Independent Clocking Quad HOTLink II™ Transceiver

Features

- **Second-generation HOTLink® technology**
- **Fibre-Channel- and Gigabit-Ethernet-compliant 8B/10B-coded or 10-bit unencoded**
- **Four independent channels**
  - **Each channel can operate at a different signaling rate**
- **8-bit encoded data transport**
  - **Equal to a throughput of 9.6 GBits/second**
- **10-bit unencoded data transport**
  - **Equal to a throughput of 12 GBits/second**
- **Selectable input clocking options**
- **Selectable output clocking options**
- **Receive framer provides alignment to COMMA or Full K28.5 detect**
  - **Single or Multibyte framer for character alignment**
  - **Low-latency option**
- **Synchronous LVTTL parallel input interface**
- **Synchronous LVTTL parallel output interface**
- **200- to1500-MBaud serial signaling rate**
- **Internal phase-locked loops (PLLs) with no external PLL components**
- **Dual differential PECL-compatible serial inputs per channel**
- **Dual differential PECL-compatible serial outputs per channel**
  - **Outputs are source-matched for 50Ω**
  - **No external bias resistors required**
  - **Controlled edge-rates**
- **Compatible with**
  - **Fiber-optic modules**
  - **copper cables**
  - **circuit board traces**
- **JTAG boundary scan**
- **Built-in Self-test (BIST) for at-speed channel testing**
- **Per-channel Link Quality Indicator**
  - **Analog signal detect**
  - **Digital signal detect**
  - **Frequency range detect**
- **Low-power 3W typical at +3.3 V_{CC}**
- **256-ball thermally enhanced BGA package**
- **0.25μ BiCMOS technology**

Functional Description

The CYP15G0403DX Quad HOTLink II™ is a point-to-point building block allowing the transfer of data over independent high-speed serial links at signaling speeds ranging from 200-to-1500 MBaud per link. Each channel operates independently with its own reference clock allowing different rates on each channel.

Each transmit channel accepts parallel characters in an Input Register, encodes each character for transport, and then converts it to serial data. Each receive channel accepts serial data and converts it to parallel data, decodes the data into characters, and presents these characters to an output register. *Figure 1* illustrates typical connections between independent host systems and corresponding CYP15G0403DX chips. As a second-generation HOTLink device, the CYP15G0403DX extends the HOTLink family to faster data rates, while maintaining serial-link compatibility with other HOTLink devices.

The transmit section of the CYP15G0403DX Quad HOTLink II consists of four independent byte-wide channels. Each channel can accept either 8-bit data characters or preencoded

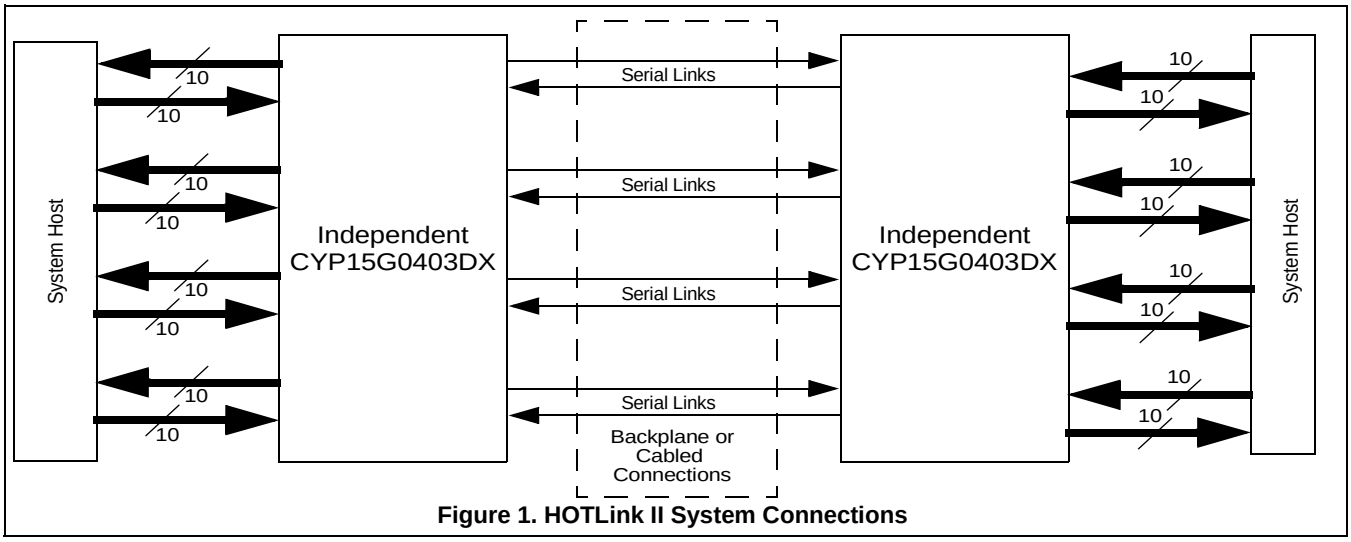


Figure 1. HOTLink II System Connections

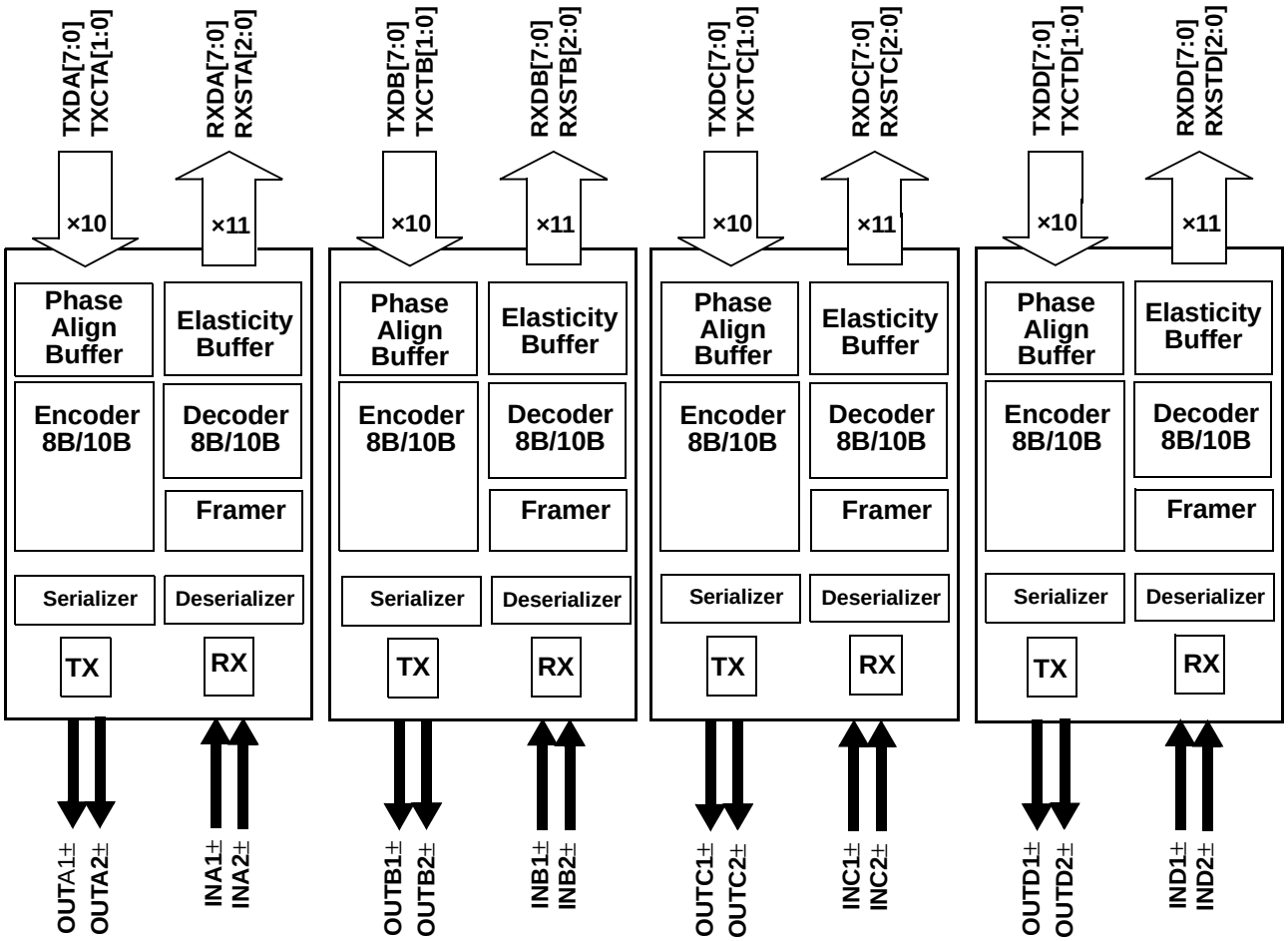
10-bit characters. Data characters are passed from the Transmit Input Register to an embedded 8B/10B Encoder. These encoded characters are then serialized and output from dual Positive ECL compatible differential transmission-line drivers at a bit-rate of either 10- or 20-times the input reference clock for that channel.

The receive section of the CYP15G0403DX Quad HOTLink II consists of four independent byte-wide channels. Each channel accepts a serial bit-stream from one of two PECL-compatible differential line receivers and, using a completely integrated PLL Clock Synchronizer, recovers the timing information necessary for data reconstruction. Each recovered bit-stream is deserialized and framed into characters, 8B/10B decoded, and checked for transmission errors. Recovered decoded characters are then written to an internal Elasticity Buffer, and presented to the destination host system. The integrated 8B/10B encoder/decoder may be bypassed. The clocking of the parallel I/O interface may be changed to match different system architectures. In additional to clocking the transmit path interfaces, the receive interface may be configured to present data relative to a recovered clock or to a local reference clock.

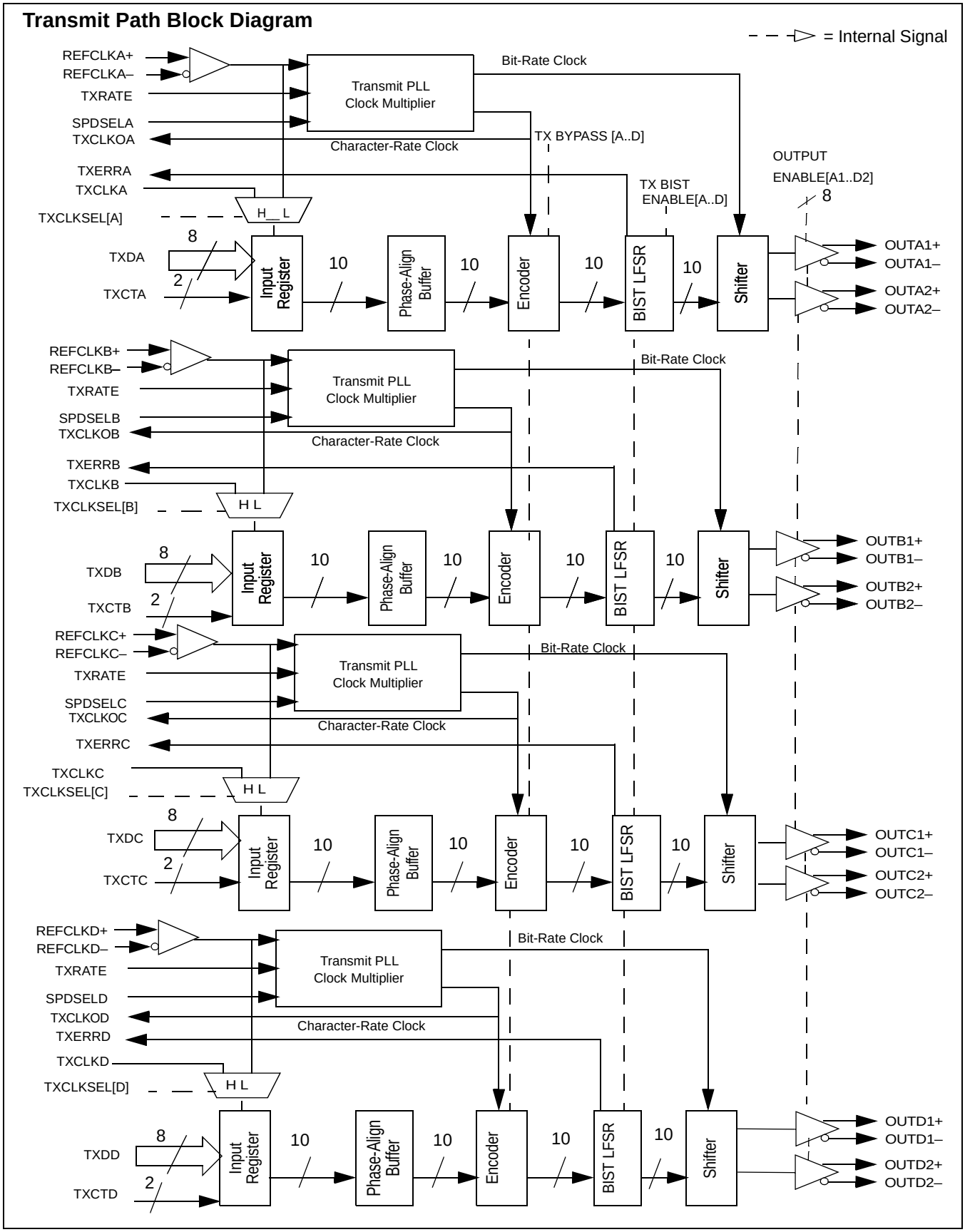
Each transmit and receive channel contains independent BIST pattern generator and pattern checkers. This BIST hardware allows full speed testing of the high-speed serial data paths in each transmit and receive section, and across the interconnecting links.

HOTLink II devices are ideal for a variety of applications where parallel interfaces can be replaced with high-speed, point-to-point serial links.Some applications include interconnecting backplanes on switches, routers, servers and video transmission systems

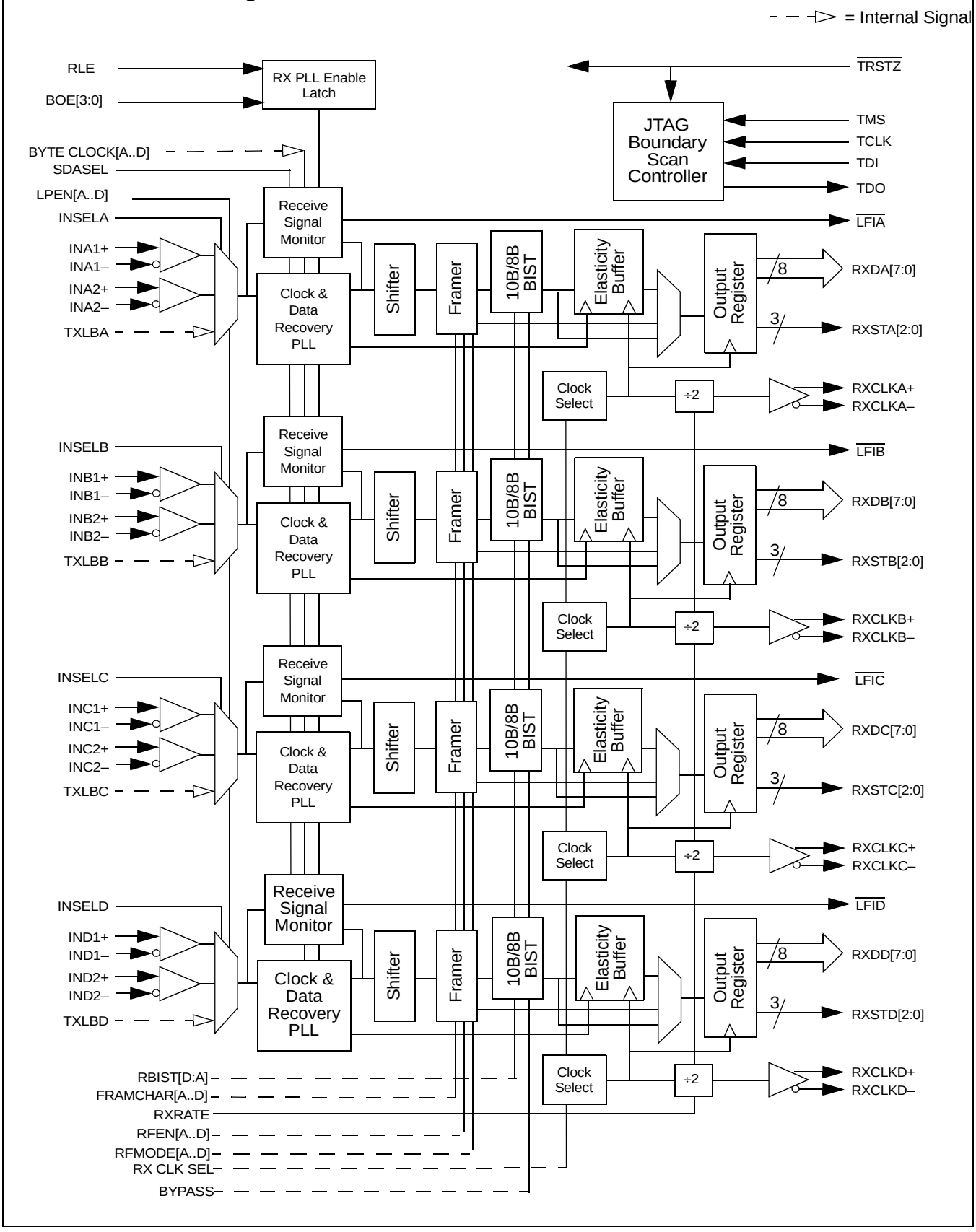
Transceiver Logic Block Diagram



Transmit Path Block Diagram



Receive Path Block Diagram



Pin Configuration (Top View)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A	INC1-	OUT C1-	INC2-	OUT C2-	**V_{CC}**	IND1-	OUT D1-	**GND**	IND2-	OUT D2-	INA1-	OUT A1-	**GND**	INA2-	OUT A2-	**V_{CC}**	INB1-	OUT B1-	INB2-	OUT B2-
B	INC1+	OUT C1+	INC2+	OUT C2+	**V_{CC}**	IND1+	OUT D1+	**GND**	IND2+	OUT D2+	INA1+	OUT A1+	**GND**	INA2+	OUT A2+	**V_{CC}**	INB1+	OUT B1+	INB2+	OUT B2+
C	TDI	TMS	IN SELC	IN SELB	**V_{CC}**	FRAM CHAR B	FRAM CHAR C	**GND**	BOE[7]	BOE[5]	BOE[3]	BOE[1]	**GND**	SDA SEL	SPD SELD	**V_{CC}**	TX RATE	RX RATE	LP END	TDO
D	TCLK	$\overline{\text{TRSTZ}}$	IN SELD	IN SELA	**V_{CC}**	FRAM CHAR A	SPD SELC	**GND**	BOE[6]	BOE[4]	BOE[2]	BOE[0]	**GND**	LP ENB	FRAM CHAR D	**V_{CC}**	LP ENA	RF ENA	RF ENB	RFMO DELE
E	**V_{CC}**	**V_{CC}**	**V_{CC}**	**V_{CC}**													**V_{CC}**	**V_{CC}**	**V_{CC}**	**V_{CC}**
F	TX ERRC	RF END	TX DC[0]	RF ENC													BIST LE	RX STB[1]	TX CLKO B	RX STB[0]
G	TX DC[7]	CLKS EL LE	TX DC[4]	TX DC[1]													SPD SELB	LP ENC	SPD SELA	RX DB[1]
H	**GND**	**GND**	**GND**	**GND**													**GND**	**GND**	**GND**	**GND**
J	TX CTC[1]	TX DC[5]	TX DC[2]	TX DC[3]													RX STB[2]	RX DB[0]	RX DB[5]	RX DB[2]
K	RX DC[2]	REFC LKC-	TXCT C[0]	$\overline{\text{LFIC}}$	]												RX DB[3]	RX DB[4]	RX DB[7]	RXCL KB+
L	RX DC[3]	REFC LK C+	TX CLKC	TX DC[6]													RX DB[6]	$\overline{\text{LFIB}}$	RX CLKB-	TX DB[6]
M	RX DC[4]	RX DC[5]	RX DC[7]	RX DC[6]													REFC LKB+	REFC LKB-	TX DB[7]	TX CLKB
N	**GND**	**GND**	**GND**	**GND**													**GND**	**GND**	**GND**	**GND**
P	RX DC[1]	RX DC[0]	RX STC[0]	RX STC[1]													TX DB[5]	TX DB[4]	TX DB[3]	TX DB[2]
R	RXST C[2]	TXCLK OC	RXCL K C+	RX CLK C-													TX DB[1]	TX DB[0]	TX CTB[1]	TX ERRB
T	**V_{CC}**	**V_{CC}**	**V_{CC}**	**V_{CC}**													**V_{CC}**	**V_{CC}**	**V_{CC}**	**V_{CC}**
U	TX DD[0]	TX DD[1]	TX DD[2]	TX CTD[1]	**V_{CC}**	RX DD[2]	RX DD[1]	**GND**	TX ERRD	RLE	REFC LK D-	TX DA[1]	**GND**	TX DA[4]	TX CTA[0]	**V_{CC}**	RX DA[2]	TX CTB[0]	RX STA[2]	RX STA[1]
V	TX DD[3]	TX DD[4]	TX CTD[0]	RX DD[6]	**V_{CC}**	RX DD[3]	RX STD[0]	**GND**	RX STD[2]	BYPA SS LE	REFC LK D+	TX CLKO A	**GND**	TX DA[3]	TX DA[7]	**V_{CC}**	RX DA[7]	RX DA[3]	RX DA[0]	RX STA[0]
W	TX DD[5]	TX DD[7]	$\overline{\text{LFID}}$	RXCL K D-	**V_{CC}**	RX DD[4]	RX STD[1]	**GND**	OELE	$\overline{\text{TX}}$ RST	RX CLKA+	TX ERRA	**GND**	TX DA[2]	TX DA[6]	**V_{CC}**	$\overline{\text{LFIA}}$	REF CLK A+	RX DA[4]	RX DA[1]
Y	TX DD[6]	TXCLK D	RX DD[7]	RXCL K D+	**V_{CC}**	RX DD[5]	RX DD[0]	**GND**	TXCLK OD	N/C	TX CLKA	RX CLKA-	**GND**	TX DA[0]	TX DA[5]	**V_{CC}**	TX CTA[1]	REF CLK A-	RX DA[6]	RX DA[5]

