CY7C954DX

ATM HOTLink® Transceiver

Features

- **Second-generation HOTLink® technology**
- **UTOPIA level I and II compatible host bus interface**
- **Three-bit Multi-PHY address capability built-in**
- **Three user-selectable Start Of Cell marker/indicators**
- **Embedded 256-character synchronous FIFOs**
- **Built-in ATM Header Error Control (HEC)**
- **Automatic Transmit-HEC insertion & Receiver-HEC check**
- **FIFO cell-level flushing of invalid ATM cells**
- **ATM Forum, Fibre Channel, and ESCON® compliant 8B/10B encoder/decoder**
- **50- to 200-MBaud serial signaling rate**
- **Internal PLLs with no external PLL components**
- **Dual differential PECL-compatible serial inputs**
- **Dual differential PECL-compatible serial outputs**
- **Compatible with fiber-optic modules and copper cables**
- **Built-In Self-Test (BIST) for link testing**
- **Link Quality Indicator**
- **Single +5.0V ±10% supply**
- **100-pin TQFP**
- **0.35µ CMOS technology**

Functional Description

The 200-MBaud CY7C954DX HOTLink Transceiver is a point-to-point communications building block allowing the transfer of data over high-speed serial links (optical fiber, balanced, and unbalanced copper transmission lines) at speeds ranging between 50 and 200 MBaud. The transmit section accepts parallel data of selectable width and converts it to serial data, while the receiver section accepts serial data and converts it to parallel data of selectable width. *Figure 1* illustrates typical connections between two independent host systems and corresponding CY7C954DX parts. As a second-generation HOTLink device, the CY7C954DX provides enhanced levels of technology, functionality, and integration over the field proven CY7B923/933 HOTLink.

The transmit section of the CY7C954DX HOTLink has been configured to accept 8-bit data characters on each clock cycle, and store the parallel data into an internal Transmit FIFO. Data is read from the Transmit FIFO and is encoded using an embedded 8B/10B encoder to improve its serial transmission characteristics. These encoded characters are then serialized and output from two Pseudo ECL (ECL referenced to +5.0V) compatible differential transmission line drivers at a bit-rate of 10 times the input reference clock.

The receive section of the CY7C954DX HOTLink 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. The recovered bit stream is deserialized and framed into characters, 8B/10B decoded, and checked for transmission errors. Recovered decoded characters are reconstructed into 8-bit data characters, written to an internal Receive FIFO, and presented to the destination host system.

For those systems requiring even greater FIFO storage capability, external FIFOs may be directly coupled to the CY7C954DX device through the parallel interface without additional glue-logic for single PHY connections.

The TTL parallel I/O interface may be configured as either a FIFO (configurable for UTOPIA emulation or for depth expansion through external FIFOs) or as a pipeline register extender. The FIFO configurations are optimized for transport of time-independent (asynchronous) 8-bit character-oriented data across a link. A Built-In Self-Test (BIST) pattern generator and checker allows for at-speed testing of the high-speed serial data paths in both the transmit and receive sections, and across the interconnecting links.

HOTLink 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 workstations, backplanes, servers, mass storage, and video transmission equipment.

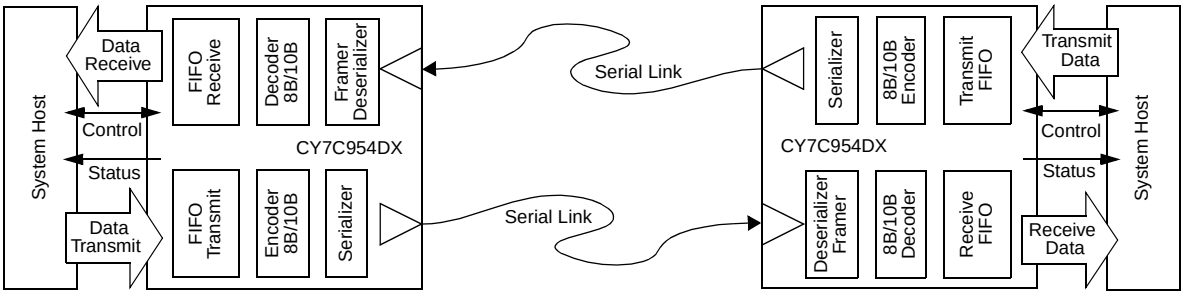
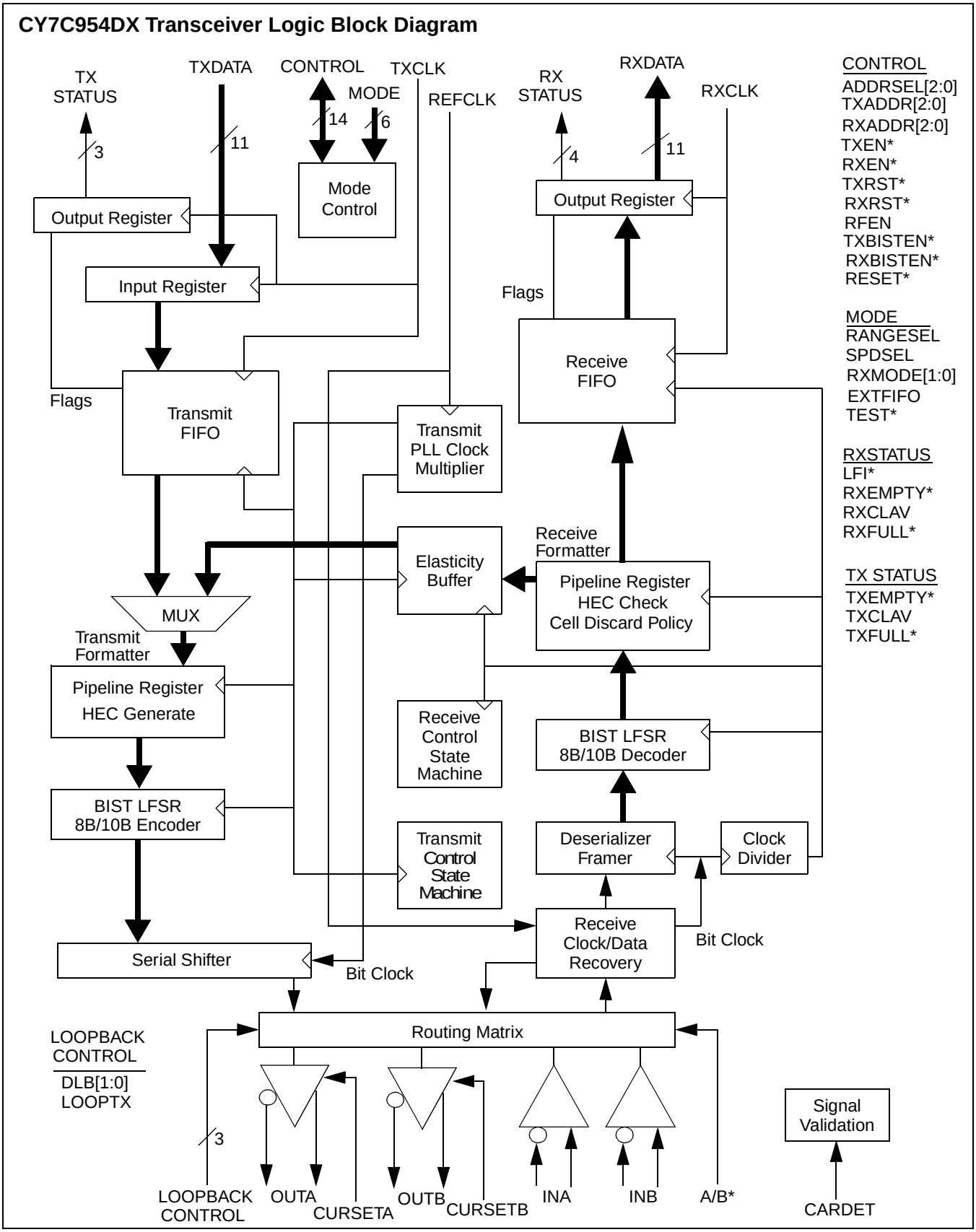


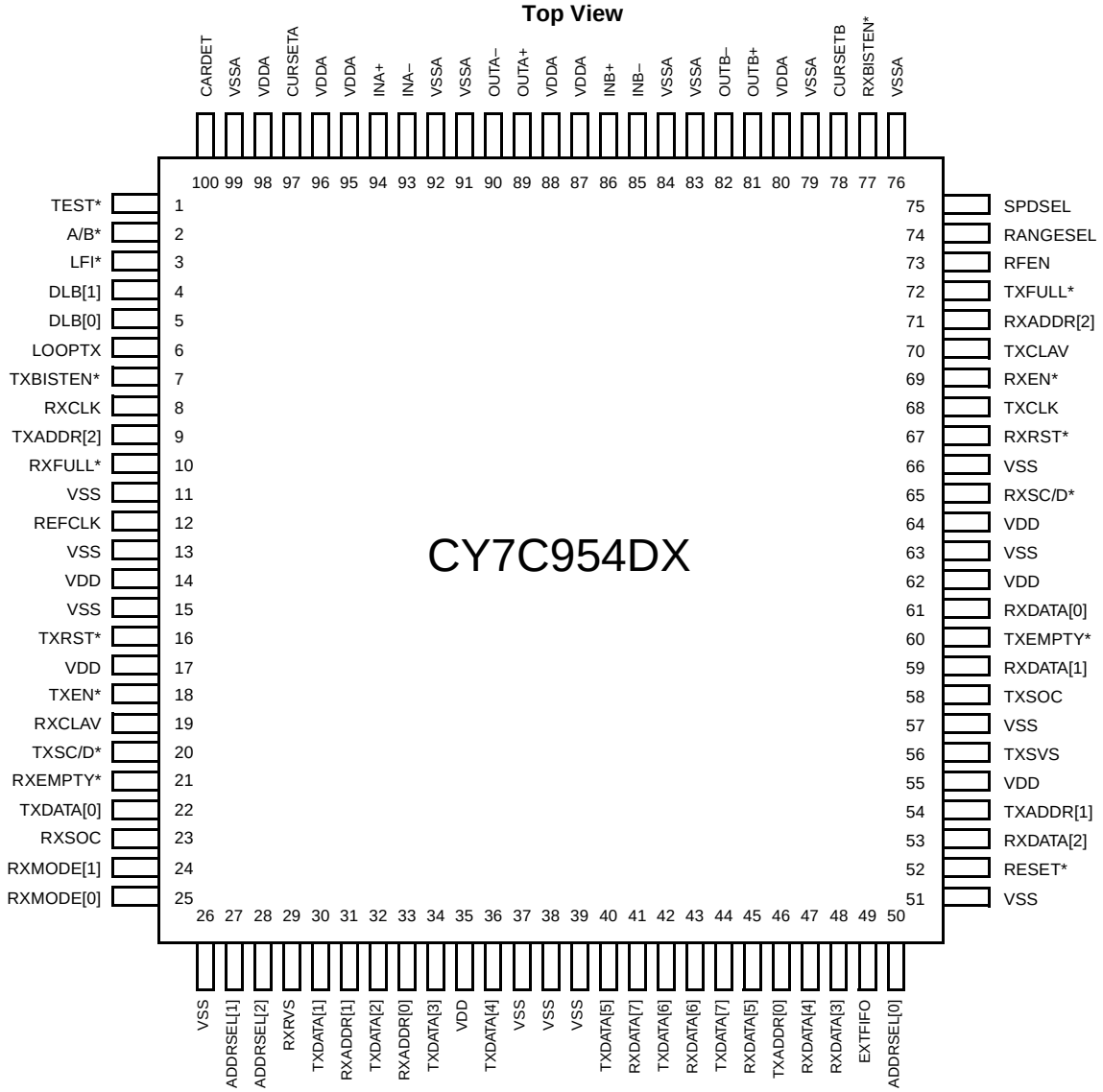
Figure 1. HOTLink System Connections

CY7C954DX Transceiver Logic Block Diagram



Pin Configuration

TQFP
Top View



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 +6.5V

DC Voltage Applied to Outputs in High-Z State ..... –0.5V to V_{DD}+0.5V

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

DC Input Voltage ..... –0.5V to V_{DD}+0.5V

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

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

Operating Range

Range	Ambient Temperature	V_{CC}
Commercial	0°C to +70°C	5.0V ± 10%

Pin Descriptions

Pin #	Name	I/O Characteristics	Signal Description
Transmit Path Signals			
44, 42, 40, 36, 34, 32, 30, 22	TXDATA[7:0]	TTL input, sampled on TXCLK↑, Internal Pull-Up	Parallel Transmit Data Input. These inputs contain data that is written to the Transmitter FIFO when TXADDR[2:0] matches ADDRSEL[2:0] and transmitter input is selected by TXEN*.
56	TXSVS	TTL input, sampled on TXCLK↑, Internal Pull-Up	Transmit Send Violation Symbol Input. This input is interpreted along with TXSOC and TXSC/D* (see *Table 1* for details).
58	TXSOC	TTL input, sampled on TXCLK↑, Internal Pull-Up	Transmit Start of Cell Input. This input is used as a message frame delimiter to indicate the beginning of a data packet. It is interpreted along with TXSVS and TXSC/D* (see *Table 1* for details).
20	TXSC/D*	TTL input, sampled on TXCLK↑, Internal Pull-Up	Transmit Special Character or Data Select Input. This input is interpreted along with TXSVS and TXSOC (see *Table 1* for details).
18	TXEN*	TTL input, sampled on TXCLK↑, Internal Pull-Up	Transmit Enable Input. Data enable for the TXDATA bus write operations. Active LOW when configured for UTOPIA timing, active HIGH when configured for Cascade timing.
9, 54, 46	TXADDR[2:0]	TTL input, sampled on TXCLK↑, Internal Pull-Up on	Transmit Address Select Input. This is the three bit Transmit Port address that is matched to ADDRSEL[2:0] to enable data transfer from the transmitting system.
68	TXCLK	TTL clock input, Internal Pull-Up	Transmit FIFO Clock. The input clock for the transmit parallel interface. Used to sample all Transmit FIFO related interface signals.
72	TXFULL*	3-state TTL output, changes following TXCLK↑	Transmit FIFO Full Status Flag. Active LOW when configured for UTOPIA timing, active HIGH when configured for Cascade timing. When TXFULL* is first asserted, the Transmit FIFO can still accept a minimum of four write cycles without loss of data. FIFO flags are updated one TXCLK cycle after an address match condition exists.
70	TXCLAV	3-state TTL output, changes following TXCLK↑	Transmit FIFO Cell Available Status Flag. Active LOW. TXCLAV is asserted LOW when the Transmit FIFO has sufficient space to insert one or more 53-byte ATM cells. TXCLAV is forced to the High-Z state only during a “full-chip” reset (i.e., while RESET* is LOW) or on the cycle after an “unmatch” in TXADDR[2:0]. (Used for polling FIFO status.) FIFO flags are updated one TXCLK cycle after an address match condition exists.
60	TXEMPTY*	3-state TTL output, changes following TXCLK↑	Transmit FIFO Empty Status Flag. Active LOW when configured for UTOPIA timing, active HIGH when configured for Cascade timing. TXEMPTY* is asserted either when no data has been loaded into the Transmit FIFO, or when the Transmit FIFO has been emptied by either a Transmit FIFO reset or by the normal transmission of the FIFO contents. When TXBISTEN* is asserted LOW, TXEMPTY* becomes the transmit BIST-loop counter indicator. In this mode TXEMPTY* is asserted for one TXCLK period at the end of each transmitted BIST sequence. FIFO flags are updated one TXCLK cycle after an address match condition exists.

