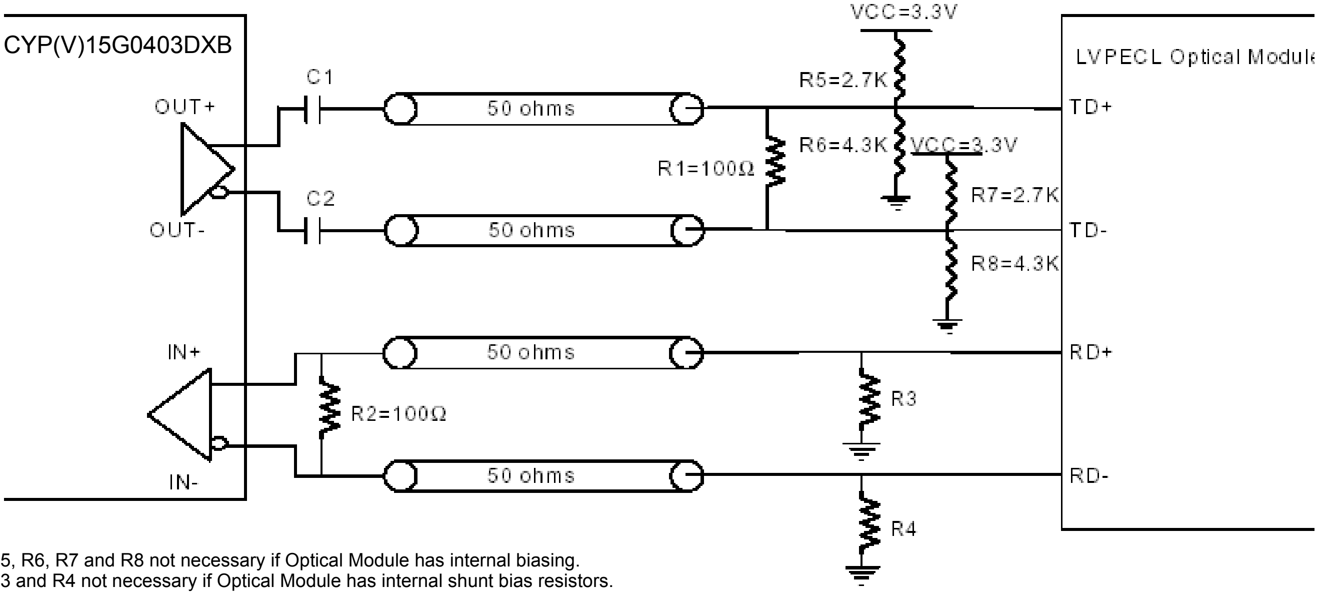
28. How do I interface the CYP(V)15G0403DXB device with an LVPECL optical module?

The interface circuit for the CYP(V)15G0403DXB device with an LVPECL optical module is shown in *Figure 9*. The circuit shown will work for an LVPECL module with no internal termination, biasing or AC coupling. System designers should ensure that the P–P swing of the LVPECL optical module outputs are within the acceptable range of CYP(V)15G0403DXB input receivers. Please refer to the Optical Module data sheet for other special requirements in the interfacing.



R1 not necessary if the optical module has internal termination
C1 and C2 are not necessary if the optical module has internal AC coupling in the TX .Similarly, C3 and C4 are not necessary if the optical module has internal AC coupling in the RX direction.

Figure 8. CYP(V)15G0403DXB Interface with CML Optical Module



R5, R6, R7 and R8 not necessary if Optical Module has internal biasing.
R3 and R4 not necessary if Optical Module has internal shunt bias resistors.
Value of R3 and R4 will be between 150 to 200 ohms. Check optical module data sheet for the value.
C1 and C2 not necessary if Optical module has internal AC coupling.
R1 not necessary if optical module has internal 100-ohm termination.

Figure 9. CYP(V)15G0403DXB Interface with LVPECL Optical Module

29. How do I transmit a special character in the transmit data path? What is the difference between the Cypress table and the Alternate table for special characters? How are these characters encoded and decoded?

