CY7B921/CY7B931, CY7B922/CY7B932, CY7B923/CY7B933

HOTLink™ Transmitter/Receiver

Features

- **Fibre Channel compliant**
- **IBM ESCON® compliant**
- **8B/10B-coded or 10-bit unencoded**
- **130- to 310-Mbps data rate**
- **TTL synchronous I/O**
- **No external PLL components**
- **Triple ECL 100K serial outputs**
- **Dual ECL 100K serial inputs**
- **Low power: 350 mW max (Tx), 500 mW max (Rx)**
- **Compatible with fiber optic modules, coaxial cable, and twisted pair media**
- **Built-In Self-Test**
- **Single +5V supply**
- **28-pin DIP/PLCC/LCC**
- **0.8µ BiCMOS**

Functional Description

The CY7B92X HOTLink Transmitter and CY7B93X HOTLink Receiver are point-to-point communications building blocks that transfer data over high-speed serial links (fiber, coax, and twisted pair) at 130 to 310 Mbits/second. *Figure 1* illustrates typical connections to host systems or controllers.

Eight bits of user data or protocol information are loaded into the HOTLink transmitter and are encoded. Serial data is shifted out of the three differential Pseudo ECL (PECL) serial ports at the bit rate (which is 10 times the byte rate).

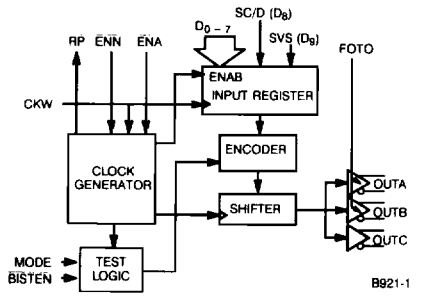
The HOTLink receiver accepts the serial bit stream at its differential line receiver inputs, and using a completely integrated PLL clock synchronizer recovers the timing information necessary for data reconstruction. The bit stream is deserialized, decoded, and checked for transmission errors. The recovered byte is presented in parallel to the receiving host along with a byte rate clock.

The 8B/10B encoder/decoder can be disabled in systems that already encode or scramble the transmitted data. I/Os are available to create a seamless interface with both asynchronous FIFOs (i.e., CY7C42X) and clocked FIFOs (i.e., CY7C44X). A Built-In Self-Test pattern generator and checker allows testing of the transmitter, receiver, and the connecting link as a part of a system diagnostic check.

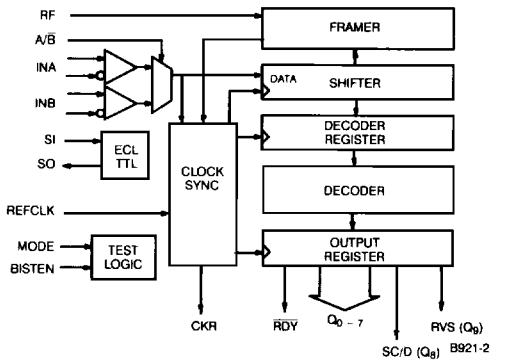
HOTLink devices are ideal for a variety of applications where a parallel interface can be replaced with a high-speed point-to-point serial link. Applications include interconnecting workstations, servers, mass storage, and video transmission equipment.

7 COMM

CY7B92X Transmitter Logic Block Diagram



CY7B93X Receiver Logic Block Diagram



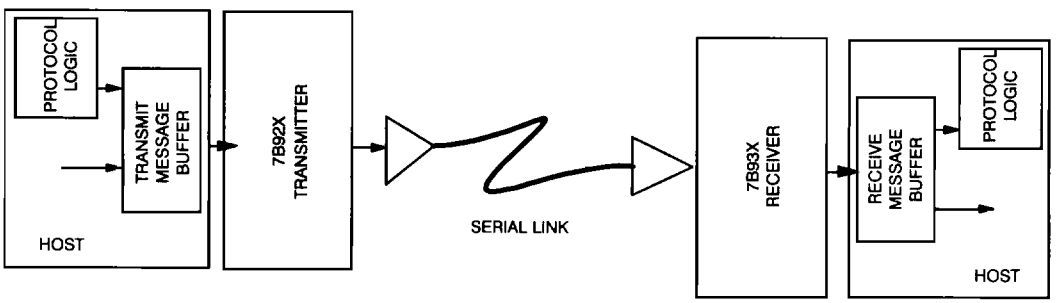
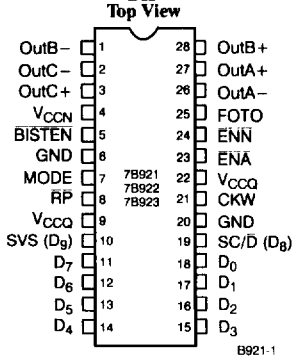


Figure 1. HOTLink System Connections

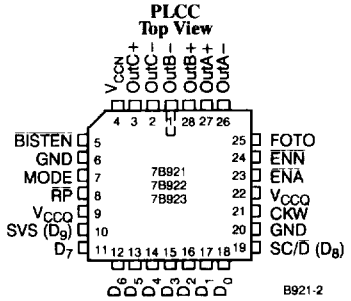
HOTLink is a registered trademark of Cypress Semiconductor Corporation.
ESCON is a registered trademark of IBM.

CY7B92X Transmitter Pin Configurations

DIP
Top View

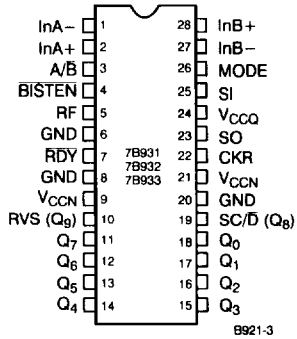


PLCC
Top View

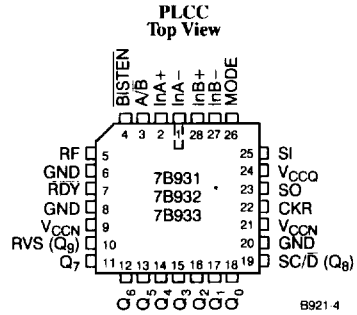


CY7B93X Receiver Pin Configurations

DIP
Top View



PLCC
Top View



Selection Guide

Transmitter Receiver	7B921 7B931	7B922 7B932	7B923 7B933
Transmission Rate (Mbits/sec)	130–170	170–240	240–310
Transmission Rate (Mbytes/sec)	13–17	17–24	24–31

Maximum Ratings

(Above which the useful life may be impaired. For user guidelines, not tested.)

Storage Temperature ..... –65 C to +150 C

Ambient Temperature with Power Applied ..... –55 C to +125 C

Supply Voltage to Ground Potential ..... –0.5V to +7.0V

DC Input Voltage ..... –0.5V to +7.0V

Output Current into TLL Outputs (LOW) ..... 30 mA

Output Current into ECL outputs (HIGH) ..... –50 mA

Static Discharge Voltage ..... >2001V (per MIL–STD–883, Method 3015)

Latch-Up Current ..... >200 mA

Operating Range

Range	Ambient Temperature	V_{CC}
Commercial	0°C to +70°C	5V ± 10%
Industrial	–40°C to +85°C	5V ± 10%
Military	–55°C to +125°C Case Temperature	5V ± 10%