Pin Descriptions (continued)

Pin #	Name	I/O Characteristics	Signal Description
16	TXRST*	TTL input, internal pull-up, sampled on TXCLK↑, Internal Pull-Up	Transmit FIFO Reset. When TXRST* is sampled asserted (LOW) for eight or more TXCLK cycles, a reset operation is started on the Transmit FIFO.
7	TXBISTEN*	TTL input, asynchronous, Internal Pull-Up	Transmitter BIST Enable. When TXBISTEN* is LOW, the transmitter generates a 511-character repeating sequence, that can be used to validate link integrity. The transmitter returns to normal operation when TXBISTEN* is HIGH. All Transmit FIFO read operations are suspended when BIST is active.
Receive Path Signals			
41, 43, 45, 47, 48, 53, 59, 61	RXDATA[7:0]	3-state TTL output, changes following RXCLK↑	Parallel Data Output. These outputs change following the rising edge of RXCLK, when enabled to output data (the device RXADDR[2:0] address matches ADDRSEL[2:0] and selected by RXEN*).
29	RXRVS	3-state TTL output, changes following RXCLK↑, Internal Pull-Up	Received Violation Symbol Indicator. In Receive mode (11), this output is the indicator that data has been received continuing errors, and is decoded in conjunction with RXSC/D* and RXSOC, per *Table 4*, to indicate the presence of specific Special Character codes in the received data stream. This output is unused for the other receive modes, except that RXRVS is used to report character mismatches when RXBISTEN* is LOW This output changes following the rising edge of RXCLK, when enabled to output data (the device RXADDR[2:0] address matches ADDRSEL[2:0] and selected by RXEN*).
23	RXSOC	3-state TTL output, changes following RXCLK↑	Receive Start Of Cell. This output is one of the indicators for the start of a cell and is decoded in conjunction with RXSC/D* and RXRVS, per *Table 4*, to indicate the presence of specific Special Character codes in the received data stream. This output changes following the rising edge of RXCLK, when enabled to output data (the device RXADDR[2:0] address matches ADDRSEL[2:0] and selected by RXEN*).
65	RXSC/D*	3-state TTL output, changes following RXCLK↑	Received Special Character or Data Indicator. This signal is use to differentiate between Special Characters and Data bytes. It is also decoded in conjunction with RXSOC and RXRVS, per *Table 4*, to indicate the presence of specific Special Character codes in the received data stream. This output changes following the rising edge of RXCLK, when enabled to output data (the device RXADDR[2:0] address matches ADDRSEL[2:0] and selected by RXEN*).
69	RXEN*	TTL input, sampled on RXCLK↑, Internal Pull-Up	Receive Enable. Data enable for the RXDATA bus write and read operations. Active LOW when configured for UTOPIA timing, active HIGH when configured for Cascade timing as determined by the EXTFIFO pin.
71,31, 33	RXADDR[2:0]	TTL input, sampled on RXCLK↑	Receive Address Input. This is the three-bit Receive Port address that is matched to ADDRSEL[2:0] to enable data transfer to the receiving system.
8	RXCLK	TTL output clock, Internal Pull-Up	Receive Clock. This clock is the Receive interface input clock and is used to control Receive FIFO read, reset, and serial register access operations.

Pin Descriptions (continued)

Pin #	Name	I/O Characteristics	Signal Description
10	RXFULL*	3-state TTL output, changes following RXCLK↑	Receive FIFO Full Flag. Active LOW when configured for UTOPIA timing (EXTFIFO is LOW), active HIGH when configured for Cascade timing (EXTFIFO is HIGH). In Receive mode (11), when the Receive FIFO is addressed, RXFULL* is asserted one RXCLK cycle after the address match, when the Receive FIFO has room for four or fewer writes. If the RXCLK input is not continuous, or if the FIFO is accessed at a rate slower than data is being received, RXFULL* may indicate loss of data. This output is not used in Receive modes (00, 01, 10).
19	RXCLAV	3-state TTL output, changes following RXCLK↑	Receive FIFO Cell Available Flag. This signal is asserted (LOW) when the Receive FIFO contains at least one ATM cell ready to be read. It is asserted one RXCLK cycle after the device RXADDR[2:0] address matches ADDRSEL[2:0] and selected by RXEN*. RXCLAV is forced to the High-Z state only during a “full-chip” reset (i.e., while RESET* is LOW) or on the cycle after an “unmatch” in RXADDR[2:0]. (Used for polling FIFO status.)
21	RXEMPTY*	3-state TTL output, changes following RXCLK↑	Receive FIFO Empty Flag. Active LOW when configured for UTOPIA timing (EXTFIFO is LOW), active HIGH when configured for Cascade timing (EXTFIFO is HIGH). In Receive mode (11), when the Receive FIFO is enabled, RXEMPTY* is asserted one RXCLK cycle after the address match, when no data remains in the Receive FIFO. Any read operation occurring when RXEMPTY* is asserted results in no change in the FIFO status, and the data from the last valid read remains on the RXDATA bus. This output functions the same RXCLAV in Receive modes (00, 01, 10).
67	RXRST*	TTL input, sampled on RXCLK↑, Internal Pull-Up	Receive FIFO Reset. When the Receive FIFO is addressed and RXRST* is sampled while asserted (LOW) for eight or more RXCLK cycles, along with the deassertion of RXEN* and an address match condition existing, a Receive FIFO reset is initiated.
73	RFEN	TTL input, asynchronous, Internal Pull-Up	Reframe Enable. Used to control when the framer is allowed to adjust the character boundaries based on detection of one or more K28.5 characters in the data stream. When HIGH, the framer is allowed to adjust the character boundaries relative to received serial data stream. When LOW, the boundary is fixed.
77	RXBISTEN*	TTL input, asynchronous, Internal Pull-Up	Receiver BIST Enable. When active, the receiver is configured to perform a character-for-character match of the incoming data stream with a 511-character BIST sequence. The result of character mismatches are indicated on RXRVS. Completion of each 511-character BIST loop is accompanied by an assertion pulse on the RXFULL* flag.
Control Signals			
6	LOOPTX	TTL asynchronous input, Internal Pull-Down	Serial-in to Serial-out LOOP Select. This input controls the LOOP-through function in which the serial data is recovered by the Clock/Data Recovery PLL and then is retransmitted using the Transmitter PLL as the bit-rate reference. It selects between the output of the Transmitter FIFO and the output of the Elasticity buffer as the input to the Transmit Encoder. When LOW, the transmit FIFO is the source of data for transmission. When HIGH, the Elasticity Buffer is the source of data for transmission.

Pin Descriptions (continued)

Pin #	Name	I/O Characteristics	Signal Description
12	REFCLK	TTL input clock	Reference Clock. This clock input is used as the timing reference for the transmit and receive PLLs. See *Table 3* for the relationships between REFCLK, SPDSEL, and RANGESEL.
75	SPDSEL	Static control input TTL levels Normally wired HIGH or LOW	Speed Select. Used to select one of two operating data rates for the device. When the operating bit rate is between 100 and 200 MBaud, SPDSEL must be HIGH. When the operating bit rate is between 50 and 100 MBaud, SPDSEL must be LOW (see *Table 3*).
74	RANGESEL	Static control input TTL levels Normally wired HIGH or LOW	Range Select. Selects the proper prescaler for the REFCLK input. See *Table 3* for the various relationships between REFCLK, SPDSEL, and RANGESEL.
49	EXTFIFO	Static control input TTL levels Normally wired HIGH or LOW	External FIFO Interface Levels and Timing Select. EXTFIFO modifies the active level of the RXEN* and TXEN* inputs and the timing of the Transmitter and Receiver data buses. While this compromises the UTOPIA multi-PHY compatibility, it adds additional FIFO capability for single PHY systems. In UTOPIA interface mode (EXTFIFO is LOW), TXEN* is assumed to be driven as a pipeline register and RXEN* is assumed to be driven by a controller for a pipeline register. In this mode the active Transmit data information is within the same clock as the clock edge that “enables” the data bus. In Cascade interface mode (EXTFIFO is HIGH), TXEN is assumed to be driven by the empty flag of an attached CY7C42X5 FIFO, and RXEN is assumed to be driven by the almost full flag of an attached CY7C42X5 FIFO. In this mode the active Transmit data information is in the clock following the clock edge that “enables” the data bus. Receive data information is always in the clock cycle following the RXEN*. EXTFIFO also modifies the output state of the Receive and Transmit FIFO flags. When configured in UTOPIA mode (EXTFIFO is LOW), all of the FIFO flags are active LOW. When configured in Cascade mode (EXTFIFO is HIGH), the Full and Empty FIFO flags are active HIGH (the TXCLAV and RXCLAV flags are always active LOW).
28,27, 50	ADDRSEL[2:0]	Static control input TTL levels Normally wired HIGH or LOW	UTOPIA Chip Address Input. These inputs select one of 7 chip addresses to which the TXADDR[2:0] and RXADDR[2:0] are compared. A match of these vectors and the appropriate TXEN* or RXEN* results in an Address match and selection operation.
24,25	RXMODE[1:0]	Static control input TTL levels Normally wired HIGH or LOW	Receive Discard Policy Select. These inputs select between the four received-cell discard modes in the receiver. See *Table 5*.
52	RESET*	TTL asynchronous input	Global Logic Reset. This input is pulsed LOW for one or more REFCLK periods to reset the internal logic.
1	TEST*	TTL asynchronous input Normally wired HIGH	Factory Test Mode Select. Used to force the part into a diagnostic test mode used for factory ATE test. This pin is tied HIGH during normal operation.
Analog I/O and Control			
89, 90, 81, 82	OUTA± OUTB±	PECL differential output	Differential Serial Data Outputs. These PECL outputs are capable of driving terminated transmission lines or commercial fiber-optic transmitter modules. An unused output pair may be powered down by leaving the outputs unconnected and strapping the associated CURSETx pin to V_{DD}.

Pin Descriptions (continued)

Pin #	Name	I/O Characteristics	Signal Description
97	CURSETA	Analog	Current-set Resistor Input for OUTA±. A precision resistor is connected between this input and a clean ground to set the output differential amplitude and currents for the OUTA± differential driver.
78	CURSETB	Analog input	Current-set Resistor Input for OUTB±. A precision resistor is connected between this input and a clean ground to set the output differential amplitude and currents for the OUTB± differential driver.
94, 93, 86, 85	INA± INB±	PECL compatible differential input	Differential Serial Data Inputs. These inputs accept the serial data stream for deserialization and decoding. Only one serial stream at a time may be fed to the receiver PLL to extract the data content. This stream is selected using the A/B* input. These inputs may also be routed to the OUTB± serial outputs using the DLB[1:0] inputs.
2	A/B*	TTL input, asynchronous, Internal Pull-Up	Receive Data Input Selector. Determines which internal or external serial bit-stream is passed to the receiver clock and data recovery circuit. See *Table 2* for details.
4, 5	DLB[1:0]	TTL input, asynchronous, Internal Pull-Down	Loop Back Select Inputs. Selects connections between serial inputs and outputs. Controls diagnostic loop-back and serial loop-through functions. See *Table 2* for details.
100	CARDET	PECL input, asynchronous	Carrier Detect Input. Used to allow an external device to signify a valid signal is being presented to the high-speed PECL input buffers, as is typical on an Optical Module. When CARDET is deasserted LOW, the LFI***** indicator asserts LOW signifying a Link Fault. This input can be tied HIGH for copper media applications.
3	LFI*	TTL output, changes following RXCLK↑	Link Fault Indication Output. Active LOW. LFI* changes synchronous with RXCLK. This output is driven LOW when the serial link currently selected by A/B* is not suitable for data recovery. This could be because 1. Serial Data Amplitude is below acceptable levels 2. Input transition density is not sufficient for PLL clock recovery 3. Input Data stream is outside an acceptable frequency range of operation 4. CARDET is LOW
Power			
80, 87, 88, 95, 96, 98	V_{DDA}		Power for PECL I/O signals and internal analog circuits.
76, 79, 83, 84, 91, 92, 99	V_{SSA}		Ground for PECL I/O signals and internal analog circuits.
4, 14, 17, 35, 55, 62, 64	V_{DD}		Power for CMOS I/O signals and internal logic circuits.
11, 13, 15, 26, 37, 38, 39, 51, 52, 57, 63, 66	V_{SS}		Ground for CMOS I/O signals and internal logic circuits.

CY7C954DX HOTLink Operation

Overview

The CY7C954DX is designed to move parallel data across both short and long distances with minimal overhead or host system intervention. This is accomplished by converting the parallel characters into a serial bit-stream, transmitting these serial bits at high speed, and converting the received serial bits back into the original parallel data format.

The CY7C954DX offers a large feature set, allowing it to be used in a wide range of host systems. Some of the configuration options are:

- ATM Cell size and header rules generate/check
- user-definable data packet or frame structure
- two-octave data-rate range
- asynchronous (FIFOed) data interface
- 8B/10B encoded serial data
- with or without HEC check/generate
- selectable cell discard policy based on 53-byte cell size
- selectable cell discard policy based on HEC errors
- embedded 256-byte (4-cell) FIFO data storage
- multi-PHY capability (three-bit address match)
- point-to-point, or point-to-multipoint data-transport

This flexibility allows the CY7C954DX to meet the data transport needs of almost any system.

Transmit Data Path

Transmit Data Interface/Transmit Data FIFO

The host interface functions in a manner similar to that of a synchronous FIFO clocked by TXCLK. The internal 256 character Transmit FIFO is always enabled. It allows the host interface to be written at any rate from DC to 50 MHz.

The interface operations support two interface timing models: UTOPIA and Cascade. The UTOPIA timing model is designed to match the active levels, bus timing, and signal sequencing called out in the ATM Forum UTOPIA specification. The Cascade timing model is designed to match a host bus that resembles a synchronous FIFO. These timing models allow the CY7C954DX to directly couple to host systems, registers, state machines, FIFOs, etc. with minimal and in many cases no external glue logic.

Encoder

Data from the Transmit FIFO is next passed to an Encoder block. The CY7C954DX contains an internal 8B/10B encoder that is used to improve the serial transport characteristics of the data.

Serializer/Line Driver

The data from the Encoder is passed to a Serializer. This Serializer operates at either 2.5, 5, or 10 times the rate of the REFCLK input. The serialized data is output from two PECL-compatible differential line drivers configured to drive transmission lines or optical modules.

Receive Data Path

Line Receiver/Deserializer/Framer

Serial data is received at one of two PECL-compatible differential line receivers. The data is passed to both a Clock and Data Recovery PLL (Phase Locked Loop) and to a Deserializer that converts serial data into parallel characters. The Framer adjusts the boundaries of these characters to match those of the original transmitted characters.

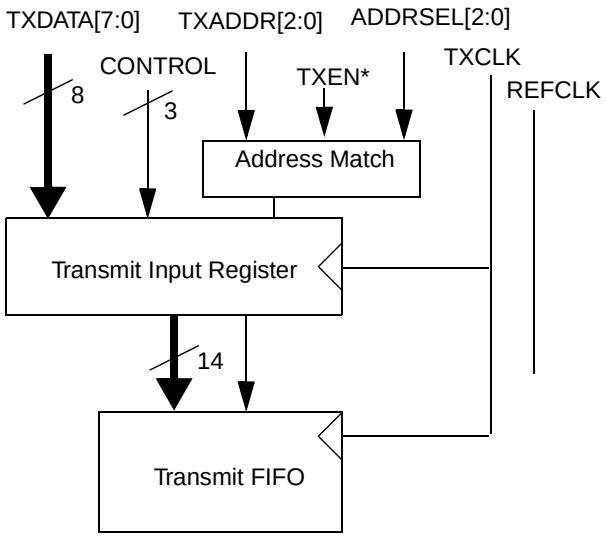


Figure 2. Transmit Input Register

Decoder

The parallel characters are passed through a 10B/8B Decoder and returned to their original form.

Receive Data Interface/Receive Data FIFO

Data from the decoder is passed directly to a 256-character Receive FIFO that allows data to be read at any rate from DC to 50 MHz. The receive interface is also configurable for both UTOPIA and Cascade timing models.

CY7C954DX HOTLink Transceiver Block Diagram Description

Transmit Input Register

The Transmit Input Register, shown in *Figure 2*, captures the data to be processed by the HOTLink Transmitter, and allows the input timing to be made compatible with asynchronous or host system buses. This bus can take the form of UTOPIA compliant interfaces, external FIFOs, state machines, or other control structures. Data present on the TXDATA[7:0] and TXSC/D* inputs are captured at the rising edge of the selected sample clock. The logical sense of the enable and FIFO flag signals depends on the intended interface convention and is set by the EXTFIFO pin.

Asynchronous interface clocking controls the writing of host bus data into the Transmit FIFO. In this configuration, all writes to the Transmit Input Register, and associated transfers to the Transmit FIFO, are controlled by TXCLK. The remainder of the transmit data path is clocked by REFCLK or synthesized derivatives of REFCLK.

UTOPIA Timing Model

The UTOPIA timing model allows multiple CY7C954DX transmitters to be addressed and accessed from a common host bus, using the protocols defined in the ATM Forum UTOPIA interface standards. It is enabled by setting EXTFIFO LOW.