Special characters are command characters that can be issued in an 8B/10B encoded system. The eight bit data presented in TXDx[7:0] can either be detected as a normal encoded character or a special character depending on the settings of TXCTx[1:0]. These settings are available in the CYP(V)15G0403DXB device data sheet. The special characters supported by HOTLink II are K28.0, K28.1, K28.2, K28.3, K28.4, K28.6, K28.7, K23.7, K27.7, K29.7 and K30.7. Each of these special characters can be transmitted by presenting either of two possible byte values, from the Cypress table or the alternate table, at the TXDx[7:0] inputs. The list of special character codes and the 8B/10B tables for them are given in the last section of the CYP(V)15G0403DXB device data sheet.

For example, K28.0 (1100001011+/0011110100-) can be transmitted by presenting either C0.0 (000 00000) or C28.0 (00011100) at the TXDx[7:0] and driving TXCTx[1:0] = “10” to transmit a special character. In the receive side, when a K28.0 is received, it can be decoded as C0.0 or C28.0, depending on the setting of the DECMODEx latch. When DECMODEx is set to ‘1’, the special characters are interpreted using the Cypress table and when DECMODEx is set to ‘0’, the characters are interpreted using the alternate table.

30. How do I power down a receive channel? How do I power down a redundant serial output buffer? How do I power down a transmit channel?

The CYP(V)15G0403DXB device supports user control of the powered up or down state of each transmit and receive channel. The receive channels are controlled by the RXPLLPDx latch via the device configuration interface. When RXPLLPDx = 0, the associated receive PLL and analog circuitry of the channel are disabled. The transmit channels are controlled by the OE1x and the OE2x latches via the device configuration interface. When a serial output driver is disabled by setting OE1x = 0 or OE2x =0, it is internally powered down to reduce device power. If both serial drivers for a channel are in this disabled state, the associated internal logic for that transmit channel is also powered down.

31. How can I perform BIST testing for the CYP(V)15G0403DXB device? What are the design considerations to allow BIST testing capability?

The CYP(V)15G0403DXB device has the capability of generating and detecting a BIST sequence, which can be used to validate both device and link operation. The BIST sequence is a 511 character sequence generated in the transmit side by a LFSR embedded in the 8B/10B encoder block. To initiate BIST testing on the transmit channel, set the TXBISTx latch to 0. The BIST sequence can be detected in the receive side and compared with a locally generated BIST LFSR sequence at the receiver. The BIST sequence is detected in a particular receive channel by setting the RXBISTx latch to 0. The validity of the BIST can be checked by the status of the RXSTx1 bit of the given channel. This bit will pulse HIGH whenever the last character of the BIST sequence is detected and valid, which is every 511 characters. Also, RXSTx2 will pulse HIGH whenever a BIST error is detected. Refer to the *Receive Character Status Bits* table in the CYP(V)15G0403DXB data sheet for the behavior of all the status bits for BIST configuration.

The major design considerations that need to be implemented to allow BIST Testing capability are:

1. Ability to read and interpret the values in RXSTx[2:0].

2. The recommended framer for BIST testing is the Cypress-Mode multibyte framer (RFMODEx[1:0] = 10). If the Alternate multibyte framer is selected (RFMODEx[1:0] = 01) and the receive channels are clocked using the recovered clock (RXCKSELx = 0), it is required to frame the receiver before the activation of BIST. Do not use the low latency framer when RFENx = 1 to receive the BIST sequence, since the low latency framer will misalign to the alias framing character present within the BIST sequence.

3. Note that if REFCLKx is selected as the receive clock in the transmitting source, each pass of the BIST sequence is preceded by a 16-character Word Sync Sequence. The CYP(V)15G0403DXB device that is receiving the BIST sequence and the CYP(V)15G0403DXB device that is generating the BIST sequence should have the same RXCKSELx setting. When transmitting or receiving BIST characters from an original HOTLink device, RXCKSELx must be 0.

32. What is a Word Sync Sequence? Can Word Sync Sequences be used to establish framing?

Yes. Word Sync Sequences can be used to establish framing. Word Sync Sequences can be generated in a transmit channel by setting TXCTx[1:0] = “11”.

When a request to send a Word Sync Sequence is issued, a 16-character sequence of K28.5 characters is generated on the associated channel. The disparity of this sequence follows a pattern of either ++--+-+-+-+-+-+- or --++-+-+-+-+-+-+.