Pin Descriptions

CY7B92X HOTLink Transmitter

Name	I/O	Description
D_{0-7}	TTL In	Parallel Data Input. Data is clocked into the Transmitter on the rising edge of CKW if $\overline{ENA}$ is LOW (or on the next rising CKW with $\overline{ENN}$ LOW). If $\overline{ENA}$ and $\overline{ENN}$ are HIGH, a Null character (K28.5) is sent.
SC/$\overline{D}$ (D_8)	TTL In	Special Character/Data Select. A HIGH on SC/$\overline{D}$ when CKW rises causes the transmitter to encode the pattern on D_{0-7} as a control code (Special Character), while a LOW causes the data to be coded using the 8B/10B data alphabet. When MODE is HIGH, SC/$\overline{D}$ (D_8) acts as D_8 input.
SVS (D_9)	TTL In	Send Violation Symbol. If SVS is HIGH when CKW rises, a Violation symbol is encoded and sent while the data on the parallel inputs is ignored. If SVS is LOW, the state of D_{0-7} and SC/$\overline{D}$ determines the code sent. In BIST mode, SVS overrides the BIST generator and forces the transmission of a Violation code. When MODE is HIGH, SC/$\overline{D}$ (D_9) acts as D_9 input.
$\overline{ENA}$	TTL In	Enable Parallel Data. If $\overline{ENA}$ is LOW on the rising edge of CKW, the data is loaded, encoded, and sent. If $\overline{ENA}$ is HIGH, the data inputs are ignored and the Transmitter will insert a Null character (K28.5) to fill the space between user data. $\overline{ENA}$ may be held HIGH/LOW continuously or it may be pulsed with each data byte to be sent. If $\overline{ENA}$ is being used for data control, $\overline{ENN}$ will normally be strapped HIGH, but can be used for BIST function control.
$\overline{ENN}$	TTL In	Enable Next Parallel Data. If $\overline{ENN}$ is LOW, the data appearing on D_{0-7} at the next rising edge of CKW is loaded, encoded, and sent. If $\overline{ENN}$ is HIGH, the data appearing on D_{0-7} at the next rising edge of CKW will be ignored and the Transmitter will insert a Null character to fill the space between user data. $\overline{ENN}$ may be held HIGH/LOW continuously or it may be pulsed with each data byte sent. If $\overline{ENN}$ is being used for data control, $\overline{ENA}$ will normally be strapped HIGH, but can be used for BIST function control.
CKW	TTL In	Clock Write. CKW is both the clock frequency reference for the multiplying PLL that generates the high-speed transmit clock, and the byte rate write signal that synchronizes the parallel data input. CKW must be connected to a crystal controlled time base that runs within the specified frequency range of the Transmitter and Receiver.
FOTO	TTL In	Fiber Optic Transmitter Off. FOTO determines the function of two of the three ECL transmitter output pairs. If FOTO is LOW, the data encoded by the Transmitter will appear at the outputs continuously. If FOTO is HIGH, OUTA± and OUTB± are forced to their "logic zero" state (OUT+ = LOW and OUT− = HIGH), causing a fiber optic transmit module to extinguish its light output. OUTC is unaffected by the level on FOTO, and can be used as a loop-back signal source for board-level diagnostic testing.
OUT A± OUT B± OUT C±	ECL Out	Differential Serial Data Outputs. These ECL 100K outputs (+5V referenced) are capable of driving terminated transmission lines or commercial fiber optic transmitter modules. Unused pairs of outputs can be wired to V_{CC} to reduce power if the output is not required. OUTA± and OUTB± are controlled by the level on FOTO, and will remain at their "logical zero" states when FOTO is asserted. OUTC± is unaffected by the level on FOTO. (OUTA+ and OUTB+ are used as a differential test clock input while in Test mode.)
MODE	3-Level In	Encoder Mode Select. The level on MODE determines the encoding method to be used. When wired LOW, MODE selects 8B/10B encoding. When wired HIGH, data inputs bypass the encoder and the bit pattern on D_{0-7}, D_8, and D_9 goes directly to the shifter. When left floating (internal resistors hold the input at $V_{CC}/2$) the internal bit-clock generator is disabled and OUTA+/OUTB+ become the differential bit clock to be used for factory test. In typical applications MODE is tied HIGH or LOW.
$\overline{BISTEN}$	TTL In	Built-In Self-Test Enable. When $\overline{BISTEN}$ is LOW and $\overline{ENA}$ and $\overline{ENN}$ are HIGH, the transmitter sends an alternating 1−0 pattern (D10.2 or D21.5). When either $\overline{ENA}$ or $\overline{ENN}$ is set LOW the transmitter begins a repeating test sequence that allows the Transmitter and Receiver to work together to test the function of the entire link. In normal use this input is held HIGH or wired to V_{CC}. The BIST generator is a free-running pattern generator that need not be initialized, but if required, the BIST sequence can be initialized by momentarily asserting SVS while $\overline{BISTEN}$ is LOW.
$\overline{RP}$	TTL Out	Read Pulse. $\overline{RP}$ is a 70% LOW duty-cycle byte-rate pulse train suitable for the read pulse in CY7C42X FIFOs. The frequency on $\overline{RP}$ is the same as CKW when enabled by $\overline{ENA}$, and duty cycle is independent of the CKW duty cycle. Pulse widths are set by logic internal to the transmitter. In BIST mode, $\overline{RP}$ will remain HIGH for all but the last byte of a test loop. $\overline{RP}$ will pulse LOW one byte time per BIST loop.
V_{CCN}		Power for output drivers.
V_{CCQ}		Power for internal circuitry.
GND		Ground.

CY7B93X HOTLink Receiver

Name	I/O	Description
Q_{0-7}	TTL Out	Q_{0-7} Parallel Data Output. Q_{0-7} contain the most recently received data. These outputs change synchronously with CKR.
SC/$\overline{D}$ (Q_8)	TTL Out	Special Character/Data Select. SC/$\overline{D}$ indicates the context of received data. HIGH indicates a Control (Special Character) code, LOW indicates a Data character. When MODE is HIGH, SC/$\overline{D}$ acts as Q_8 output.
RVS (Q_9)	TTL Out	Received Violation Symbol. A HIGH on RVS indicates that a code rule violation has been detected in the received data stream. A LOW shows that no error has been detected. In BIST mode, a LOW on RVS indicatescorrect operation of the Transmitter, Receiver, and link on a byte-by-byte basis. When MODE is HIGH, RVS acts as Q_9 output.
$\overline{RDY}$	TTL Out	Data Output Ready. A LOW pulse on $\overline{RDY}$ indicates that new data has been received and is ready to be delivered. A missing pulse on $\overline{RDY}$ shows that the received data is the Null character (normally inserted by the transmitter as a pad between data inputs). In BIST mode $\overline{RDY}$ will remain LOW for all but the last byte of a test loop and will pulse HIGH one byte time per BIST loop.
CKR	TTL Out	Clock Read. This byte rate clock output is phase and frequency aligned to the incoming serial data stream. $\overline{RDY}$, Q_{0-7}, SC/$\overline{D}$ (Q_8), and RVS (Q_9) all switch synchronously with the rising edge of this output.
INA± INB±	Diff In	Differential Serial Data Inputs. The differential signal at the receiver end of the communication link is connectedto the differential pairs INA± or INB±. Either the INA pair or the INB pair can be used as the main data input and the other can serve as a loop-back channel or as an alternative data input selected by the state of A/$\overline{B}$. INB± is used as the test clock while in Test mode.
A/$\overline{B}$	ECL in	Serial Data Input Select. This ECL 100K (+5V referenced) input selects INA or INB as the active data input. If A/$\overline{B}$ is HIGH, INA is connected to the shifter and signals connected to INA will be decoded. If A/$\overline{B}$ is LOW INB is selected.
SI	ECL in	Status In. The ECL 100K (+5V referenced) signal appearing on SI is translated to a TTL signal at SO. SI is typically used to translate the Carrier Detect output from a fiber optic receiver.
SO	TTL Out	Status Out. SO is the TTL translated output of SI. It is typically used to translate the Carrier Detect output from a fiber optic receiver.
RF	TTL In	Reframe Enable. RF controls the Framer logic in the Receiver. When RF is held HIGH, each SYNC (K28.5) symbol detected in the shifter will frame the data that follows. When RF is held LOW, the reframing logic is disabled. The incoming data stream is then continuously de-serialized and decoded using byte boundaries set by the internal byte counter. Bit errors in the data stream will not cause alias SYNC characters to reframe the data erroneously.
REFCLK	TTL In	Reference Clock. REFCLK is the clock frequency reference for the clock/data synchronizing PLL. REFCLK sets the approximate center frequency for the internal PLL to track the incoming bit stream. REFCLK must be connected to a crystal-controlled time base that runs within the frequency limits of the Tx/Rx pair, and the frequency must be the same as the transmitter CKW frequency (within CKW±0.1%).
MODE	TTL In	Decoder Mode Select. The level on the MODE pin determines the decoding method to be used. When tied LOW, MODE selects 8B/10B decoding. When tied HIGH, registered shifter contents bypass the decoder and are sent to Q_{0-7}, SC/$\overline{D}$ and RVS directly. When left floating (internal resistors hold the MODE pin at $V_{CC}/2$) the internal bit clock generator is disabled and INB± becomes the bit rate test clock to be used for factory test. In typical applications, MODE is tied HIGH or LOW.
$\overline{BISTEN}$	TTL In	Built-In Self-Test Enable. When $\overline{BISTEN}$ is LOW the Receiver awaits a D0.0 (sent once per BIST loop) character and begins a continuous test sequence that tests the functionality of the Transmitter, the Receiver, and the link connecting them. In BIST mode the status of the test can be monitored with $\overline{RDY}$ and RVS outputs. In normal use $\overline{BISTEN}$ is held HIGH or wired to V_{CC}.
V_{CCN}		Power for output drivers.
V_{CCQ}		Power for internal circuitry.
GND		Ground

CY7B92X HOTLink Transmitter Block Diagram Description

Input Register

The Input register holds the data to be processed by the HOTLink transmitter and allows the input timing to be made consistent with a standard FIFOs. The Input register is clocked by CKW and loaded with information on the D_{0-7}, SC/$\overline{D}$ (D_8), and SVS (D_9) pins. Two enable inputs ($\overline{ENA}$ and $\overline{ENN}$) allow the user to choose when data is to be sent. Asserting $\overline{ENA}$ (Enable, LOW) causes the inputs to be loaded on the rising edge of CKW. If $\overline{ENN}$ (Enable Next, LOW) is asserted when CKW rises, the data present on the inputs will be loaded into the input register on the next rising edge of CKW. These two inputs allow proper timing and function for compatibility with either asynchronous FIFOs or clocked FIFOs without external logic, as shown in *Figure 2.*

In BIST mode, the Input register becomes the signature pattern generator by logically converting the parallel input register into a Linear Feedback Shift Register(LFSR). When enabled, this LFSR will generate all possible input patterns in a predictable but pseudo-random sequence that can be matched to an identical LFSR in the Receiver.

Encoder

The Encoder transforms the input data held by the Input register into a form more suitable for transmission on a serial interface link. The code used is specified by ANSI X3T9.3 (Fibre Channel) and the IBM ESCON channel (code tables are at the end of this data-sheet). The eight D_{0-7} data inputs are converted to either a DATA symbol or a Special Character, depending upon the state of the SC/$\overline{D}$ input. If SC/$\overline{D}$ is HIGH, the data inputs represent a control code and is encoded using the Special Character code tables. If SC/$\overline{D}$ is LOW, the data inputs are converted using the DATA code table. If a byte time passes with the inputs disabled, the Encoder will output a Special Character Comma K28.5 (or SYNC) that will maintain link synchronization. Strings of SYNC will be decoded in the Receiver as Null characters, thus simplifying the system control logic for FIFO interfaces. SVS input forces the transmission of a specified Violation symbol to allow the user to check error handling system logic in the controller.

The 8B/10B coding function of the Encoder can be bypassed for systems that include an external coder or scrambler function as part of the controller. This bypass is controlled by the MODE select pin. When in bypass mode, D_{0-7}, SC/$\overline{D}$ (D_8), and SVS (D_9) become the ten inputs to the Shifter.

Shifter

The Shifter accepts parallel data from the Encoder once each byte time and shifts it to the serial interface output buffers using a PLL multiplied bit clock that runs at ten (10) times the byte clock rate. Timing for the parallel transfer is controlled by the counter included in the Clock Generator and is not affected by signal levels or timing at the input pins.