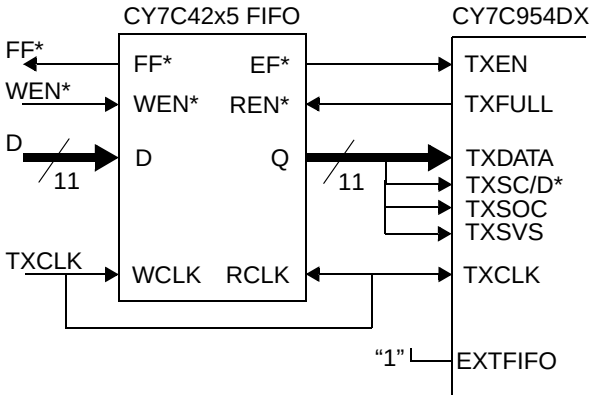


Figure 3. External FIFO Depth Expansion of the CY7C954DX Transmit Data Path

In UTOPIA timing, the TXEMPTY* and TXFULL* outputs and TXEN* input are all active LOW signals. If the CY7C954DX is addressed by TXADDR[2:0] matching ADDRSEL[2:0], it becomes "selected" when TXEN* is asserted LOW. Following selection, data is written into the Transmit FIFO on every clock cycle where TXEN* remains LOW.

Cascade Timing Model

The Cascade timing model is a variation of the UTOPIA timing model. The multi-PHY polling schemes used by UTOPIA-II do not work in the Cascade mode. This mode is intended for point-to-point (single PHY) use only. Here the TXEMPTY and TXFULL* outputs, and TXEN input are all active HIGH signals. Cascade timing makes use of the same address and selection sequences as UTOPIA timing, but write data accesses use a delayed write. This delayed write is necessary to allow direct coupling to external FIFOs, or to state machines that initiate a write operation one clock cycle before the data is available on the bus.

Cascade timing is enabled by setting EXTFIFO HIGH.

When used for FIFO depth expansion, Cascade timing allows the size of the internal Transmit FIFO to be expanded to an almost unlimited depth. It allows a CY7C42x5 series synchronous FIFO to be attached to the transmit interface without any extra logic, as shown in *Figure 3*.

Transmit FIFO

The Transmit FIFO is used to buffer data captured in the input register for later processing and transmission. This FIFO is sized to hold 256 14-bit characters. When the Transmit FIFO is enabled, and a Transmit FIFO write is enabled (the device is selected and TXEN* is sampled asserted) data is captured in the transmit input register and stored into the Transmit FIFO. All Transmit FIFO write operations are clocked by TXCLK.

The Transmit FIFO presents Full, Half-Full, and Empty FIFO flags. These flags are provided synchronous to TXCLK to allow operation with a Moore-type external controlling state machine. When configured for Cascade timing, the timing and active levels of these signals are also designed to support direct expansion to Cypress CY7C42x5 synchronous FIFOs.

The Transmit FIFO can be clocked at any rate from DC to 50 MHz. This gives the Transmit FIFO a maximum bandwidth of 50 million characters per second. Since the serial output speed is limited to 20 million characters per second at its fastest operating rate, there is ample time to service multiple HOTLinks with a single controller.

The read port of the Transmit FIFO is connected to a logic block that performs data formatting and validation. All data read operations from the Transmit FIFO are controlled by a Transmit Control State Machine that operates synchronous to REFCLK.

Transmit Formatter and Validation

The Transmit Formatter and validation logic controls the timing for the transfer of data from the Transmit Input Register, Transmit FIFO, or Elasticity Buffer.

Transmit Data Formatting

The CY7C954DX supports a number of protocol enhancements over a raw physical-layer device. These enhancements are made possible in part through the use of the Transmit and Receive FIFOs. These FIFOs allow the CY7C954DX to manage the data stream to a much greater extent than was possible before. In addition to the standard 8B/10B encoding used to improve serial data transmission, the CY7C954DX also supports:

- marking of packet or cell boundaries using TXSOC
- calculation and optional insertion of Header Error Check byte (HEC).

Both of these capabilities are supported for 8-bit encoded characters, and use of the TXSOC bit. This bit is interpreted, along with TXSC/D* and TXSVS to trigger HEC operations and create three unique Start of Cell Markers. All three bits determine how the data associated with them is processed for transmission. These operations are listed in *Table 1*.

The entries in *Table 1* where TXSOC is LOW generate the same characters in the serial data stream as a standard CY7B923 HOTLink Transmitter, which uses the ANSI standard 8B/10B character set. The data, command, and exception character encoding are listed in the Data and Special Character code tables (*Tables 8* and *9*) found near the end of this data sheet.

Table 1. Transmit Data Formatting

TXSOC	TXSC/D*	TXSVS	Data Format Operation
0	0	0	Normal data encode
0	0	1	Replace character with C0.7 Exception
0	1	0	Normal command encode
0	1	1	Replace character with C0.7 Exception
1	0	0	Send Start of Cell Marker Type I (C8.0) + Data Character
1	0	1	Replace character with C0.7 Exception
1	1	0	Send Start of Cell Marker Type II (C9.0) + Data Character
1	1	1	Send Start of Cell Marker Type III (C10.0) + Data Character

When the TXSOC bit (as read from the Transmit FIFO) is HIGH, an extra character is inserted into the data stream. This extra character is always a Special Character code (see *Table 9*) that is used to inform the remote receiver that the immediately following character should be interpreted differently from its normal meaning. The associated character present on TXDATA[x:0] is always encoded as a data character.

The 100b (TXSOC = 1, TXSC/D* = 0, and TXSVS = 0), 110b and 111b combinations are used as markers for the start of a cell, frame, or packet of data being sent across the interface. When a character is read from the Transmit FIFO with these bits set, a C8.0, C9.0, or C10.0 Special Character code is sent to the encoder prior to sending the associated data character.

The 101b encoding has the same function as the 001b and 011b normal data modes. It instructs the encoder to discard the associated data character and to replace it with a C0.7 Exception character.

The 110b encoding might be used for a context-based Start of Cell marker (SOC with one of three modifiers), or it could be used to expand the command space beyond that available with the default 8B/10B code (SC/D* with a modifier). The 8B/10B code normally supports a *data* space of 256 data characters, and a *command* (non-data) space of twelve command characters (C0.0-C11.0 in *Table 9*). For those data links where these few commands are not sufficient, the 110b encoding can be used to mark the associated data as an extended command. This expands the command space to 256 commands (in addition to some of the present twelve). When a character is read from the Transmit FIFO with these bits set, a C9.0 Special Character code is sent to the encoder prior to sending the data character.

Note: Since this character is interpreted as a “Start of Cell” marker, care should be taken in its placement. If the receiver is in Receive Mode (00, 01,10) placements that create “illegal ATM cells” will be discarded.

The 111b encoding might be used for a different context-based Start of Cell marker (SOC with one of three modifiers).

When a character is read from the Transmit FIFO with these bits set, a C10.0 Special Character is sent to the encoder prior to sending the associated data character.

Header Error Check Generation and insertion

If HEC generation is enabled (although not really a “Receive Mode”, this function is enabled by 00b on the RXMODE[1:0] pins; see *Table 5*) the transmitter will overwrite the 5th byte of each ATM cell with the appropriate internally generated HEC code. This code is a CRC of the first four bytes in the ATM header (the first four bytes after the Transmit Start of Cell Marker) as is defined by the ATM Forum spec I413.

Encoder Block

The Encoder logic block performs two primary functions: encoding the data for serial transmission and generating BIST (Built-In Self Test) patterns to allow at-speed link and device testing.

BIST LFSR

The Encoder logic block operates on data stored in a register. This register accepts information directly from the Transmit FIFO, the Transmit Input Register, the 10/8 Byte-Packer, or from the Transmit Control State Machine when it inserts special characters into the data stream.

This same register is converted into a Linear-Feedback Shift-Register (LFSR) when the Built-In Self-Test (BIST) pattern generator is enabled (TXBISTEN* is LOW). When enabled, this LFSR generates a 511-character sequence that includes all Data and Special Character codes, including the explicit violation symbols. This provides a predictable but pseudo-random sequence that can be matched to an identical LFSR in the Receiver.

The specific patterns generated are described in detail in the Cypress application note “HOTLink Built-In Self-Test.” The sequence generated by the CY7C954DX is identical to that in the CY7B923, CY7C924, and CY7B929, allowing interoperable systems to be built when used at compatible serial signaling rates and appropriate ATM Cell handling logic, since none of these are ATM aware.

Encoder

The data passed through the Transmit FIFO and formatter, or as received directly from the Transmit Input Register, is seldom in a form suitable for transmission across a serial link. The characters must usually be processed or transformed to guarantee:

- a minimum transition density (to allow the serial receiver PLL to extract a clock from the data stream),
- a DC-balance in the signaling (to prevent baseline wander),
- run-length limits in the serial data (to limit the bandwidth of the link), and
- some way to allow the remote receiver to determine the correct character boundaries (framing).

The CY7C954DX contains an integrated 8B/10B encoder that accepts 8-bit data characters and converts these into 10-bit transmission characters that have been optimized for transport on serial communications links. The operation of the 8B/10B encoding algorithm is described in detail later in this datasheet, and the complete encoding tables are listed in *Tables 8* and *9*.

The transmit data characters (as passed through the Transmit FIFO and formatter) are converted to either a 10-bit Data symbol or a 10-bit Special Character, depending upon the state of the TXSC/D* input. If TXSC/D* is HIGH, the data inputs represent a Special Character code and are encoded using the Special Character encoding rules in *Table 9*. If TXSC/D* is LOW, the data inputs are encoded using the Data Character encoding in *Table 8*.

If this bit (TXSVS) is HIGH, the respective character is replaced with an SVS (C0.7) character. This can be used to check error handling system-logic in the receiver controller or for proprietary applications. This will cause the entire ATM cell to be discarded, except in Receive Mode (11), which will pass the error character to down stream logic.

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.

Transmit Shifter

The Transmit Shifter accepts 10-bit parallel data from the Encoder block once each character time, and shifts it out the serial interface output buffers using a PLL-multiplied bit-clock. This bit-clock runs at 2.5, 5, or 10 times the REFCLK rate as

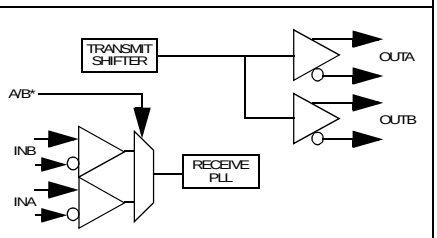
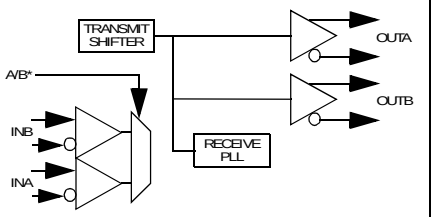
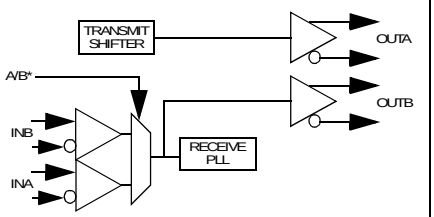
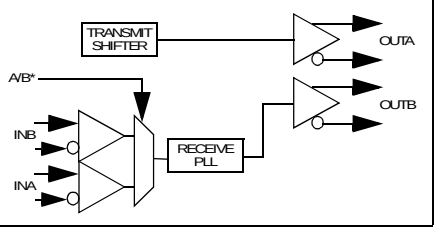
selected by RANGESEL and SPDSEL (see *Table 3*). Timing for the parallel transfer is controlled by the counter and dividers in the Clock Multiplier PLL and is not affected by signal levels or timing at the input pins.

Bits in each character are shifted out LSB first, as required by ANSI and IEEE standards for 8B/10B coded serial data streams.

Routing Matrix

The Routing Matrix is a set of precision multiplexors that allow various combinations of Transmit Shifter, buffered INA± or INB± serial line receiver inputs, or a reclocked serial line receiver input to be transmitted from the OUTB± serial data outputs. The signal routing for the transmit serial outputs is controlled primarily by the DLB[1:0] inputs as listed in *Table 2*.

Table 2. Transmit Data Routing Matrix

DLB[1]	DLB[0]	Data Connections
0	0	
0	1	
1	0	
1	1	

Serial Line Drivers

The serial interface PECL-compatible Output Drivers (ECL referenced to +5V) are the transmission line drivers for the serial media. OUTA± receives its data direct from the transmit shifter, while OUTB± receives its data from the Routing Matrix. These two outputs (OUTA± and OUTB±) are capable of direct connection to +5V optical modules, and can also directly drive DC- or AC-coupled transmission lines.

The PECL-compatible Output Drivers can be viewed as programmable current sources. The output voltage is determined by the output current and the load impedance Z_{LOAD}. The desired output voltage swing is therefore controlled by the current-set resistor R_{CURSET} associated with that driver. Different R_{CURSET} values are required for different line impedance/amplitude combinations. The output swing is designed to center around V_{DD}–1.33V. Each output must be externally biased to V_{DD}–1.33V.

This differential output-swing can be specified two ways: either as a peak-to-peak voltage into a single-end load, or as an absolute differential voltage into a differential load.

When specified into a single-ended load (one of the outputs switching into a load), the single output will both source and sink current as it changes between its HIGH and LOW levels. The voltage difference between this HIGH level and LOW level determine the peak-to-peak signal-swing of the output. This amplitude relationship is controlled by the load impedance on the driver, and by the resistance of the R_{CURSET} resistor for that driver, as listed in *Eq. 1*

$$R_{CURSET} = \frac{180 \times Z_{LOAD}}{V_{OPP}} \quad \text{Eq. 1}$$

In *Eq. 1*, V_{OPP} is the difference in voltage levels at one output of the differential driver when that output is driving HIGH and LOW, Z_{LOAD} is that load seen by the one output when it is sourcing and sinking current. With a known load impedance and a desired signal swing, it is possible to calculate the value of the associated CURSETA or CURSETB resistor that sets this current.

Unused differential output drivers should be left open, and can reduce their power dissipation by connecting their respective CURSETx input to V_{DD}.

Transmit PLL Clock Multiplier

The Transmit PLL Clock Multiplier accepts an external clock at the REFCLK input, and multiples that clock by 2.5, 5, or 10 to generate a bit-rate clock for use by the transmit shifter. It also provides a character-rate clock used by the Transmit Controller state machine.

The clock multiplier PLL can accept a REFCLK input between 10 MHz and 40 MHz, however, this clock range is limited by the operation mode of the CY7C954DX as selected by the SPDSEL and RANGESEL inputs. The operating serial signalling rate and allowable range of REFCLK frequencies is listed in *Table 3*.

Table 3. Speed Select and Range Select Settings

SPDSEL	RANGESEL	Serial Data Rate (MBaud)	REFCLK Frequency (MHz)
LOW	LOW	50–100	10–20
LOW	HIGH	50–100	20–40
HIGH	LOW	100–200	10–20
HIGH	HIGH	100–200	20–40

Transmit Control State Machine

The Transmit Control State Machine responds to multiple inputs to control the data stream passed to the encoder. It operates in response to:

- the state of the LOOPTX inputs
- the presence of data in the Transmit FIFO
- the contents of the Transmit FIFO
- the contents of the Elasticity Buffer
- the state of the transmitter BIST enable (TXBISTEN*)

These signals are used by the Transmit Control State Machine to control the data formatter, read access to the Transmit FIFO and Elasticity Buffer, the Byte-Packer, and BIST. They determine the content of the characters passed to the Encoder and Transmit Shifter.

Elasticity Buffer

A short (8-character) FIFO is contained between the receive and transmit paths. This FIFO is used to separate the time domains of the received serial data stream and the outbound transmit data stream. This permits retransmission of received data without worry of jitter gain or jitter transfer. This allows error-free transmission of the same data, when configured in daisy-chain or ring configurations, to an unlimited number of destinations.

This Elasticity Buffer is enabled when the LOOPTX input is asserted HIGH. This directs the receiver to place all non-C5.0 (K28.5) characters into the Elasticity Buffer. LOOPTX also directs the Transmit Control State Machine to read data from the Elasticity Buffer instead of from the Transmit FIFO.

While retransmitting data from the Elasticity Buffer, the Transmit FIFO is available for preloading of data to be transmitted. Once LOOPTX is deasserted (LOW), normal data transmission from the Transmit FIFO resumes.