33. Can I perform a Serial IN to Serial OUT loopback in the same channel of the CYP(V)15G0403DXB device? If so, under what conditions can it be performed?

The CYP(V)15G0403DXB has limited capability to perform this type of loopback without external hardware since it is usually implemented in point-to-point links where the incoming data stream has a frequency offset with transmitter clock (REFCLKx).

If the device is configured in 10-bit unencoded data transfer mode (ENCBYPx = 0, DECBYPx = 0, DECMODEx = 1 and RXCKSELx = 0), this loopback within the same channel cannot be done without external hardware. If RXCLKx is connected to TXCLKx of the same channel, errors may occur because the frequency of TXCLKx could deviate from the frequency of REFCLKx. Also, RXCLKx cannot be connected to REFCLKx for the reason described in *Question 13*. However, this loopback within the same channel can be performed with a pure lab-testing setup if the incoming serial input stream has a 0 PPM frequency offset from REFCLKx. Note that in real applications, the incoming data stream will not have a 0 PPM frequency offset from REFCLKx.

When RXCKSELx = 1 and the 8B/10B encoder and decoder are enabled, the RXCLKx± follows the REFCLKx (RXCKSELx= 1). In this mode a Serial IN to Serial OUT loopback can be performed provided the application can tolerate the addition and deletion of K28.5 characters from the data stream. Moreover, note that the disparity of the serial output characters in this operation might be different from the disparity of the serial input characters.

34. Can I perform a Serial IN to Serial OUT loopback across multiple channels of the CYP(V)15G0403DXB device? How can I use the CYP(V)15G0403DXB device in a repeater application?

Yes. Serial IN to Serial OUT loopback can be performed across multiple channels by supplying the receive clock of one channel as the reference clock to another channel.

In a repeater application, a similar set-up to the diagram in *Figure 10* could be used. The parallel data (RXDA[9:0]) and receive clock (RXCLKA+) are sent to the repeater logic and then passed back into the transmitter of the other channel as TXDB[9:0] and TXCLKB, respectively. RXCLKA- is connected directly to the reference clock of the other channel.

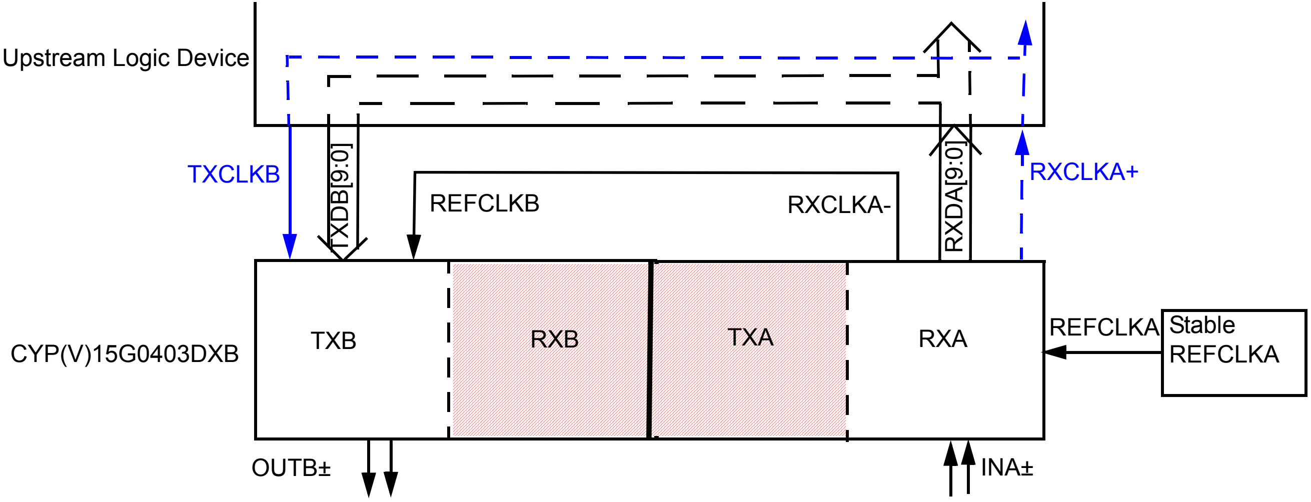


Figure 10. Serial In Serial Out Loopback Across Multiple Channels in a Repeater Application

35. When I bypass the 10B/8B decoder, can I use the Elasticity Buffer? Can I use the Elasticity buffer when I operate the CYP(V)15G0403DXB device in 10-bit parallel data transfer mode (unencoded mode)?