OutA, OutB, OutC

The serial interface ECL output buffers (100K referenced to +5v) are the drivers for the serial media. They are all connected to the Shifter and contain the same serial data. Two of the output pairs (OUTA± and OUTB±) are controllable by the FOTO input and can be disabled by the system controller to force a logical zero (i.e., "light off") at the outputs. The third output pair (OUTC±) is not affected by FOTO and will supply a continuous data stream suitable for loop-back testing of the subsystem.

OUTA± and OUTB± will respond to FOTO input changes within a few bit times. However, since FOTO is not synchronized with the transmitter data stream, the outputs will be forced off or turned on at arbitrary points in a transmitted byte. This function is intended to augment an external laser safety controller and as an aid for Receiver PLL testing, and thus need not be synchronized.

In wire-based systems, control of the outputs may not be required, and FOTO can be strapped LOW. The three outputs are intended to add system and architectural flexibility by offering identical serial bit streams with separate interfaces for redundant connections or for multiple destinations. Unneeded outputs can be wired to V_{CC} to disable and power down the unused output circuitry.

Clock Generator

The clock generator is an embedded phase-locked loop (PLL) that takes a byte-rate reference clock (CKW) and multiplies by ten (10) to create a bit rate clock for driving the serial shifter. The byte rate reference comes from CKW, the rising edge of which clocks data into the Input register. This clock must be a crystal referenced pulse stream that has a frequency between the minimum and maximum specified for the HOTLink Transmitter/Receiver pair. (Each Transmit/Receive pair; 7B921/931, 7B922/932, 7B923/933 have a specified range of operating frequencies.) Signals controlled by this block form the bit clock and the timing signals that control internal data transfers between the Input register and the Shifter.

The read pulse ($\overline{RP}$) is derived from the feedback counter used in the PLL multiplier. It is a byte-rate pulse stream with the proper phase and pulse widths to allow transfer of data from an asynchronous FIFO. Pulse width is independent of CKW duty cycle, since proper phase and duty cycle is maintained by the PLL. The $\overline{RP}$ pulse stream will insure correct data transfers between asynchronous FIFOs and the transmitter input latch with no external logic.

Test Logic

Test logic includes the initialization and control for the Built-In Self-Test (BIST) generator, the multiplexer for Test mode clock distribution,and control logic to properly select the data encoding. Test logic is discussed in more detail in the CY7B92X HOTLink TransmitterOperating Mode Description.

CY7B93X HOTLink Receiver Block Diagram Description

Differential Inputs

This pair of differential line receivers are the inputs for the serial data stream. INA± or INB± can be selected with the A/$\overline{B}$ input. INA± is selected with A/$\overline{B}$ HIGH and INB± is selected with A/$\overline{B}$ LOW. The threshold of A/$\overline{B}$ is compatible with the ECL 100K signals from ECL fiber optic interface modules. The differential threshold of INA± and INB± will accommodate wire interconnect with filtering losses or transmission line attenuation greater than 20 db ($V_{DIF} \geq 50$mv) or can be directly connected to fiber optic interface modules (any ECL logic family, not limited to ECL 100K) with up to 1.2 volts of differential signal. The common mode tolerance will accommodate a wide range of signal termination voltages. The highest HIGH input that can be tolerated is $V_{IN} = V_{CC}$, and the lowest LOW input that can be interpreted correctly is V_{IN} = GND+2.5V.

ECL-TTL Translator

This positive-referenced ECL-to-TTL translator is provided to eliminate external logic between an ECL fiber-optic interface module "carrier detect" output and the TTL input in the control logic. The input threshold is compatible with ECL 100K levels (+5V referenced). It can also be used as part of the link status indication logic for wire connected systems.

Clock Sync

The Clock Synchronizer function is performed by an embedded phase-lockedloop (PLL) that tracks the frequency of the incoming bit stream and aligns the phase of its internal bit rate clock to the serial data transitions. This block contains the logic to transfer the data from the Shifter to the Decode register once every byte. The counter that controls this transfer is initialized by the Framer logic. CKR is a buffered output derived from the bit counter used to control Decode register and Output register transfers.

Clock output logic is designed so that when reframing causes the counter sequence to be interrupted, the period and pulse width of CKR will never be less than expected. Reframing may stretch the period of CKR by up to 90%, and either CKR Pulse Width HIGH or Pulse Width LOW may be stretched, depending on when reframe occurs.

The REFCLK input provides a byte-rate reference frequency to improve PLL acquisition time and limit unlocked frequency excursions of the CKR when no data is present at the serial inputs. The frequency of REFCLK is required to be within ±0.1% of the frequency of the clock that drives the transmitter CKW pin.

Framer

Framer logic checks the incoming bit stream for the pattern that defines the byte boundaries. This combinatorial logic filter looks for the X3T9.3 symbol defined as a Special Character Comma (K28.5). When it is found, the free-running bit counter in the Clock Sync block is synchronously reset to its initial state, thus framing the data correctly on the correct byte boundaries. The Fibre Channel specification optionally allows this 10-bit pattern (001111 1000 or 110000 0111) to be detected using only a 7-bit detector (but restricts usage of other Data and Special Character codes). Framer logic in the Receiver will completely decode all ten (10) bits of K28.5 to reframe, and thus remove the limitations on code sequences.

The Framer can be inhibited by holding the RF input LOW. When RF rises, $\overline{RDY}$ will be inhibited until a K28.5 has been detected, after which $\overline{RDY}$ will resume its normal function.

Shifter

The Shifter accepts serial inputs from the Differential inputs one bit at a time, as clocked by the Clock Sync logic. Data is transferred to the Framer on each bit, and to the Decode register once per byte.

Decode Register

The Decode register accepts data from the Shifter once per byte as determined by the logic in the Clock Sync block. It is presented to the Decoder and held until it is transferred to the output latch.

Decoder

Parallel data is transformed from ANSI X3T9.3 8B/10B codes back to "raw data" in the Decoder. This block uses the standard decoder patterns shown in the Valid Data Characters and Valid Special Character Codes and Sequences sections of this datasheet. Data patterns are signaled by a LOW on the SC/$\overline{D}$ output and Special Character patterns are signaled by a HIGH on the SC/$\overline{D}$ output. Unused patterns or disparity errors are signaled as errors by a HIGH on the RVS output and by specific Special Character codes.

Output Register

The Output register holds the recovered data (Q_{0-7}, SC/$\overline{D}$, and RVS) and aligns it with the recovered byte clock (CKR). This synchronization insures proper timing to match a FIFO interface or other logic that requires glitch free and specified output behavior. Outputs are changed synchronously with the rising edge of CKR.

In BIST mode, this register becomes the signature pattern generator and checker by logically converting the parallel output register into a Linear Feedback Shift Register (LFSR) pattern generator. When enabled, this LFSR will generate all possible code patterns in a predictable but pseudo-random sequence that can be matched to an identical LFSR in the Transmitter. When synchronized, it checks each byte in the Decoder with each byte generated by the LFSR and shows errors at RVS. Patterns generated by the LFSR are compared after being buffered to the output pins and then fed back to the comparators, allowing test of the entire receive function.

In BIST mode, the LFSR is initialized by the first occurrence of the transmitter BIST loop start code D0.0 (D0.0 is sent only once per BIST loop). Once the BIST loop has been started, RVS will be HIGH for pattern mismatches between the received sequence and the internally generated sequence. Code rule violations or running disparity errors that occur as part of the BIST loop will not cause an error indication. $\overline{RDY}$ will pulse HIGH once per BIST loop and can be used to check test pattern progress. If it is suspected that the receiver pattern generator has lost sync with the transmitter BIST pattern, the receiver BIST generator can be reinitialized by leaving and re-entering BIST mode.

Test Logic

Test logic includes the initialization and control for the Built-In Self-Test (BIST) generator, the multiplexer for Test mode clock distribution, and control logic for the decoder. Test logic is discussed in more detail in the CY7B93X HOTLink Receiver Operating Mode Description.

CY7B92X/CY7B93X Electrical Characteristics Over the Operating Range[1]

Parameter	Description	Test Conditions	Min.	Max.	Units
Transmitter TTL-Compatible Pins: D_{0-7}, SC/$\overline{D}$, SVS, $\overline{ENA}$, $\overline{ENN}$, CKW, FOTO, $\overline{BISTEN}$, $\overline{RP}$ Receiver TTL- Compatible Pins: Q_{0-7}, SC/$\overline{D}$, RVS, $\overline{RDY}$, CKR, REFCLK, RF, $\overline{BISTEN}$, SO					
V_{OHT}	Output HIGH Voltage	I_{OH} = − 2 mA	2.4		V
V_{OLT}	Output LOW Voltage	I_{OL} = 4 mA		0.45	V
I_{OST}	Output Short Circuit Current	V_{OUT} = 0V[2]	−15	− 90	mA
V_{IHT}	Input HIGH Voltage		2.0	V_{CC}	V
V_{ILT}	Input LOW Voltage		− 0.5	0.8	V
I_{IHT}	Input HIGH Current	V_{IN} = V_{CC}	− 10	+10	µA
I_{ILT}	Input LOW Current	V_{IN} = 0.0V		− 500	µA
Transmitter ECL-Compatible Output Pins: OUTA+, OUTA−, OUTB+, OUTB−, OUTC+, OUTC−					
V_{OHE}	Output HIGH Voltage (V_{CC} referenced)	Load = 50 ohms to V_{CC} − 2V	V_{CC}−1.03	V_{CC}−0.88	V
V_{OLE}	Output LOW Voltage (V_{CC} referenced)	Load = 50 ohms to V_{CC} − 2V	V_{CC}−1.81	V_{CC}− 1.63	V
Receiver ECL-Compatible Input Pins: A/$\overline{B}$, SI					
V_{IHE}	Input HIGH Voltage		V_{CC}−1.17	V_{CC}−0.88	V
V_{ILE}	Input LOW Voltage		V_{CC}−1.81	V_{CC}−1.48	V
I_{IHE}	Input HIGH Current	V_{IN} = V_{IHE} Max.		+500	µA
I_{ILE}	Input LOW Current	V_{IN} = V_{ILL} Min.	+0.5		µA
Differential Line Receiver Input Pins: INA+, INA−, INB+, INB−					
V_{DIFF}	Input Differential Voltage |(IN+) − (IN−)|		50	1200	mV
V_{IHH}	Highest Input HIGH Voltage			V_{CC}	V
V_{ILL}	Lowest Input LOW Voltage		2.5		V
Miscellaneous					
I_{CCT}	Transmitter Power Supply Current	V_{CC} = Max., T_A = Max., Freq. = Max. (One ECL output pair loaded with 50 ohms to V_{CC} − 2.0V, others tied to V_{CC})		TBD	mA
I_{CCR}	Receiver Power Supply Current	V_{CC} = Max., T_A = Max., Freq. = Max.		TBD	mA