Serial Line Receivers

Two differential line receivers, INA± and INB±, are available for accepting serial data streams, with the active input selected using the A/B* input.

The DLB[1:0] inputs allow the transmit Serializer output to be selected as a third input serial stream (DLB[1:0]=01 is the diagnostic loopback function). The serial line receiver inputs are all differential, and will accommodate wire interconnect with filtering losses or transmission line attenuation greater than 9 dB ($V_{DIF} \geq 200$ mV, or 400 mV peak-to-peak differential) or can be directly connected to +5V fiber-optic interface modules with appropriate terminations (any ECL logic family, not limited to ECL 100K). The common-mode tolerance of these line receivers accommodates a wide range of signal termination voltages. Input levels less than about 300 mV will cause LFI* to be asserted LOW, indicating a line fault, but the input should still decode data correctly.

As can be seen in *Table 2*, these inputs are configured to allow single-pin control for most applications. For those systems requiring selection of only INA± or INB±, the DLB[1:0] signals can be tied LOW, and the A/B selection can be performed using only A/B*. For those systems requiring only a single input and a local loopback, the A/B* can be tied HIGH or LOW, DLB[1] signal can be tied LOW and DLB[0] can be used for loopback control.

The level-restored (10) and reclocked (11) settings make use of one of the transmit data outputs. When configured for level-restored or reclocked data, the selected input is retransmitted on OUTB±. The level-restored connection simply buffers the input signal allowing a “bus-like” connection to be constructed without concern for multi-drop PECL signal layout issues.

The reclocked connection buffers a PLL-filtered copy of the selected input data stream. This removes most of the high-frequency jitter that accumulates on a signal when sent over long transmission lines. Unlike data retransmitted from the Elasticity Buffer, the output data stream is clocked by a recovered clock, not by a derivative of the local REFLCK input. This allows a data source to provide data to multiple recipients, but can suffer from jitter peaking when communicated through several PLLs. The reclocked connection may be required when sending non-8B/10B coded data streams, or data streams that cannot tolerate the data forwarding policies of the Elasticity Buffer.

This reclocked output stream may also be beneficial in systems requiring very low latency. The internal data delays for a reclocked serial stream are a small number of bits, while data sent through the Elasticity Buffer incurs a delay of a small number of characters.

Signal Detect

The selected Line Receiver (that routed to the clock and data recovery PLL) is simultaneously monitored for:

- analog amplitude (>400 mV pk-pk) on selected input,
- transition density,
- received data stream outside normal frequency range (±400 ppm),
- and carrier detected.

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 LFI* (Link Fault Indicator) output, which changes synchronous to RXCLK. While link status is monitored internally at all times, it is necessary to have transitions on RXCLK to allow this signal to change externally.

Clock/Data Recovery

The extraction of a bit-rate clock and recovery of data bits from the received serial stream is performed within the Clock/Data Recovery (CDR) block. The clock extraction function is performed by a high-performance embedded phase-locked loop (PLL) that tracks the frequency of the incoming bit stream and aligns the phase of its internal bit-rate clock to the transitions in the serial data stream.

The CDR makes use of the clock present at the REFCLK input. It is used to ensure that the VCO (within the CDR) is operating at the correct frequency (rather than some harmonic of the bit rate), to improve PLL acquisition time, and to limit unlocked frequency excursions of the CDR VCO when no data is present at the 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 for the range controls, the CDR PLL will track REFCLK 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 REFCLK is required to be within ±400 ppm of the frequency of the clock that drives the

REFCLK signal at the remote transmitter to ensure a lock to the incoming data stream.

For systems using multiple or redundant connections, the LFI* output can be used to select an alternate data stream. When an LFI* indication is detected, external logic can toggle selection of the INA± and INB± inputs through the A/B* input. When a port switch takes place, it is necessary for the PLL to reacquire the new serial stream and frame to the incoming characters.

Clock Divider

This block contains the clock division logic used to transfer the data from the Deserializer/Framer to the Decoder once every character (once every ten or twelve bits) clock. This counter is free running and generates outputs at the bit-rate divided by 10.

Deserializer/Framer

The CDR circuit extracts bits from the 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 C5.0 (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.

The framer operates in one of three different modes, as selected by the RFEN input. When RFEN is first asserted (HIGH), the framer is allowed to reset the internal character boundaries on any detected C5.0 character.

Once RFEN has been HIGH for greater than approximately 2000 character clock cycles, the multi-byte framer is enabled. This requires two C5.0 characters, within a span of five characters, with both C5.0 characters located on identical 10-bit character boundary locations, before the framer is allowed to reset the internal character boundary.

If RFEN is LOW, the framer is disabled and no changes are made to character boundaries.

The framer in the CY7C954DX operates by shifting the internal character position to align with the character clock. This ensures that the recovered clock will not contain any significant phase changes/hops during normal operation or framing, and allows the recovered clock to be replicated and distributed to other circuits using PLL-based logic elements.

Decoder Block

The decoder logic block performs two primary functions: decoding the received transmission characters back into Data and Special Character codes, and comparing generated BIST patterns with received characters to permit at-speed link and device testing.

10B/8B Decoder

The framed parallel output of the Deserializer is passed to the 10B/8B Decoder where, if the Decoder is enabled, it is transformed from a 10-bit transmission character back to the original Data and Special Character codes. This block uses the standard decoder patterns in *Tables 8* and *9* of this data sheet. Data patterns are indicated by a LOW on RXSC/D*, and Special Character codes are indicated by a HIGH. Invalid patterns or disparity errors are signaled as errors by a HIGH on RXRVS, and by specific Special Character codes.

BIST LFSR

The output register of the Decoder block is normally used to accumulate received characters for delivery to the Receive Formatter block. When configured for BIST mode (RXBISTEN* is LOW), this register becomes a signature pattern generator and checker by logically converting to a Linear-Feedback Shift-Register (LFSR). When enabled, this LFSR generates a 511-character sequence that includes all Data and Special Character codes, including the explicit violation symbols. This provides a predictable but pseudo-random sequence that can be matched to an identical LFSR in the Transmitter. When synchronized with the received data stream, it checks each character in the Decoder with each character generated by the LFSR and indicates compare errors at the RXRVS output of the Receive Output Register.

The LFSR is initialized by the BIST hardware to the BIST loop start code of D0.0 (D0.0 is sent only once per BIST loop). Once the start of the BIST loop has been detected by the receiver, RXRVS is asserted for pattern mismatches between the received characters and the internally generated character sequence. Code rule violations or running disparity errors that occur as part of the BIST loop do not cause an error indication. RXFULL* pulses asserted for one RXCLK cycle per BIST loop and can be used to check test pattern progress.

The specific patterns checked by the receiver are described in detail in the Cypress application note “HOTLink Built-In Self-Test.” The sequence compared by the CY7C954DX is identical to that in the CY7B933 and CY7B929, allowing interoperable systems to be built when used at compatible serial signaling rates.

If a large number of errors are detected, the receive BIST state machine aborts the compare operations and resets the LFSR to the D0.0 state to look for the start of the BIST sequence again.

Receive Formatter

The protocol enhancements of the transmit path are mirrored in the receive path logic. In addition to the standard 10B/8B decoding used for character reception and recovery, the CY7C954DX also supports:

- marking of packet or cell boundaries using RXSOC, RXSC/D*, and RXRVS
- ability to accept or discard data based received cell size
- ability to accept or discard data based received HEC calculation

The entries in *Table 4* show how the RXSOC, RXSC/D*, and RXRVS bits are encoded to indicate the reception of specific characters and character combinations. Normal Data and Special Character code characters are indicated by RXSOC being LOW (0). This allows the standard Special Characters codes to also be reported and output.

Individual character errors that are not part of one of the supported sequences (i.e., Start of Cell Marker) are marked by the 011b (RXSOC = 0, RXSC/D* = 1, and RXRVS = 1) decode.

Anytime RXSOC is reported HIGH (1) at least one of the C8.0, C9.0, or C10.0 characters was received as a valid character. If the immediately following character is a valid Data character, then the corresponding combination of RXSOC, RXSC/D*, and RXRVS indicate the type of information received. If the immediately following character is a Special Character code of

Table 4. Receive Data Formatting

RXSOC	RXSC/D*	RXRVS	Data Format Indication
0	0	0	Normal data character
0	0	1	Reserved
0	1	0	Normal command character
0	1	1	Received C0.7 Exception character or other character exception (as listed in*Table 9*)
1	0	0	Received Start of Cell Marker Type I (C8.0) + Data Character
1	0	1	Received Illegal Sequence
1	1	0	Received Start of Cell Marker Type II (C9.0) + Data Character
1	1	1	Received Start of Cell Marker Type III (C10.0) + Data Character

any type (even a C5.0), then a 101b is posted to indicate an illegal sequence was received.

An illegal sequence can be caused by a remote transmitter sending incorrect information, or by the data getting corrupted during transmission. When such an error is detected and the 101b status bits posted, the associated data field is set to the Special Character code that was received without error (C8.0, C9.0, or C10.0 reported as D8.0, D9.0, or D10.0 along with the 101b status). This information is provided to assist in debugging link or protocol faults.

Note: Since an error in an ATM cell causes the cell to be discarded, except in Receive mode (11), these error indications will not be visible.

The 100b indication is used to mark the associated Data character as the first character of a new Cell (SOC Marker Type I), packet, cell, or other data construct used by the system. The Data characters and Special Character codes that follow this marker are written to the Receive FIFO (depending on the Receiver Discard Policy in effect; see *Table 5*).

The 110b indication is used to mark the associated Data character as the first character of a new Cell (SOC Marker Type II). This marker is treated internally the same as the 100b Start-Of-Cell marker, which allows it to be used to mark the boundary of any user-specific information. As a boundary or cell marker, the immediately following data can be a data field, a header, a stream identifier, a transaction number, a packet length indicator, or any of a number of pieces of information connected to a data transfer.

The 111b indication is used to mark the associated Data character as the first character of a new Cell (SOC Marker Type III). This marker is treated internally the same as the 100b and 110b Start-Of-Cell markers, which allows it to be used to mark the boundary of any user-specific information.

The 100b, 110b and 110b, indicators can be used interchangeably; i.e., they can be used for Start of Cell markers, Extended Command Markers, General Cell routing indicators, or any number of user specified functions.

If the transmit FIFO is allowed to empty, the transmitter will fill the unused data spaces with C5.0 (k28.5). A receiver set to allow cells longer than 53 bytes will assume that eight consecutive C5.0 characters are equivalent to a “virtual SOC” and will release the cell into the FIFO for further processing. Care should be taken to avoid intra-cell gaps that might result in unintentional termination of the ATM cell, because for receive modes that discard cells less than 53 bytes, this “virtual SOC” could result in cell loss.

Receive Control State Machine

The Receive Control State Machine responds to multiple input conditions to control the routing and handling of received characters. It controls the staging of characters across various registers and the Receive FIFO. It also interprets all embedded Special Character codes, and converts the appropriate ones to specific bit combinations in the Receive FIFO. It controls the various discard policies and error control within the receiver. It operates in response to:

- the received character stream
- the room for additional data in the Receive FIFO
- the state of the receiver BIST enable (RXBISTEN*)
- the state of LOOPTX

These signals are used by the Receive Control State Machine to control the Receive Formatter, write access to the Receive FIFO, write access to the Elasticity Buffer, the Receive Output register, and BIST. They determine the content of the characters passed to each of these destinations,

The Receive Control State Machine always operates synchronous to the recovered character clock (bit-clock/10).

Discard Policies

The Receive Control State Machine has the ability to selectively discard specific characters and cells from the data stream that are determined by the present configuration as being unnecessary. When discarding is enabled, it reduces the host system overhead necessary to keep the Receive FIFO from overflowing and losing data.

The discard policy is configured as part of the operating mode and is set using the RXMODE[1:0] inputs. The four discard policies are listed in *Table 5*.

Policy 0 (00b) is the most stringent and also most closely approximates the rules for an ATM data link. In this mode, every cell is checked for a valid Header Error Check byte (transmitted cells have the HEC calculated and inserted into the 5th byte after the Start of Cell marker) and cell lengths are checked.

The receiver will calculate the HEC code for each ATM header and compare it with the 5th byte of each cell. If the calculated HEC code does not match the one that is received, the ATM cell will be discarded.

Long cell truncation is enabled and the HOTLink will truncate long cells to 53 bytes, discarding the remainder. The receiver counts incoming bytes upon receipt of a Start of Cell command code. Once 53 bytes of data are received, a new Start of Cell is assumed. The remainder creates a short cell fragment in the receiver which will be flushed because of HEC check, or length.

Short cell flushing is enabled, so any ATM cells that are shorter than 53 bytes will be discarded. The RXFIFO will begin counting incoming bytes upon receipt of the Start of Cell command code. If another Start of Cell command code is received before

Table 5. Receiver Discard Policies

Policy#	RXMODE[1]	RXMODE[0]	Policy Description
0	0	0	HEC-Gen=ON, (Insert calculated HEC into 5th byte of Transmitted Cell.) Discard Cells with Received HEC-Check=Error, Truncate Cells >53 bytes, Discard Short Cells<53 bytes
1	0	1	HEC-Gen=OFF (Ignore Transmitter HEC) Discard Cells with Received HEC-Check=Error, Truncate Cells >53 bytes, Discard Short Cells<53 bytes
2	1	0	HEC-Gen=OFF, No Received HEC-Check, Truncate Cells >64 bytes, Discard Short Cells<53 bytes
3	1	1	HEC-Gen=OFF, No Received HEC-Check, No Cell byte counting

53 bytes of data have been accumulated, the cell will be flushed from the FIFO.

Policy 1 (01b) is similar to Policy 0, except that the Transmitter does not insert HEC, but the Receiver checks and discards cells with defective HEC. Long cells are truncated and short cells are discarded.

Policy 2 (10b) further relaxes the restrictions on the data format by not generating or checking HEC. Cells longer than 64 bytes are truncated and short cells are discarded.

Policy 3 (11b) accepts all data that is received and puts it into the Receive FIFO, without checking HEC or Cell byte count.

The Receive FIFO is used to buffer data captured from the selected serial stream for later processing by the host system. This FIFO is sized to hold 256 characters. It is written to by the Receive Control State Machine. When data is present in the Receive FIFO (as indicated by the RXFULL*, RXCLAV (RXHALF* in this mode), and RXEMPTY* Receive FIFO status flags), it can be read from the Output Register by asserting RXADDR[2:0] matching ADDRSEL[2:0] and RXEN*.

All K28.5 (C5.0) characters are automatically discarded at the Receiver FIFO. These characters are used for fill when the Transmitter FIFO goes empty and are not part of any ATM protocol structures.

If eight contiguous C5.0 (K28.5) characters are received, a “virtual SOC” is asserted, which might truncate ATM cells. To prevent this occurrence, the Transmitter FIFO should not be allowed to run empty during ATM cell transmission.

The read port on the Receive FIFO may be configured for the same two timing models as the transmit interface: UTOPIA and Cascade. The UTOPIA timing model has active LOW RXEMPTY* and RXFULL* status flags, and an active LOW RXEN* enable. When configured for Cascade operation (not compatible with UTOPIA status polling), these same signals are all active HIGH. Either timing model supports connection

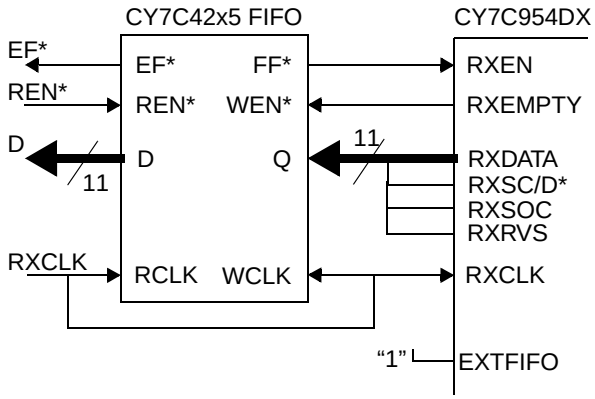


Figure 4. External FIFO Depth Expansion of the CY7C954DX Receive Data Path

to various host bus interfaces, state machines, or external FIFOs for depth expansion (see *Figure 4*).

The Receive FIFO presents Full, Half-Full, and Empty FIFO status flags. These flags are provided synchronous to RXCLK to allow operation with a Moore-type external controlling state machine.