Pin Configuration (Bottom View)

	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
A	OUT B2-	INB2-	OUT B1-	INB1-	V_{CC}	OUT A2-	INA2-	GND	OUT A1-	INA1-	OUT D2-	IND2-	GND	OUT D1-	IND1-	V_{CC}	OUT C2-	INC2-	OUT C1-	INC1-
B	OUT B2+	INB2+	OUT B1+	INB1+	V_{CC}	OUT A2+	INA2+	GND	OUT A1+	INA1+	OUT D2+	IND2+	GND	OUT D1+	IND1+	V_{CC}	OUT C2+	INC2+	OUT C1+	INC1+
C	TDO	LP END	RX RATE	TX RATE	V_{CC}	SPD SELD	SDA SEL	GND	BOE[1]	BOE[3]	BOE[5]	BOE[7]	GND	FRAM CHAR C	FRAM CHAR B	V_{CC}	IN SELB	IN SELC	TMS	TDI
D	RFMO DELE	RF ENB	RF ENA	LP ENA	V_{CC}	FRAM CHAR D	LP ENB	GND	BOE[0]	BOE[2]	BOE[4]	BOE[6]	GND	SPD SELC	FRAM CHAR A	V_{CC}	IN SELA	IN SELD	$\overline{TRSTZ}$	TCLK
E	V_{CC}	V_{CC}	V_{CC}	V_{CC}													V_{CC}	V_{CC}	V_{CC}	V_{CC}
F	RX STB[0]	TX CLKO B	RX STB[1]	BIST LE													RF ENC	TX DC[0]	RF END	TX ERRC
G	RX DB[1]	SPD SELA	LP ENC	SPD SELB													TX DC[1]	TX DC[4]	CLKS EL LE	TX DC[7]
H	GND	GND	GND	GND													GND	GND	GND	GND
J	RX DB[2]	RX DB[5]	RX DB[0]	RX STB[2]													TX DC[3]	TX DC[2]	TX DC[5]	TX CTC[1]
K	RXCL KB+	RX DB[7]	RX DB[4]	RX DB[3]													$\overline{LFIC}$	TXCT C[0]	REFC LKC-	RX DC[2]
L	TX DB[6]	RX CLKB-	$\overline{LFIB}$	RX DB[6]													TX DC[6]	TX CLKC	REFC LK C+	RX DC[3]
M	TX CLKB	TX DB[7]	REFC LKB-	REFC LK+													RX DC[6]	RX DC[7]	RX DC[5]	RX DC[4]
N	GND	GND	GND	GND													GND	GND	GND	GND
P	TX DB[2]	TX DB[3]	TX DB[4]	TX DB[5]													RX STC[1]	RX STC[0]	RX DC[0]	RX DC[1]
R	TX ERRB	TX CTB[1]	TX DB[0]	TX DB[1]													RX CLK C-	RXCL K C+	TXCLK OC	RXST C[2]
T	V_{CC}	V_{CC}	V_{CC}	V_{CC}													V_{CC}	V_{CC}	V_{CC}	V_{CC}
U	RX STA[1]	RX STA[2]	TX CTB[0]	RX DA[2]	V_{CC}	TX CTA[0]	TX DA[4]	GND	TX DA[1]	REFC LK D-	RLE	TX ERRD	GND	RX DD[1]	RX DD[2]	V_{CC}	TX CTD[1]	TX DD[2]	TX DD[1]	TX DD[0]
V	RX STA[0]	RX DA[0]	RX DA[3]	RX DA[7]	V_{CC}	TX DA[7]	TX DA[3]	GND	TX CLKO A	REFC LK D+	BYPA SS LE	RX STD[2]	GND	RX STD[0]	RX DD[3]	V_{CC}	RX DD[6]	TX CTD[0]	TX DD[4]	TX DD[3]
W	RX DA[1]	RX DA[4]	REF CLK A-	$\overline{LFIA}$	V_{CC}	TX DA[6]	TX DA[2]	GND	TX ERRA	RX CLKA+	$\overline{TX}$ RST	OELE	GND	RX STD[1]	RX DD[4]	V_{CC}	RXCL K D-	$\overline{LFID}$	TX DD[7]	TX DD[5]
Y	RX DA[5]	RX DA[6]	REF CLK A+	TX CTA[1]	V_{CC}	TX DA[5]	TX DA[0]	GND	RX CLKA-	TX CLKA	N/C	TXCLK OD	GND	RX DD[0]	RX DD[5]	V_{CC}	RXCL K D+	RX DD[7]	TXCLK D	TX DD[6]

Pin Descriptions

Name	I/O Characteristics	Signal Description
Transmit Path Data and Status Signals		
TXERRA TXERRB TXERRC TXERRD	LVTTL Output, changes relative to REFCLKx[1]	**Transmit Path Error**. When signal is HIGH, indicates detection of transmit Phase-Align Buffer underflow or overflow. If an underflow or overflow condition is detected, TXERRx is high until either a Word Sync Sequence is transmitted on that channel, or $\overline{\text{TXRST}}$ is reset LOW to recenter the transmit Phase-Align Buffers. When BIST is enabled for a transmit channel, BIST progress is presented on the associated TXERRx output. Once every 511 character times, the associated TXERRx signal pulses HIGH for one transmit-character clock period to indicate a pass through the BIST bit sequence.
TXCTA[1:0] TXCTB[1:0] TXCTC[1:0] TXCTD[1:0]	LVTTL Input, synchronous, sampled by the associated TXCLKx]↑ or REFCLKx↑[1]	**Transmit Control**. These inputs are captured on the rising edge of the transmit interface clock and are passed to the data encoder. They identify the type of character captured on the associated TXDx[7:0] inputs. If the encoder is bypassed, these inputs are input data bits. When the encoder is enabled, these inputs determine if the TXDx[7:0] character is encoded as Data, a Special Character code, or replaced with an Special Character codes (see *Table 3* for details).
TXDA[7:0] TXDB[7:0] TXDC[7:0] TXDD[7:0]	LVTTL Input, synchronous, sampled by the associated TXCLKx↑ or REFCLKx↑[1]	**Transmit Data Inputs**. These data inputs are captured on the rising edge of the transmit interface clock and passed to the encoder. When the encoder is enabled, TXDx[7:0] is the data to be encoded for transport over the serial link.
Transmit Path Control and Clock Signals		
TXRST	LVTTL Input, asynchronous, internal pull-up, sampled by TXCLKx↑ or REFCLKx↑[1]	**Transmit Clock Phase Reset When LOW**. The transmit Phase-Align Buffers are allowed to adjust their data-transfer timing to allow error free transfer of data from the input register to the encoder. When $\overline{\text{TXRST}}$ is HIGH, the internal phase relationship between the associated input clock and the internal character-rate clock is fixed. During Phase Reset alignment period, one or more characters may be added to or lost from the associated transmit paths This process may continue until phase of recovered clock and the reference clock are matched. $\overline{\text{TXRST}}$ must be LOW for a minimum of two consecutive rising edges of TXCLKx or REFCLKx↑ to ensuring a correct reset.
TXCLKOA TXCLKOB TXCLKOC TXCLKOD	LVTTL Output	**Transmit Clock Output**. These output clocks are synthesized by each channel's transmit PLL and operate synchronous to the internal transmit character clock. Each clock operates at either the same frequency as the associated REFCLKx, or at twice the frequency of the associated REFCLKx as selected by TXRATE. TXCLKOx is always equal to the transmit VCO bit-clock frequency ÷10. These output clocks have no fixed phase relationship to REFCLKx.
TXRATE	LVTTL Input, Static Control input, internal pull-down	**Transmit PLL Clock Rate Select**. When TXRATE = HIGH, the Transmit PLL multiplies all REFCLKx inputs by 20 to generate the serial bit-rate clock. When TXRATE = LOW, each transmit PLL multiples the associated REFCLKx by 10 to generate the serial bit-rate clock. See *Table 5* for a list of operating serial rates. When REFCLKx is selected for clocking of the receive parallel interface, the TXRATE input also determines if the clock on the RXCLKx± outputs are a full or half-rate clock. When TXRATE = HIGH, these output clocks are half-rate clocks and follow the frequency and duty cycle of the REFCLKx input. When TXRATE = LOW, these output clocks are full-rate clocks and follow the frequency and duty cycle of the REFCLKx input.
TXCLKA TXCLKB TXCLKC TXCLKD	LVTTL Clock Input, internal pull-down	**Transmit Path Input Clocks**. When selected as an input sample clock, each transmit clock will be frequency-coherent to its TXCLKOx clock, but not have a fixed phase reala-tionship. The internal operating phase of each input clock relative to its associated REFCLKx is adjusted when $\overline{\text{TXRST}}$ = LOW and locked when $\overline{\text{TXRST}}$ = HIGH.
Receive Path Data Signals		
RXDA[7:0] RXDB[7:0] RXDC[7:0] RXDD[7:0]	LVTTL Output, synchronous to the selected RXCLKx↑ output or REFCLKx↑[1] input	**Parallel Data Output**. These outputs change following the falling edge of the receive interface clock.

Note:

1. When REFCLKx is configured for half-rate operation (TXRATE = HIGH), these inputs are sampled relative to both the rising and falling edges of the associated REFCLKx.

Pin Descriptions (continued)

Name	I/O Characteristics	Signal Description
RXSTA[2:0] RXSTB[2:0] RXSTC[2:0] RXSTD[2:0]	LVTTL Output, synchronous to the selected RXCLKx↑ output or REFCLKx↑ [1] input	**Parallel Status Output**. These outputs change following the falling edge of the receive interface clock. When the decoder is bypassed, RXSTx[1:0] become the two low-order bits of the 10-bit received character, while RXSTx[2] = HIGH indicates the presence of the selected framing character in the output register.
Receive Control and Clock Signals		
RXCLKA± RXCLKB± RXCLKC± RXCLKD±	Three-state, LVTTL Output clock	**Receive Character Clock Output**. These true and complement clocks are the Receive interface clocks which are used to control timing of data output transfers. These clocks are output continuously at either at half character rate or at the character rate of the data being received, as selected by RXRATE. When configured such that all output data paths are clocked by the REFCLKx instead of a recovered clock, the RXCLKx± output drivers present a buffered form of the associated REFCLKx. RXCLKx± are buffered forms of REFCLKx that are delayed in phase to align with the data. This phase difference allows the user to select the optimal clock (REFCLKx or RXCLKx) for setup/hold timing for their specific interface.
RXRATE	LVTTL Input Static Control Input, internal pull-down	**Receive Clock Rate Select**. When LOW, the RXCLKx± recovered clock outputs are complementary clocks operating at the recovered character rate. Data for the associated receive channels should be latched on the rising edge of RXCLKx+ or falling edge of RXCLKx–. When HIGH, the RXCLKx± recovered clock outputs are complementary clocks operating at half the character rate. Data for the associated receive channels should be latched alternately on the rising edge of RXCLKx+ and RXCLKx–. When operated with REFCLKx clocking of the received parallel data outputs, the RXRATE input is not used.
RFENA RFENB RFENC RFEND	LVTTL input, asynchronous, internal pull-down	**Reframe Enable**. When RFENx is HIGH, the associated channel's framer is enabled to frame per the presently enabled framing mode.
FRAMCHARA FRAMCHARB FRAMCHARC FRAMCHARD	3-Level Select [2] Static Control Input	**Framing Character Select**. Used to control the character or portion of a character used for character framing of each channel's received data streams. When LOW, the framer looks for an 8-bit positive COMMA character in the data stream. When MID, the framer looks for both positive and negative disparity versions of the 8-bit COMMA character. When HIGH, the framer looks for both positive and negative disparity versions of the K28.5 character.
Device Control Signals		
SPDSELA SPDSELB SPDSELC SPDSELD	3-Level Select [2], static configuration input	**Serial Rate Select**. These inputs specify the operating signaling-rate range of each channel's transmit and receive PLLs. LOW = 200–400 MBaud MID = 400–800 MBaud HIGH = 800–1500 MBaud
REFCLKA± REFCLKB± REFCLKC± REFCLKD±	Differential LVPECL or single-ended LVTTL input clock	**Reference Clock**. This clock input is used as the timing reference for the transmit and receive PLLs. This input clock may also be selected to clock the transmit and receive parallel interfaces. For LVTTL input clock, connect clock source to a REFCLK input and float the other REFCLK input. For an LVPECL input level input clock has to be a differential clock, using both inputs.

Note:

2. 3-Level select inputs are used for static configuration. They are ternary inputs that make use of logic levels of LOW, MID, and HIGH. The LOW level is usually implemented by direct connection to V_{SS} (ground). The HIGH level is usually implemented by direct connection to V_{CC} (power). When not connected or allowed to float, a 3-Level select input will self-bias to the MID level.

Pin Descriptions (continued)

Name	I/O Characteristics	Signal Description
Latch Control Signals and Bus		
BOE[7:0]	LVTTL Input, asynchronous, internal pull-up	**Latch Control Data Bus**. These inputs are passed to the output enable latch when OELE is HIGH, and captured in this latch when OELE returns LOW. These inputs are passed to the BIST enable latch when BISTLE is HIGH, and captured in this latch when BISTLE returns LOW. These inputs are passed to the Receive channel enable/decoder mode latch when RLE is HIGH, and captured in this latch when RLE returns LOW. These inputs are passed to the transmit and receive clock select latch when CLKSELLE is HIGH, and captured in this latch when CLKSELLE returns LOW. These inputs are passed to the encoder/decoder bypass latch when BYPASSLE is HIGH, and captured in this latch when BYPASSLE returns LOW. These inputs are passed to the reframe mode enable latch when RFMODELE is HIGH, and captured in this latch when RFMODELE returns LOW.
CLKSELLE	LVTTL input, asynchronous, internal pull-up	**Clock Select Latch Enable**. Selects the clock source, that is used to write data into the channel's transmit input register and the receive clock-source is used to transfer data to the output registers. When CKSELLE is HIGH, the signals on the BOE[7:0] inputs directly control the channel input and output register clock. When BOE[x] is HIGH the input or output register is clocked by the associated REFCLKx. When BOE[x] is LOW, the input register clock is TXCLKx, and the output registers are clocked by the channel's recovered clock (see *Table 4* for details).
BYPASSLE	LVTTL input, asynchronous, internal pull-up	**Encoder/Decoder Latch Enable**. When BYPASSLE = HIGH, signals on the BOE[7:0] inputs directly control the encoder and decoder enables for each channel. When BOE[x] is LOW, the encoder or decoder is bypassed and raw 10-bit characters are transmitted or received (see *Table 4* for details).
OELE	LVTTL Input, asynchronous, internal pull-up	**Serial Driver Output Enable Latch Enable**. When OELE = HIGH, the signals on the BOE[7:0] inputs directly control the OUTxy± differential drivers. When the BOE[x] input is HIGH, the associated OUTxy± differential driver is enabled. When the BOE[x] input is LOW, the associated OUTxy± differential driver is powered down. When OELE returns LOW, the last values present on BOE[7:0] are captured in the internal Output Enable latch. The specific mapping of BOE[7:0] signals to transmit output enables is listed in *Table 4*. When the latch is reset by $\overline{\text{TRSTZ}}$, all outputs are reset to disable all outputs.
RFMODELE	LVTTL Input, asynchronous, internal pull-up	**Reframe Mode Latch Enable**. When RFMODELE = HIGH, the signals on the BOE[7:0] inputs directly control the type of character framing used to adjust the character boundaries. This signal operates in conjunction with the type of framing character selected. When BOE[x,x-1] = 00, the low-latency framer is selected. This will frame on each occurrence of the selected framing character in the received data stream. This mode of framing stretches the recovered clock for one or multiple cycles to align that clock with the recovered data. When BOE[x,x-1] = 01, the alternate mode multi-byte parallel framer is selected. This requires detection of the selected framing character of the allowed disparities in the received data stream, on identical 10-bit boundaries, on four directly adjacent characters. The recovered character clock remains in the same phasing regardless of character offset. When BOE[x,x-1]=10, the Cypress-mode multi-byte parallel framer is selected. This requires a pair of the selected framing character, on identical 10-bit boundaries, within a span of 50 bits, before the character boundaries are adjusted. The recovered character clock remains in the same phasing regardless of character offset. See *Table 4*. BOE[x,x-1] = 11 is reserved for test. When RFMODELE returns low the last value present on BOE[7:0] are captures in the internal reframe mode latch. If the device is reset by $\overline{\text{TRSTZ}}$, the latch defaults to the 01 mode to activate the alternate mode multi-byte framer.
BISTLE	LVTTL Input, asynchronous, internal pull-up	**Transmit and Receive BIST Latch Enable**. When BISTLE = HIGH, the signals on the BOE[7:0] inputs directly control the transmit and receive BIST enables. When BOE[x] input is LOW, the associated transmit or receive channel is configured to generate or compare the BIST sequence. When the BOE[x] input is HIGH, the associated transmit or receive channel is configured for normal data transmission or reception. When BISTLE returns LOW, the last values present on BOE[7:0] are captured in the internal BIST Enable latch. The specific mapping of BOE[7:0] signals to transmit and receive BIST enables is listed in *Table 4*. If the device is reset by $\overline{\text{TRSTZ}}$, the latch is set to an all HIGH state to disable BIST on all transmit and receive channels.