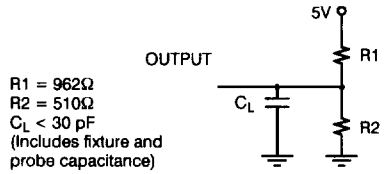
Capacitance[3]

Parameters	Description	Test Conditions	Max.	Units
C_{IN}	Input Capacitance	T_A = 25°C, f_0 = 1 MHz, V_{CC} = 5.0V	10	pF

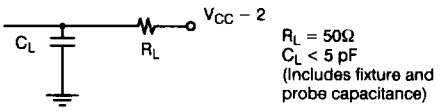
Notes:

1. See the last page of this specification for Group A subgroup testing information.
2. Tested on one output at a time, output shorted for less than one second, less than 10% duty cycle.
3. Tested initially and after any design or process changes that may affect these parameters.

AC Test Loads and Waveforms

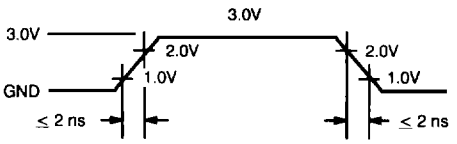


(a) TTL AC Test Load[4]



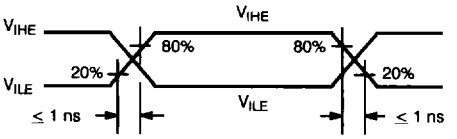
(b) ECL AC Test Load[4]

B921-5



(c) TTL Input Test Waveform

B921-6



(d) ECL Input Test Waveform

B921-7

7B921/2/3 Transmitter Switching Characteristics Over the Operating Range[1]

Parameters	Description	7B921		7B922		7B923		Units
		Min.	Max.	Min.	Max.	Min.	Max.	
t_{CKW}	Write Clock Cycle	56	76	42	57	32	43	ns
t_B	Bit Time[5]	5.6	7.6	4.2	5.7	3.2	4.3	ns
t_{CPWH}	CKW Pulse Width HIGH	9		9		9		ns
t_{CPWL}	CKW Pulse Width LOW	9		9		9		ns
t_{SD}	Data Set-Up Time[6]	5		5		5		ns
t_{HD}	Data Hold Time[6]	0		0		0		ns
t_{SEND}	Enable Set-Up Time (to capture data)[7]	5		5		5		ns
t_{SENP}	Enable Set-Up Time (to assure correct $\overline{RP}$)[8]	$7^1/_4t_B+4$		$7^1/_4t_B+4$		$7^1/_4t_B+4$		ns
t_{HEN}	Enable Hold Time	0		0		0		ns
t_{PDR}	Read Pulse Alignment[9]	$(-^1/_4t_B-3)$	$(+^1/_4t_B+3)$	$(-^1/_4t_B-3)$	$(+^1/_4t_B+3)$	$(-^1/_4t_B-3)$	$(+^1/_4t_B+3)$	ns
t_{PPWH}	Read Pulse HIGH[9]	$3t_B-3$		$3t_B-3$		$3t_B-3$		ns
t_{PPWL}	Read Pulse LOW[9]	$7t_B-3$		$7t_B-3$		$7t_B-3$		ns

Notes:

4. Cypress uses constant current (ATE) load configurations and forcing functions. This figure is for reference only.
5. Transmitter t_B is calculated as $t_{CKW}/10$. The byte rate is one tenth of the bit rate.
6. Data includes D_{0-7}, SC/$\overline{D}$ (D_8), and SVS (D_9).
7. t_{SEND} minimum timing assures correct Data load on rising edge of CKW, but not proper $\overline{RP}$ function or timing.
8. t_{SENP} minimum timing insures correct $\overline{RP}$ pulse width and correct Data load on rising edge of CKW.
9. Loading on $\overline{RP}$ pin is ≤2 mA and <15 pF.

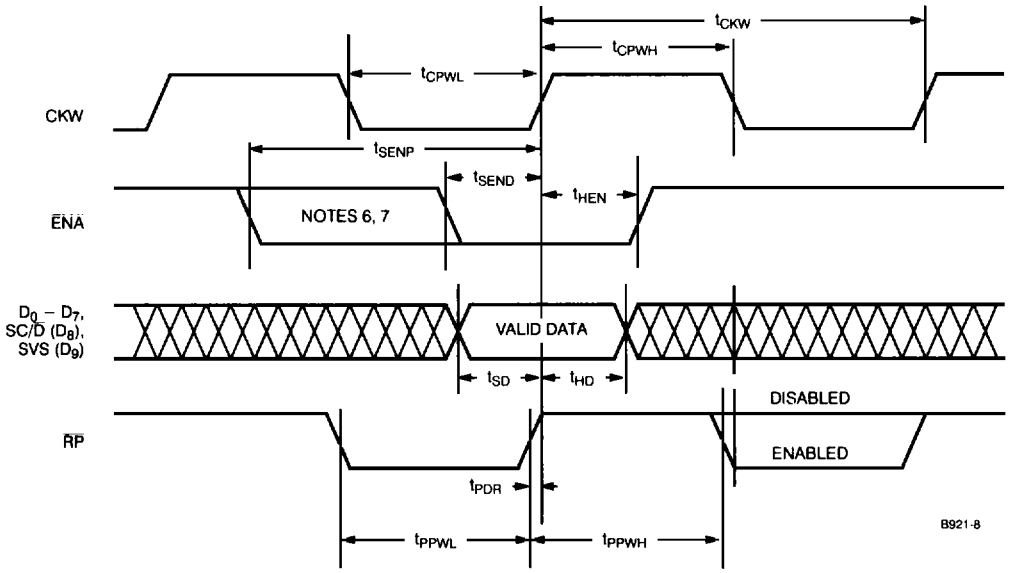
7B931/2/3 Receiver Switching Characteristics Over the Operating Range[1]

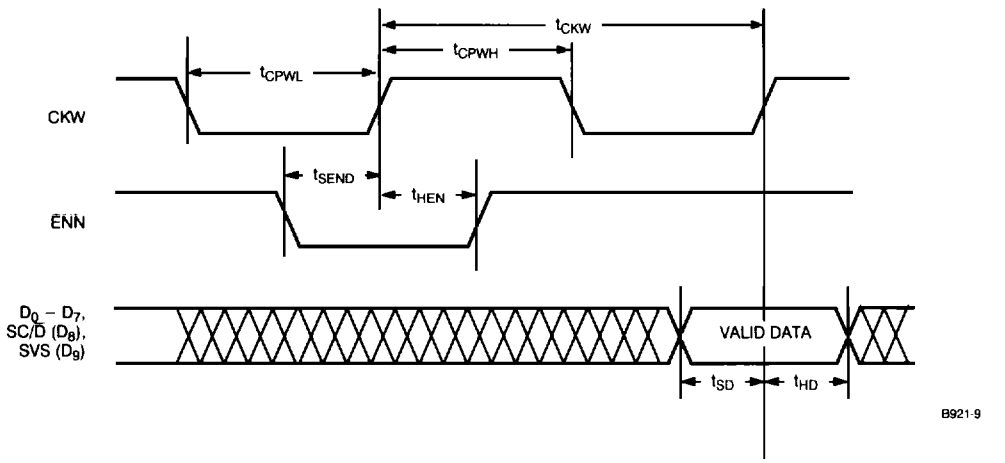
Parameters	Description	7B931		7B932		7B933		Units
		Min.	Max.	Min.	Max.	Min.	Max.	
t_{CKR}	Read Clock Period (No Serial Data Input), REFCLK as Reference[10]	−1	+1	−1	+1	−1	+1	%
t_B[11]	Bit Time	5.6	7.6	4.2	5.7	3.2	4.3	ns
t_{CPRH}	Read Clock Pulse HIGH	$5t_B-3$		$5t_B-3$		$5t_B-3$		ns
t_{CPRL}	Read Clock Pulse LOW	$5t_B-3$		$5t_B-3$		$5t_B-3$		ns
t_{RH}	$\overline{RDY}$ Hold Time	t_B-3		t_B-3		t_B-3		ns
t_{PRL}	$\overline{RDY}$ Pulse Width LOW	$6t_B-3$		$6t_B-3$		$6t_B-3$		ns
t_{PRH}	$\overline{RDY}$ Pulse Width HIGH	$4t_B-3$		$4t_B-3$		$4t_B-3$		ns
t_A	Data Access Time[12, 13]	$2t_B-3$	$2t_B+3$	$2t_B-3$	$2t_B+3$	$2t_B-3$	$2t_B+3$	ns
t_{ROH}	Data Hold Time[12, 13]	t_B-3		t_B-3		t_B-3		ns
t_{CKX}	REFCLK Clock Period Referenced to CKW of Transmitter[14]	−0.1	+0.1	−0.1	+0.1	−0.1	+0.1	%
t_{CPXH}	REFCLK Clock Pulse HIGH	9		9		9		ns
t_{CPXL}	REFCLK Clock Pulse LOW	9		9		9		ns
t_{DS}	Propagation Delay SI to SO (note ECL and TTL thresholds)[15]		15		15		15	ns

Notes:

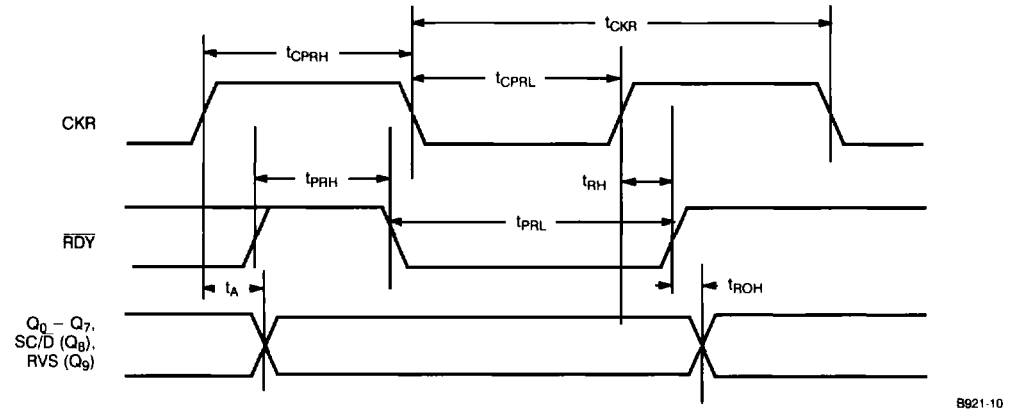
10. The period of t_{CKR} will match the period of the transmitter CKW when the receiver is receiving serial data. When data is interrupted, CKR may drift to one of the range limits above.
11. Receiver t_B is calculated as $t_{CKR}/10$ if no data is being received, or $t_{CKW}/10$ if data is being received. See note 5.
12. Data includes Q_{0-7}, SC/$\overline{D}$ (Q_8), and RVS (Q_9).
13. t_A and t_{ROH} specifications are only valid if all outputs (CKR, $\overline{RDY}$, Q_{0-7}, SC/$\overline{D}$, and RVS) are loaded with the same DC and AC load.
14. REFCLK has no phase or frequency relationship with CKR and only acts as a centering reference to reduce clock synchronization time. REFCLK must be within 0.1% of the transmitter CKW frequency, necessitating a ±500-PPM crystal.
15. The ECL switching threshold is the midpoint between the ECL–V_{OH}, and V_{OL} specification (approximately V_{CC} − 1.35V). The TTL switching threshold is 1.5V.

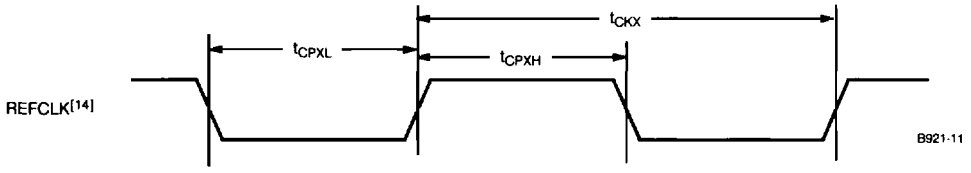
Switching Waveforms for the CY7B92X HOTlink Transmitter

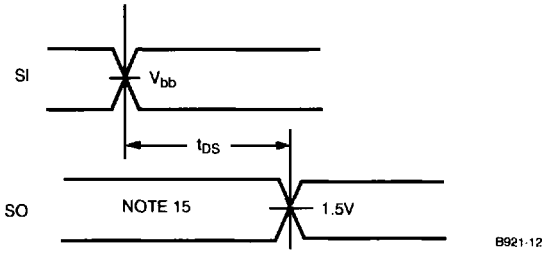




Switching Waveforms for the CY7B93X HOTlink Receiver







7 COMM

CY7B92X HOTlink Transmitter Operating Mode Description

The CY7B92X Transmitter operating with the CY7B93X Receiver forms a general-purpose data communication subsystem capable of transporting user data at up to 30 Mbytes per second over several types of serial interface media. In normal operation, the Transmitter can operate in either of two modes. The Encoded mode allows a user to send and receive eight (8) bit data and control information without first converting it to transmission characters. The Bypass mode is used for systems in which the encoding and decoding is performed on an external protocol controller.

In either mode, data is loaded into the input register of the Transmitter on the rising edge of CKW. The input timing and functional response of the Transmitter input can be made to match timing and function of either an asynchronous FIFO or a clocked FIFO by an appropriate connection of input signals (See *Figure 2*).

Encoded Mode Operation

In Encoded mode the input data is interpreted as eight bits of data (D_0 – D_7), a context control bit (SC/$\overline{D}$), and a system diagnostic input bit (SVS). If the context of the data is to be normal message data, the SC/$\overline{D}$ input will be LOW, and the data will be encoded using the valid data character set described in the Valid Data Characters section of this datasheet. If the context of the data is to be control or protocol information, the SC/$\overline{D}$ input will be HIGH, and the data will be encoded using the valid special character set described in the Valid Special Character Codes and Sequences section. Special characters include all protocol characters necessary to encode packets for Fibre Channel, ESCON, proprietary systems, and for diagnostic purposes.

The diagnostic characters and sequences available as Special Characters include those for Fibre Channel link testing, as well as codes to be used for testing system response to link errors and timing. The Violation symbol can be explicitly sent as part of a user data packet (i.e., send CE0; D_{7-0} = 1110 0000 and SC/$\overline{D}$ = 1), or can be sent in response to an external system using the SVS input. This will allow system diagnostic logic to evaluate the errors in an unambiguous manner, and will not require any modification to the transmission interface to force transmission errors for testing purposes.

Bypass Mode Operation

In Bypass mode the input data is interpreted as ten (10) bits (D_{0-7}, SC/$\overline{D}$ (D_8), and SVS (D_9)) of pre-encoded transmission data to be

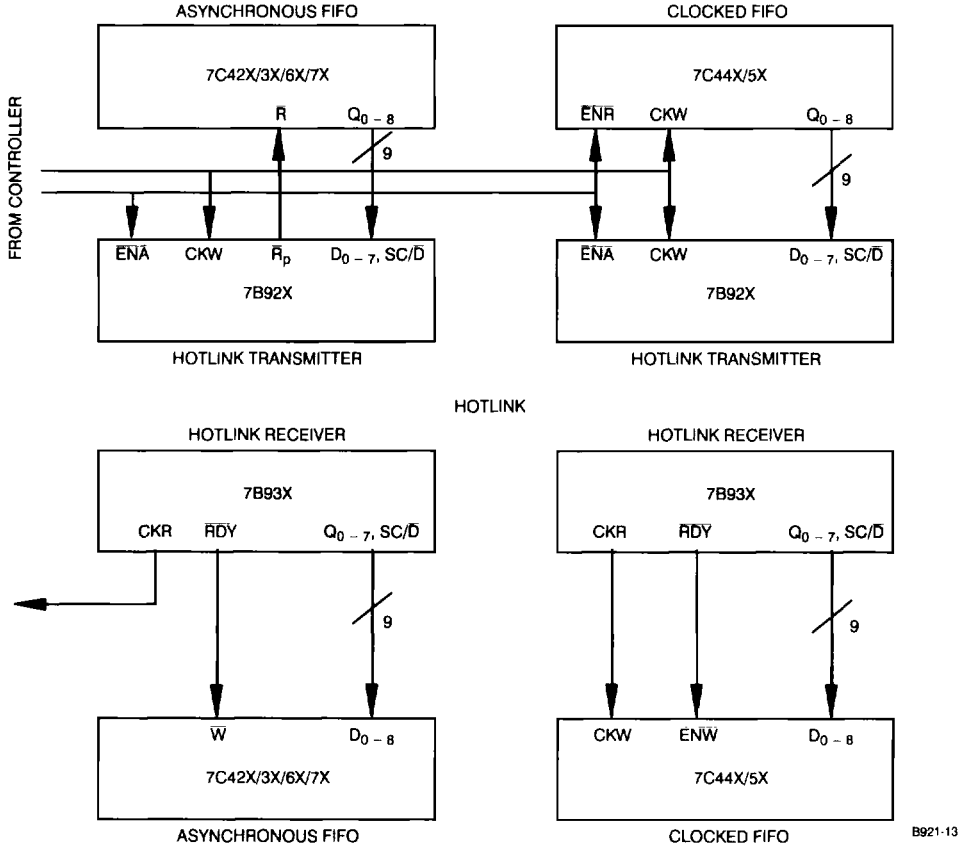


Figure 2. Seamless FIFO Interface

serialized and sent over the link. This data can use any encoding method suitable to the designer. The only restrictions upon the data encoding method is that it contain suitable transition density for the Receiver PLL data synchronizer, and that it be compatible with the transmission media.

Data loaded into the Input register on the rising edge of CKW will be loaded into the Shifter on the following rising edge of CKW. It will then be shifted to the outputs one bit at a time using the internal clock generated by the clock generator. The first bit of the transmission character will appear at the output (OUTA±, OUTB±, and OUTC±) immediately upon loading the Shifter.

While in either the Encoded mode or Bypass mode, if a CKW edge arrives when the inputs are not enabled ($\overline{ENA}$ and $\overline{ENN}$ both HIGH), the Encoder will insert a pad character K28.5 (e.g., C05) to maintain proper link synchronization (in Bypass mode the proper sense of running disparity cannot be guaranteed for the first pad character, but is correct for all pad characters that follow). This automatic insertion of pad characters can be inhibited by insuring that the Transmitter is always enabled (i.e., $\overline{ENA}$ is hard-wired LOW).

ECL Output Functional and Connection Options

The three pairs of ECL outputs all contain the same information and are intended for use in systems with multiple connections. Each output pair may be connected to a different serial media and may be a different length, link type, or interface technology. For systems that do not require all three output pairs, the unused pairs should be wired to V_{CC} to minimize the power dissipated by the output circuit, and to minimize unwanted noise generation.

In systems that require the outputs to be shut off during some periods when link transmission is prohibited (e.g., for laser safety functions), the FOTO input can be asserted. While it is possible to insure that the output state of the ECL drivers is LOW (i.e., light is off) by sending all 0's in Bypass mode, it is often inconvenient to insert this level of control into the data transmission channel, and it is impossible in Encoded mode. FOTO is provided to simplify and augment this control function (typically found in laser-based transmission systems). FOTO will force OUTA± and OUTB± to go LOW, while allowing OUTC± to continue to function normally (OUTC is typically used as a diagnostic feedback and cannot be disabled). This separation of function allows various system configurations without undue load on the control function or data channel logic.