Receive Output Register

The Receive Output Register changes in response to the rising edge of RXCLK. The Receive FIFO status flag outputs of this register are placed in a High-Z state when the CY7C954DX is not addressed (RXADDR[2:0] doesn’t match ADDRSEL[2:0]). The RXDATA bus output drivers are enabled when the device is addressed (RXADDR[2:0] matches ADDRSEL[2:0]), and RXEN* is sampled asserted, initiating a Receive FIFO read cycle.

CY7C954DX DC Electrical Characteristics Over the Operating Range

Parameter	Description	Test Conditions	Min.	Max.	Unit
TTL Outputs					
V_{OHT}	Output HIGH Voltage	I_{OH} = –2 mA, V_{DD} = Min	2.4		V
V_{OLT}	Output LOW Voltage	I_{OL} = 8 mA, V_{DD} = Min		0.4	V
I_{OST}	Output Short Circuit Current	V_{OUT} = 0V[1]	–30	–80	mA
I_{OZL}	High-Z Output Leakage Current		–20	20	µA
TTL Inputs					
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}		+40	µA
I_{ILT}	Input LOW Current	V_{IN} = 0.0V	–40		µA
I_{IHPD}	Input HIGH Current	V_{IN} = V_{CC}, Pins with internal pull-down		+300	µA
I_{ILPU}	Input LOW Current	V_{IN} = 0.0V, Pins with internal pull-up	–300		µA
Transmitter PECL-Compatible Output Pins: OUTA+, OUTA–, OUTB+, OUTB–					
V_{OHE}	Output HIGH Voltage (V_{DD} referenced)	Load = 50Ω to V_{CC} – 1.33V R_{CURSET}= 10k (± 1% tolerance)	V_{DD} – 1.03	V_{DD} – 0.83	V
V_{OLE}	Output LOW Voltage (V_{DD} referenced)	Load = 50Ω to V_{CC} – 1.33V R_{CURSET}= 10k (± 1% tolerance)	V_{DD} – 2	V_{DD} – 1.62	V
V_{ODIF}	Output Differential Voltage |(OUT+) – (OUT–)|	Load = 50Ω to V_{CC} – 1.33V R_{CURSET}= 10k (± 1% tolerance)	600	1100	mV
Receiver Single-ended PECL-Compatible Input Pin: CARDET					
V_{IHE}	Input HIGH Voltage (V_{DD} referenced)		V_{DD} – 1.165	V_{DD}	V
V_{ILE}	Input LOW Voltage (V_{DD} referenced)		2.5	V_{DD} – 1.475	V
I_{IHE}	Input HIGH Current	V_{IN} = V_{IHE}(min.)		+40	µA
I_{ILE}	Input LOW Current	V_{IN} = V_{ILE}(max.)	–40		µA
Differential Line Receiver Input Pins: INA+, INA–, INB+, INB–					
V_{DIFF}	Input Differential Voltage |(IN+) – (IN–)|		200	2500	mV
V_{IHH}	Highest Input HIGH Voltage			V_{DD}	V
V_{ILL}	Lowest Input LOW Voltage		2.5		V
I_{IHH}	Input HIGH Current	V_{IN} = V_{IHH} Max.		750	µA
I_{ILL}[2]	Input LOW Current	V_{IN} = V_{ILL} Min.	–200		µA
Miscellaneous			**Typ.**	**Max.**	
I_{DD} [3]	Power Supply Current	Freq. = Max.	170	250	mA

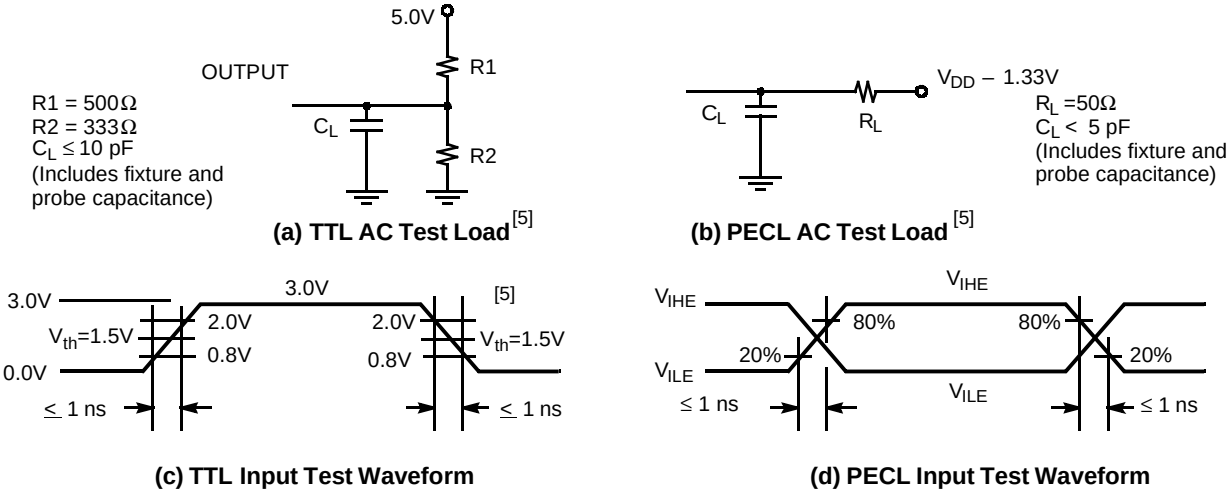
Capacitance[4]

Parameter	Description	Test Conditions	Max.	Unit
C_{INTTL}	TTL Input Capacitance	T_A = 25°C, f_0 = 1 MHz, V_{DD} = 5.0V	7	pF
C_{INPECL}	PECL input Capacitance	T_A = 25°C, f_0 = 1 MHz, V_{DD} = 5.0V	4	pF

Notes:

1. Tested one output at a time, output shorted for less than one second, less than 10% duty cycle.
2. To guarantee positive currents for all PECL voltages, an external pull-down resistor must be present.
3. Maximum I_{DD} is measured with V_{DD} = MAX, RFEN = LOW, and outputs unloaded. Typical I_{DD} is measured with V_{DD} = 5.0V, T_A = 25°C, RFEN = LOW, and outputs unloaded.
4. Tested initially and after any design or process changes that may affect these parameters, but not 100% tested.

AC Test Loads and Waveforms



(a) TTL AC Test Load [5]

(b) PECL AC Test Load [5]

(c) TTL Input Test Waveform

(d) PECL Input Test Waveform

CY7C954DX Transmitter TTL Switching Characteristics Over the Operating Range

Parameter	Description	7C954DX		Unit
		Min.	Max.	
f_{TS}	TXCLK Clock Cycle Frequency		50	MHz
t_{TXCLK}	TXCLK Period	20		ns
t_{TXCPWH}	TXCLK HIGH Time	6.5		ns
t_{TXCPWL}	TXCLK LOW Time	6.5		ns
t_{TXCLKR}	TXCLK Rise Time	0.7	2	ns
t_{TXCLKF}	TXCLK Fall Time	0.7	2	ns
t_{TXA}	Flag Access Time From TXCLK↑ to Output	2	15	ns
t_{TXDS}	Transmit Data Set-Up Time to TXCLK↑	4		ns
t_{TXDH}	Transmit Data Hold Time from TXCLK↑	1		ns
t_{TXENS}	Transmit Enable Set-Up Time to TXCLK↑	4		ns
t_{TXENH}	Transmit Enable Hold Time from TXCLK↑	1		ns
t_{TXRSS}	Transmit FIFO Reset (TXRST*) Set-Up Time to TXCLK↑	4		ns
t_{TXRSH}	Transmit FIFO Reset (TXRST*) Hold Time from TXCLK↑	1		ns
t_{TXAMS}	Transmit Address Match Set-Up Time to TXCLK↑	4		ns
t_{TXAMH}	Transmit Address Match Hold Time from TXCLK↑	1		ns
t_{TXOZA}	Sample of Address Match TRUE by TXCLK↑, Output High-Z to Active HIGH or LOW	0		ns
t_{TXOE}	Sample of Address Match TRUE by TXCLK↑ to Output Valid	1.5	20	ns
t_{TXOAZ}	Sample of Address Match FALSE by TXCLK↑ to Output in High-Z	1.5	20	ns

Note:
5. Cypress uses constant current (ATE) load configurations and forcing functions. This figure is for reference only.

CY7C954DX Receiver TTL Switching Characteristics Over the Operating Range

Parameter	Description	7C954DX		Unit
		Min.	Max.	
f_{RIS}	RXCLK Clock Cycle Frequency		50	MHz
t_{RXCLKP}	RXCLK Input Period	20		ns
t_{RXCPWH}	RXCLK Input HIGH Time	6.5		ns
t_{RXCPWL}	RXCLK Input LOW Time	6.5		ns
$t_{RXCLKIR}$	RXCLK Input Rise Time	0.25	2	ns
$t_{RXCLKIF}$	RXCLK Input Fall Time	0.25	2	ns
t_{RXENS}	Receive Enable Set-Up Time to RXCLK↑	4		ns
t_{RXENH}	Receive Enable Hold Time from RXCLK↑	1		ns
t_{RXRSS}	Receive FIFO Reset (RXRXT*) Set-Up Time to RXCLK↑	4		ns
t_{RXRSH}	Receive FIFO Reset (RXRXT*) Hold Time from RXCLK↑	1		ns
t_{RXAMS}	Receive Address Match Set-Up Time to RXCLK↑	4		ns
t_{RXAMH}	Receive Address Match Hold Time from RXCLK↑	1		ns
t_{RXA}[6]	Flag and Data Access Time From RXCLK↑ to Output	1.5	15	ns
t_{RXOZA}[6]	Sample of Address Match TRUE by RXCLK↑, Output High-Z to Active HIGH or LOW, or Sample of RXEN* Asserted by RXCLK↑, Output High-Z to Active HIGH or LOW	0		ns
t_{RXOE}[6]	Sample of Address Match TRUE by RXCLK↑ to Output Valid[6], or Sample of RXEN* Asserted by RXCLK↑ to RXDATA Outputs Valid	1.5	20	ns
t_{RXOAZ}[6]	Sample of Address Match FALSE by RXCLK↑ to Output in High-Z[6], or Sample of RXEN* Asserted by RXCLK↑ to RXDATA Outputs in High-Z	1.5	20	ns

CY7C954DX REFCLK Input Switching Characteristics Over the Operating Range

Parameter	Description	Conditions		Min.	Max.	Unit
		SPDSEL	RANGESEL			
f_{REF}	REFCLK Clock Frequency—40 to 100 MBaud, 8-bit mode, REFCLK = 2x character rate	0	0	8	20	MHz
	REFCLK Clock Frequency—40 to 100 MBaud, 8-bit mode, REFCLK = 4x character rate	0	1	16	40	MHz
	REFCLK Clock Frequency—100 to 200 MBaud, 8-bit mode, REFCLK = character rate	1	0	10	20	MHz
t_{REFCLK}	REFCLK Period			25	120	ns
t_{REFH}	REFCLK HIGH Time			6.5		ns
t_{REFL}	REFCLK LOW Time			6.5		ns
t_{REFRX}	REFCLK Frequency Referenced to Received Clock Period [7]			–0.04	+0.04	%

Notes:

6. Parallel data output specifications are only valid if all outputs are loaded with similar DC and AC loads.
7. REFCLK has no phase or frequency relationship with the recovered byte clock, which sets the Receiver operating frequency, and only acts as a centering reference to reduce clock synchronization time. REFCLK must be within ±0.04% of the transmitter PLL reference (REFCLK) frequency, necessitating a ±200-PPM crystal.

CY7C954DX Receiver Switching Characteristics Over the Operating Range

Parameter	Description	Condition	7C954DX		Unit
			Min.	Max.	
t_B[8]	Bit Time		5.0	20.0	ns
t_{IN-J}	IN± Peak-to-Peak Input Jitter Tolerance[4, 10, 11]			0.5	UI
t_{SA}	Static Alignment[4, 9]			600	ps
t_{EFW}	Error Free Window[4, 10, 12]		0.65		UI

CY7C954DX Transmitter Switching Characteristics Over the Operating Range

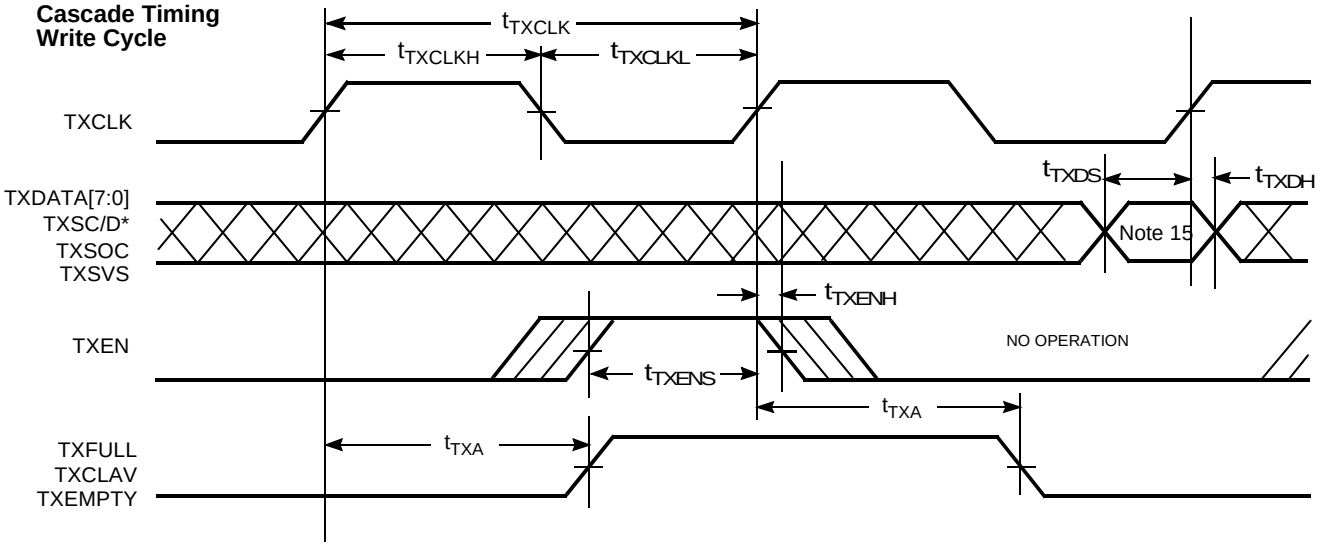
Parameter	Description	Condition	7C954DX		Unit
			Min.	Max.	
t_B[8]	Bit Time		5.0	20.0	ns
t_{RISE}	PECL Output Rise Time 20–80% (PECL Test Load)[4]		200	1700	ps
t_{FALL}	PECL Output Fall Time 80–20% (PECL Test Load)[4]		200	1700	ps
t_{DJ}	Deterministic Jitter (peak-peak)[4, 13]			0.02	UI
t_{RJ}	Random Jitter (σ)[4, 14]			0.008	UI
t_{JT}	Transmitter Total Output Jitter (pk-pk)[4]			0.08	UI

Notes:

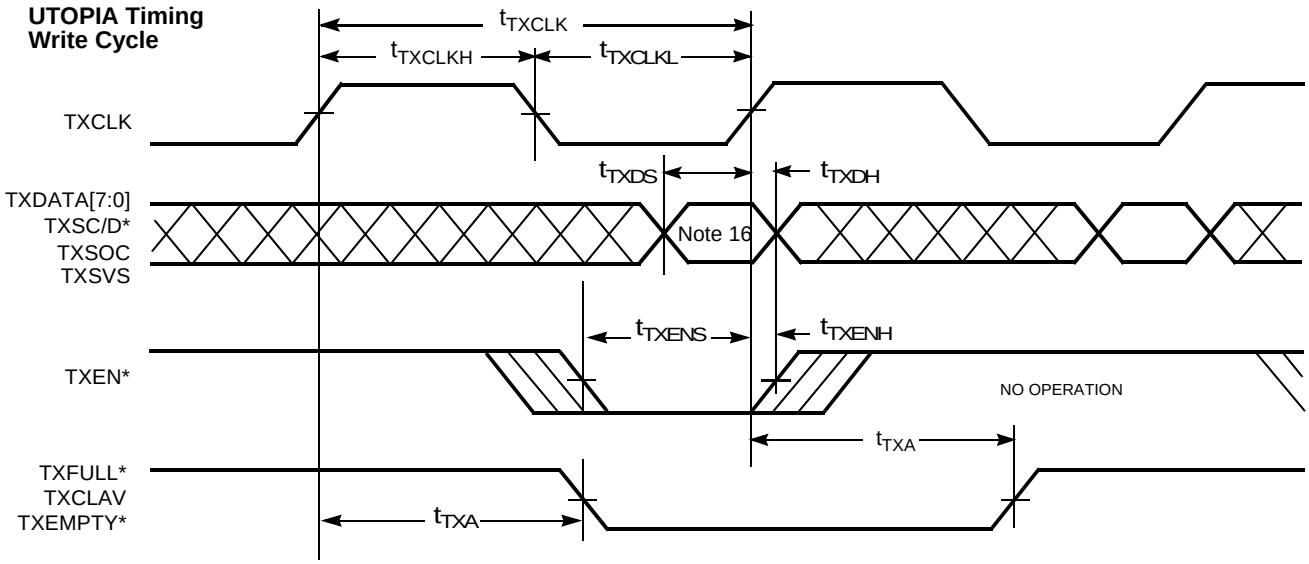
8. The PECL switching threshold is the midpoint between the V_{OHE}, and V_{OLE} specification (approximately V_{DD} – 1.33V).
9. Static alignment is a measure of the alignment of the Receiver sampling point to the center of a bit. Static alignment is measured by the absolute difference of the left and right edge shifts ($|t_{SH\_L} - t_{SH\_R}|$) of one bit until a character error occurs.
10. Receiver UI (Unit Interval) is calculated as ($1/f_{REF}$) if no data is being received, or ($1/f_{REF}$) of the remote transmitter if data is being received. In an operating link this is equivalent to 10 * t_B.
11. The specification is sum of 25% duty cycle distortion (DCD), 10% Data dependent Jitter (DDJ), 15% Random Jitter (RJ).
12. Error Free Window is a measure of the time window between bit centers where an input data transition may occur without causing a bit sampling error. EFW is measured over the operating range, input jitter < 50% Dj.
13. While sending continuous K28.5s, outputs loaded to 50Ω to V_{DD}–1.33V, over the operating range.
14. While sending continuous K28.7s, after 100,000 samples measured at the cross point of differential outputs, time referenced to REFCLK input, over the operating range.