No. The Elasticity Buffer cannot be used when the CYP(V)15G0403DXB device is operating in 10-bit parallel data transfer (unencoded mode). When the 10B/8B decoder is bypassed by setting DECBYPx = 0 and DECMODEx = 1, it also bypasses the Elasticity Buffer.

36. Can I use the reference clock to clock out the parallel data from the output register (RXCKSELx = 1) when I bypass the 10B/8B decoder?

No. When RXCKSELx = 1, the Elasticity Buffer must be used to absorb frequency differences between the recovered clock and the reference clock. However, when the decoder is bypassed, the Elasticity Buffer is also bypassed. Therefore, setup and hold times between the receive clock (selected to be REFCLK) and data may be violated, resulting in errors.

37. What do the status bits (RXSTx[2:0]) and $\overline{\text{LFIx}}$ indicate when there is no data at the serial inputs?

When the device is powered up and when there is no data at the serial inputs, the status bits RXSTx[2:0] indicate either a 000, 100, 110, 101, 001 or 011 depending on the data characters that are formed by the interpretation of the noise at the serial inputs. $\overline{\text{LFI}}$ will be LOW (active) when there is no data at the serial inputs. Hence, the only way an upstream device can tell that there is no valid serial input data at the serial inputs is by monitoring $\overline{\text{LFIx}}$.

38. What do the terms θ_{JC}, θ_{JA} and θ_{CA}mean when referring to thermal package characteristics?

Junction to Case thermal resistance (θ_{JC}), Case to Ambient thermal resistance (θ_{CA}), and Junction to Ambient thermal resistance (θ_{JA}) are the three parameters generally used to characterize the thermal performance of a package. θ_{JC} is mainly a function of the thermal properties of the materials constituting the package, which is defined as the temperature difference between the silicon die and a reference point on the package when dissipating 1 Watt of power. θ_{JA} includes not only the internal package resistance but also the convective thermal resistance from the package to the ambient. θ_{CA} is the case to ambient thermal resistance and is mainly dependent on the surface area available for convection and radiation and the ambient conditions among other factors. This can be controlled at the user end by using heat-sinks that provide a greater surface area and better conduction path or by using air or liquid cooling. Some important equations that relate to these characteristics are as follows:

junction temperature $T_J = (P_D * \theta_{JA}) + T_A$
where T_A is the ambient temperature and P_D is the power dissipated by the device
$\theta_{JA} = \theta_{JC} + \theta_{CA}$

39. What are the thermal characteristics of the CYP(V)15G0403DXB device? What is the junction temperature of the CYP(V)15G0403DXB device?

The thermal characteristics of the CYP(V)15G0403DXB device is provided in *Table 9*. Refer to *Question 38* for definitions of the parameters.

Table 9. Thermal Parameters for the CYP(V)15G0403DXB Package

Parameter	Air Flow	CYP(V)15G0403DXB
θ_{JA}(°C/W)	0 m/s	14.1
	1 m/s	12.2
	2 m/s	10.4
θ_{JC}(°C/W)	N/A	2

40. Is the CYP(V)15G0403DXB device SMPTE compliant?

The CY**V**15G0403DXB device supports SMPTE 259M (Levels C and D) and SMPTE 292M and therefore is SMPTE compliant. Cypress has a reference design with a user-friendly graphical user interface to prove this. This reference board supports all SMPTE pathological test patterns (as recommended in SMPTE EG-34 1999), SMPTE functional test patterns and SMPTE parallel loopback tests. For more details on the characterization data obtained from the evaluation of this reference design, refer to the Cypress application note entitled *SD-SDI and HD-SDI Checkfield Testing on HOTLink II Transceivers for SMPTE Pathological Conditions*.

As stated in the introduction, the difference between the CYP15G0403DXB and the CYV15G0403DXB device is that the latter satisfies SMPTE 259M and SMPTE 292M pathological test requirements per SMPTE EG 34-1999. Thus, the CY**P**15G0403DXB is not SMPTE compliant.