Transmitter Test Mode Description

The CY7B92X Transmitter offers two types of test mode operation, BIST mode and Test mode. In a normal system application, the Built-In Self-Test (BIST) mode can be used to check the functionality of the Transmitter, the Receiver, and the link connecting them. This mode is available with minimal impact on user system logic, and can be used as part of the normal system diagnostics. Typical connections and timing are shown in *Figure 3*.

BIST Mode

BIST mode functions as follows:

1. Set $\overline{BISTEN}$ LOW to begin test pattern generation. Transmitter begins sending bit rate ...1010...
2. Set either $\overline{ENA}$ or $\overline{ENN}$ LOW to begin pattern sequence generation (use of Enable pin not being used for normal FIFO interface can minimize logic delays between the FIFO and transmitter).
3. Allow the Transmitter to run through several BIST loops or until the Receiver test is complete. $\overline{RP}$ will pulse HIGH once per BIST loop, and can be used to count the number of test pattern loops.
4. When testing is completed, set $\overline{BISTEN}$ HIGH and $\overline{ENA}$ and $\overline{ENN}$ HIGH and resume normal function.

Note: It may be advisable to send violation characters to test the RVS output in the Receiver. This can be done by explicitly sending a violation with the SVS input, or allowing the transmitter BIST loop to run while the Receiver runs in normal mode. The BIST loop includes deliberate violation symbols and will test the RVS function adequately.

BIST mode is intended to check the entire function of the Transmitter (except the Transmitter input pins and the bypass function in the Encoder), the serial link, and the Receiver. It augments normal factory ATE testing and provides the designer with a rigorous test mechanism to check the link transmission system without requiring any significant system overhead.

When in Bypass mode, the BIST logic will function in the same way as in the Encoded mode. MODE = HIGH and $\overline{BISTEN}$ = LOW causes the Transmitter to switch to Encoded mode and begin sending the BIST pattern, as if MODE = LOW. When $\overline{BISTEN}$ returns to HIGH, the Transmitter resumes normal BYPASS operation. In Test mode the BIST function works as in the Normal mode.

Test Mode

The MODE input pin selects between three transmitter functional modes. When wired to HIGH, the D_{0-7}, SVS, and SC/$\overline{D}$ inputs bypass the Encoder and load directly from the Input register into the Shifter. When wired to LOW, the inputs are encoded using the 8B/10B codes and sequences shown at the end of this datasheet. Since the Transmitter is usually hard wired to Encoded or Bypass mode, a third function is provided for the MODE pin. Test mode is used for factory or incoming device test. Test mode is selected by floating the MODE pin (internal resistors hold the MODE pin at $V_{CC}/2$.)

Test mode causes the Transmitter to function in its Encoded mode, but with OutA+/OutB+ (used as a differential test clock input) as the bit rate clock input instead of the internal PLL-generated bit clock. In this mode, inputs are clocked by CKW and transfers between the Input register and Shifter are timed by the internal counters. The phase and pulse width of $\overline{RP}$ are controlled by phases of the bit counter (PLL feedback counter) as in Normal mode. Input and output patterns can be synchronized with internal logic by observing the state of $\overline{RP}$ or the device can be initialized to match an ATE test pattern using the following technique:

1. Assert Test mode for several test clock cycles to establish normal counter sequence.
2. Assert $\overline{BISTEN}$ for one or more test clock cycles.
3. Deassert $\overline{BISTEN}$ and the next test clock cycle will reset the counter.
4. Proceed with pattern, voltage, and timing tests.

Test mode is intended to allow logical, DC, and AC testing of the Transmitter without requiring that the tester check output data patterns at the 300-MHz bit rate, or accommodate the PLL lock, tracking, and frequency range characteristics that are required when the HOTLink part operates in its normal mode. To use OutA+/OutB+ as the test clock input, the FOTO input is held HIGH while in Test mode. This forces the two outputs to go to an "ECL LOW," which can be ignored while the test system creates a differential input signal at some higher voltage.

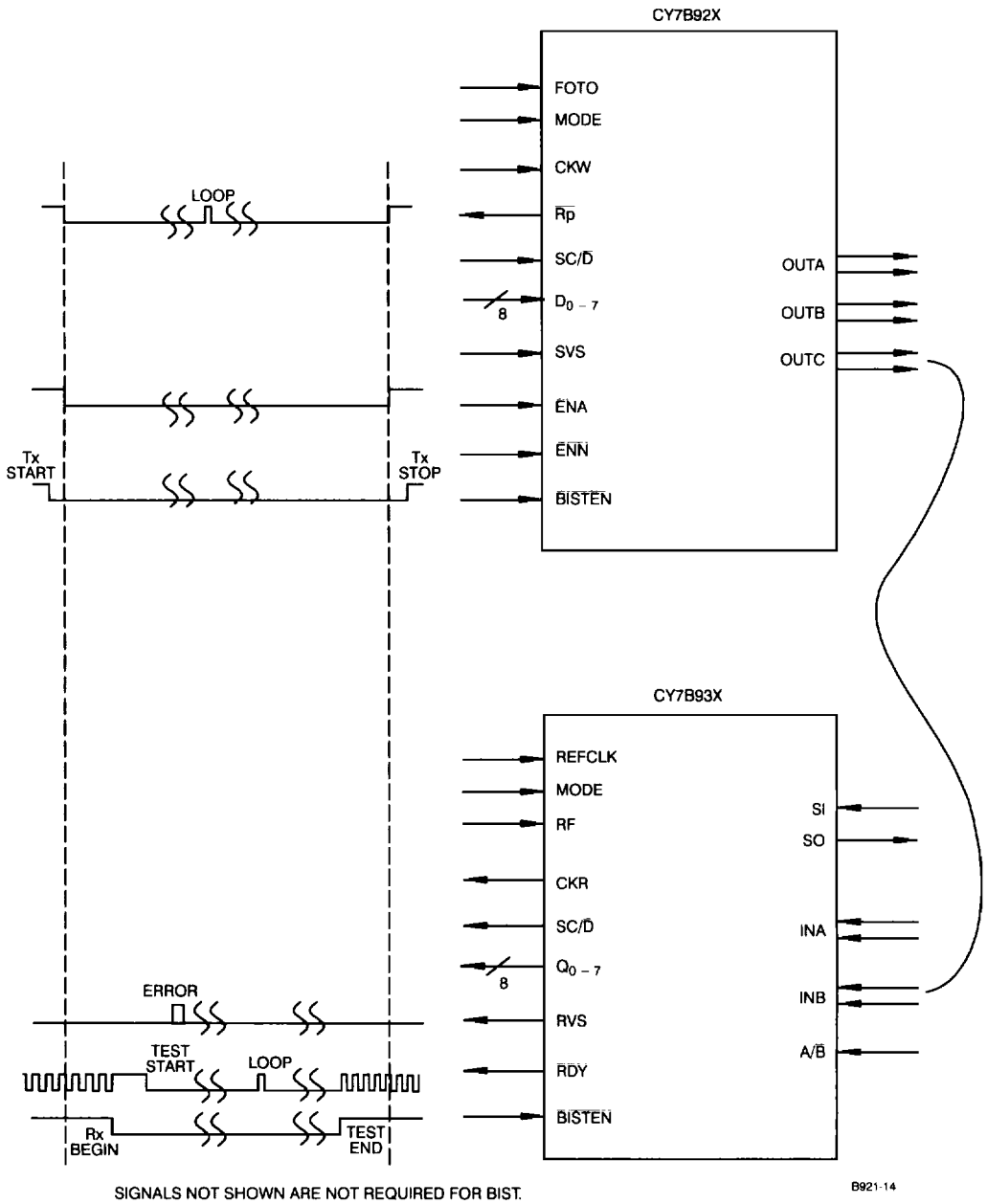


Figure 3. Built-In Self-Test Illustration

CY7B93X HOTlink Receiver Operating Mode Description

The CY7B93X Receiver operating with the CY7B92X Transmitter forms a general-purpose data communication subsystem capable of transporting user data at up to 30 Mbytes per second over several types of serial interface media. In normal user operation, the Receiver can operate in either of two modes. The Encoded mode allows a user system to send and receive 8-bit data and control information without first converting it to transmission characters. The Bypass mode is used for systems in which the encoding and decoding is performed by an external protocol controller.

In either mode, serial data is received at one of the differential line receiver inputs and routed to the Shifter and the Clock Synchronizer. The PLL in the Clock Sync aligns the internally generated bit rate clock with the incoming data stream and clocks the data into the shifter. At the end of a byte time (ten bit times), the data accumulated in the shifter is transferred to the Decode register.

To properly align the incoming bit stream to the intended byte boundaries, the bit counter in Clock Sync must be initialized. Framer logic block checks the incoming bit stream for the unique pattern that defines the byte boundaries. This combinatorial logic filter looks for the X3T9.3 symbol defined as "Special Character Comma" (K28.5). Once K28.5 is found, the free running bit counter in the Clock Sync block is synchronously reset to its initial state, thus "framing" the data to the correct byte boundaries.

Since noise-induced errors can cause the incoming data to be corrupted, and since many combinations of error and legal data can create an alias K28.5, an option is included to disable resynchronization of the bit counter. The Framer will be inhibited when the RF input is held LOW. When RF rises, $\overline{RDY}$ will be inhibited until a K28.5 has been detected, and $\overline{RDY}$ will resume its normal function. Data will continue to flow through the Receiver while $\overline{RDY}$ is inhibited.