CY7C954DX HOTLink Transmitter Switching Waveforms

Cascade Timing
Write Cycle



UTOPIA Timing
Write Cycle

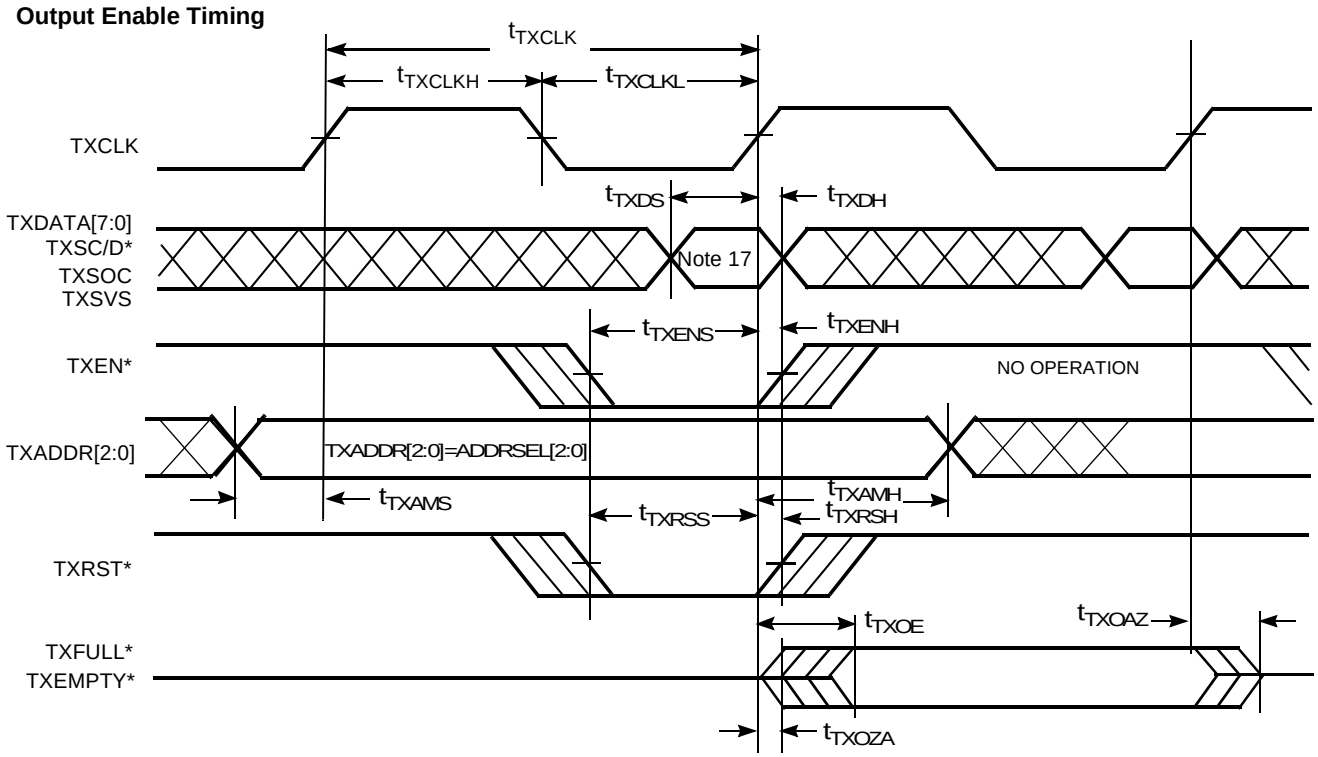


Notes:

15. When transferring data to the Transmit FIFO from a depth expanded external FIFO, the data is captured from the external FIFO one clock cycle following the actual enable.
16. When writing data from a UTOPIA compliant interface, the write data is captured on the same clock cycle as the data.

CY7C954DX HOTLink Transmitter Switching Waveforms (continued)

Output Enable Timing

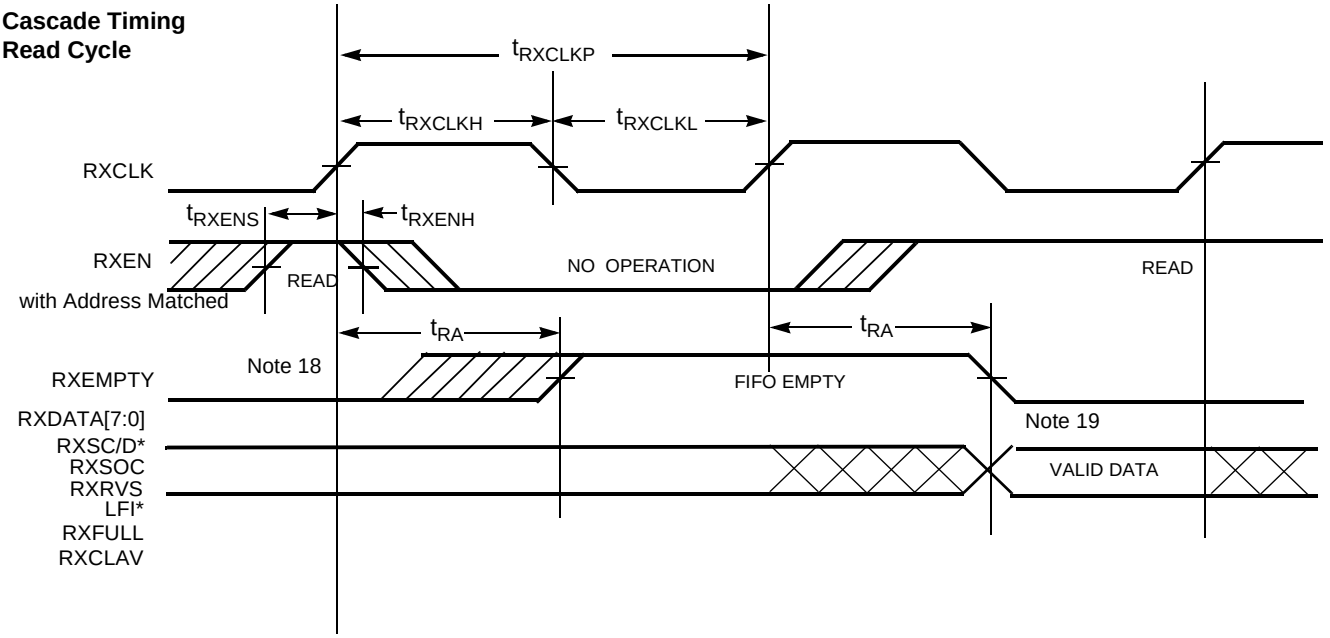


Note:

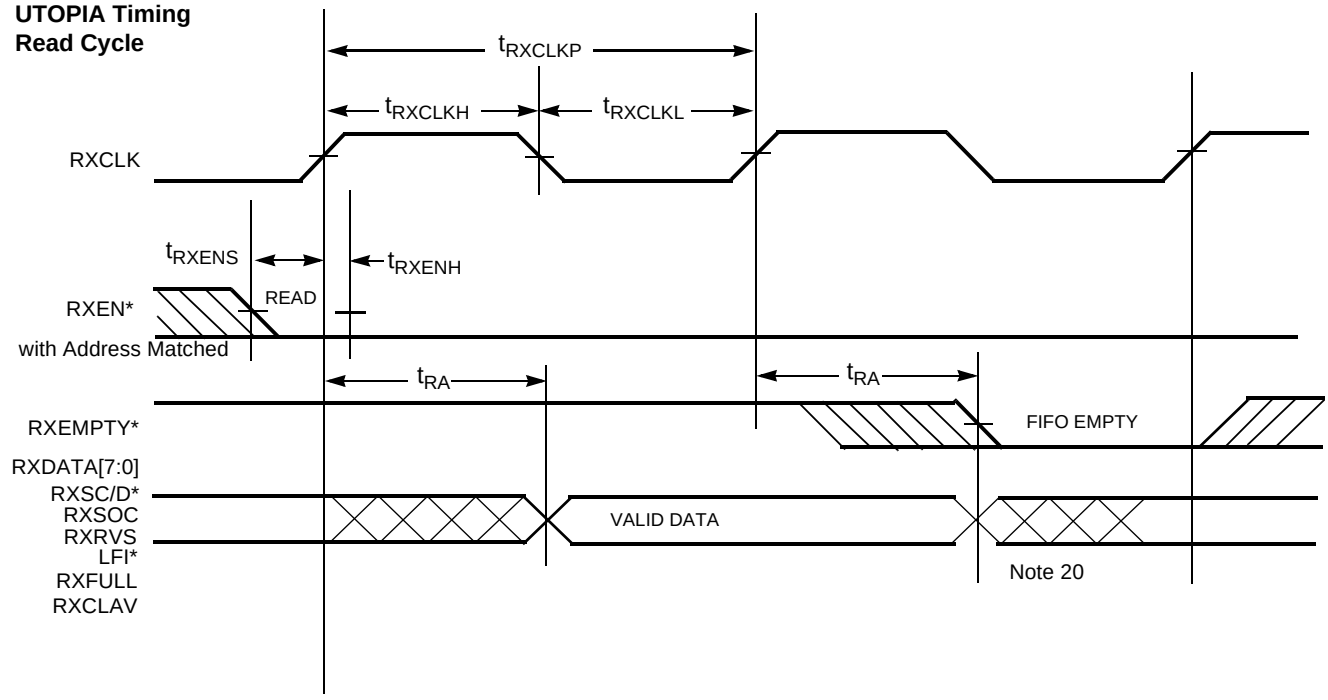
17. Transmit FIFO Writes are permitted while the status flag outputs are High-Z, however operation in this mode is not encouraged since this may mask a FIFO full condition, causing data to be lost.

CY7C954DX HOTLink Receiver Switching Waveforms

Cascade Timing
Read Cycle



UTOPIA Timing
Read Cycle

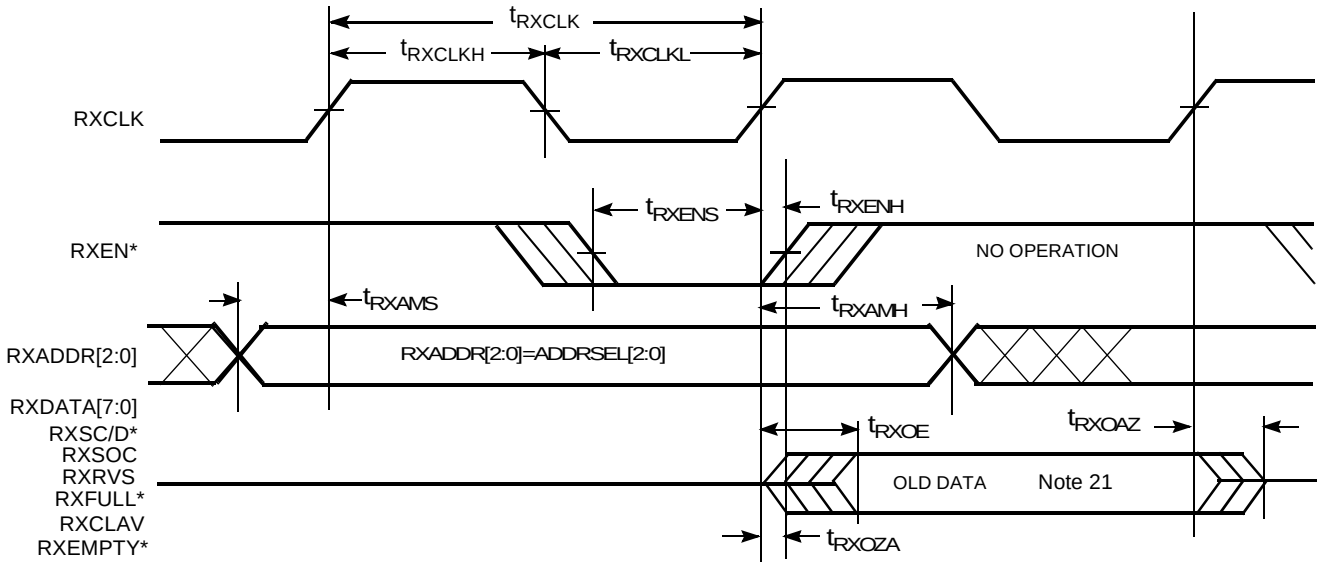


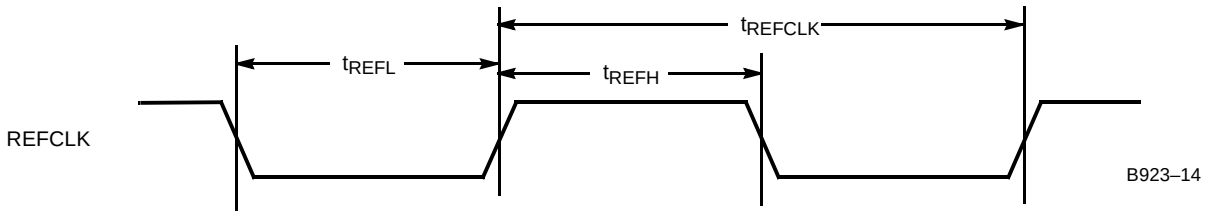
Notes:

18. When transferring data from the Receive FIFO to a depth expanded external FIFO, the data is sent to the external FIFO on the same clock cycle an RXEMPTY indicates the data is available.
19. On inhibited reads, or if the Receive FIFO goes empty, the data outputs do not change.
20. When reading data from a UTOPIA compliant interface, the data is captured on the same clock cycle as the FIFO flag indicates data available, and not when the FIFO indicates empty.

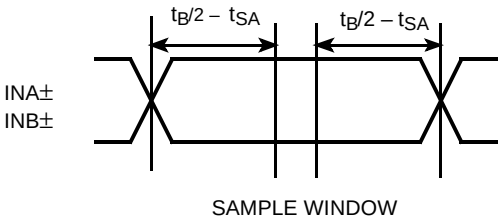
CY7C954DX HOTLink Receiver Switching Waveforms (continued)

Output Enable Timing

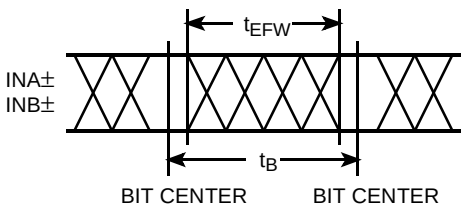




Static Alignment



Error-Free Window



Note:
21. Receive FIFO Reads are inhibited while the outputs are High-Z.