41. Is the CYP(V)15G0403DXB device DVB-ASI compliant? What are the design considerations to use the CYP(V)15G0403DXB device for DVB-ASI?

Yes. The CYP(V)15G0403DXB device is DVB-ASI compliant. Refer to the application note entitled *Implementing DVB-ASI Serial Interfaces Using HOTLink* for more details on how to implement this.

42. What are the recommended cable drivers and equalizers to use with the CYP(V)15G0403DXB device in a SMPTE/DVB interface that uses 300m of coaxial cable transmission link?

As stated in the answer to *Question 40*, the CYV15G0403DXB is SMPTE compliant, while the CYP15G0403DXB is not. Therefore, the CYV15G0403DXB can be used in both SMPTE and DVB interfaces, while the CYP15G0403DXB can be used in the DVB interface, but not the SMPTE interface.

Cable Driver: Any cable driver that has an acceptable receive input swing beyond the CYP(V)15G0403DXB device output P–P swing can interface with the CYP(V)15G0403DXB device. If the cable driver has LVPECL inputs, then the CYP(V)15G0403DXB device outputs can be DC-coupled to the cable driver inputs. In all other cases, AC coupling and external biasing might be necessary. Examples of cable drivers that are compatible with the CYP(V)15G0403DXB device are National's CLC007 and Gennum's GS9028, GS1528.

Equalizer: Equalizer's are not necessary for short cable lengths, but become necessary at longer cable lengths like 300m, where the signal is affected by frequency dependent attenuation due to the cables. The CYP(V)15G0403DXB device can interface to any cable equalizer that has an output swing that is higher than the receive input sensitivity of the CYP(V)15G0403DXB device. Depending on the power supply voltage and DC levels of equalizer outputs, AC coupling might be necessary. Examples of equalizers that are compatible with the CYP(V)15G0403DXB device are National's CLC014 and Gennum's GS9024, GS1524.

43. What are the design considerations that need to be made to use the CYP(V)15G0403DXB device for GbE?

The CYP(V)15G0403DXB device fully supports Ten-bit-interface (TBI or GPCS) mapping of GbE. With some external logic, HOTLink II can support GMII mapping of GbE. Refer to the *Using HOTLink II for Gigabit Ethernet* application note for more details.

44. What are the distances over copper, twisted-pair cable and coaxial cable that can be driven by the CYP(V)15G0403DXB device without an external cable driver or an equalizer? What is the rule of thumb to determine the maximum serial transmission media length?

The CYP(V)15G0403DXB device is a frequency agile device that supports multiple data rates and standards. Moreover, it can be used with different types of media including PCB, Coaxial Cable, Dual Coaxial Cable, Unshielded Twisted Pair and Shielded Twisted Pair. Media from different vendors have different pass band characteristics. Therefore, it is not possible to specify the maximum transmission length using CYP(V)15G0403DXB transceivers for a particular type of media at a given data rate.

However, the cable or media length that can be used for transmission can be estimated using rules of thumb derived from fundamental physics. The factors that will affect the maximum length of transmission media are the following:

1. Output Buffer Amplitude (V_{ODIF})—Can be obtained from the device data sheet.

2. Input Receiver Sensitivity (V_{DIFFS})—Can be obtained from the device data sheet.

3. Attenuation/Unit Length of the Media Used for Different Frequencies—This can be obtained from the manufacturer of the media material (cable).

4. Operating Data Rate—This is the rate at which the serial data switches (Baud rate).

The maximum attenuation that can be tolerated by the link depends on the output amplitude of the transmitter and the input sensitivity of the receiver. The max loss that can be tolerated is given by:

Loss (dB)= 20 * log (V_{ODIF}/V_{DIFFS})

For the CYP(V)15G0403DXB device, the typical V_{ODIF} is 600 mV. The minimum input sensitivity is 100 mV. Therefore, the typical loss that can be tolerated from the above equation is 15.56 dB.

Let the baud rate be F_o Mbps. The highest fundamental frequency for NRZ digital data at this baud rate will be $F_o/2$ MHz (based on 101010... pattern). The majority of the spectral content for digital signals lies within the fifth harmonic of the fundamental frequency. Hence, if the media provides an attenuation of less than 15.56 db for all frequencies within $5F_o/2$ MHz, the CYP(V)15G0403DXB device should be able to transmit and receive data over the link. The attenuation of the medium with respect to frequency can be obtained from the manufacturer of the medium.