Encoded Mode Operation

In Encoded mode the serial input data is decoded into eight bits of data ($Q_0 - Q_7$), a context control bit ($SC/\overline{D}$), and a system diagnostic output bit (RVS). If the pattern in the Decode register is found in the Valid Data Characters table, the context of the data is decoded as normal message data and the $SC/\overline{D}$ output will be LOW. If the incoming bit pattern is found in the Valid Special Character Codes and Sequences table, it is interpreted as "control" or "protocol information," and the $SC/\overline{D}$ output will be HIGH. Special characters include all protocol characters defined for use in packets for Fibre Channel, ESCON, and other proprietary and diagnostic purposes.

The Violation symbol that can be explicitly sent as part of a user data packet (i.e., Transmitter sending CE0; D_{7-0} = 1110 0000 and $SC/\overline{D}$ = 1; or SVS = 1) will be decoded and indicated in exactly the same way as a noise-induced error in the transmission link. This function will allow system diagnostics to evaluate the error in an unambiguous manner, and will not require any modification to the receiver data interface for error-testing purposes.

Bypass Mode Operation

In Bypass mode the serial input data is not decoded, and is transferred directly to the Output register's 10 bits (Q_{0-7}, Q_8, and Q_9). It is assumed that the data has been pre-encoded prior to transmission, and will be decoded in subsequent logic external to HOTLink. This data can use any encoding method suitable to the designer. The only restrictions upon the data encoding method is that it contain suitable transition density for the Receiver PLL data synchronizer, and that it be compatible with the transmission media.

The framer function in Bypass mode is identical to Encoded mode, so a K28.5 pattern can still be used to re-frame the serial bit stream.

Parallel Output Function

The 10 outputs (Q_{0-7}, $SC/\overline{D}$, and RVS) all transition simultaneously, and are aligned with $\overline{RDY}$ and CKR with timing allowances to interface directly with either an asynchronous FIFO or a clocked FIFO. Typical FIFO connections are shown in *Figure 2.*

Data outputs can be clocked into the system using either the rising or falling edge of CKR, or the rising or falling edge of $\overline{RDY}$. If CKR is used, $\overline{RDY}$ can be used as an enable for the receiving logic. A LOW pulse on $\overline{RDY}$ shows that new data has been received and is ready to be delivered. The signal on $\overline{RDY}$ is a 60%–LOW duty cycle byte-rate pulse train suitable for the write pulse in asynchronous FIFOs such as the CY7C42X, or the enable write input on Clocked FIFOs such as the CY7C44X. HIGH on $\overline{RDY}$ shows that the received data is the null character (normally inserted by the transmitter as a pad between data inputs) and should be ignored.

When the Transmitter is disabled it will continuously send pad characters (K28.5). To assure that the receive FIFO will not be overfilled with these dummy bytes, the $\overline{RDY}$ pulse output is inhibited during fill strings. Data at the Q_{0-7} outputs will reflect the correct received data, but will not appear to change, since a string of K28.5s all are decoded as Q_{7-0} = 0000 0101 and $SC/\overline{D}$ = 1 (C05). When new data appears (not K28.5), the $\overline{RDY}$ output will resume normal function.

Fill characters are defined as any K28.5 followed by another K28.5. All fill characters will not cause $\overline{RDY}$ to pulse. Any K28.5 followed by any other character (including violation or illegal characters) will be interpreted as usable data and will cause $\overline{RDY}$ to pulse.

As noted above, $\overline{RDY}$ can also be used as an indication of correct framing of received data. While the Receiver is awaiting receipt of a K28.5 with RF HIGH, the $\overline{RDY}$ outputs will be inhibited. When $\overline{RDY}$ resumes, the received data will be properly framed and will be decoded correctly.

Code rule violations and reception errors will be indicated as follows:

	RVS	$SC/\overline{D}$	Qouts
1. Good Data code received with good RD	0	0	00–FF
2. Good Special Character code received with good RD	0	1	00–0B
3. Unassigned code received	1	1	E0
4. –K28.5+ received when RD was +	1	1	E1
5. +K28.5– received when RD was –	1	1	E2
6. Good code received with wrong RD	1	1	E4

Receiver Test Mode Description

The CY7B93x Receiver offers two types of test mode operation, BIST mode and Test mode. In a normal system application, the Built-In Self-Test (BIST) mode can be used to check the functionality of the Transmitter, the Receiver and the link connecting them. This mode is available with minimal impact on user system logic, and can be used as part of the normal system diagnostics. Typical connections and timing are shown in *Figure 3.*

BIST Mode

BIST Mode function is as follows:

1. Set $\overline{BISTEN}$ LOW to enable self-test generation and await $\overline{RDY}$ LOW indicating that the initialization code has been received.

2. Monitor RVS and check for any byte time with the pin HIGH to detect pattern mismatches. $\overline{RDY}$ will pulse HIGH once per BIST loop, and can be used to check test pattern progress. Q_{0-7} and SC/$\overline{D}$ will show the expected pattern and may be useful for debug purposes.
3. When testing is completed, set $\overline{BISTEN}$ HIGH and resume normal function.

Note: A specific test of the RVS output may be required to assure an adequate test. To perform this test, it is only necessary to have the Transmitter send violation (SVS = HIGH) for a few bytes before beginning the BIST test sequence. Alternatively, the Receiver could enter BIST mode after the Transmitter has begun sending BIST loop data, or be removed before the Transmitter finishes sending BIST loops, each of which contain several deliberate violations and should cause RVS to pulse HIGH.

BIST mode is intended to check the entire function of the Transmitter, serial link, and Receiver. It augments normal factory ATE testing and provides the user system with a rigorous test mechanism to check the link transmission system, without requiring any significant system overhead.

When in Bypass mode, the BIST logic will function in the same way as in the Encoded mode. MODE = HIGH and $\overline{BISTEN}$ = LOW causes the Receiver to switch to Encoded mode and begin checking the decoded received data of the BIST pattern, as if MODE = LOW. When $\overline{BISTEN}$ returns to HIGH, the Receiver resumes normal Bypass operation. In Test mode the BIST function works as in the normal mode.

Test Mode

The MODE input pin selects between three receiver functional modes. When wired HIGH, the Shifter contents bypass the Decoder and go directly from the Decoder latch to the Q_{0-7}, RVS, and SC/$\overline{D}$ inputs of the Output latch. When wired LOW, the outputs are decoded using the 8B/10B codes shown at the end of this datasheet. The third function is Test mode, used for factory or incoming device test. This mode can be selected by leaving the MODE pin open (internal circuitry forces an open pin to $V_{CC}/2$).

Test mode causes the Receiver to function in its Encoded mode, but with INB± as the bit rate Test clock instead of the PLL VCO. In this mode, transfers between the Shifter, Decoder register and Output register are controlled by their normal logic, but with an external bit rate clock instead of the PLL (the recovered bit clock). Internal logic and test pattern inputs can be synchronized by sending a SYNC pattern and allowing the Framer to align the logic to the bit stream. The flow is as follows:

1. Assert Test mode for several test clock cycles to establish normal counter sequence.
2. Assert RF to enable reframing.
3. Input a repeating sequence of bits representing K28.5 (Sync).
4. $\overline{RDY}$ falling shows the byte boundary established by the K28.5 input pattern.
5. Proceed with pattern, voltage and timing tests as is convenient for the test program and tester to be used.

Internal PLL dividers can be checked in Test mode by asserting RF = HIGH. In this mode, the outputs on Q_0, Q_1, and Q_2 will reflect the state of the internal counters. These counters cannot be initialized, but their output duty cycle is defined (Q_0 = 1024:1, Q_1 = 102:1, Q_2 = 103:1 as set by the PLL divider constants) and easily tested.

Test mode is intended to allow logical, DC, and AC testing of the Receiver without requiring that the tester generate input data at the 300-MHz bit rate or accommodate the PLL lock, tracking and frequency range characteristics that are required when the part operates in its normal mode.

X3T9.3 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, bit by bit. Information received over a serial link is collected ten bits at a time, and those Transmission Characters that are used for data (Data Characters) are decoded into the correct eight-bit codes. The 10-bit Transmission Code supports all 256 8-bit combinations. Some of the remaining Transmission Characters (Special Characters) are used for functions other than data transmission.

The primary rationale for 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 make clock recovery possible at the Receiver. Such encoding also greatly increases the likelihood of detecting any single or multiple bit errors that may occur during transmission and reception of information. In addition, some Special Characters of the Transmission Code selected by Fibre Channel Standard consist of a distinct and easily recognizable bit pattern (the Special Character Comma) that assists a Receiver in achieving word alignment on the incoming bit stream.

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 that the order of bits is reversed):

```
Data Byte Name  D5.2
          Bits: ABCDE FGH
                10100 010
```

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

```
     Bits: abcdei  fghj
           101001  0101
```

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 the SC/$\overline{D}$ pin is LOW) or a Special Character (c is set to K, and the SC/$\overline{D}$ pin is 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 val-

ue 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 4,488,739. 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 Level (FC_PH/91–001R2.13, X3T9.3/90–071). Working draft proposed for American National Standard for Information Systems, Rev 2.13 December 4, 1991.

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 that 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 CE1 and CE2 can be used to force the transmission of a specific Special Character with a specific running disparity as required for some special sequences in X3T9.3.

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 (fghi) 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.

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 TransmissionCharacter 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 TransmissionCharacter'svalidity, 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 TransmissionCharacter.