CY7C954DX HOTLink Transceiver Operation

The interconnection of two or more CY7C954DX Transceivers form a general-purpose communications subsystem capable of transporting user data at up to 20 MBytes per second over several types of serial interface media. The CY7C954DX is highly configurable with multiple modes of operation.

In the transmit section of the CY7C954DX, data moves from the input register, through the Transmit FIFO, to the 8B/10B encoder. The encoded data is then shifted serially out the OUTx± differential PECL compatible drivers. The bit-rate clock is generated internally from a 2.5x, 5x, or 10x PLL clock multiplier. A more complete description is found in the section *CY7C954DX HOTLink Transmit-Path Operating Mode Description*.

In the receive section of the CY7C954DX, serial data is sampled by the receiver on one of the INx± differential line receiver inputs. The receiver clock and data recovery PLL locks onto the selected serial bit stream and generates an internal bit-rate sample clock. The bit stream is deserialized, decoded, and presented to the Receive FIFO, along with a character clock. The data in the FIFO can then be read either slower or faster than the incoming character rate. A more complete description is found in the section *CY7C954DX HOTLink Receive-Path Operating Mode Description*.

The Transmitter and Receiver parallel interface timing and functionality can be configured to be a UTOPIA level I or II compliant interface, or for single PHY (point-to-point interfaces) to Cascade directly to external FIFOs for depth expansion.

The HOTLink Transceiver serial interface provides a seamless interface to various types of media. A minimal number of external passive components are required to properly terminate transmission lines and provide PECL loads. For power supply decoupling, a single capacitor (in the range of 0.02 µF to 0.1 µF) is required per power/ground pair. Additional information on interfacing these components to various media can be found in the “HOTLink Design Considerations” application note.

CY7C954DX HOTLink Transmit-Path Operating Mode Descriptions

The HOTLink Transmitter data interface is an asynchronous parallel data register, enabled by a match between the TXADDR[2:0] and the strapped value on ADDRSEL[2:0], and qualified by TXEN.

Input Register Mapping

TXDATA input bus is mapped into characters including a TXSOC and TXSVS bit for protocol mapping, eight bits of data and a TXSC/D* bit to select either control or data characters.

If the TXSVS bit is HIGH (and either TXSOC or TXSC/D* is LOW), an SVS (C0.7) character is passed to the encoder, regardless of the contents of the other TXDATA inputs. If the TXSVS bit is LOW, the associated TXDATA character is encoded per the remaining bits in that character.

When TXSOC is LOW, the TXSC/D* bit controls the encoding of the data bits TXDATA[7:0] of each character. It is used to identify if the input character represents data or a Special Character code. If the TXSC/D* input is LOW, the character is encoded using the Data Character codes listed in *Table 8*. If the TXSC/D* input is HIGH, the character is encoded using the Special Character codes listed in *Table 9*.

This input structure allows transmission of normal data streams, while offering the added benefits of three types of embedded cell markers. The Serializer operates synchronous to REFCLK, which is multiplied by 10 to generate the serial data bit-clock.

Embedded Cell Marker

Embedded cell markers are used to mark the start of cells or frames of information passed from one end of the link to the other. These markers are set by asserting TXSOC HIGH, with TXSC/D* and TXSVS in combinations of HIGH or LOW (see *Table 1*), along with the remaining data on the TXDATA bus. When the character accompanying this marker is read from the output end of the Transmit FIFO, a C8.0 (K23.7), C9.0 (K27.7), or C10.0 (K29.7) character is inserted into the data stream prior to the following data characters being read from the Transmit FIFO.

CY7C954DX HOTLink Receive-Path

The HOTLink Receiver is a Serial-in to Parallel-out converter with some data processing capability.

In this mode, serial data is received at one of the differential line receiver inputs and routed to the Deserializer and Framer. The PLL in the clock and data recovery block is used to extract a bit-rate clock from the transitions in the data stream, and uses that clock to capture bits from the serial stream. These bits are passed to the Deserializer where they are formed into 10-bit characters.

To align the incoming bit stream to the proper character boundaries, the Framer must be enabled by asserting RFEN HIGH. The Framer logic-block checks the incoming bit stream for the unique pattern that defines the character boundaries. This logic filter looks for the ANSI X3.230 symbol defined as a “Special Character Comma” (K28.5 or C5.0). Once a K28.5 is found, the Framer captures the offset of the data stream from the present character boundaries, and resets the boundary to reflect this new offset, thus framing the data to the correct character boundaries.

Since noise-induced errors can cause the incoming data to be corrupted, and since many combinations of corrupt and legal data can create an aliased K28.5, the framer may also be disabled by deasserting RFEN LOW.

An option exists in the framer to require multiple K28.5 characters, meeting specific criteria, before the character boundaries are reset. This multi-byte mode of the Framer is enabled by keeping RFEN asserted HIGH for greater than approximately 2000 character clock cycles. For multi-byte framing, the receiver must find a pair of K28.5 characters, both on identical 10-bit boundaries, within a 5-character span (50 bits).

The deserializer operates synchronous to the recovered bit-clock, which is divided by 10 to generate the Receive FIFO write clock. Data words are read from the Receive FIFO, using the external RXCLK input, when addressed by RXADDR[2:0] matching ADDRSEL[2:0] and selected by RXEN*.

Embedded Cell Marker

Three types of embedded cell marker are available to mark the start of cells or frames of information passed from one end of the link to the other. When a C8.0 (K23.7) character is detected

in the data stream it is discarded and the following character is written to the Receive FIFO along with RXSOC set HIGH, and RXSC/D* and RXRVS set LOW. When the character accompanying this marker is read from the Receive FIFO with these same bits set, it indicates the Start of Cell Marker Type I.

When a C9.0 (K27.7) character is detected in the data stream it is discarded, and the following character is written to the Receive FIFO along with both RXSOC and RXSC/D* set HIGH, and RXRVS set LOW indicating Start of Cell Marker Type II.

When a C10.0 (K29.7) character is detected in the data stream it is discarded, and the following character is written to the Receive FIFO along with RXSOC, RXSC/D*, and RXRVS set HIGH indicating Start of Cell Marker Type III.

BIST Operation and Reporting

The CY7C954DX HOTLink Transceiver incorporates the same Built-In Self-Test (BIST) capability used with the CY7B923/CY7B933 and CY7C954 HOTLink components. This link diagnostic uses a linear-feedback shift-register (LFSR) to generate a 511-character repeating sequence that is compared, character-for-character, at the receiver.

BIST mode is intended to check the entire high-speed serial link at full link-speed, without the use of specialized and expensive test equipment. The complete sequence of patterns used in BIST are documented in the “HOTLink Built-In Self-Test” application note.

BIST Enable Inputs

There are separate BIST enable inputs for the transmit and receive paths of the CY7C954DX. These inputs are both active LOW; i.e., BIST is enabled in its respective section of the device when the BIST enable input is determined to be at a logic-0 level. Both BIST enable inputs are asynchronous; i.e., they are synchronized inside the CY7C954DX to the internal state machines.

BIST Transmit Path

The transmit path operation with BIST is controlled by the TXBISTEN* input and overrides most other inputs (see *Figure 5*). When TXBISTEN* is recognized internally, all reads from the Transmit FIFO are suspended and the BIST generator is enabled to sequence out the 511-character repeating BIST sequence. If the Transmit Control State Machine was in the middle of an atomic operation (e.g., sending an SOC of any type) the Data Character associated with the Special Character code must be transmitted prior to recognition of the TXBISTEN* signal and suspension of FIFO data processing. If the recognition occurs in the middle of a cell, the remaining data for that cell is not transmitted at that time, but remains in the Transmit FIFO. Once the TXBISTEN* signal is removed, the data in the Transmit FIFO is again available for transmission.

With transmit BIST enabled, the Transmit FIFO remains available for loading of data. It may be written up to its normal maximum limit while the BIST operation takes place. To allow re-

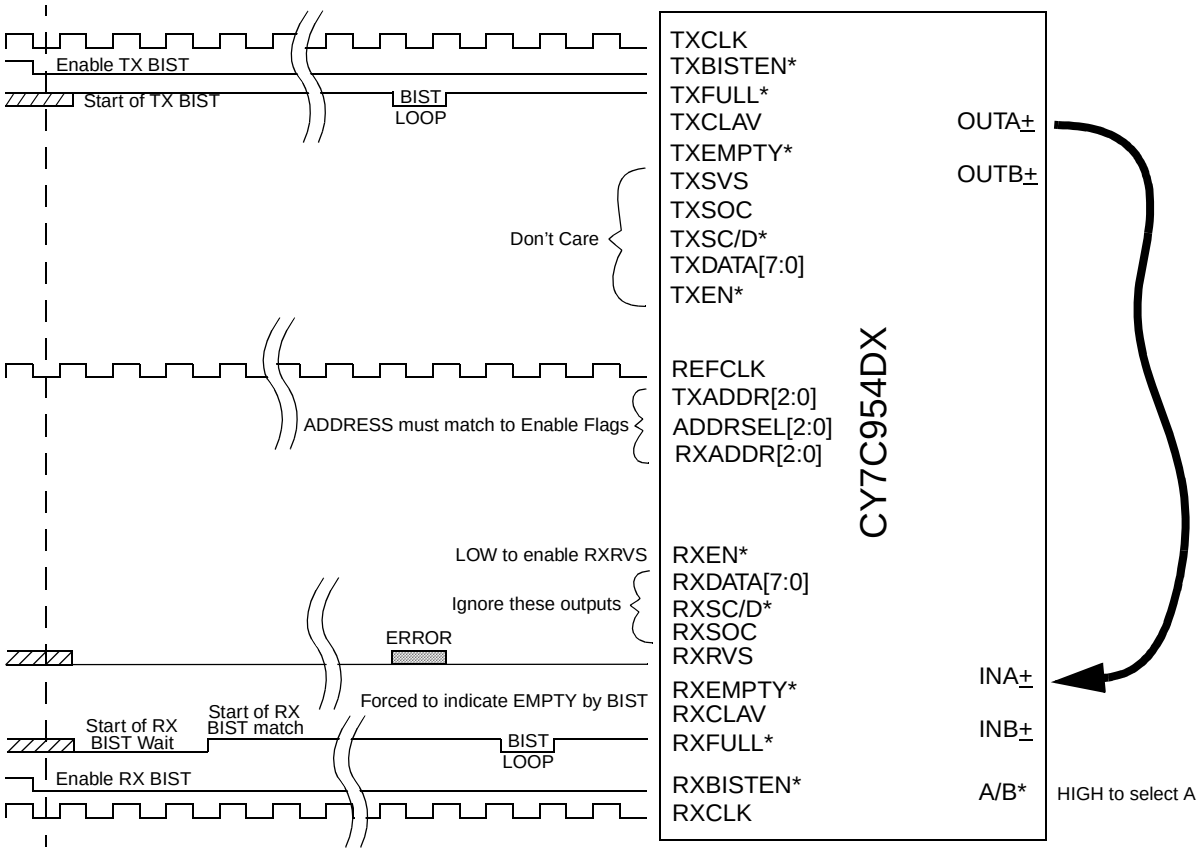


Figure 5. Built-In Self-Test Illustration

moval of stale data from the Transmit FIFO, it may also be reset during a BIST operation. The reset operation proceeds as documented, with the exception of the information presented on the TXEMPTY* FIFO status flag. Since this flag is used to present BIST loop status, it will reflect the state of the transmit BIST loop status until TXBISTEN* is no longer recognized internally. The completion of the reset operation may still be monitored through the TXFULL* FIFO status flag.

The TXEMPTY* flag, when used for transmit BIST progress indication, continues to reflect the active HIGH or active LOW settings determined by the UTOPIA or Cascade timing model selected by the EXTFIFO input.; i.e., when configured for the Cascade timing model, the TXEMPTY and TXFULL FIFO flags are active HIGH, when configured for the UTOPIA timing model the TXEMPTY* and TXFULL* FIFO flags are active LOW. *Figure 5* uses the UTOPIA conventions for the illustration.

When TXBISTEN* is first recognized, the TXEMPTY* flag is clocked to a reset state, regardless of the addressed state of the Transmit FIFO (if TXADDR[2:0] matches ADDRSEL[2:0] or not), but is not driven out of the part unless the MATCH has been sampled TRUE. Following this, on each completed pass through the BIST loop, the TXEMPTY* flag is set for one TXCLK period.

The TXEMPTY* flag remains set until the interface is addressed and the state of TXEMPTY* has been observed. If the device is not addressed (if TXADDR[2:0] does not match ADDRSEL[2:0]), the flag remains set internally regardless of the number of TXCLK clock cycles that are processed. If the device status is not polled on a sufficiently regular basis, it possible for the host system to miss some of these BIST loop indications.

A pass through the loop is defined as that condition where the encoder generates the D0.0 state. Depending on the initial state of the BIST LFSR, the first pass through the loop may occur at substantially less than 511 character periods. Following the first pass, as long as TXBISTEN* remains LOW, all remaining passes are exactly 511 characters in length.

BIST Receive Path

The receive path operation in BIST is similar to that of the transmit path. When RXBISTEN* is recognized internally, all writes to the Receive FIFO are suspended. If the receive data state machine was in the middle of processing a multi-character sequence or other atomic operation (e.g., a start of cell marker and its associated data), the characters associated with the atomic operation are discarded and not written to the Receive FIFO.

Any data present in the Receive FIFO when RXBISTEN* is recognized remains in the FIFO but is NOT available for reading through the host parallel interface until the BIST operation is complete. This is because the error output indicator for receive BIST operations is the RXRVS pin, which is normally associated with the RXDATA bus. To prevent read operations while BIST is in operation, the RXEMPTY* and RXCLAV flags are forced to indicate an Empty condition. Once RXBISTEN* has been removed and recognized internally, the Receive FIFO status flags are updated to reflect the current content status of the Receive FIFO.

To allow removal of stale data from the Receive FIFO, it may be reset during a BIST operation. The reset operation proceeds as documented, with the exception that the RXEMPTY* and RXCLAV status flags already indicate an empty condition. The RXFULL* flag is used to present BIST progress. The active (asserted) state on RXFULL* (and RXEMPTY*) remain controlled by the present operating mode and interface timing model (UTOPIA or Cascade).

When RXBISTEN* has been recognized, RXFULL* becomes the receive BIST loop indicator. When RXBISTEN* is first recognized, the RXFULL* flag is clocked to a set state, regardless of the addressed state of the Receive FIFO (regardless of the match between TXADDR[2:0] and ADDRSEL[2:0]). Following this, RXFULL* remains set until the receiver detects the start of the BIST pattern. Then RXFULL* is deasserted for the duration of the BIST pattern, pulsing asserted for one RXCLK period on the last symbol of each BIST loop. If 14 of 28 consecutive characters are received in error, RXFULL* returns to the set state until the start of a BIST sequence is again detected.

Just like the BIST status flag on the transmit data path, the RXFULL* flag captures the asserted states, and keeps them until they are read. This means that if the status flag is not read on a regular basis, events may be lost.

The detection of errors is presented on the RXRVS output. Unlike the RXFULL* FIFO status flag, the active state of this outputs is not controlled by the EXTFIFO input. With the Receive FIFO enabled, this output will operate the same as the RXFULL* flag, with respect to preserving the detection state of an error until it is read. An error indication that occurs while the RXEN* is deasserted will be “remembered” until RXEN* is asserted.

Unlike the RXFULL* flag, which only needs the CY7C954DX to be addressed (RXADDR[2:0] matching ADDRSEL[2:0] sampled TRUE by RXCLK) to enable the RXFULL* three-state driver, and an RXCLK to “read” the flag, the RXRVS output requires a selection (assertion of RXEN* while addressed) to enable the RXDATA bus three-state drivers. The selection process is necessary to ensure that a multi-PHY implementation does not enable two drivers onto the RXRVS output at the same time.

Bus Interfacing

The parallel transmit and receive host interfaces to the CY7C954DX are configurable. The choices are UTOPIA or Cascade control modes.