45. Describe the operation of the Elasticity Buffer. How can I disable and enable the elasticity buffer? How are K28.5s inserted to and deleted from the character stream when the frequency difference between the recovered clock and system REFCLK is absorbed?

The elasticity buffer is a 10-character-deep FIFO, present in each channel of the device, that is used to absorb frequency differences and/or phase differences between two different clock domains.

When the 8B/10B encoder is enabled, the elasticity buffer is also enabled when RXCKSELx = 1. The write clock for the elasticity buffer is always the recovered character rate clock that follows the clock recovered from the incoming serial stream of data. The read clock can be selected as the internal character rate clock that follows the REFCLK of that channel by setting RXCKSELx to 1.

Rate matching: The elasticity buffer can be used to absorb frequency differences between the incoming serial data and the read clock of the elasticity buffer. Note that the manner in which the insertion and deletion takes place might affect upstream logic if the selected framing character is used as a special command.

Insertion of framing character: The elasticity buffer will insert framing characters into the data stream whenever the read clock is faster than the write clock (max. frequency offset tolerated is 200 ppm). Insertion of a K28.5 character will take place by adding a K28.5 character following a received K28.5 framing character.

Deletion of framing character: The elasticity buffer will delete framing characters from the data stream whenever the read clock is slower than the write clock (max. frequency offset tolerated is 200 ppm). Deletion of a framing character will take place by deleting the framing character on reception.

The elasticity buffer is disabled whenever the 10B/8B decoder block is disabled (by setting DECBYP = 0 and DECMODEx = 1).

46. Can the 3-Level select static control inputs for SPDSELx be controlled by an FPGA/CPLD output?

Controlling 3-Level select inputs for SPDSELx with an FPGA/CPLD output is not recommended, since the DC levels for the three level input HIGH and LOW have less noise margin than LVTTL DC levels. Moreover, controlling the FPGA/CPLD output in three-state mode to achieve the MID state for the 3-Level inputs is not a recommended practice.

For a given application, the SPDSELx inputs are meant to be static. To achieve a HIGH state, tie a strong pull-up (100 ohms) resistor to the 3-Level select pin. To achieve a LOW state, tie a strong pull-down resistor (100 ohms) to the 3-Level select pin. To achieve a MID state, leave the pin floating. In the prototype design stage, customers might want to have the options for pull-up as well as pull-down resistor to provide more flexibility.

47. What is the purpose of the Use Local Clock signal ($\overline{\text{ULCx}}$)? In what situations can it be used?

When $\overline{\text{ULCx}}$ is LOW, the RXPLL is forced to lock to REFCLKx± instead of the received serial data stream. While $\overline{\text{ULCx}}$ is LOW, the $\overline{\text{LFIx}}$ for the associated channel is LOW, indicating a link fault. When $\overline{\text{ULCx}}$ is HIGH, the RXPLL performs Clock and Data Recovery functions on the input data streams.

This function is used in applications where a stable RXCLKx± is needed in the absence of valid incoming serial data. An example of its use is shown in *Figure 11*. When an optical signal is lost or extremely attenuated during transmission, the RXPLL will not be able to lock to the incoming serial data stream. In order to maintain a stable RXCLKx±, $\overline{\text{ULCx}}$ needs to change to 0. To automatically perform this change, the Signal Detect control signal (SD) of the optical module can be connected to $\overline{\text{ULCx}}$. In this case, when there is a loss of signal, SD will become LOW, forcing $\overline{\text{ULCx}}$ LOW. The RXPLL will lock to REFCLKx. When there is valid incoming data, the error signal will be deasserted, returning ULCx to 1. The RXPLL will start locking to the serial data stream again.