Pin Descriptions (continued)

Name	I/O Characteristics	Signal Description
RLE	LVTTL Input, asynchronous, internal pull-up	**Receive Channel Power-Control/Decoder Special Character Table**. When RLE is HIGH, the signals on the BOE[7:0] inputs directly control the power enables for the receive PLLs, analog logic and the decoder special character table. When a BOE[6,4,2,0] input is HIGH, the PLL and analog logic associated with matching receive channel A through D are active. When a BOE[6,4,2,0] input is LOW, the PLL and analog logic associated with matching receive channel A through D are placed in a power down mode. When RLE is HIGH, the signals on the BOE[7,5,3,1] directly control which special character table the decoder will use. When BOE[x] = LOW, the alternate table is used. When BOE[x] = HIGH, the Cypress table is used. The specific mapping of BOE[7:0] signals to the associated channel is listed in *Table 4*. When RLE returns LOW, the last values present on BOE[7:0] are captured in the internal RX PLL Enable and decoder special character table latches. If the device is reset by TRSTZ, the latch is reset to disable all receive channels and disable the use of the alternate special character table.
Analog I/O and Control		
OUTA1± OUTB1± OUTC1± OUTD1±	CML Differential Output	**Primary Differential Serial Data Outputs**. These PECL-compatible CML outputs are capable of driving terminated transmission lines or standard fiber-optic transmitter modules. These outputs must be AC-coupled for PECL-compatible connections.
OUTA2± OUTB2± OUTC2± OUTD2±	CML Differential Output	**Secondary Differential Serial Data Outputs**. These PECL-compatible CML outputs are capable of driving terminated transmission lines or standard fiber-optic transmitter modules. These outputs must be AC-coupled for PECL-compatible connections.
INA1± INB1± INC1± IND1±	LVPECL Differential Input	**Primary Differential Serial Data Inputs**. These inputs accept the serial data stream for deserialization and decoding. The INx1± serial streams are passed to the receiver Clock and Data Recovery (CDR) circuits to extract the data content when INSELx = HIGH.
INA2± INB2± INC2± IND2±	LVPECL Differential Input	**Secondary Differential Serial Data Inputs**. These inputs accept the serial data stream for deserialization and decoding. The INx2± serial streams are passed to the receiver CDR circuits to extract the data content when INSELx = LOW.
INSELA INSELB INSELC INSELD	LVTTL Input, asynchronous	**Receive Input Selector**. Determines which external serial bit stream is passed to the receiver CDR circuit. When HIGH, the INx1± input is selected. When LOW, the INx2± input is selected.
SDASEL	3-Level Select [2], static configuration input	**Signal Detect Amplitude Level Select**. Allows selection of one of three predefined amplitude trip points for a valid signal indication, as listed in *Table 6*.
LPENA LPENB LPENC LPEND	LVTTL Input, asynchronous, internal pull-up	**Loop-Back-Enable**. When HIGH, the transmit serial data from the associated channel is internally routed to the associated receiver CDR circuit. All enabled serial drivers on the selected channel are forced to differential logic-1, and the serial data inputs are ignored.
$\overline{\text{LFIA}}$ $\overline{\text{LFIB}}$ $\overline{\text{LFIC}}$ LFID	LVTTL Output, synchronous to the selected RXCLKx↑ output or REFCLKx↑ [1] input, asynchronous to receive channel enable/disable	**Link Fault Indication Output**. Active LOW. $\overline{\text{LFI}}$ is the logical OR of four internal conditions: 1. received serial data frequency outside expected range; 2. analog amplitude below expected levels; 3. transition density lower than expected; and 4. receive channel disabled.
JTAG Interface		
TMS	LVTTL Input, internal pull-up	**Test Mode Select**. Used to control access to the JTAG Test Modes. If maintained HIGH for ≥ 5 TCLK cycles, the JTAG test controller is reset.
TCLK	LVTTL Input, internal pull-down	JTAG Test Clock
TDO	Three-State LVTTL Output	**Test Data Out**. JTAG data output buffer which is High-Z while JTAG test mode is not selected.

Pin Descriptions (continued)

Name	I/O Characteristics	Signal Description
TDI	LVTTL Input, internal pull-up	**Test Data In**. JTAG data input port.
$\overline{\text{TRSTZ}}$	LVTTL Input, internal pull-up	**Test Port and Device Reset**. Active LOW. Initializes the JTAG controller and all state machines and counters in the device. When asserted (LOW), this input asynchronously resets the JTAG test access port controller. When sampled LOW by the rising edge of REFCLKx, this input resets the internal state machines and sets the Elasticity Buffer pointers to a nominal offset. When the reset is removed ($\overline{\text{TRSTZ}}$ sampled HIGH by REFCLKx↑), the status and data outputs will become deterministic in less than 16 REFCLKx cycles. The BISTLE, OELE, CLKSELLE, BYPASSLE and RLE latches are reset by $\overline{\text{TRSTZ}}$.
Power		
V_{CC}		+3.3V power
GND		Signal and power ground for all internal circuits

PIN	PIN NAME	PIN TYPE
A01	INC1-	Serial Differential Receiver (-)
A02	OUTC1-	CML Differential Output (-)
A03	INC2-	Serial Differential Receiver (-)
A04	OUTC2-	CML Differential Output (-)
A05	VCC	
A06	IND1-	Serial Differential Receiver (-)
A07	OUTD1-	CML Differential Output (-)
A08	GND	
A09	IND2-	Serial Differential Receiver (-)
A10	OUTD2-	CML Differential Output (-)
A11	INA1-	Serial Differential Receiver (-)
A12	OUTA1-	CML Differential Output (-)
A13	GND	
A14	INA2-	Serial Differential Receiver (-)
A15	OUTA2-	CML Differential Output (-)
A16	VCC	
A17	INB1-	Serial Differential Receiver (-)
A18	OUTB1-	CML Differential Output (-)
A19	INB2-	Serial Differential Receiver (-)
A20	OUTB2-	CML Differential Output (-)
B01	INC1+	Serial Differential Receiver (+)
B02	OUTC1+	CML Differential Output (+)
B03	INC2+	Serial Differential Receiver (+)
B04	OUTC2+	CML Differential Output (+)
B05	VCC	
B06	IND1+	Serial Differential Receiver (+)
B07	OUTD1+	CML Differential Output (+)
B08	GND	
B09	IND2+	Serial Differential Receiver (+)
B10	OUTD2+	CML Differential Output (+)
B11	INA1+	Serial Differential Receiver (+)
B12	OUTA1+	CML Differential Output (+)
B13	GND	
B14	INA2+	Serial Differential Receiver (+)
B15	OUTA2+	CML Differential Output (+)
B16	VCC	
B17	INB1+	Serial Differential Receiver (+)
B18	OUTB1+	CML Differential Output (+)
B19	INB2+	Serial Differential Receiver (+)
B20	OUTB2+	CML Differential Output (+)
C01	TDI	LVTTL input with pull-up
C02	TMS	LVTTL input w/ pull-up
C03	INSELC	LVTTL input
C04	INSELB	LVTTL input
C05	VCC	
C06	FRAMCHARB	3-level select
C07	FRAMCHARC	3-level select
C08	GND	
C09	BOE<7>	LVTTL input w/ pull-up
C10	BOE<5>	LVTTL input w/ pull-up
C11	BOE<3>	LVTTL input w/ pull-up
C12	BOE<1>	LVTTL input w/ pull-up
C13	GND	
C14	SDASEL	3-level select
C15	SPDSELD	3-level select
C16	VCC	
C17	TXRATE	LVTTL input w/ pull-down
C18	RXRATE	LVTTL input w/ pull-down
C19	LPEND	LVTTL input w/ pull-down
C20	TDO	three-state LVTTL Output
D01	TCLK	LVTTL input w/ pull-down
D02	/TRSTZ	LVTTL input w/ pull-up
D03	INSELD	LVTTL input
D04	INSELA	LVTTL input
D05	VCC	
D06	FRAMCHARA	3-level select
D07	SPDSELC	3-level select
D08	GND	
D09	BOE<6>	LVTTL input w/ pull-up
D10	BOE<4>	LVTTL input w/ pull-up
D11	BOE<2>	LVTTL input w/ pull-up
D12	BOE<0>	LVTTL input w/ pull-up
D13	GND	
D14	LPENB	LVTTL input w/ pull-down
D15	FRAMCHARD	3-level select
D16	VCC	
D17	LPENA	LVTTL input w/ pull-down
D18	RFENA	LVTTL input w/ pull-down
D19	RFENB	LVTTL input w/ pull-down
D20	RFMODELEL	VTTL input w/pull up
E01	VCC	
E02	VCC	
E03	VCC	
E04	VCC	
E17	VCC	
E18	VCC	
E19	VCC	
E20	VCC	
F01	TXERRC	LVTTL output
F02	RFEND	LVTTL input w/ pull-down

PIN	PIN NAME	PIN TYPE
F03	TXDC[0]	LVTTL input
F04	RFENC	LVTTL input w/ pull-down
F17	BISTLE	LVTTL input w/ pull-up
F18	RXSTB[1]	LVTTL output
F19	TXCLKOB	LVTTLOUT_FAST
F20	RXSTB[0]	LVTTL output
G01	TXDC[7]	LVTTL input
G02	CLKSELLE	LVTTL input w/ pull-up
G03	TXDC[4]	LVTTL input
G04	TXDC[1]	LVTTL input
G17	SPDSELB	3-level select.
G18	LPENC	LVTTL input w/ pull-down
G19	SPDSELA	3-level select.
G20	RXDB[1]	LVTTL output
H01	GND	
H02	GND	
H03	GND	
H04	GND	
H17	GND	
H18	GND	
H19	GND	
H20	GND	
J01	TXCTC[1]	LVTTL input
J02	TXDC[5]	LVTTL input
J03	TXDC[2]	LVTTL input
J04	TXDC[3]	LVTTL input
J17	RXSTB[2]	LVTTL output
J18	RXDB[0]	LVTTL output
J19	RXDB[5]	LVTTL output
J20	RXDB[2]	LVTTL output
K01	RXDC[2]	LVTTL output
K02	REFCLKC-	Differential reference clock input (-)
K03	TXCTC[0]	LVTTL input
K04	/LFIC	LVTTL output
K17	RXDB[3]	LVTTL output
K18	RXDB[4]	LVTTL output
K19	RXDB[7]	LVTTL output
K20	RXCLKB+	LVTTLOUT_FAST_ind
L01	RXDC[3]	LVTTL output
L02	REFCLKC+	Differential reference clock input (+)
L03	TXCLKC	LVTTL input w/ pull-down
L04	TXDC[6]	LVTTL input
L17	RXDB[6]	LVTTL output
L18	/LFIB	LVTTL output
L19	RXCLKB-	LVTTLOUT_FAST
L20	TXDB[6]	LVTTL input
M01	RXDC[4]	LVTTL output
M02	RXDC[5]	LVTTL output
M03	RXDC[7]	LVTTL output
M04	RXDC[6]	LVTTL output
M17	REFCLKB+	Differential reference clock input (+)
M18	REFCLKB-	Differential reference clock input (-)
M19	TXDB[7]	LVTTL input
M20	TXCLKB	LVTTL input w/ pull-down
N01	GND	
N02	GND	
N03	GND	
N04	GND	
N17	GND	
N18	GND	
N19	GND	
N20	GND	
P01	RXDC[1]	LVTTL output
P02	RXDC[0]	LVTTL output
P03	RXSTC[0]	LVTTL output
P04	RXSTC[1]	LVTTL output
P17	TXDB[5]	LVTTL input
P18	TXDB[4]	LVTTL input
P19	TXDB[3]	LVTTL input
P20	TXDB[2]	LVTTL input
R01	RXSTC[2]	LVTTL output
R02	TXCLKOC	LVTTLOUT_FAST
R03	RXCLKC+	LVTTLOUT_FAST
R04	RXCLKC-	LVTTLOUT_FAST
R17	TXDB[1]	LVTTL input
R18	TXDB[0]	LVTTL input
R19	TXCTB[1]	LVTTL input
R20	TXERRB	LVTTL output
T01	VCC	
T02	VCC	
T03	VCC	
T04	VCC	
T17	VCC	
T18	VCC	
T19	VCC	
T20	VCC	
U01	TXDD[0]	LVTTL input
U02	TXDD[1]	LVTTL input
U03	TXDD[2]	LVTTL input
U04	TXCTD[1]	LVTTL input

PIN	PIN NAME	PIN TYPE
U05	VCC	
U06	RXDD[2]	LVTTL output
U07	RXDD[1]	LVTTL output
U08	GND	
U09	TXERRD	LVTTL output
U10	RLE	LVTTL input w/ pull up
U11	REFCLKD-	Differential reference clock input (-)
U12	TXDA[1]	LVTTL input
U13	GND	
U14	TXDA[4]	LVTTL input
U15	TXCTA[0]	LVTTL input
U16	VCC	
U17	RXDA[2]	LVTTL output
U18	TXCTB[0]	LVTTL input
U19	RXSTA[2]	LVTTL output
U20	RXSTA[1]	LVTTL output
V01	TXDD[3]	LVTTL input
V02	TXDD[4]	LVTTL input
V03	TXCTD[0]	LVTTL input
V04	RXDD[6]	LVTTL output
V05	VCC	
V06	RXDD[3]	LVTTL output
V07	RXSTD[0]	LVTTL output
V08	GND	
V09	RXSTD[2]	LVTTL output
V10	BYPASSLE	LVTTL input w/ pull up
V11	REFCLKD+	Differential reference clock input (+)
V12	TXCLKOA	LVTTLOUT_FAST_ind
V13	GND	
V14	TXDA[3]	LVTTL input
V15	TXDA[7]	LVTTL input
V16	VCC	
V17	RXDA[7]	LVTTL output
V18	RXDA[3]	LVTTL output
V19	RXDA[0]	LVTTL output
V20	RXSTA[0]	LVTTL output
W01	TXDD[5]	LVTTL input
W02	TXDD[7]	LVTTL input
W03	/LFID	LVTTL output
W04	RXCLKD-	LVTTLOUT_FAST
W05	VCC	
W06	RXDD[4]	LVTTL output
W07	RXSTD[1]	LVTTL output
W08	GND	
W09	OELE	LVTTL input w/ pull up
W10	/TXRST	LVTTL input w/ pull-up
W11	RXCLKA+	LVTTLOUT_FAST
W12	TXERRA	LVTTL output
U13	GND	
W14	TXDA[2]	LVTTL input
W15	TXDA[6]	LVTTL input
W16	VCC	
W17	/LFIA	LVTTL output
W18	REFCLKA+	Differential reference clock input (+)
W19	RXDA[4]	LVTTL output
W20	RXDA[1]	LVTTL output
Y01	TXDD[6]	LVTTL input
Y02	TXCLKD	LVTTL input w/ pull-down
Y03	RXDD[7]	LVTTL output
Y04	RXCLKD+	LVTTLOUT_FAST_ind
Y05	VCC	
Y06	RXDD[5]	LVTTL output
Y07	RXDD[0]	LVTTL output
Y08	GND	
Y09	TXCLKOD	LVTTLOUT_FAST
Y10	NC	N/C, VGND pad place holder
Y11	TXCLKA	LVTTL input w/pull down
Y12	RXCLKA-	LVTTLOUT_FAST
Y13	GND	
Y14	TXDA[0]	LVTTL input
Y15	TXDA[5]	LVTTL input
Y16	VCC	
Y17	TXCTA[1]	LVTTL input
Y18	REFCLKA-	Differential reference clock input (-)
Y19	RXDA[6]	LVTTL output
Y20	RXDA[5]	LVTTL output

CYP15G0403DX HOTLink II Operation

The CYP15G0403DX is a configurable device designed to transfer of large quantities of data, using high-speed serial links. This device supports four single-byte channels.

CYP15G0403DX Transmit Data Path

Input Register

The bits in the Input Register for each channel support different bit assignments, based on if the input data is encoded or unencoded. These assignments are shown in *Table 1*.

Table 1. Input Register Bit Assignments [3]

Signal Name	Unencoded	Encoded
TXDx[0] (LSB)	DINx[0]	TXDx[0]
TXDx[1]	DINx[1]	TXDx[1]
TXDx[2]	DINx[2]	TXDx[2]
TXDx[3]	DINx[3]	TXDx[3]
TXDx[4]	DINx[4]	TXDx[4]
TXDx[5]	DINx[5]	TXDx[5]
TXDx[6]	DINx[6]	TXDx[6]
TXDx[7]	DINx[7]	TXDx[7]
TXCTx[0]	DINx[8]	TXCTx[0]
TXCTx[1] (MSB)	DINx[9]	TXCTx[1]

Note:
3. LSB shifted out first.

Each input register captures eight data bits and two control bits on each input clock cycle. When the encoder is bypassed, the control bits are part of the pre-encoded 10-bit character.

When the Encoder is enabled, the TXCTx[1:0] bits are interpreted along with the associated TXDx[7:0] character to generate a specific 10-bit transmission character.

Data from each input register is passed to the associated Phase-Align buffer. These buffers are used to absorb clock phase differences between the selected input clock and the internal character clock.

Initialization of these phase-align buffers takes place when the $\overline{\text{TXRST}}$ input is sampled LOW by TXCLKx↑. When $\overline{\text{TXRST}}$ is returned HIGH, the present input clock phase relative to REFCLKx↑ is set. $\overline{\text{TXRST}}$ is an asynchronous input, but is sampled by each TXCLKx↑ to synchronize it to the internal transmit path state machines. $\overline{\text{TXRST}}$ must be sampled LOW by a min.-mum of two consecutive TXCLKx↑ clocks to ensure the reset operation is initiated correctly on the associated channels.

Once the phase relationship if TXCLKx is set, these clocks are allowed to skew in phase up to half a character period in either direction relative to REFCLKx↑; i.e., ±180°. This phase shift allows the delay paths of the character clocks to change due to operating voltage and temperature.

If the phase offset, between the initialized location of the input clock and REFCLKx↑, exceeds the skew handling capabilities of the Phase-Align Buffer, an error is reported on that channel TXERRx output. This output indicates a continuous error until the Phase-Align Buffer is reset. Until the buffer fault is cleared, the transmitter for the associated channel outputs a continuous C0.7 character to indicate to the remote receiver that an error condition is present in the link.

It is possible to reset each Phase-Align Buffer individually and with minimal disruption of the serial data stream. When a Phase-Align Buffer error is present, the transmission of a Word Sync Sequence will re-center the buffer and will clear the error flag condition. **Note**. K28.5 characters may be added or lost from the data stream during reset operation. When used with non-Cypress devices that require a complete 16-character Word Sync Sequence for proper receive Elasticity Buffer operation, it is recommend that the reset sequence be followed by a Word Sync Sequence to ensure proper operation.

Encoder

Each character, received from the input register or phase-align buffer, is then passed to the Encoder logic. This block interprets each character and any associated control bits, and outputs a 10-bit transmission character.