Detection of a code violation does not necessarily show that the TransmissionCharacter 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. The following table shows an example of this behavior:

	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	+

Valid Transmission Characters

Naming notation and examples:

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	000101	45
. .	. .	. .	. .
D30.7	111	11110	FE
D31.7	111	11111	FF

Valid Data Characters (SC/$\overline{D}$ = LOW)

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

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

Valid Data Characters (SC/$\overline{D}$ = LOW) (continued)

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

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

Valid Data Characters (SC/$\overline{D}$ = LOW) (continued)

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

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

Valid Data Characters (SC/$\overline{D}$ = LOW) (continued)

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

Data Byte Name	Bits		Current RD−		Current RD+	
	HGF	EDCBA	abcdei	fghj	abcdei	fghj
D0.7	111	00000	100111	0001	011000	1110
D1.7	111	00001	011101	0001	100010	1110
D2.7	111	00010	101101	0001	010010	1110
D3.7	111	00011	110001	1110	110001	0001
D4.7	111	00100	110101	0001	001010	1110
D5.7	111	00101	101001	1110	101001	0001
D6.7	111	00110	011001	1110	011001	0001
D7.7	111	00111	111000	1110	000111	0001
D8.7	111	01000	111001	0001	000110	1110
D9.7	111	01001	100101	1110	100101	0001
D10.7	111	01010	010101	1110	010101	0001
D11.7	111	01011	110100	1110	110100	1000
D12.7	111	01100	001101	1110	001101	0001
D13.7	111	01101	101100	1110	101100	1000
D14.7	111	01110	011100	1110	011100	1000
D15.7	111	01111	010111	0001	101000	1110
D16.7	111	10000	011011	0001	100100	1110
D17.7	111	10001	100011	0111	100011	0001
D18.7	111	10010	010011	0111	010011	0001
D19.7	111	10011	110010	1110	110010	0001
D20.7	111	10100	001011	0111	001011	0001
D21.7	111	10101	101010	1110	101010	0001
D22.7	111	10110	011010	1110	011010	0001
D23.7	111	10111	111010	0001	000101	1110
D24.7	111	11000	110011	0001	001100	1110
D25.7	111	11001	100110	1110	100110	0001
D26.7	111	11010	010110	1110	010110	0001
D27.7	111	11011	110110	0001	001001	1110
D28.7	111	11100	001110	1110	001110	0001
D29.7	111	11101	101110	0001	010001	1110
D30.7	111	11110	011110	0001	100001	1110
D31.7	111	11111	101011	0001	010100	1110

Valid Special Character Codes and Sequences (SC/$\overline{D}$ = HIGH)[16]

S.C. Byte Name	S.C. Code Name		Bits HGF	Bits EDCBA	Current RD− abcdei	Current RD− fghj	Current RD+ abcdei	Current RD+ fghi
K28.0	C0.0	(C00)	000	00000	001111	0100	110000	1011
K28.1	C1.0	(C01)	000	00001	001111	1001	110000	0110
K28.2	C2.0	(C02)	000	00010	001111	0101	110000	1010
K28.3	C3.0	(C03)	000	00011	001111	0011	110000	1100
K28.4	C4.0	(C04)	000	00100	001111	0010	110000	1101
K28.5	C5.0	(C05)	000	00101	001111	1010	110000	0101
K28.6	C6.0	(C06)	000	00110	001111	0110	110000	1001
K28.7	C7.0	(C07)	000	00111	001111	1000	110000	0111
K23.7	C8.0	(C08)	000	01000	111010	1000	000101	0111
K27.7	C9.0	(C09)	000	01001	110110	1000	001001	0111
K29.7	C10.0	(C0A)	000	01010	101110	1000	010001	0111
K30.7	C11.0	(C0B)	000	01011	011110	1000	100001	0111
Reserved	C12.0	(C0C)	000	01100				
:	:	:	:	:				
Reserved	C31.0	(C1F)	000	11111				
Idle	C0.1	(C20)	001	00000	-K28.5+,D21.4,D21.5,D21.5,repeat[17]			
R_RDY	C1.1	(C21)	001	00001	K28.5+,D21.4,D10.2,D10.2,repeat[18]			
EOFxx	C2.1	(C22)	001	00010	-K28.5,Dn.xxx[19]		+K28.5,Dn.xxx[19]	
Reserved	C3.1	(C23)	001	00011				
:	:	:	:	:				
Reserved	C31.6	(CDF)	110	11111				
Exception	C0.7	(CE0)	111	00000	<Code Rule Violation>[20]			
K28.0	C1.7	(CE1)	111	00001	001111	1010[21]	001111	1010[21]
+K28.5	C2.7	(CE2)	111	00010	110000	0101[22]	110000	0101[22]
Reserved	C3.7	(CE3)	111	00011				
Exception	C4.7	(CE4)	111	00100	<Running Disparity Violation>[23]			
Reserved	C5.7	(CE5)	111	00101				
:	:	:	:	:				
Reserved	C31.7	(CFF)	111	11111				

Notes:

16. Notation for Special Character Byte 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).
17. C20 = Transmit Negative K28.5 (−K28.5+) disregarding Current RD when input is held for only one byte time. If held longer, transmitter begins sending the repeating transmit sequence −K28.5+, D21.4, D21.5, D21.5, (repeat all four bytes)... defined in X3T9.3 as the primitive signal "Idle word." This Special Character input must be held for four (4) byte times or multiples of four bytes or it will be truncated by the new data.
18. C21 = Transmit Negative K28.5 (−K28.5+) disregarding Current RD when input is held for only one byte time. If held longer, transmitter begins sending the repeating transmit sequence −K28.5+, D21.4, D10.2, D10.2, (repeat all four bytes)... defined in X3T9.3 as the primitive signal "Receiver_Ready (R_RDY)." This Special Character input must be held for four (4) byte times or multiples of four bytes or it will be truncated by the new data.

    Receiver will never output this Special Character, since K28.5 is decoded as C05, and the subsequent bytes are decoded as data.

Notes (continued):

19. C22 = 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 X3T9.3 "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 C22−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 C22−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.

    Receiver will never output this Special Character, since K28.5 is decoded as C05, and the subsequent bytes are decoded as data.

20. CE0 = 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 SVS = HIGH.

    Receiver will only output this Special Character if the Transmission Character being decoded is not found in the tables.

21. CE1 = Transmit Negative K28.5 (−K28.5+) disregarding Current RD.

    Receiver will only output this Special Character if K28.5 is received with the wrong running disparity. Receiver will output CE1 if −K28.5 is received with RD+, otherwise K28.5 is decoded as C05.

22. CE2 = Transmit Positive K28.5 (+K28.5−) disregarding Current RD.

    Receiver will only output this Special Character if K28.5 is received with the wrong running disparity. Receiver will output CE2 if +K28.5 is received with RD−, otherwise K28.5 is decoded as C05.

23. CE4 = Transmit the same deliberate code rule violation as is sent by asserting CE0.

    Receiver will only output this Special Character if the Transmission Character being decoded is found in the tables, but Running Disparity does not match.

Ordering Information

Ordering Code	Package Type	Operating Range
CY7B921−DC	D22	Commercial
CY7B921−JC	J64	
CY7B921−LC	L64	
CY7B921−PC	P21	
CY7B921−JI	J64	Industrial
CY7B921−PI	P21	
CY7B921−DMB	D22	Military
CY7B921−LMB	L64	

Ordering Code	Package Type	Operating Range
CY7B931−DC	D22	Commercial
CY7B931−JC	J64	
CY7B931−LC	L64	
CY7B931−PC	P21	
CY7B931−JI	J64	Industrial
CY7B931−PI	P21	
CY7B931−DMB	D22	Military
CY7B931−LMB	L64	

Ordering Code	Package Type	Operating Range
CY7B922−DC	D22	Commercial
CY7B922−JC	J64	
CY7B922−LC	L64	
CY7B922−PC	P21	
CY7B922−JI	J64	Industrial
CY7B922−PI	P21	
CY7B922−DMB	D22	Military
CY7B922−LMB	L64	

Ordering Code	Package Type	Operating Range
CY7B932−DC	D22	Commercial
CY7B932−JC	J64	
CY7B932−LC	L64	
CY7B932−PC	P21	
CY7B932−JI	J64	Industrial
CY7B932−PI	P21	
CY7B932−DMB	D22	Military
CY7B932−LMB	L64	

Ordering Code	Package Type	Operating Range
CY7B923−DC	D22	Commercial
CY7B923−JC	J64	
CY7B923−LC	L64	
CY7B923−PC	P21	
CY7B923−JI	J64	Industrial
CY7B923−PI	P21	
CY7B923−DMB	D22	Military
CY7B923−LMB	L64	

Ordering Code	Package Type	Operating Range
CY7B933−DC	D22	Commercial
CY7B933−JC	J64	
CY7B933−LC	L64	
CY7B933−PC	P21	
CY7B933−JI	J64	Industrial
CY7B933−PI	P21	
CY7B933−DMB	D22	Military
CY7B933−LMB	L64	

MILITARY SPECIFICATIONS
Group A Subgroup Testing

DC Characteristics

Parameters	Subgroups
V_{OHT}	1, 2, 3
V_{OLT}	1, 2, 3
V_{OHE}	1, 2, 3
V_{OLE}	1, 2, 3
I_{OST}	1, 2, 3
V_{IHT}	1, 2, 3
V_{ILT}	1, 2, 3
V_{IHE}	1, 2, 3
V_{ILE}	1, 2, 3
I_{IHT}	1, 2, 3
I_{ILT}	1, 2, 3
I_{IHE}	1, 2, 3
I_{ILE}	1, 2, 3
I_{CC}	1, 2, 3
V_{DIFF}	1, 2, 3
V_{IHH}	1, 2, 3
V_{ILL}	1, 2, 3

Switching Characteristics

Parameters	Subgroups
t_{CKR}	9, 10, 11
t_{CKW}	9, 10, 11
t_{CKX}	9, 10, 11
t_{B}	9, 10, 11
t_{CPWH}	9, 10, 11
t_{CPWL}	9, 10, 11
t_{CPRH}	9, 10, 11
t_{CPRL}	9, 10, 11
t_{CPXH}	9, 10, 11
t_{CPXL}	9, 10, 11
t_{RH}	9, 10, 11
t_{DS}	9, 10, 11
t_{PRH}	9, 10, 11
t_{PRL}	9, 10, 11
t_{A}	9, 10, 11
t_{SD}	9, 10, 11
t_{HD}	9, 10, 11
t_{ROH}	9, 10, 11
t_{SEND}	9, 10, 11
t_{SENP}	9, 10, 11
t_{HEN}	9, 10, 11
t_{PDR}	9, 10, 11
t_{PPWH}	9, 10, 11
t_{PPWL}	9, 10, 11

Document #: 38–00189