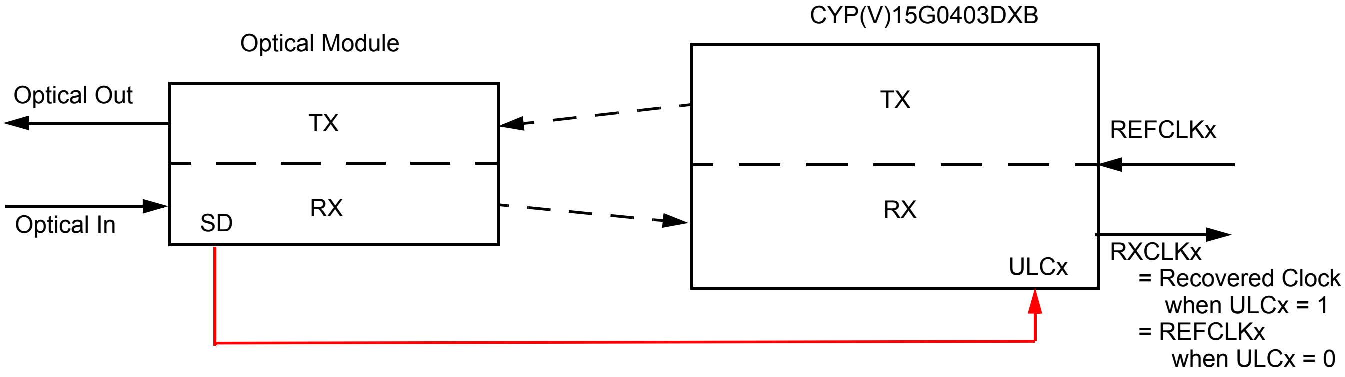


Figure 11. Example of ULCx While Using an Optical Module

48. How do I configure the CYP(V)15G0403DXB for different protocols?

Refer to the application note entitled *Configuring the CYP15G0403DXB HOTLink II Device for Multiple Protocols*.

49. Can channel bonding be performed using the CYP15G0403DXB?

No. The device cannot perform channel bonding without external hardware. CYP15G0401DXB and CYP15G0201DXB have the capability to perform channel bonding.

50. What are the possible rates of RXCLKx depending on the values of RXRATEx, TXRATEx, and RXCKSELx?

The rates of RXCLKx are shown in *Table 10*. Note that when REFCLKx is a half-rate clock and RXCKSEL = 1, RXCLKx is a half-rate clock following REFCLKx independently of RXRATEx.

Table 10.RXCLKx Description Based on REFCLK Rate, RXCKSELx, and RXRATEx

REFCLK Rate (TXRATE)	RXCKSELx	RXRATEx	RXCLKx Description
Full Rate (0)	0	0	Full rate, following Recovered Clock
Full Rate (0)	0	1	Half rate, following Recovered Clock
Full Rate (0)	1	0	Full Rate, following REFCLKx
Full Rate (0)	1	1	Half Rate, following REFCLKx
Half Rate (1)	0	0	Full rate, following Recovered Clock
Half Rate (1)	0	1	Half rate, following Recovered Clock
Half Rate (1)	1	0 or 1	Half rate, following REFCLKx

51. Describe the timing parameters t_{RREFDA}, t_{RREFDW}, $t_{REFxDV+}$, $t_{REFxDV-}$, t_{RXDV+}, t_{RXDV-}.

Table 11.Description of Timing Parameters

Parameter	RXCLKx Description	Description
t_{RREFDA}	Full Rate, following REFCLK	Time from rising edge of REFCLKx to the start of valid data
	Half Rate, following REFCLK	Time from rising or falling edge of REFCLKx to the start of valid data
t_{RREFDW}	Full or Half Rate following REFCLK	Duration of one cycle of valid data
$t_{REFxDV+}$	Full Rate, following REFCLK	Time from rising edge of RXCLKx+ to the time that data is no longer valid
	Half Rate, following REFCLK	Time from rising or falling edge of RXCLKx± to time that data is no longer valid
$t_{REFxDV-}$	Full Rate, following REFCLK	Time from presence of valid data to the rising edge of RXCLKx+
	Half Rate, following REFCLK	Time from presence of valid data to the rising or falling edge of RXCLKx±
t_{RXDV+}	Full Rate, following Recovered Clock	Time from rising edge of RXCLKx+ to the time that data is no longer valid
	Half Rate, following Recovered Clock	Time from rising or falling edge of RXCLKx± to time that data is no longer valid
t_{RXDV-}	Full Rate, following Recovered Clock	Time from presence of valid data to the rising edge of RXCLKx+
	Half Rate, following Recovered Clock	Time from presence of valid data to the rising or falling edge of RXCLKx±

Approved AN060_B 4/7/04 kkv