The operational mode controls the generated transmission character

- the 10-bit preencoded character accepted in the input register
- the 10-bit equivalent of the 8-bit Data character accepted in the input register
- the 10-bit equivalent of the 8-bit Special Character code accepted in the input register
- the 10-bit equivalent of the C0.7 violation character if a Phase-Align Buffer overflow or underflow error is present
- a character that is part of the 511-character BIST sequence
- a K28.5 character generated as an individual character or as part of the 16-character Word Sync sequence.

Data Encoding

Raw data, as received from the Transmit Input Register must be processed to guarantee

- a minimum transition density to ensure low noise clock extraction from the data stream.
- to reduce DC term in the original data stream.
- run-length limits to reduce the low frequency bandwidth requirements of the serial stream.
- to provide a means to framing the remote receiver.

When the Encoder is enabled, the characters to be transmitted are converted from Data or Special Character to 10-bit transmission characters, using an integrated 8B/10B encoder. When directed to encode the character as a Special Character code, the encoder will use the Special Character encoding rules listed in *Table 15*. When directed to encode the character as a Data character, it is encoded using the Data Character encoding rules in *Table 14*.

The 8B/10B encoder is standards-compliant with ANSI/NCITS ASC X3.230-1994 (Fibre Channel), IEEE 802.3z Gigabit Ethernet, the IBM® ESCON® and FICON™ channels, and ATM Forum standards for data transport.

The Special Character codes that may be generated are listed in *Table 15*.

The CYP15G0403DX is designed to support two independent Special Character code tables. This allows the CYP15G0403DX to operate in mixed environments with other CYP15G0403DXs using the enhanced Cypress command code set, or the command sets of other devices. Even when used in an environment that uses non-Cypress Special Character codes, the selective use of Cypress command codes can permit operation where running disparity and error handling must be managed.

Following conversion of each input character from 8 bits to a 10-bit transmission character, it is passed to the Transmit Shifter and is shifted out LSB first, as required by ANSI and IEEE standards for 8B/10B coded serial data streams.

Transmit Modes

Encoder Bypass

When the Encoder is bypassed, the character captured in the TXDx[7:0] and TXCTx[1:0] inputs is passed directly to the transmit shifter without modification. With the encoder bypassed, the TXCTx[1:0] inputs are considered part of the data character and do not perform a control function on the TXDx[7:0] bits. The bit usage and mapping of these control bits when the Encoder is bypassed is shown in *Table 2*.

Table 2. Encoder Bypass Mode

Signal Name	Bus Weight	10B Name
TXDx[0] (LSB)	2^0	$a^{[3]}$
TXDx[1]	2^1	b
TXDx[2]	2^2	c
TXDx[3]	2^3	d
TXDx[4]	2^4	e
TXDx[5]	2^5	i
TXDx[6]	2^6	f
TXDx[7]	2^7	g
TXCTx[0]	2^8	h
TXCTx[1] (MSB)	2^9	j

When the encoder is enabled, the TXCTx[1:0] data control bits control the interpretation of the TXDx[7:0] bits and the characters generated by them. These bits are interpreted as listed in *Table 3*.

Table 3. Transmit modes

TXCTx[1]	TXCTx[0]	Characters Generated
0	0	Encoded data character
0	1	K28.5 fill character
1	0	Special character code
1	1	16-character Word Sync sequence

Word Sync Sequence

When TXCTx[1:0] = 11, a 16-character sequence of K28.5 characters, known as a Word Sync Sequence, is generated on the associated channel. This sequence of K28.5 characters may start with either a positive or negative disparity K28.5. The disparity of the second and third K28.5 characters in this sequence are reversed from what normal 8B/10B coding rules would generate. The remaining K28.5 characters in the sequence follow all 8B/10B coding rules. The disparity of the generated K28.5 characters in this sequence follow a pattern of either ++−−+−+−+−+−+− or −−++−+−+−+−+−+. The generation of this character, once it has been started, it cannot be stopped until all 16 characters have been sent. The content of the associated input registers are ignored for the duration of this 16-character sequence. At the end of this sequence, if the TXCTx[1:0] = 11 the K28.5 character sequence generation is started again.

Transmit BIST

The transmitter interfaces contain internal pattern generators that can be used to validate link operation. These generators are enabled by the associated BOE[x] signals as listed in *Table 4* when the BISTLE latch enable input is HIGH. When enabled, a register in the associated transmit channel becomes a pattern generator by converting to a Linear Feedback Shift Register (LFSR). This LFSR generates a 511-character sequence that includes all Data and Special Character codes, including the explicit violation symbols. This provides a predictable yet pseudo-random sequence that can be matched to an identical LFSR in the attached Receiver.

When the BISTLE signal is HIGH, any BOE[x] input that is LOW enables the BIST generator in that transmit channel or the BIST checker in the receive channel. When BISTLE returns LOW, the values of all BOE[x] signals are captured in the BIST Enable Latch. These values remain in the BIST Enable Latch until BISTLE is returned high. When the system is reset by TRSTZ, the default is to disable BIST.

All data and data-control information present at the associated TXDx[7:0] and TXCTx[1:0] inputs are ignored when BIST is active. If the receive channels are configured for reference clock operation, each pass is preceded by a 16-character Word Sync Sequence that provides time for Elasticity Buffer centering and clock frequency adjustments.

Serial Output Drivers

The serial outputs use differential CML drivers to provide a source-matched driver for transmission lines. These drivers accept data from the Transmit Shifters. These outputs have signal swings equivalent to that of standard PECL drivers, and are capable of driving AC-coupled optical modules or AC-coupled transmission lines.

Each output can be enabled or disabled separately through the BOE[7:0] input, as controlled by the OELE latch-enable signal. When OELE is HIGH, the signals present on the BOE[7:0] inputs are passed through the Serial Output Enable latch to control the serial output drivers. The BOE[7:0] input associated with a specific OUTxy± driver is listed in *Table 4*.

When OELE is HIGH and BOE[x] is HIGH, the associated serial driver is enabled.

When OELE is HIGH and BOE[x] is LOW, the associated driver is powered down. If both outputs for a channel are disabled, the internal logic for that channel is also powered down.

When OELE returns LOW, the values present on the BOE[7:0] inputs are latched in the Output Enable Latch, and remain there until OELE returns HIGH. When a disabled transmit channel is re-enabled, the data on the serial outputs may not meet all timing specifications for up to 10ms.

Table 4. Control Latches Signal Map

BOE Input	Output Controlled (OELE)	BIST Channel Enable (BISTLE)	Receive PLL Enable Decoder Code (RLE)	Encoder/ Decoder (BYPASSLE)	Clock Select Enable (CLKSELLE)	Reframe Mode (RFMODELE)
BOE[7]	OUTD2±	Transmit D	Decoder D	Transmit D	Transmit D	ReframeD1
BOE[6]	OUTD1±	Receive D	RXPLL D	Receive D	Receive D	ReframeD0
BOE[5]	OUTC2±	Transmit C	Decoder C	Transmit C	Transmit C	Reframe C1
BOE[4]	OUTC1±	Receive C	RXPLL C	Receive C	Receive C	ReframeC0
BOE[3]	OUTB2±	Transmit B	Decoder B	Transmit B	Transmit B	Reframe B1
BOE[2]	OUTB1±	Receive B	RXPLL B	Receive B	Receive B	ReframeB0
BOE[1]	OUTA2±	Transmit A	Decoder A	Transmit A	Transmit A	Reframe A1
BOE[0]	OUTA1±	Receive A	RXPLL A	Receive A	Receive A	Reframe A0

Transmit PLL Clock Multiplier

Each Transmit PLL Clock Multiplier accepts a character-rate or half-character-rate external clock for REFCLKx input, and that clock is multiplied by 10 or 20, as selected by TXRATE, to generate a bit-rate clock for use by the transmit shifter. It also provides a character-rate clock used by the transmit paths, and outputs this character rate clock as TXCLKOx.

Each clock multiplier PLL can accept a REFCLKx input between 10 MHz and 150 MHz, however, this clock range is limited by the operating mode of the CYP15G0403DX clock multiplier TXRATE and by the level on the associated SPDSEL[A:D] input.

SPDSEL[A:D] inputs that selects one of three operating ranges for the serial data outputs and inputs of the associated channel. The operating serial signaling-rate and allowable range of REFCLKx frequencies are listed in *Table 5*.

Table 5. Operating Speed Settings

SPDSELx	TXRATE	REFCLKx Frequency (MHz)	Signaling Rate (MBaud)
LOW	1	20	200–400
	0	20–40	
MID (Open)	1	20–40	400–800
	0	40–80	
HIGH	1	40–75	800–1500
	0	80–150	

The REFCLKx± inputs are differential inputs with each input internally biased to $V_{CC}/2$. The REFCLKx input interfaces directly to TTL, LVTTL, or LVCMOS clock source.

When both the REFCLKx+ and REFCLKx– inputs are connected, the clock source must be a differential clock. This can be a differential LVPECL clock that is DC-or AC-coupled.

By connecting the REFCLKx– input to an external voltage source, it is possible to adjust the reference point of the REFCLKx+ input for alternate logic levels.

CYP15G0403DX Receive Data Path

Serial Line Receivers

Two differential line receivers, INx1± and INx2±, are available on each channel for accepting serial data streams. The active line receiver on a channel is selected using the associated INSELx input. The serial line receiver inputs are differential and only require is a 200mV peak-to-peak signal. It can be DC- or AC-coupled to +3.3V powered fiber-optic interface modules with a PECL output. 5V optical modules should be AC-coupled. The common-mode tolerance of these line receivers will accommodates a wide range of input signal voltages. For AC-coupled signals the receiver provides an average value DC restoration.

The local internal loopback allows the serial transmit data outputs to be routed internally back to the Clock and Data Recovery circuit associated with each channel. When configured for local loopback, all transmit serial driver outputs are forced to output a differential logic-1. This prevents local diagnostic patterns from being broadcast to attached remote receivers.

Signal Detect/Link Fault

Each selected Line Receiver is simultaneously monitored for

- analog amplitude
- transition density
- range controls reporting the received data stream inside a normal frequency range (±200 ppm)
- receive channel enabled.

All of these conditions must be valid for the Signal Detect block to indicate a valid signal is present. This status is presented on the LFIx (Link Fault Indicator) output associated with each receive channel, which changes synchronous to the selected receive interface clock.

Analog Amplitude

The analog amplitude level detection is adjustable to allow operation with highly attenuated signals, or in high-noise environments. This adjustment is made through the SDASEL signal, a 3-level select[2] input, which sets the trip point for the detection of a valid signal at one of three levels, as listed in *Table 6*. This control input effects the analog monitors for all receive channels.

Table 6. Analog Amplitude Detect Valid Signal Levels

SDASEL	Typical signal with peak amplitudes above
LOW	140 mV p-p differential
MID (Open)	280 mV p-p differential
HIGH	420 mV p-p differential

The Signal Detect monitors are active for the present line receiver, as selected by the associated INSELx input. When configured for local loopback, no input receivers are selected, and the LFI output for each channel reports only the receive VCO frequency out-of-range and transition density status of the associated transmit signal. In local loopback the associated analog amplitude monitor is disabled.

Transition Density

The transition detection logic checks for the absence of transitions spanning greater than six transmission characters or 60 bits. If no transitions are present in the data received, the transition detection logic for that channel will assert LFIx. The LFIx output remains asserted until at least one transition is detected in each of three adjacent received characters.

Range Controls

The receive-VCO range-control monitors report the frequency status of the received signal. They also determine if the receive CDR circuits should align the receive VCO clock to the data stream or to the associated REFCLKx input. This function prevents the receive VCO from tracking an out-of-specification received signal.

When the range-control monitor for a channel indicates that the signaling rate is within specification, the phase detector in the receive PLL is configured to track the transitions in the received data stream. In this mode the LFIx output for the associated channel is HIGH. If the range-control monitor indicates that the received data stream signaling-rate is out of specification, the phase detector is configured to track the local REFCLKx input, and the associated LFIx output is asserted LOW.

The specific trip points for this compare function are listed in *Table 7*. Because the compare function operates with two asynchronous clocks, there is a small uncertainty in the measurement. The switch points are asymmetric to provide operational hysteresis.

Table 7. Receive Signaling Rate Range Control criteria

Current RX PLL Tracking Source	Frequency Difference Between Transmit Character Clock & RX VCO	Next RX PLL Tracking Source
Selected data stream (LFIx = HIGH)	< 1708 ppm	Data Stream
	1708–1953 ppm	Indeterminate
	> 1953 ppm	REFCLKx
REFCLKx (LFIx = LOW)	< 488 ppm	Data Stream
	488–732 ppm	Indeterminate
	> 732 ppm	REFCLKx

Receive Channel Enabled

The CYP15G0403DX contains four receive channels that can be independently enabled and disabled. Each channel can be enabled or disabled separately through the BOE[6,4,2,0] inputs, as controlled by the RLE latch-enable signal. When RLE is HIGH, the signals present on the BOE[] inputs are passed through the Receive Channel Enable latch to control the PLLs and logic of the associated receive channel. The BOE[] input associated with a specific receive channel is listed in *Table 4*.

When RLE is HIGH and BOE[x] is HIGH, the associated receive channel enabled to receive and decode a serial stream. When RLE is HIGH and BOE[x] is LOW, the associated receive channel is disabled and powered down. Any disabled channel will indicate a constant link fault condition on the LFIx output. When RLE returns LOW, the values present on the BOE[6,4,2,0] inputs are latched in the Receive Channel Enable Latch, and remain there until RLE returns HIGH to to resample the input again. **Note**. When a disabled receive channel is re-enabled, the status of the associated LFIx output and data on the parallel outputs for the associated channel may be indeterminate for up to 10ms.

Clock and Data Recovery

The extraction of a bit-rate clock and recovery of bits from each received serial stream is performed by a separate CDR block within each receive channel. The clock extraction function is performed by embedded phase-locked loops that track the frequency and phase of the transitions in the incoming bit streams.

Each CDR accepts a character-rate or half-character-rate reference clock from the associated REFCLKx input. This REFCLKx input is used

- to ensure that the VCO in each CDR is operating at the correct frequency.
- to improve PLL acquisition time.
- to limit unlocked frequency excursions of the VCO when no data is present at the selected serial inputs.

Regardless of the type of signal present, the CDR will attempt to recover a data stream from it. If the frequency of the recovered data stream is outside the limits set by the range control monitors, the CDR PLL will track REFCLKx instead of the data stream. When the frequency of the selected data stream returns to a valid frequency, the CDR PLL is allowed to track the received data stream. The frequency of REFCLKx is required to be within ±200 ppm of the frequency of the clock that drives the REFCLK input of the *remote* transmitter to ensure a lock to the incoming data stream.

For systems using multiple or redundant connections, the LFIx output can be used to select an alternate data stream. When an LFIx indication is detected, external logic can toggle selection INx1± and INx2± inputs. When a port switch takes place, the receive PLL must reacquire the new serial stream and frame to the incoming character boundaries.

Deserializer/Framer

Each CDR circuit extracts bits from the associated serial data stream and clocks these bits into the Shifter/Framer at the bit-clock rate. When enabled, the Framer examines the data stream looking for one or more COMMA or K28.5 characters at all possible bit positions. The location of this character in the data stream is used to determine the character boundaries of all following characters.

Framing Character

The CYP15G0403DX allows selection of different framing characters on each channel. Three combinations of framing characters are supported to meet the requirements of different interfaces. The selection of the framing character is made through the FRAMCHARx inputs.

The FRAMCHARx signals are 3-level select [2] inputs that allow selection of one of three different framing characters or character combinations. The specific bit combinations of these framing characters are listed in *Table 8*. When the specific bit combination of the selected framing character is detected by the framer, the boundaries of the characters present in the received data stream are known.

Table 8. Framing Character Selector

FRAMCHARx	Bits detected in framer	
	Character Name	Bits Detected
LOW	COMMA+	00111110XX [4]
MID	COMMA+ COMMA–	00111110XX [4] or 11000001XX
HIGH	–K28.5 +K28.5	0011111010 or 1100000101

Note:

4. The standard definition of a COMMA contains only seven bits. However, since all valid COMMA characters within the 8B/10B character set also have the eighth bit as an inversion of the seventh bit, the compare pattern is extended to a full 8 bits to reduce the possibility of a framing error.

Framer

The framer on each channel operates in one of three different modes. The framer is controlled by RFENx. When the framer is disabled, no combination of received bits will alter the frame information.

When the low-latency framer is selected (BOE[x,x-1] = 00), the framer operates by stretching the recovered character clock until it aligns with the received character boundaries. In this mode the framer starts its alignment process on the first detection of the selected framing character. To reduce the impact on external circuits that use the recovered clock, the clock period is not stretched by more than two bit-periods. When operated with a character-rate output clock, the output of properly framed characters may be delayed by up to nine character-clock cycles from the detection of the selected framing character. When operated with a half-character-rate output clock, the output of properly framed characters may be delayed by up to 14 character-clock cycles from the detection of the framing character.

When BOE(x,x-1) = 1x, the Cypress-mode multi-byte framer is selected. The required detection of multiple framing characters makes the associated link much more robust. In this mode, the framer does not adjust the character clock boundary, but instead aligns the character to the already recovered character clock. This ensures that the recovered clock will not contain any phase changes during normal operation. This process allows the recovered clock to be replicated and distributed to other external circuits. In this framing mode the character boundaries are only adjusted 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.

When BOE(x,x-1) = 01, the alternate-mode multi-byte framer is enabled. Like the Cypress-mode multi-byte framer, multiple framing characters must be detected before the character boundary is adjusted. 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.

10B/8B Decoder Block

The decoder logic block performs three primary functions:

- decoding the received transmission characters to Data and Special Character codes
- comparing generated BIST patterns with received characters to permit at-speed link testing.

10B/8B Decoder

The framed parallel output of each deserializer shifter is passed to the 10B/8B decoder where the input data is transformed from a 10-bit transmission character back to the original Data and Special Character code. This block uses the 10B/8B decoder patterns in *Table 14* and *Table 15.* Received Special Code characters are decoded using *Table 15*. Valid data characters are indicated by a 000b bit-combination on the associated RXSTx[2:0] status bits, and Special Character codes are indicated by a 001b bit-combination of these status bits. Framing characters, invalid patterns, disparity errors, and synchronization status are presented as alternate combinations of these status bits.

The 10B/8B decoder uses one of two look-up tables, and it could be bypassed.

Receive BIST Operation

The receiver interfaces contain internal pattern generators that can be used to validate system operation. These generators are enabled by the associated BOE[x] signals listed in *Table 4* when the BISTLE latch enable input is HIGH. When enabled, a register in the associated receive channel becomes a signature pattern generator and checker by logically converting to a Linear Feedback Shift Register (LFSR). This LFSR generates a 511-character sequence that includes all Data and Special Character codes, including the explicit violation symbols. This provides a predictable yet pseudo-random sequence that can be matched to an identical LFSR in the attached Transmitter. When synchronized with the received data stream, the associated receiver checks each character in the Decoder with each character generated by the LFSR and indicates compare errors and BIST status at the RXSTx[2:0] bits of the output register.

When the BISTLE signal is HIGH, any BOE[x] input that is LOW enables the BIST generator/checker in the selected receive channel. When BISTLE returns LOW, the values of all BOE[x] signals are captured in the BIST Enable Latch. These values remain in the BIST Enable Latch until BISTLE is returned high to sample input again. All captured signals in the BIST Enable Latch are set HIGH and BIST is disabled following a device reset by TRSTZ.

The LFSR is initialized by the BIST hardware once the BIST enabled for that receive channel. This sets the BIST LFSR to the BIST-loop start-code of D0.0. The code D0.0 is sent only once per BIST loop. The status of the BIST and any character mismatches are presented on the RXSTx[2:0] status outputs.

Code rule violations or running disparity errors that occur as part of the BIST loop do not cause an error indication. RXSTx[2:0] indicates 010b or 100b for one character period per BIST loop to indicate loop completion. This status signal can be used to check BIST pattern progress. These same status values are presented even when the decoder is bypassed and BIST is enabled on a receive channel.

The specific status reported by the BIST state machine are listed in *Table 11*. These same codes are reported on the receive status outputs.

The specific patterns checked by each receiver are described in detail in the Cypress application note “HOTLink Built-In Self-Test.” The sequence compared by the CYP15G0403DX is identical to that in the CY7B933, CY7C924DX and CYP15G0401DX allowing interoperable systems to be built when used at compatible serial signaling rates.

If the number of invalid characters received ever exceeds the number of valid characters by 16, the receive BIST state machine aborts the compare operations and resets the LFSR to the D0.0 state.

When the receive paths are configured for common clock operation, each pass must be preceded by a 16-character Word Sync Sequence to allow output buffer alignment and management of clock frequency variations.

The BIST state machine requires the characters to be correctly framed for it to detect the BIST sequence. If the framer is enabled and configured for low-latency operation, the framer can align to characters within the BIST sequence. If either of the multi-byte framers are enabled, it is generally necessary to frame the receiver before BIST is enabled, and then disable framing. If the receive outputs are clocked relative to REFCLKx, the transmitter precedes every 511 character BIST sequence with a 16-character Word Sync Sequence.

Receive Elasticity Buffer

Each receive channel contains an Elasticity Buffer that is designed to support clock tolerance management. These buffers allow data to be read using an Elasticity Buffer Read clock that is asynchronous in both frequency and phase from the Elasticity Buffer Write clock.

If the chip is configured for operation with a recovered clock, the elasticity buffer is bypassed.

Each Elasticity Buffer is a minimum of 5 characters deep, and supports a 11-bit-wide data path. It is capable of supporting a decoded character and three status bits for each character present in the buffer. The Write clock for these buffers is always the recovered clock for the associated Read channel.

The Read clock for the Elasticity Buffers may come from two selectable sources:

- a character-rate REFCLKx↑
- a recovered clock from the receive channel.

Receive Modes

When the receive channel is clocked by REFCLKx, the RXCLKx± outputs present a buffered and delayed form of REFCLKx. In this mode, the receive Elasticity Buffers are enabled. For REFCLKx clocking, for example, the Elasticity Buffers are enabled to insert K28.5 characters and delete framing characters as appropriate.

The insertion of a K28.5 or deletion of a framing character can occur at any time on any channel, however, the actual timing on these insertions and deletions is controlled in part by the how the transmitter sends its data. Insertion of a K28.5 character can only occur when the receiver has a framing character in the Elasticity Buffer. Likewise, to delete a framing character, one must also be in the Elasticity Buffer. To prevent a buffer overflow or underflow on a receive channel, a minimum density of framing characters must be present in the received data streams.

The Elasticity buffer may be reset by a device reset operation initiated through the TRSTZ input, however, following such an event the CYP15G0403DX will normally require a framing event before it will correctly decode characters.

When the receive channel output register is clocked by a recovered clock, no characters are added or deleted; the receiver Elasticity Buffer is bypassed.

Each receive channel presents an 11-signal output bus consisting of an 8-bit data bus and a 3-bit status bus.

The signals present on this output bus are modified by the present operating mode of the CYP15G0403DX as selected BYPASSLE. This mapping is shown in *Table 9*.

Table 9. Output Register Bit Assignments

Signal Name	BYPASS ACTIVE	DECODER
RXSTx[2] (LSB)	COMDETx	RXSTx[2]
RXSTx[1]	DOUTx[0] [5]	RXSTx[1]
RXSTx[0]	DOUTx[1]	RXSTx[0]
RXDx[0]	DOUTx[2]	RXDx[0]
RXDx[1]	DOUTx[3]	RXDx[1]
RXDx[2]	DOUTx[4]	RXDx[2]
RXDx[3]	DOUTx[5]	RXDx[3]
RXDx[4]	DOUTx[6]	RXDx[4]
RXDx[5]	DOUTx[7]	RXDx[5]
RXDx[6]	DOUTx[8]	RXDx[6]
RXDx[7] (MSB)	DOUTx[9]	RXDx[7]

Note:
5. LSB received first.

When the 10B/8B decoder is bypassed, the framed 10-bit value is presented to the associated output register, along with a status output indicating if the character in the output register is one of the selected framing characters. The bit usage and mapping of the external signals to the raw 10B coded character is shown in *Table 10*.

The COMDETx status outputs operate the same regardless of the bit combination selected for character framing by the FRAMCHARx input. They are HIGH when the character in the output register contains the selected framing character at the proper character boundary, and LOW for all other bit combinations.

When the low-latency framer and half-rate receive port clocking are also enabled, the framer stretches the recovered clock to the nearest 20-bit boundary such that the rising edge of RXCLKx+ occurs when COMDETx is present on the associated output bus.

Table 10. Decoder Bypass Mode

Signal Name	Bus Weight	10B Name
RXSTx[2] (LSB)	COMDETx	
RXSTx[1]	2^0 [5]	a [5]
RXSTx[0]	2^1	b
RXDx[0]	2^2	c
RXDx[1]	2^3	d
RXDx[2]	2^4	e
RXDx[3]	2^5	i
RXDx[4]	2^6	f
RXDx[5]	2^7	g
RXDx[6]	2^8	h
RXDx[7] (MSB)	2^9	j

When the standard framer is enabled and half-rate receive port clocking is also enabled, the output clock is not modified when framing is detected, but a single pipeline stage may be added or subtracted from the data stream by the framer logic such that the rising edge of RXCLKx+ occurs when COMDET is present on the associated output bus.

This adjustment only occurs when the framer is enabled. When the framer is disabled, the clock boundaries are not adjusted, and COMDETx may be active during the rising edge of RXCLKx– (if an odd number of characters were received following the initial framing).

Receive Status Bits

When the 10B/8B decoder is enabled, each character presented at the output register includes three associated status bits. These bits are used to identify

- if the contents of the data bus are valid,
- the type of character present,
- the state of receive BIST operations,
- character violations.

These conditions normally overlap; for example, a valid data character received with incorrect running disparity is not reported as a valid data character. It is instead reported as a decoder violation of some specific type. This implies a hierarchy or priority level to the various status bit combinations. The hierarchy and value of each status are listed in *Table 11*.

A second status mapping, listed in *Table 11*, is used when the receive channel is configured for BIST operation. This status is used to report receive BIST status and progress.

BIST Status State Machine

When a receive path is enabled to look for and compare the received data stream with the BIST pattern, the RXSTx[2:0] bits identify the present state of the BIST compare operation.

The BIST state machine has multiple states, as shown in Figure 2 and *Table 11*. When the receive PLL detects an out-of-lock condition, the BIST state is forced to the Start-of-BIST state, regardless of the present state of the BIST state machine. If the number of detected errors ever exceeds the number of valid matches by greater than 16, the state machine is forced to the WAIT_FOR_BIST state where it monitors the interface for the first character of the next BIST sequence (D0.0). Also, if the Elasticity Buffer ever hits and overflow/underflow condition, the status is forced to the BIST_START until the buffer is re centered (approximately nine character periods).

JTAG Support

The CYP15G0403DX contains a JTAG port to allow system level diagnosis of device interconnect. Only JTAG boundary scan is supported. This capability is present on the LVTTL inputs and outputs and the REFCLKx± clock input. The high-speed serial ports are not supported.

3-Level Select Inputs

Each 3-level select input reports as two bits in the scan register. These bits report the LOW, MID, and HIGH state of the associated input as 00, 10, and 11 respectively

JTAG ID

The JTAG device ID for the CYP15G0403DX is “0C810069x.”

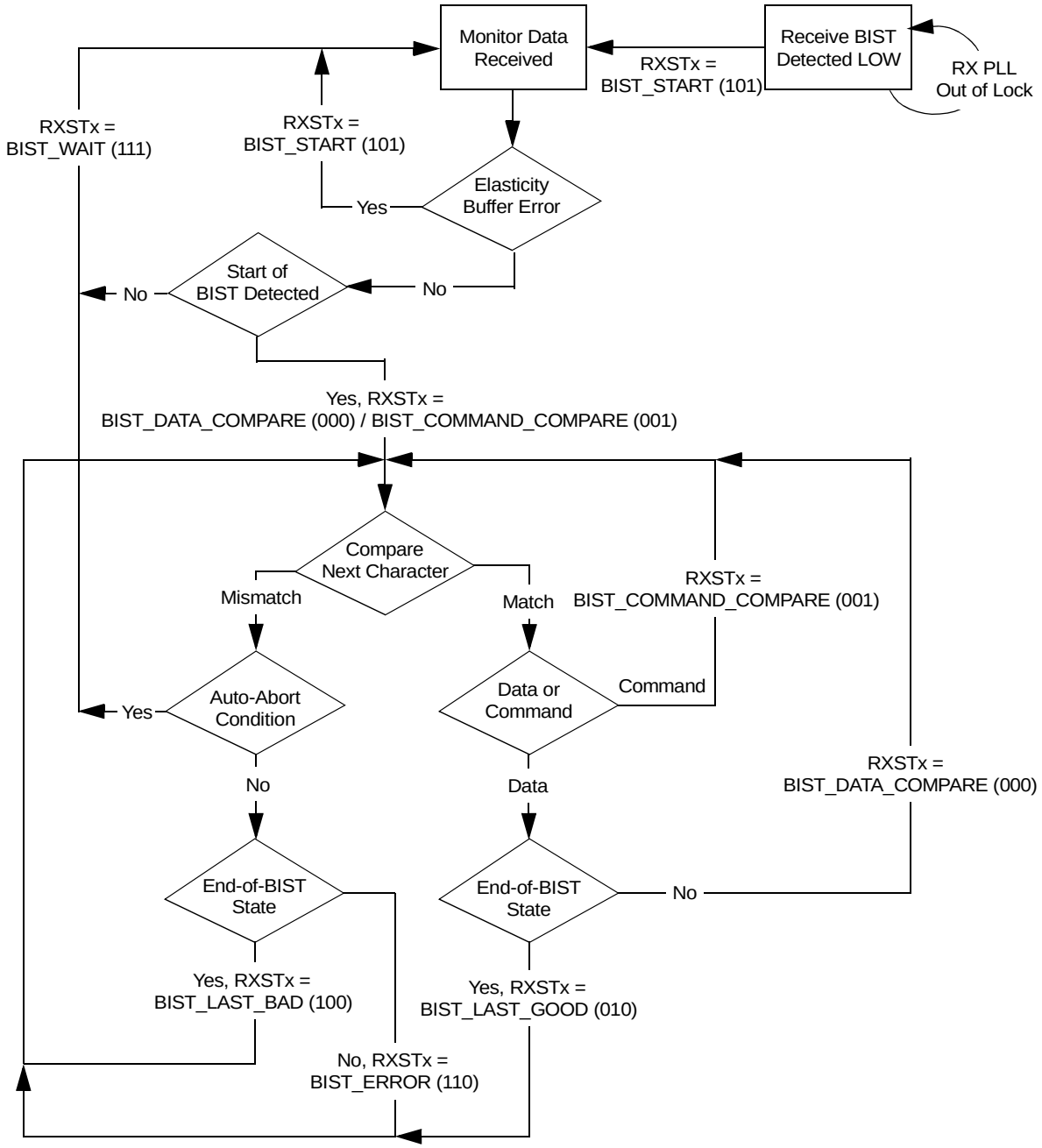


Figure 2. Receive BIST State Machine

Table 11. Receive Character Status Bits

RXSTx[2:0]	Priority	Description	
		Normal Status	Receive BIST Status (Receive BIST = Enabled)
000	7	**Normal Character Received**. The valid Data character on the output bus meets all the formatting requirements of Data characters listed in *Table 14*.	**BIST Data Compare**. Character compared correctly
001	7	**Special Code Detected**. The valid special character on the output bus meets all the formatting requirements of Special Code characters listed in *Table 15*, but is not the presently selected framing character or a decoder violation indication.	**BIST Command Compare**. Character compared correctly

Table 11. Receive Character Status Bits (continued)

010	2	**Receive Elasticity Buffer Underrun/Overrun Error**. The receive buffer was not able to add/drop a K28.5 or framing character	**BIST Last Good**. Last Character of BIST sequence detected and valid.
011	5	**Framing Character Detected**. This indicates that a character matching the patterns identified as a framing character was detected. The decoded value of this character is present in the associated output bus.	
100	4	**Codeword Violation**. The character on the output bus is a C0.7. This indicates that the received character cannot be decoded into any valid character.	**BIST Last Bad**. Last Character of BIST sequence detected invalid.
101	1	**Loss of Sync**. This indicates a PLL Out of Lock condition	**BIST Start**. Receive BIST is enabled on this channel, but character compares have not yet commenced. This also indicates a PLL Out of Lock condition, and Elasticity Buffer overflow/underflow conditions.
110	6	**Running Disparity Error**. The character on the output bus is a C4.7, C1.7, or C2.7.	**BIST Error**. While comparing characters, a mismatch was found in one or more of the decoded character bits.
111	3	Reserved	**BIST Wait**. The receiver is comparing characters. but has not yet found the start of BIST character to enable the LFSR.

X3.230 Codes and Notation Conventions

Information to be transmitted over a serial link is encoded eight bits at a time into a 10-bit Transmission Character and then sent serially. Information received over a serial link is collected ten bits at a time, and those Transmission Characters that are used for data characters are decoded into the correct eight-bit codes. The 10-bit Transmission Code supports all 256 8-bit combinations. Some of the Special Transmission Characters are used for other than data functions.

The primary use of a Transmission Code is to improve the transmission characteristics of a serial link. The encoding defined by the Transmission Code ensures that sufficient transitions are present in the serial bit stream to recover the clock at the Receiver. Such encoding increases the likelihood of detecting bit errors that may occur. Some Special Characters of the Transmission Code used by Fibre Channel Standard consist of an easily recognizable bit pattern that assists a Receiver in achieving word alignment.

Notation Conventions

The documentation for the 8B/10B Transmission Code uses letter notation for the bits in an 8-bit byte. Fibre Channel Standard notation uses a bit notation of A, B, C, D, E, F, G, H for the 8-bit byte for the raw 8-bit data, and the letters a, b, c, d, e, i, f, g, h, j for encoded 10-bit data. There is a correspondence between bit A and bit a, B and b, C and c, D and d, E and e, F and f, G and g, and H and h. Bits i and j are derived, respectively, from (A,B,C,D,E) and (F,G,H).

The bit labeled A in the description of the 8B/10B Transmission Code corresponds to bit 0 in the numbering scheme of the FC-2 specification, B corresponds to bit 1, as shown below.

FC-2 bit designation—	7	6	5	4	3	2	1	0
HOTLink D/Q designation—	7	6	5	4	3	2	1	0
8B/10B bit designation—	H	G	F	E	D	C	B	A

To clarify this correspondence, the following example shows the conversion from an FC-2 Valid Data Byte to a Transmission Character (using 8B/10B Transmission Code notation).

```
FC-2  45
      Bits: 7654 3210
            0100 0101
```

Converted to 8B/10B notation. (**Note** (carefully). The order of bits is reversed.)

```
Data Byte Name   D5.2
                 Bits:ABCDEFGH
                      10100 010
```

Translated to a transmission Character in the 8B/10B Transmission Code.

```
Bits: abcdeifghj
      1010010101
```

Each valid Transmission Character of the 8B/10B Transmission Code has been given a name using the following convention: cxx.y, where c is used to show whether the Transmission Character is a Data Character (c is set to D, and SC/$\overline{D}$ = LOW) or a Special Character (c is set to K, and SC/$\overline{D}$ = HIGH). When c is set to D, xx is the decimal value of the binary number composed of the bits E, D, C, B, and A in that order, and the y is the decimal value of the binary number composed of the bits H, G, and F in that order. When c is set to K, xx and y are derived by comparing the encoded bit patterns of the Special Character to those patterns derived from encoded Valid Data bytes and selecting the names of the patterns most similar to the encoded bit patterns of the Special Character.

Under the above conventions, the Transmission Character used for the examples above, is referred to by the name D5.2. The Special Character K29.7 is so named because the first six bits (abcdei) of this character make up a bit pattern similar to that resulting from the encoding of the unencoded 11101 pattern (29), and because the second four bits (fghj) make up a bit pattern similar to that resulting from the encoding of the unencoded 111 pattern (7).

Note. This definition of the 10-bit Transmission Code is based on (and is in basic agreement with) the following references, which describe the same 10-bit transmission code.

A.X. Widmer and P.A. Franaszek. “A DC-Balanced, Partitioned-Block, 8B/10B Transmission Code” *IBM Journal of Research and Development*, 27, No. 5: 440–451 (September, 1983).

U.S. Patent 4,486,739. Peter A. Franaszek and Albert X. Widmer. “Byte-Oriented DC Balanced (0.4) 8B/10B Partitioned Block Transmission Code” (December 4, 1984).

Fibre Channel Physical and Signaling Interface (ANS X3.230–1994 ANSI FC–PH Standard).

IBM Enterprise Systems Architecture/390 ESCON I/O Interface (document number SA22–7202).

8B/10B Transmission Code

The following information describes how the tables shall be used for both generating valid Transmission Characters (encoding) and checking the validity of received Transmission Characters (decoding). It also specifies the ordering rules to be followed when transmitting the bits within a character and the characters within the higher-level constructs specified by the standard.

Transmission Order

Within the definition of the 8B/10B Transmission Code, the bit positions of the Transmission Characters are labeled a, b, c, d, e, i, f, g, h, j. Bit “a” shall be transmitted first followed by bits b, c, d, e, i, f, g, h, and j in that order. (**Note**. Bit i shall be transmitted between bit e and bit f, rather than in alphabetical order.)

Valid and Invalid Transmission Characters

The following tables define the valid Data Characters and valid Special Characters (K characters), respectively. The tables are used for both generating valid Transmission Characters (encoding) and checking the validity of received Transmission Characters (decoding). In the tables, each Valid-Data-byte or Special-Character-code entry has two columns that represent two (not necessarily different) Transmission Characters. The two columns correspond to the current value of the running disparity (“Current RD–” or “Current RD+”). Running disparity is a binary parameter with either the value negative (–) or the value positive (+).

After powering on, the Transmitter may assume either a positive or negative value for its initial running disparity. Upon transmission of any Transmission Character, the transmitter will select the proper version of the Transmission Character based on the current running disparity value, and the Transmitter shall calculate a new value for its running disparity based on the contents of the transmitted character. Special

Character codes C1.7 and C2.7 can be used to force the transmission of a specific Special Character with a specific running disparity as required for some special sequences in X3.230.

After powering on, the Receiver may assume either a positive or negative value for its initial running disparity. Upon reception of any Transmission Character, the Receiver shall decide whether the Transmission Character is valid or invalid according to the following rules and tables and shall calculate a new value for its Running Disparity based on the contents of the received character.

The following rules for running disparity shall be used to calculate the new running-disparity value for Transmission Characters that have been transmitted (Transmitter's running disparity) and that have been received (Receiver's running disparity).

Running disparity for a Transmission Character shall be calculated from sub-blocks, where the first six bits (abcdei) form one sub-block and the second four bits (fghj) form the other sub-block. Running disparity at the beginning of the 6-bit sub-block is the running disparity at the end of the previous Transmission Character. Running disparity at the beginning of the 4-bit sub-block is the running disparity at the end of the 6-bit sub-block. Running disparity at the end of the Transmission Character is the running disparity at the end of the 4-bit sub-block.

Running disparity for the sub-blocks shall be calculated as follows:

1. Running disparity at the end of any sub-block is positive if the sub-block contains more ones than zeros. It is also positive at the end of the 6-bit sub-block if the 6-bit sub-block is 000111, and it is positive at the end of the 4-bit sub-block if the 4-bit sub-block is 0011.
2. Running disparity at the end of any sub-block is negative if the sub-block contains more zeros than ones. It is also negative at the end of the 6-bit sub-block if the 6-bit sub-block is 111000, and it is negative at the end of the 4-bit sub-block if the 4-bit sub-block is 1100.
3. Otherwise, running disparity at the end of the sub-block is the same as at the beginning of the sub-block.

Use of the Tables for Generating Transmission Characters

The appropriate entry in the table shall be found for the Valid Data byte or the Special Character byte for which a Transmission Character is to be generated (encoded). The current value of the Transmitter's running disparity shall be used to select the Transmission Character from its corresponding column. For each Transmission Character transmitted, a new value of the running disparity shall be calculated. This new value shall be used as the Transmitter's current running disparity for the next Valid Data byte or Special Character byte to be encoded and transmitted. *Table 12* shows naming notations and examples of valid transmission characters.

Use of the Tables for Checking the Validity of Received Transmission Characters

The column corresponding to the current value of the Receiver's running disparity shall be searched for the received Transmission Character. If the received Transmission Character is found in the proper column, then the Transmission Character is valid and the associated Data byte or Special Character code is determined (decoded). If the received Transmission Character is not found in that column, then the Transmission Character is invalid. This is called a code violation. Independent of the Transmission Character's validity, the received Transmission Character shall be used to calculate a new value of running disparity. The new value shall be used as the Receiver's current running disparity for the next received Transmission Character.

Table 12. Valid Transmission Characters

Data			
	D_{IN} or Q_{OUT}		
Byte Name	765	43210	Hex Value
D0.0	000	00000	00
D1.0	000	00001	01
D2.0	000	00010	02
. .	. .	. .	. .
D5.2	010	00010 1	45
. .	. .	. .	. .
D30.7	111	11110	FE
D31.7	111	11111	FF

Detection of a code violation does not necessarily show that the Transmission Character in which the code violation was detected is in error. Code violations may result from a prior error that altered the running disparity of the bit stream which did not result in a detectable error at the Transmission Character in which the error occurred. *Table 13* shows an example of this behavior.

Maximum Ratings

Above which the useful life may be impaired.

Storage Temperature	–65°C to +150°C
Ambient Temperature with Power Applied	–55°C to +125°C
Maximum Supply Voltage	–0.5V to +3.8V
Output Current into LVTTL Outputs (LOW)	30 mA
DC Input Voltage	–0.5V to V_{CC}+0.5V
Static Discharge Voltage (per MIL-STD-883, Method 3015)	> 2000 V
Latch-Up Current	> 200 ma

Operating Range

Range	Ambient Temperature	V_{CC}
Commercial	0°C to +70°C	+3.3V ± 5%
Industrial	–40°C to +85°C	+3.3V ± 5%

CYP15G0403DX DC Electrical Characteristics

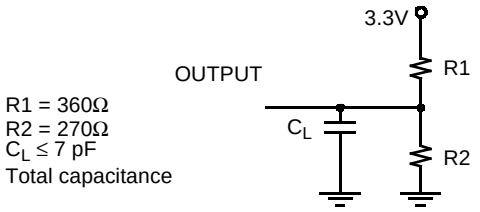
Parameter	Description	Test Conditions	Min.	Max.	Unit
LVTTL-compatible Outputs					
V_{OHT}	Output HIGH Voltage	I_{OH} = –4 mA, V_{CC} = Min.	2.4	V_{CC}	V
V_{OLT}	Output LOW Voltage	I_{OL} = 4 mA, V_{CC} = Min.	–0.5	0.8	V
I_{OST}	Output Short Circuit Current	V_{OUT} = 0V [6]	–15	–45	mA
I_{OZL}	High-Z Output Leakage Current		–20	20	µA
LVTTL-compatible Inputs					
V_{IHT}	Input HIGH Voltage		2.0	V_{CC} + 0.3	V
V_{ILT}	Input LOW Voltage		–0.5	0.8	V
I_{IHT}	Input HIGH Current	REFCLK Input, V_{IN} = V_{CC}		1.5	ma
		Other Inputs, V_{IN} = V_{CC}		+40	µA
I_{ILT}	Input LOW Current	REFCLK Input, V_{IN} = 0.0V		–1.5	ma
		Other Inputs, V_{IN} = 0.0V		–40	µA
I_{IHPDT}	Input HIGH Current with internal pull-down	V_{IN} = V_{CC}		+200	µA
I_{ILPUT}	Input LOW Current with internal pull-up	V_{IN} = 0.0V		–200	µA
LVDIFF Inputs: REFCLKx±					
V_{DIFF} [7]	Input Differential Voltage		400	V_{CC}	mV
V_{IHHP}	Highest Input HIGH Voltage		1.0	V_{CC}	V
V_{ILLP}	Lowest Input LOW voltage		GND	V_{CC}/2	V
V_{COM} [8]	Common Mode Range		1.0	V_{CC} – 1.2V	V
3-Level Inputs					
V_{IHH}	3-Level Input HIGH Voltage	Min. ≤ V_{CC} ≤ Max.	0.87 * V_{CC}	V_{CC}	V
V_{IMM}	3-Level Input MID Voltage	Min. ≤ V_{CC} ≤ Max.	0.47 * V_{CC}	0.53 * V_{CC}	V
V_{ILL}	3-Level Input LOW Voltage	Min. ≤ V_{CC} ≤ Max.	0.0	0.13 * V_{CC}	V
I_{IHH}	Input HIGH Current	V_{IN} = V_{CC}		200	µA
I_{IMM}	Input MID current	V_{IN} = V_{CC}/2	–50	50	µA
I_{ILL}	Input LOW current	V_{IN} = GND		–200	µA
Differential CML Serial Outputs: OUTA1±, OUTA2±, OUTB1±, OUTB2±, OUTC1±, OUTC2±, OUTD1±, OUTD2±					
V_{OHC}	Output HIGH Voltage	100Ω differential load	V_{CC} – 0.5	V_{CC} – 0.2	V
		150Ω differential load	V_{CC} – 0.5	V_{CC} – 0.2	V
V_{OLC}	Output LOW Voltage	100Ω differential load	V_{CC} – 1.1	V_{CC} – 0.7	V
		150Ω differential load	V_{CC} – 1.1	V_{CC} – 0.7	V
V_{ODIF}	Output Differential Voltage |(OUT+) – (OUT–)|	100Ω differential load	450	800	mV
		150Ω differential load	560	1000	mV

Notes:

6. Tested one output at a time, output shorted for less than one second, less than 10% duty cycle.
7. This is the minimum difference in voltage between the true and complement inputs required to ensure detection of a logic-1 or logic-0.
8. The common mode range defines the allowable input voltage range of both input signal. The input signals must be different by the 100 mv and in this voltage range. This is the voltage range in which zero-crossing detection operates between true and complement inputs.

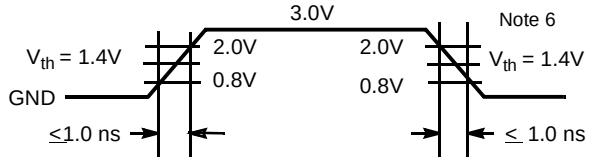
CYP15G0403DX DC Electrical Characteristics (continued)

Parameter	Description	Test Conditions		Min.	Max.	Unit
Differential Serial Line Receiver Inputs: INA1±, INA2±, INB1±, INB2±, INC1±, INC2±, IND1±, IND2±						
$V_{DIFF[7]}$	Input Differential Voltage |(IN+) – (IN–)|			100	1200	mV
V_{IHE}	Highest Input HIGH Voltage				V_{CC}	V
V_{ILE}	Lowest Input LOW Voltage			V_{CC} – 2.0		V
I_{IHE}	Input HIGH Current	$V_{IN} = V_{IHE}$ Max.			1000	µA
I_{ILE}	Input LOW Current	$V_{IN} = V_{ILE}$ Min.		–700		µA
$VI_{COM[8]}$	Common Mode Input Range	$((V_{CC} – 2.0V) + 0.5)$ min., $((V_{CC} – 0.5V)$ max.		+1.25	+3.25	V
Miscellaneous				**Typ.**	**Max.**	
I_{CC} [9]	Power Supply Current	Freq. = Max.	Commercial	860	1000	mA
			Industrial	TBD	TBD	mA

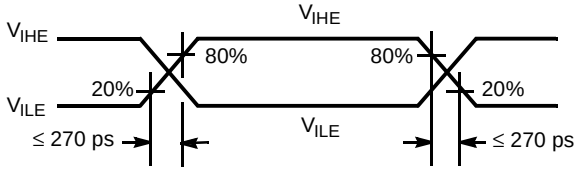


(a) LVTTL Input AC Test Load[10]

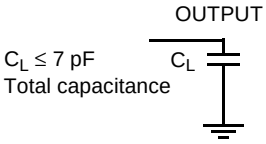
(b) CML Input Test Load[11]



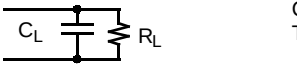
(c) LVTTL Input Test Waveform



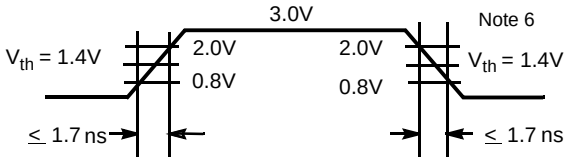
(d) PECL Input Test Waveform



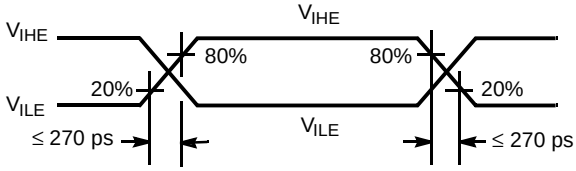
(a) LVTTL Output AC Test Load[10]



(b) CML Input Test Load[11]



(c) LVTTL Output Test Waveform



(d) PECL Output Test Waveform

Notes:

9. Maximum I_{CC} is measured with V_{CC} = 3.465V, frequency = 150 MHz, RFENx = LOW, T_A = –45°C. Typical I_{CC} is measured under similar conditions except with V_{CC} = 3.3V, T_A = 25°C. The serial channels sending a constant 01 pattern and the outputs unloaded.
10. Cypress uses constant current (ATE) load configurations and forcing functions. This figure is for reference only.
11. The LVTTL switching threshold is 1.4V. All timing references are made where the signal edges cross the threshold voltage.

CYP15G0403DX Transmitter LVTTL Switching Characteristics Over the Operating Range

Parameter	Description	Min.	Max.	Unit
f_{TS}	TXCLKx Clock Cycle Frequency	20	150	MHz
t_{TXCLK}	TXCLKx Period	6.66	50	ns
t_{TXCLKH}	TXCLKx HIGH Time	2.2		ns
t_{TXCLKL}	TXCLKx LOW Time	2.2		ns
t_{TXCLKR} [12, 14, 15]	TXCLKx Rise Time		1.7	ns
t_{TXCLKF} [12, 14, 15]	TXCLKx Fall Time		1.7	ns
t_{TXDS}	Transmit Data Set-up Time to TXCLKx↑	2		ns
t_{TXDH}	Transmit Data Hold Time from TXCLKx↑	1		ns
f_{TOS}	TXCLKOx Clock Cycle Frequency = 1x or 2x REFCLKx Frequency	20	150	MHz
t_{TXCLKO}	TXCLKOx Period	6.66	50	ns
$t_{TXCLKOD}$	TXCLKOx Duty Cycle	40	60	%
$t_{TXclk0d+}$	TXCLKO+ Duty Cycle Centered with 60% HIGH Time	–0.7	+0.7	ns
$t_{TXclk0d-}$	TXCLKO– Duty Cycle Centered with 40% HIGH Time	0.0	+1.5	ns

CYP15G0403DX Receiver LVTTL Switching Characteristics Over the Operating Range

Parameter	Description	Min.	Max.	Unit
f_{RS}	RXCLKx Clock Output Frequency	20	150	MHz
t_{RXCLKP}	RXCLKx Period	6.66	50	ns
t_{RXCLKH}	RXCLKx HIGH Time (RXRATE = LOW)	1.5	24	ns
	RXCLKx HIGH Time (RXRATE = HIGH)	5	25	ns
t_{RXCLKL}	RXCLKx LOW Time (RXRATE = LOW)	1.5	24	ns
	RXCLKx LOW Time (RXRATE = HIGH)	5	25	ns
t_{RXCLKD}	RXCLKx Duty Cycle Centered at 50%	–1.0	+1.0	ns
t_{RXCLKR} [12]	RXCLKx Rise Time		1.2	ns
t_{RXCLKF} [12]	RXCLKx Fall Time		1.2	ns
t_{RXDv-} [13]	Status and Data Valid Time to RXCLKX with RXCSEL MID or HIGH	5UI – 1.5		ns
t_{RXDv+} [13]	Status and Data Valid Time to RXCLDX	5UI – 1.8		ns
t_{RXDv-} [13]	Status and Data Valid Time to RXCLDX with half rate recovered clock	5UI – 1.0		ns
t_{RXDv+} [13]	Status and Data Valid Time to RXCLDX with half rate recovered clock	5UI – 1.8		ns

CYP15G0403DX REFCLKx Switching Characteristics Over the Operating Range

Parameter	Description	Min.	Max.	Unit
f_{REF}	REFCLKx Clock Frequency	20	150	MHz
t_{REFCLK}	REFCLKx Period	6.6	50	ns
t_{REFH}	REFCLKx HIGH Time (TXRATE = HIGH)	5.9	35	ns
	REFCLKx HIGH Time (TXRATE = LOW)	2.9	35	ns
t_{REFL}	REFCLKx LOW Time (TXRATE = HIGH)	5.9	35	ns
	REFCLKx LOW Time (TXRATE = LOW)	2.9	35	ns
t_{REFD} [11]	REFCLKx Duty Cycle	30	70	%
t_{REFR} [12, 14, 15]	REFCLKx Rise Time (20%–80%)		2	ns
t_{REFF} [12, 14, 15]	REFCLKx Fall Time (20%–80%)		2	ns

Notes:

12. Tested initially and after any design or process changes that may affect these parameters, but not 100% tested.
13. Parallel data output specifications are only valid if all outputs are loaded with similar DC and AC loads.
14. The ratio of rise time to falling time must not vary by greater than 2:1.
15. For a given operating frequency, neither rise or fall specification can be greater than 20% of the clock-cycle period or the data sheet maximum time.

CYP15G0403DX REFCLKx Switching Characteristics Over the Operating Range (continued)

Parameter	Description	Min.	Max.	Unit
t_{TREFDS}	Transmit Data Setup Time to REFCLKx (TXCKSEL = LOW)	2		ns
t_{TREFDH}	Transmit Data Hold Time from REFCLKx (TXCKSEL= LOW)	1		ns
t_{RREFDA}	Receive Data Access Time to REFCLKx (RXCKSEL = LOW)		9.5	ns
t_{RREFDV}	Receive Data Valid Time from REFCLKx (RXCKSEL = LOW)	4.0		ns
$t_{REFADV-}$	Receive Data Valid Time to REFCLKx (RXCKSEL = LOW)	1.5		ns
$t_{REFADV+}$	Receive Data Valid Time from REFCLKx	1.5		ns
$t_{REFCDV-}$	Received Data Valid Time to RXCLKx	3		ns
$t_{REFCDV+}$	Received Data valid Time from RXCLKx	0.5		ns
t_{REFRX}	REFCLKx Frequency Referenced to Received Clock Period[20]	–0.02	+0.02	%

CYP15G0403DX Transmit Serial Outputs and TX PLL Characteristics Over the Operating Range

Parameter	Description	Condition	Min.	Max.	Unit
t_B	Bit Time		5000	660	ps
t_{RISE}	CML Output Rise Time 20–80% (CML Test Load)[12]	SPDSELx = HIGH	50	270	ps
		SPDSELx = MID	100	500	ps
		SPDSELx =LOW	200	1000	ps
t_{FALL}	CML Output Fall Time 80–20% (CML Test Load)[12]	SPDSELx = HIGH	50	270	ps
		SPDSELx = MID	100	500	ps
		SPDSELx =LOW	200	1000	ps
t_{DJ}	Deterministic Jitter (peak-peak)[12, 18]			0.1	UI
t_{TJ}	Total Jitter (σ)[12, 19]	0.2–1.0 Gbps		0.2	UI
		1.0–1.5 Gbps		192	ps
t_{TXLOCK}	Transmit PLLx lock to REFCLKx		TBD	TBD	

CYP15G0403DX Receive Serial Inputs and CDR PLL Characteristics Over the Operating Range

Parameter	Description	Min.	Max.	Unit
t_{RXLOCK}	Receive PLL lock to input data stream (cold start)		10	ms
	Receive PLL lock to input data stream		2500	UI
$t_{RXUNLOCK}$	Receive PLL Unlock Rate	TBD	TBD	ns
t_{SA}	Static Alignment[12, 16]			ps
t_{EFW}	Error Free Window [12, 17, 18]	0.75		UI

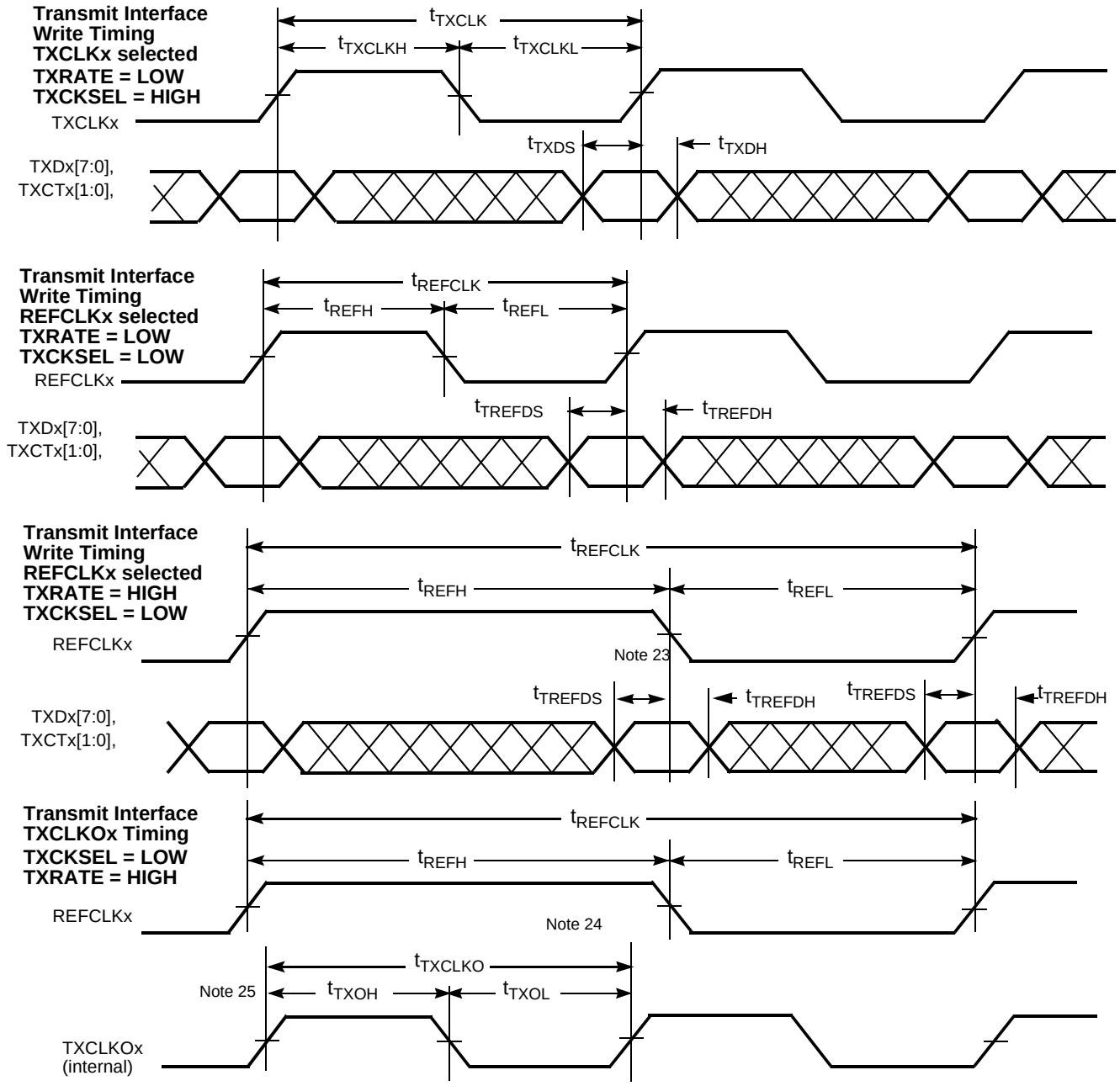
Capacitance[12]

Parameter	Description	Test Conditions	Max.	Unit
C_{INTTL}	TTL Input Capacitance	T_A = 25°C, f_0 = 1 MHz, V_{CC} = 3.3V	7	pF
C_{INPECL}	PECL input Capacitance	T_A = 25°C, f_0 = 1 MHz, V_{CC} = 3.3V	4	pF

Notes:

16. Static alignment is a measure of the alignment of the Receiver sampling point to the center of a bit. Static alignment is measured by sliding one bit edge in 3,000 nominal transitions until a character error occurs.
17. Receiver Unit Interval (UI) is calculated as $1/(f_{REF} * 20)$ (when RXRATE = HIGH) or $1/(f_{REF} * 10)$ (when RXRATE = LOW) if no data is being received, or $1/(f_{REF} * 20)$ (when RXRATE = HIGH) or $1/(f_{REF} * 10)$ (when RXRATE = LOW) of the remote transmitter if data is being received. In an operating link this is equivalent to t_B.
18. Error-free Window (EFW) is a measure of the time window between bit centers where a transition may occur without causing a bit sampling error. EFW is measured over the operating range, input jitter < 50% Dj.
19. The duty cycle specification is a simultaneous condition with the t_{REFH} and t_{REFL} parameters. This means that at faster character rates the REFCLKx duty cycle cannot be as large as 30%–70%,
20. REFCLKx has no phase or frequency relationship with the recovered clock(s) and only acts as a centering reference to reduce clock synchronization time. REFCLKx must be within ±200 PPM (±0.02%) of the transmitter PLL reference (REFCLKx) frequency, necessitating a ±100-PPM crystal.
21. While sending continuous K28.5s, outputs loaded to a balanced 100Ω load, over the operating range.
22. While sending continuous K28.7s, after 100,000 samples measured at the cross point of differential outputs, time referenced to REFCLKx input, over the operating range.

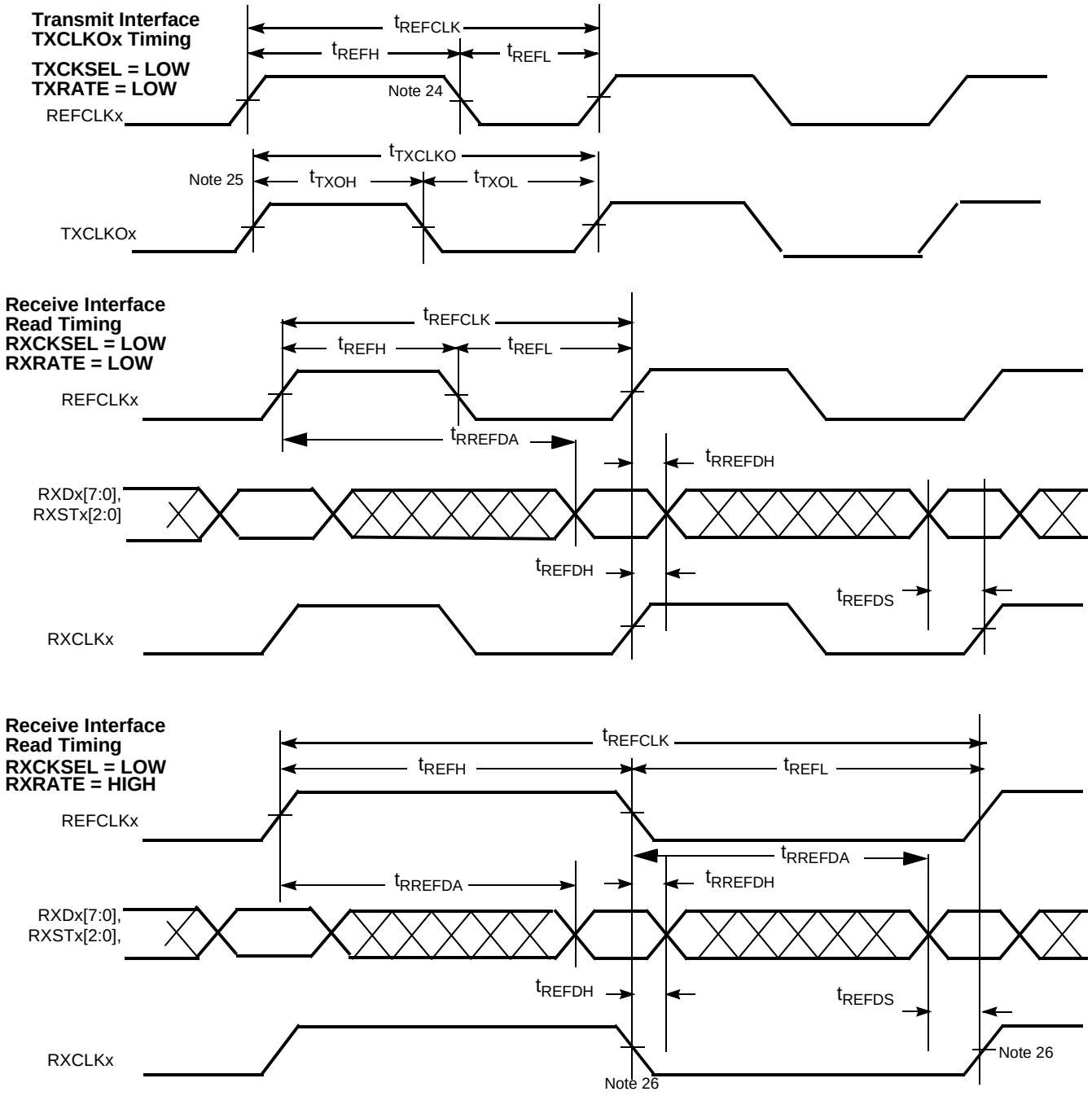
CYP15G0403DX HOTLink II Transmitter Switching Waveforms



Note:

23. When REFCLKx is configured for half-rate operation (TXRATE = HIGH) and data is captured using REFCLKx instead of a TXCLKx clock. Data is captured using both the rising and falling edges of REFCLKx.
24. The TXCLKOx output remains at the character rate regardless of the state of TXRATE and does not follow the duty cycle of REFCLKx.
25. The rising edge of TXCLKOx output has no direct phase relationship to the REFCLKx input.

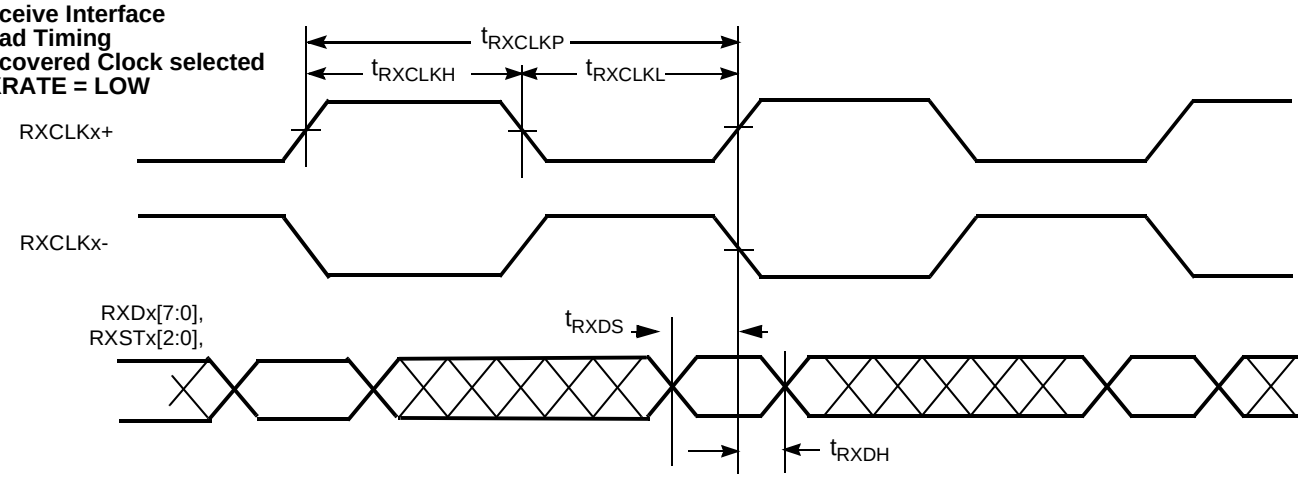
CYP15G0403DX HOTLink II Transmitter Switching Waveforms (continued)



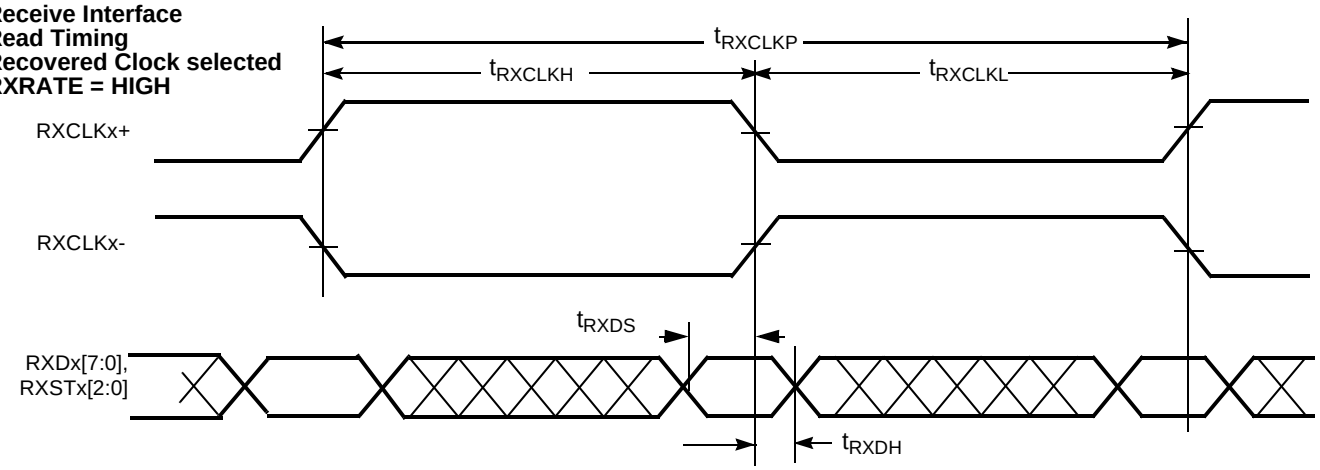
Note:

26. When operated with a half-rate REFCLKx, the set-up and hold specifications for data relative to RXCLKx are relative to both rising and falling edges of the respective clock output.

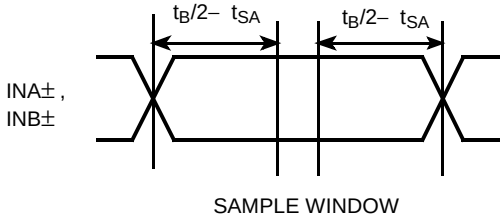
Receive Interface Read Timing Recovered Clock selected RXRATE = LOW



Receive Interface Read Timing Recovered Clock selected RXRATE = HIGH



Static Alignment



Error-Free Window

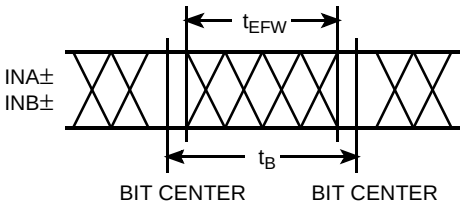


Table 13. Code Violations Resulting from Prior Errors

	RD	Character	RD	Character	RD	Character	RD
Transmitted data character	–	D21.1	–	D10.2	–	D23.5	+
Transmitted bit stream	–	101010 1001	–	010101 0101	–	111010 1010	+
Bit stream after error	–	101010 1011	+	010101 0101	+	111010 1010	+
Decoded data character	–	D21.0	+	D10.2	+	Code Violation	+

Table 14. Valid Data Characters (TXCTx[0] = 0, RXSTx[2:0] = 000)

Data Byte Name	Bits	Current RD–	Current RD+	Data Byte Name	Bits	Current RD–	Current RD+
	HGF EDCBA	abcdei fghj	abcdei fghj		HGF EDCBA	abcdei fghj	abcdei fghj
D0.0	000 00000	100111 0100	011000 1011	D0.1	001 00000	100111 1001	011000 1001
D1.0	000 00001	011101 0100	100010 1011	D1.1	001 00001	011101 1001	100010 1001
D2.0	000 00010	101101 0100	010010 1011	D2.1	001 00010	101101 1001	010010 1001
D3.0	000 00011	110001 1011	110001 0100	D3.1	001 00011	110001 1001	110001 1001
D4.0	000 00100	110101 0100	001010 1011	D4.1	001 00100	110101 1001	001010 1001
D5.0	000 00101	101001 1011	101001 0100	D5.1	001 00101	101001 1001	101001 1001
D6.0	000 00110	011001 1011	011001 0100	D6.1	001 00110	011001 1001	011001 1001
D7.0	000 00111	111000 1011	000111 0100	D7.1	001 00111	111000 1001	000111 1001
D8.0	000 01000	111001 0100	000110 1011	D8.1	001 01000	111001 1001	000110 1001
D9.0	000 01001	100101 1011	100101 0100	D9.1	001 01001	100101 1001	100101 1001
D10.0	000 01010	010101 1011	010101 0100	D10.1	001 01010	010101 1001	010101 1001
D11.0	000 01011	110100 1011	110100 0100	D11.1	001 01011	110100 1001	110100 1001
D12.0	000 01100	001101 1011	001101 0100	D12.1	001 01100	001101 1001	001101 1001
D13.0	000 01101	101100 1011	101100 0100	D13.1	001 01101	101100 1001	101100 1001
D14.0	000 01110	011100 1011	011100 0100	D14.1	001 01110	011100 1001	011100 1001
D15.0	000 01111	010111 0100	101000 1011	D15.1	001 01111	010111 1001	101000 1001
D16.0	000 10000	011011 0100	100100 1011	D16.1	001 10000	011011 1001	100100 1001
D17.0	000 10001	100011 1011	100011 0100	D17.1	001 10001	100011 1001	100011 1001
D18.0	000 10010	010011 1011	010011 0100	D18.1	001 10010	010011 1001	010011 1001
D19.0	000 10011	110010 1011	110010 0100	D19.1	001 10011	110010 1001	110010 1001
D20.0	000 10100	001011 1011	001011 0100	D20.1	001 10100	001011 1001	001011 1001
D21.0	000 10101	101010 1011	101010 0100	D21.1	001 10101	101010 1001	101010 1001
D22.0	000 10110	011010 1011	011010 0100	D22.1	001 10110	011010 1001	011010 1001
D23.0	000 10111	111010 0100	000101 1011	D23.1	001 10111	111010 1001	000101 1001
D24.0	000 11000	110011 0100	001100 1011	D24.1	001 11000	110011 1001	001100 1001
D25.0	000 11001	100110 1011	100110 0100	D25.1	001 11001	100110 1001	100110 1001
D26.0	000 11010	010110 1011	010110 0100	D26.1	001 11010	010110 1001	010110 1001
D27.0	000 11011	110110 0100	001001 1011	D27.1	001 11011	110110 1001	001001 1001
D28.0	000 11100	001110 1011	001110 0100	D28.1	001 11100	001110 1001	001110 1001
D29.0	000 11101	101110 0100	010001 1011	D29.1	001 11101	101110 1001	010001 1001
D30.0	000 11110	011110 0100	100001 1011	D30.1	001 11110	011110 1001	100001 1001
D31.0	000 11111	101011 0100	010100 1011	D31.1	001 11111	101011 1001	010100 1001
D0.2	010 00000	100111 0101	011000 0101	D0.3	011 00000	100111 0011	011000 1100
D1.2	010 00001	011101 0101	100010 0101	D1.3	011 00001	011101 0011	100010 1100
D2.2	010 00010	101101 0101	010010 0101	D2.3	011 00010	101101 0011	010010 1100

Table 14. Valid Data Characters (TXCTx[0] = 0, RXSTx[2:0] = 000) (continued)

Data Byte Name	Bits	Current RD–	Current RD+	Data Byte Name	Bits	Current RD–	Current RD+
	HGF EDCBA	abcdei fghj	abcdei fghj		HGF EDCBA	abcdei fghj	abcdei fghj
D3.2	010 00011	110001 0101	110001 0101	D3.3	011 00011	110001 1100	110001 0011
D4.2	010 00100	110101 0101	001010 0101	D4.3	011 00100	110101 0011	001010 1100
D5.2	010 00101	101001 0101	101001 0101	D5.3	011 00101	101001 1100	101001 0011
D6.2	010 00110	011001 0101	011001 0101	D6.3	011 00110	011001 1100	011001 0011
D7.2	010 00111	111000 0101	000111 0101	D7.3	011 00111	111000 1100	000111 0011
D8.2	010 01000	111001 0101	000110 0101	D8.3	011 01000	111001 0011	000110 1100
D9.2	010 01001	100101 0101	100101 0101	D9.3	011 01001	100101 1100	100101 0011
D10.2	010 01010	010101 0101	010101 0101	D10.3	011 01010	010101 1100	010101 0011
D11.2	010 01011	110100 0101	110100 0101	D11.3	011 01011	110100 1100	110100 0011
D12.2	010 01100	001101 0101	001101 0101	D12.3	011 01100	001101 1100	001101 0011
D13.2	010 01101	101100 0101	101100 0101	D13.3	011 01101	101100 1100	101100 0011
D14.2	010 01110	011100 0101	011100 0101	D14.3	011 01110	011100 1100	011100 0011
D15.2	010 01111	010111 0101	101000 0101	D15.3	011 01111	010111 0011	101000 1100
D16.2	010 10000	011011 0101	100100 0101	D16.3	011 10000	011011 0011	100100 1100
D17.2	010 10001	100011 0101	100011 0101	D17.3	011 10001	100011 1100	100011 0011
D18.2	010 10010	010011 0101	010011 0101	D18.3	011 10010	010011 1100	010011 0011
D19.2	010 10011	110010 0101	110010 0101	D19.3	011 10011	110010 1100	110010 0011
D20.2	010 10100	001011 0101	001011 0101	D20.3	011 10100	001011 1100	001011 0011
D21.2	010 10101	101010 0101	101010 0101	D21.3	011 10101	101010 1100	101010 0011
D22.2	010 10110	011010 0101	011010 0101	D22.3	011 10110	011010 1100	011010 0011
D23.2	010 10111	111010 0101	000101 0101	D23.3	011 10111	111010 0011	000101 1100
D24.2	010 11000	110011 0101	001100 0101	D24.3	011 11000	110011 0011	001100 1100
D25.2	010 11001	100110 0101	100110 0101	D25.3	011 11001	100110 1100	100110 0011
D26.2	010 11010	010110 0101	010110 0101	D26.3	011 11010	010110 1100	010110 0011
D27.2	010 11011	110110 0101	001001 0101	D27.3	011 11011	110110 0011	001001 1100
D28.2	010 11100	001110 0101	001110 0101	D28.3	011 11100	001110 1100	001110 0011
D29.2	010 11101	101110 0101	010001 0101	D29.3	011 11101	101110 0011	010001 1100
D30.2	010 11110	011110 0101	100001 0101	D30.3	011 11110	011110 0011	100001 1100
D31.2	010 11111	101011 0101	010100 0101	D31.3	011 11111	101011 0011	010100 1100

Table 14. Valid Data Characters (TXCTx[0] = 0, RXSTx[2:0] = 000) (continued)

Data Byte Name	Bits	Current RD–	Current RD+	Data Byte Name	Bits	Current RD–	Current RD+
	HGF EDCBA	abcdei fghj	abcdei fghj		HGF EDCBA	abcdei fghj	abcdei fghj
D0.4	100 00000	100111 0010	011000 1101	D0.5	101 00000	100111 1010	011000 1010
D1.4	100 00001	011101 0010	100010 1101	D1.5	101 00001	011101 1010	100010 1010
D2.4	100 00010	101101 0010	010010 1101	D2.5	101 00010	101101 1010	010010 1010
D3.4	100 00011	110001 1101	110001 0010	D3.5	101 00011	110001 1010	110001 1010
D4.4	100 00100	110101 0010	001010 1101	D4.5	101 00100	110101 1010	001010 1010
D5.4	100 00101	101001 1101	101001 0010	D5.5	101 00101	101001 1010	101001 1010
D6.4	100 00110	011001 1101	011001 0010	D6.5	101 00110	011001 1010	011001 1010
D7.4	100 00111	111000 1101	000111 0010	D7.5	101 00111	111000 1010	000111 1010
D8.4	100 01000	111001 0010	000110 1101	D8.5	101 01000	111001 1010	000110 1010
D9.4	100 01001	100101 1101	100101 0010	D9.5	101 01001	100101 1010	100101 1010
D10.4	100 01010	010101 1101	010101 0010	D10.5	101 01010	010101 1010	010101 1010
D11.4	100 01011	110100 1101	110100 0010	D11.5	101 01011	110100 1010	110100 1010
D12.4	100 01100	001101 1101	001101 0010	D12.5	101 01100	001101 1010	001101 1010
D13.4	100 01101	101100 1101	101100 0010	D13.5	101 01101	101100 1010	101100 1010
D14.4	100 01110	011100 1101	011100 0010	D14.5	101 01110	011100 1010	011100 1010
D15.4	100 01111	010111 0010	101000 1101	D15.5	101 01111	010111 1010	101000 1010
D16.4	100 10000	011011 0010	100100 1101	D16.5	101 10000	011011 1010	100100 1010
D17.4	100 10001	100011 1101	100011 0010	D17.5	101 10001	100011 1010	100011 1010
D18.4	100 10010	010011 1101	010011 0010	D18.5	101 10010	010011 1010	010011 1010
D19.4	100 10011	110010 1101	110010 0010	D19.5	101 10011	110010 1010	110010 1010
D20.4	100 10100	001011 1101	001011 0010	D20.5	101 10100	001011 1010	001011 1010
D21.4	100 10101	101010 1101	101010 0010	D21.5	101 10101	101010 1010	101010 1010
D22.4	100 10110	011010 1101	011010 0010	D22.5	101 10110	011010 1010	011010 1010
D23.4	100 10111	111010 0010	000101 1101	D23.5	101 10111	111010 1010	000101 1010
D24.4	100 11000	110011 0010	001100 1101	D24.5	101 11000	110011 1010	001100 1010
D25.4	100 11001	100110 1101	100110 0010	D25.5	101 11001	100110 1010	100110 1010
D26.4	100 11010	010110 1101	010110 0010	D26.5	101 11010	010110 1010	010110 1010
D27.4	100 11011	110110 0010	001001 1101	D27.5	101 11011	110110 1010	001001 1010
D28.4	100 11100	001110 1101	001110 0010	D28.5	101 11100	001110 1010	001110 1010
D29.4	100 11101	101110 0010	010001 1101	D29.5	101 11101	101110 1010	010001 1010
D30.4	100 11110	011110 0010	100001 1101	D30.5	101 11110	011110 1010	100001 1010
D31.4	100 11111	101011 0010	010100 1101	D31.5	101 11111	101011 1010	010100 1010
D0.6	110 00000	100111 0110	011000 0110	D0.7	111 00000	100111 0001	011000 1110
D1.6	110 00001	011101 0110	100010 0110	D1.7	111 00001	011101 0001	100010 1110
D2.6	110 00010	101101 0110	010010 0110	D2.7	111 00010	101101 0001	010010 1110

Table 14. Valid Data Characters (TXCTx[0] = 0, RXSTx[2:0] = 000) (continued)

Data Byte Name	Bits	Current RD–	Current RD+	Data Byte Name	Bits	Current RD–	Current RD+
	HGF EDCBA	abcdei fghj	abcdei fghj		HGF EDCBA	abcdei fghj	abcdei fghj
D3.6	110 00011	110001 0110	110001 0110	D3.7	111 00011	110001 1110	110001 0001
D4.6	110 00100	110101 0110	001010 0110	D4.7	111 00100	110101 0001	001010 1110
D5.6	110 00101	101001 0110	101001 0110	D5.7	111 00101	101001 1110	101001 0001
D6.6	110 00110	011001 0110	011001 0110	D6.7	111 00110	011001 1110	011001 0001
D7.6	110 00111	111000 0110	000111 0110	D7.7	111 00111	111000 1110	000111 0001
D8.6	110 01000	111001 0110	000110 0110	D8.7	111 01000	111001 0001	000110 1110
D9.6	110 01001	100101 0110	100101 0110	D9.7	111 01001	100101 1110	100101 0001
D10.6	110 01010	010101 0110	010101 0110	D10.7	111 01010	010101 1110	010101 0001
D11.6	110 01011	110100 0110	110100 0110	D11.7	111 01011	110100 1110	110100 1000
D12.6	110 01100	001101 0110	001101 0110	D12.7	111 01100	001101 1110	001101 0001
D13.6	110 01101	101100 0110	101100 0110	D13.7	111 01101	101100 1110	101100 1000
D14.6	110 01110	011100 0110	011100 0110	D14.7	111 01110	011100 1110	011100 1000
D15.6	110 01111	010111 0110	101000 0110	D15.7	111 01111	010111 0001	101000 1110
D16.6	110 10000	011011 0110	100100 0110	D16.7	111 10000	011011 0001	100100 1110
D17.6	110 10001	100011 0110	100011 0110	D17.7	111 10001	100011 0111	100011 0001
D18.6	110 10010	010011 0110	010011 0110	D18.7	111 10010	010011 0111	010011 0001
D19.6	110 10011	110010 0110	110010 0110	D19.7	111 10011	110010 1110	110010 0001
D20.6	110 10100	001011 0110	001011 0110	D20.7	111 10100	001011 0111	001011 0001
D21.6	110 10101	101010 0110	101010 0110	D21.7	111 10101	101010 1110	101010 0001
D22.6	110 10110	011010 0110	011010 0110	D22.7	111 10110	011010 1110	011010 0001
D23.6	110 10111	111010 0110	000101 0110	D23.7	111 10111	111010 0001	000101 1110
D24.6	110 11000	110011 0110	001100 0110	D24.7	111 11000	110011 0001	001100 1110
D25.6	110 11001	100110 0110	100110 0110	D25.7	111 11001	100110 1110	100110 0001
D26.6	110 11010	010110 0110	010110 0110	D26.7	111 11010	010110 1110	010110 0001
D27.6	110 11011	110110 0110	001001 0110	D27.7	111 11011	110110 0001	001001 1110
D28.6	110 11100	001110 0110	001110 0110	D28.7	111 11100	001110 1110	001110 0001
D29.6	110 11101	101110 0110	010001 0110	D29.7	111 11101	101110 0001	010001 1110
D30.6	110 11110	011110 0110	100001 0110	D30.7	111 11110	011110 0001	100001 1110
D31.6	110 11111	101011 0110	010100 0110	D31.7	111 11111	101011 0001	010100 1110

Table 15. Valid Special Character Codes and Sequences (TXCTx = special character code or RXSTx[2:0] = 001)[27, 28]

S.C. Code Name	S.C. Byte Name						Current RD– abcdei fghj	Current RD+ abcdei fghj
	Cypress			Alternate				
	S.C. Byte Name[29]		Bits HGF EDCBA	S.C. Byte Name[29]		Bits HGF EDCBA		
K28.0	C0.0	(C00)	000 00000	C28.0	(C1C)	000_11100	001111 0100	110000 1011
K28.1 [30]	C1.0	(C01)	000 00001	C28.1	(C3C)	001_11100	001111 1001	110000 0110
K28.2 [30]	C2.0	(C02)	000 00010	C28.2	(C5C)	010_11100	001111 0101	110000 1010
K28.3	C3.0	(C03)	000 00011	C28.3	(C7C)	011_11100	001111 0011	110000 1100
K28.4 [30]	C4.0	(C04)	000 00100	C28.4	(C9C)	100_11100	001111 0010	110000 1101
K28.5 [30, 31]	C5.0	(C05)	000 00101	C28.5	(CBC)	101_11100	001111 1010	110000 0101
K28.6 [30]	C6.0	(C06)	000 00110	C28.6	(CDC)	110_11100	001111 0110	110000 1001
K28.7 [30, 32]	C7.0	(C07)	000 00111	C28.7	(CFC)	111_11100	001111 1000	110000 0111
K23.7	C8.0	(C08)	000 01000	C23.7	(CF7)	111_10111	111010 1000	000101 0111
K27.7	C9.0	(C09)	000 01001	C27.7	(CFB)	111_11011	110110 1000	001001 0111
K29.7	C10.0	(C0A)	000 01010	C29.7	(CFD)	111_11101	101110 1000	010001 0111
K30.7	C11.0	(C0B)	000 01011	C30.7	(CFE)	111_11110	011110 1000	100001 0111
End of Frame Sequence								
EOFxx	C2.1	(C22)	001 00010	C2.1	(C22)	001 00010	–K28.5,Dn.xxx0[33]	+K28.5,Dn.xxx1[33]
Code Rule Violation and SVS Tx Pattern								
Exception [32, 34]	C0.7	(CE0)	111 00000	C0.7	(CE0)	111 00000	100111 1000	011000 0111
–K28.5[35]	C1.7	(CE1)	111 00001	C1.7	(CE1)	111 00001	001111 1010	001111 1010
+K28.5 [36]	C2.7	(CE2)	111 00010	C2.7	(CE2)	111 00010	110000 0101	110000 0101
Running Disparity Violation Pattern								
Exception [37]	C4.7	(CE4)	111 00100	C4.7	(CE4)	111 00100	110111 0101	001000 1010

Notes:

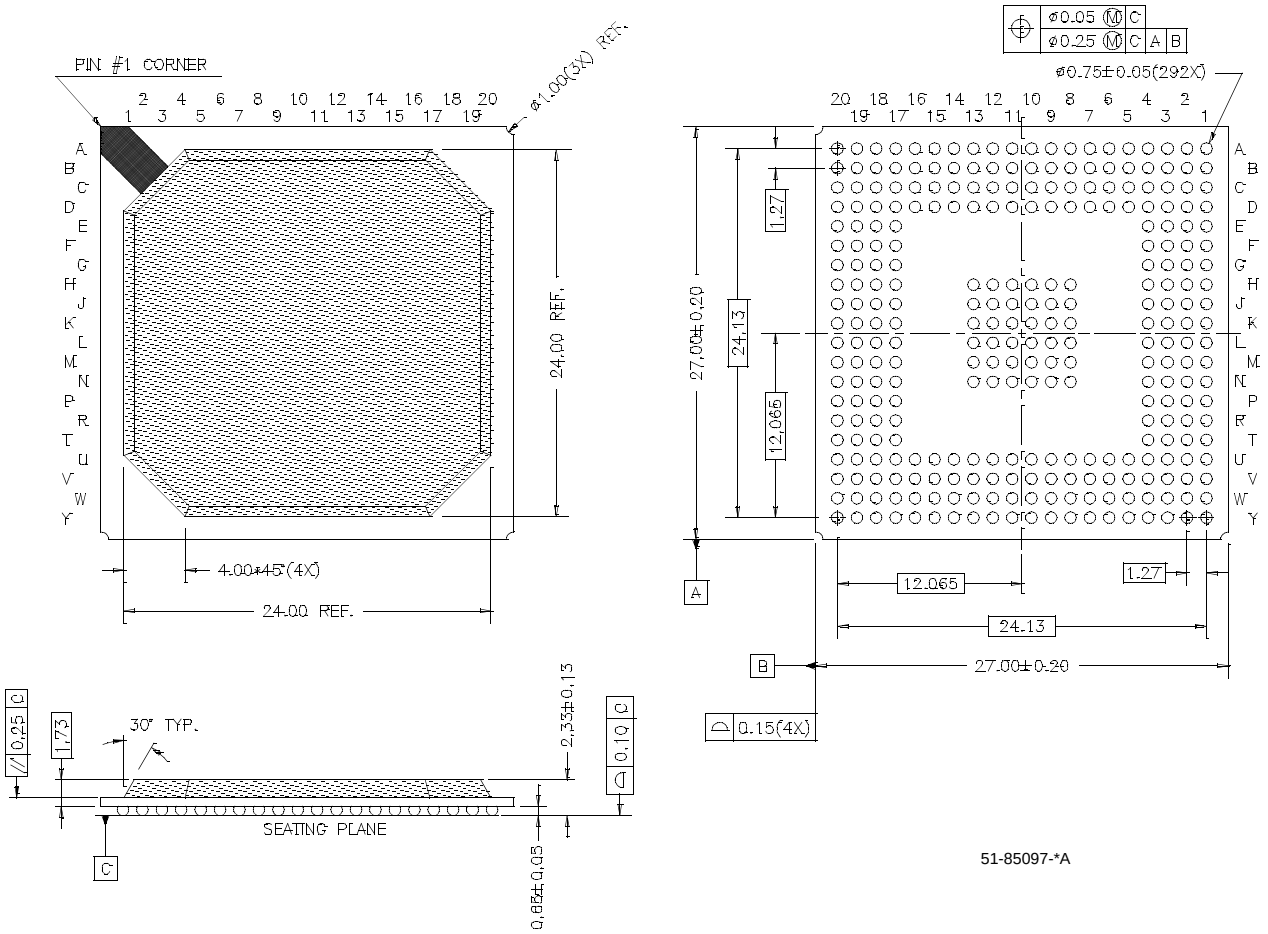
27. All codes not shown are reserved.
28. Notation for Special Character Code Name is consistent with Fibre Channel and ESCON naming conventions. Special Character Code Name is intended to describe binary information present on I/O pins. Common usage for the name can either be in the form used for describing Data patterns (i.e., C0.0 through C31.7), or in hex notation (i.e., Cnn where nn = the specified value between 00 and FF).
29. Both the Cypress and alternate encodings may be used for data transmission to generate specific Special Character Codes. The decoding process for received characters generates Cypress codes or Alternate codes as selected by the BOE[7:0] configuration inputs.
30. These characters are used for control of ESCON interfaces. They can be sent as embedded commands or other markers when not operating using ESCON protocols.
31. The K28.5 character is used for framing operations by the receiver. It is also the pad or fill character transmitted to maintain the serial link when no user data is available.
32. Care must be taken when using this Special Character code. When a C7.0 is followed by a D11.x or D20.x, or when an SVS (C0.7) is followed by a D11.x, an alias K28.5 sync character is created. These sequences can cause erroneous framing and should be avoided while RFENx = HIGH.
33. C2.1 = Transmit either –K28.5+ or +K28.5– as determined by Current RD and modify the Transmission Character that follows, by setting its least significant bit to 1 or 0. If Current RD at the start of the following character is plus (+) the LSB is set to 0, and if Current RD is minus (–) the LSB becomes 1. This modification allows construction of X3.230 “EOF” frame delimiters wherein the second data byte is determined by the Current RD.
    For example, to send “EOFdt” the controller could issue the sequence C2.1–D21.4– D21.4–D21.4, and the HOTLink Transmitter will send either K28.5–D21.4–D21.4–D21.4 or K28.5–D21.5– D21.4–D21.4 based on Current RD. Likewise to send “EOFdti” the controller could issue the sequence C2.1–D10.4–D21.4–D21.4, and the HOTLink Transmitter will send either K28.5–D10.4–D21.4– D21.4 or K28.5–D10.5–D21.4– D21.4 based on Current RD. The receiver will never output this Special Character, since K28.5 is decoded as C5.0, C1.7, or C2.7, and the subsequent bytes are decoded as data.
34. C0.7 = Transmit a deliberate code rule violation. The code chosen for this function follows the normal Running Disparity rules. Transmission of this Special Character has the same effect as asserting TXSVS = HIGH. The receiver will only output this Special Character if the Transmission Character being decoded is not found in the tables.
35. C1.7 = Transmit Negative K28.5 (–K28.5+) disregarding Current RD. The receiver will only output this Special Character if K28.5 is received with the wrong running disparity. The receiver will output C1.7 if –K28.5 is received with RD+, otherwise K28.5 is decoded as C5.0 or C2.7.
36. C2.7 = Transmit Positive K28.5 (+K28.5–) disregarding Current RD. The receiver will only output this Special Character if K28.5 is received with the wrong running disparity. The receiver will output C2.7 if +K28.5 is received with RD–, otherwise K28.5 is decoded as C5.0 or C1.7.
37. C4.7 = Transmit a deliberate code rule violation to indicate a Running Disparity violation. The receiver will only output this Special Character if the Transmission Character being decoded is found in the tables, but Running Disparity does not match. This might indicate that an error occurred in a prior byte.

Ordering Information

Speed	Ordering Code	Package Name	Package Type	Operating Range
Standard	CYP15G0403DX-BGC	BG256	256-lead Thermally Enhanced Ball Grid Array	Commercial
Standard	CYP15G0403DX-BGI	BG256	256-lead Thermally Enhanced Ball Grid Array	Industrial

Package Diagram

256-Lead PBGA (27 x 27 x 2.33 mm) BG256



HOTLink, HOTLink II, and MultiFrame are either trademarks or registered trademarks of Cypress Semiconductor. IBM, ESCON, and FICON are either trademarks or registered trademarks of International Business Machines. All product and company names mentioned in this document may be the trademarks of their respective holders.

Document Title: CYP15G0403DXA Quad HOTLink II® Transceiver (Preliminary) Document Number: 38-02033				
REV.	**ECN NO.**	**Issue Date**	**Orig. of Change**	**Description of Change**
**	113238	03/18/02	TPS	New Data Sheet
*A	114539	03/26/02	BSS	Rev ** has the incorrect package drawing in the datasheet. It should be *A for spec 51-85097