Both modes have internal Transmit and Receive FIFOs which can be written or read at any rate up to the maximum 50-MHz clock rate of the FIFOs. Internal operations of the CY7C954DX do not use the external TXCLK or RXCLK, but instead make use of REFCLK for transmit path operations and a recovered character clock for receive path operations.

The UTOPIA timing model is based on the ATM Forum UTOPIA interface standards. This timing model is that of a FIFO with active LOW FIFO status flags and read/write enables.

The Cascade timing model is a modification of the UTOPIA configuration that changes the flags and FIFO read/write enables to active HIGH. This model is present primarily to allow depth expansion of the internal FIFO by direct coupling to external CY7C42x5 synchronous FIFOs. To allow this direct coupling, the cycle-to-cycle timing between the transmit and receive enables (TXEN* and RXEN*) are also modified to ensure correct data transfer.

These two configurations of bus operation and timing/control can all be used with or without external FIFOs. Depending on the specific mode selected, the amount of external hardware necessary to properly couple the CY7C954DX to state machines or external FIFOs is minimal in all cases, and may be zero if the proper configuration is selected.

With only minor exceptions, all configurations rely on the common UTOPIA concepts of addressing and selection to control the enabled/disabled state of the output drivers, and when data can be written to or read from the part.

UTOPIA Interface Background

The UTOPIA interface is defined by the ATM Forum as the bus interface between the ATM and PHY layer devices of an ATM system. This interface is defined as 8 bits or 16 bits wide, with the later reserved mainly for high-speed physical interfaces (PHYs) such as 622 Mbps OC-12. Due to the limited speed range of the CY7C954, only the 8-bit interface is implemented.

UTOPIA-1 was the original UTOPIA specification (created in 1993) which covers transport of:

- 155.52 Mbps (scrambled SONET/ OC-3)
- 155.52 Mbps (8B/10B block coded at 194.4 MBaud)
- 100 Mbps (4B/5B encoded TAXI)
- 44.736 Mbps (DS-3/T3)
- 51.84 Mbps (OC-1)

The UTOPIA-1 interface has a maximum clock rate of 25 MHz. All AC timing and pin descriptions are covered in the UTOPIA-1 Specification, Version 2.01.

UTOPIA-2 was created as an addendum to the UTOPIA-1 specification. In this revision, the parallel interface was extended to both 33 MHz and 50 MHz to accommodate PCI bus architectures in ATM designs. A method of addressing was added to allow up to thirty one devices (PHYs) to share a common host bus. Also, a description of a management interface was added (not supported by this device).

The CY7C954DX contains all pins necessary to support the UTOPIA-1 and up to seven UTOPIA-2 devices using the TXADDR[2:0], RXADDR[2:0], and ADDRSEL[2:0] pins. The full thirty-one device space can be accessed by use of an external address decoder (connected to one of the TXADDR[2:0] and RXADDR[2:0] pins). The maximum bus speed supports the full 50-MHz I/O rate for emerging high-performance systems.

UTOPIA Address Match and Selection

All actions on a UTOPIA-2 interface are controlled by the Address Match and selection states of the interface. These states control the read and write access to the Receive and Transmit FIFOs, access to the FIFO status flags, and reset of the Transmit and Receive FIFOs. The CY7C954DX supports the concept of an “address match” using a three-bit chip address input (ADDRSEL[2:0]) and a pair of data port address vectors (TXADDR[2:0] and RXADDR[2:0]).

Address Match and FIFO Flag Access

The CY7C954DX makes use of a three-bit Address Select vector, which is compared to a TXADDR and RXADDR input to generate address-match conditions. When these inputs match, as is required to implement an ATM address compare on both the TXADDR and RXADDR buses. This allows multiple CY7C954DX devices to share a common bus, with device output three-state controls being managed by either an address match condition (TXADDR[2:0] matches ADDRSEL[2:0]), or by a selection state.

The Transmit and Receive FIFO flag output drivers are enabled in any TXCLK, or RXCLK cycle following TXADDR[2:0] matching ADDRSEL[2:0] being sampled TRUE by the rising edge of the respective clock. The TXADDR[2:0] and RXADDR[2:0] inputs are sampled separately by the clocks for the transmit and receive interfaces, which allows these clocks to be both asynchronous to each other, and to operate at different clock rates. An example of both Transmit and Receive FIFO flag access is shown in *Figure 6*.

When the Transmit FIFO is enabled by the rising edge of TXCLK and TXADDR[2:0] matches ADDRSEL[2:0], the output drivers for the TXCLAV, TXFULL* and TXEMPTY* FIFO flags are enabled. When TXADDR[2:0] doesn’t match ADDRSEL[2:0] at the rising edge of TXCLK, these same output drivers are disabled.

Device Selection

The concept of selection is used to control the access to the transmit and receive parallel-data ports of the device. There are five primary types of selection:

- Transmit data selection
- Receive data selection
- Transmit FIFO Flag selection
- Receive FIFO Flag selection
- Continuous selection (for either or both transmit and receive interfaces)

In addition to these normal selection types, there are two additional sequences that are used to control the internal Transmit and Receive FIFOs reset operations:

- Transmit reset sequence
- Receive reset sequence

Of these selection types, the transmit data selection and transmit reset sequence states are mutually exclusive and cannot exist at the same time. The receive data selection and receive

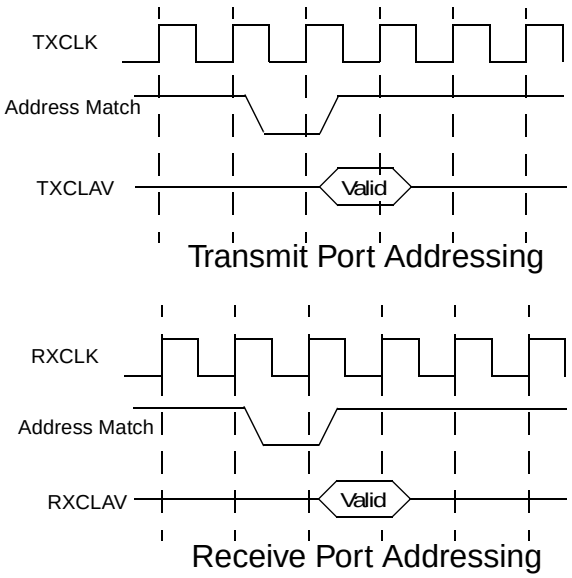


Figure 6. FIFO Flag Driver Enables

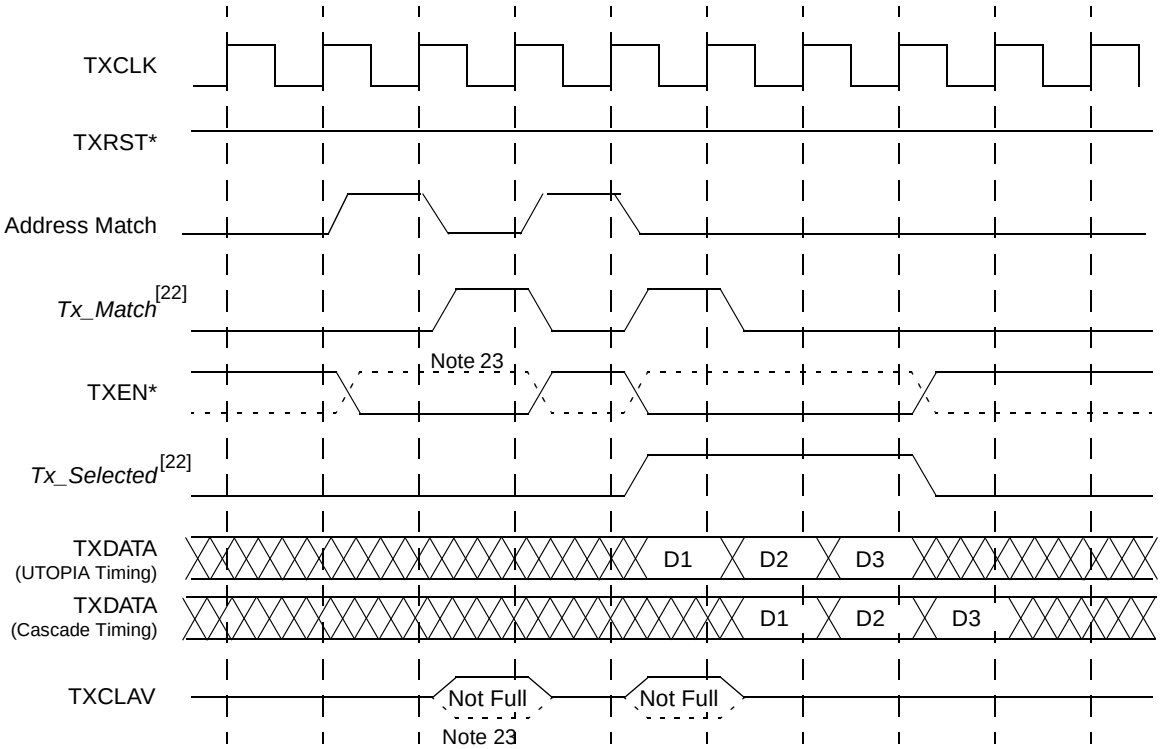


Figure 7. Transmit Selection

Notes:
22. Signals labeled in italics are internal to the CY7C954DX.
23. Signals shown as dotted lines represent the differences in timing and active state of signals when operated in Cascade Timing.

reset sequence states are also mutually exclusive and cannot exist at the same time. Either transmit state can exist at the same time as either receive state.

All normal forms of selection require that an Address Match condition must exist (TXADDR[2:0] or RXADDR[2:0] matching ADDRSEL[2:0]) either at the same time as the selection control signal being sampled asserted, or one or more clock cycles prior to the selection control signal being sampled asserted.

Transmit Timing and Control

UTOPIA Flag Selection

When TXADDR[2:0] matches ADDRSEL[2:0] and TXRST* is sampled HIGH by the rising edge of TXCLK, a Tx_Match condition is generated. This Tx_Match condition continues until TXADDR[2:0] doesn't match ADDRSEL[2:0] or TXRST* is sampled LOW at the rising edge of TXCLK. When a Tx_Match (or Tx_RstMatch) condition is present, the TXCLAV, TXEMPTY*, and TXFULL* output drivers are enabled. When a Tx_Match (or Tx_RstMatch) condition is not present, these same drivers are disabled (High-Z).

UTOPIA Data Selection

The selection state of the Transmit FIFO is entered when a Tx_Match condition is present, and TXEN* transitions from HIGH to LOW. Once selected, the Transmit FIFO remains selected until TXEN* is sampled HIGH by the rising edge of TXCLK. In the selected state, data present on the TXDATA inputs is captured and stored in the Transmit FIFO. This transmit interface selection process is shown in *Figure 7*.

Receive Timing and Control

UTOPIA Flag Selection

When RXADDR[2:0] matches ADDRSEL[2:0] and RXRST* is sampled HIGH by the rising edge of RXCLK input, an Rx_Match condition is generated. This Rx_Match condition continues until RXADDR[2:0] doesn't match ADDRSEL[2:0] or RXRST* is sampled LOW at the rising edge of RXCLK input. When an Rx_Match (or Rx_RstMatch) condition is present, the RXCLAV, RXEMPTY* and RXFULL* output drivers are enabled. When an Rx_Match (or Rx_RstMatch) condition is not present, these same drivers are disabled (High-Z).

The UTOPIA Data selection state of the Receive FIFO is entered when an Rx_Match condition is present, and RXEN* transitions from HIGH to LOW. Once selected, the Receive FIFO remains selected until RXEN* is sampled HIGH by the rising edge of RXCLK input. The selected state initiates a read cycle from the Receive FIFO, and enables the Receive FIFO data onto the RXDATA bus. This receive interface selection process is shown in *Figure 8*.

Continuous Selection

Continuous Selection is a specialized form of selection which does not require sequenced assertion of Address Match and TXEN* or RXEN* to select the device for data transfers. In this Continuous Selection mode, the TXADDR[2:0] or RXADDR[2:0] matches ADDRSEL[2:0] and associated TXEN* or RXEN* en-

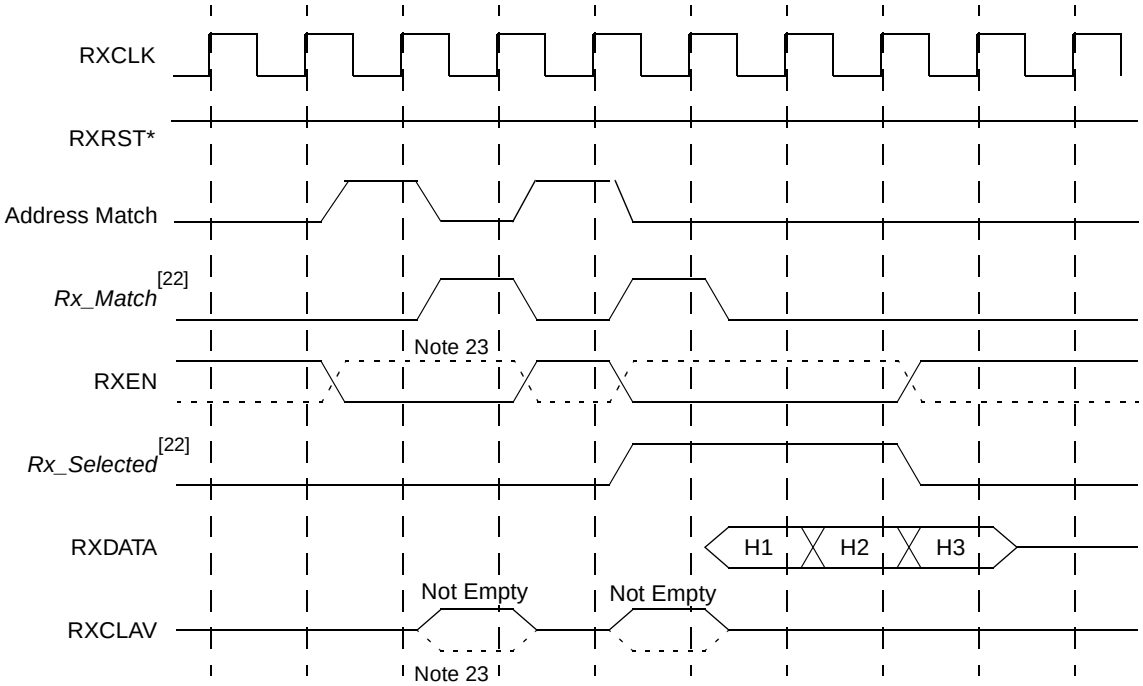


Figure 8. Receive Selection

able signal must be deasserted for one clock cycle when the device is powered up. So long as these signals remain asserted, the device remains selected and data is accepted and presented on every clock cycle.

Note: The use of continuous selection makes it impossible to reset the respective internal FIFOs.

FIFO Reset Address Match

When TXADDR[2:0] matches ADDRSEL[2:0] and TXRST* are both LOW, and this condition is sampled by eight consecutive rising edges of TXCLK, a Tx_RstMatch condition is generated. This Tx_RstMatch condition continues until TXADDR[2:0] doesn't match ADDRSEL[2:0] or TXRST* is sampled HIGH by the rising edge of TXCLK. When a Tx_RstMatch (or Tx_Match) condition is present, the TXEMPTY*, and TXFULL* output drivers are enabled (just as in a normal Tx_Match condition). When a Tx_RstMatch (or Tx_Match) condition is not present, these same drivers are disabled (HIGH-Z). The Transmit FIFO reset Address Match is shown in *Figure 9*. Note that although TXRST* remains LOW for more than one clock cycle, the Tx_RstMatch does not because the TXADDR[2:0] doesn't match ADDRSEL[2:0].

When RXADDR[2:0] matches ADDRSEL[2:0] and RXRST* is LOW, and this condition is sampled by eight consecutive rising edges of RXCLK, an Rx_RstMatch condition is generated. This Rx_RstMatch condition continues until RXADDR[2:0] doesn't match ADDRSEL[2:0] or RXRST* is sampled HIGH, at the rising edge of RXCLK. When an Rx_RstMatch (or Rx_Match) condition is present, the RXEMPTY* and RXFULL* output drivers are enabled. When an Rx_RstMatch (or Rx_Match) condition is not present, these same drivers are disabled (High-Z). The Receive FIFO reset Address Match is shown in *Figure 10*. Note that while the FIFO flags remain asserted for more than one clock cycle, this is due to an Rx_Match condition, not a continuation of the Rx_RstMatch.

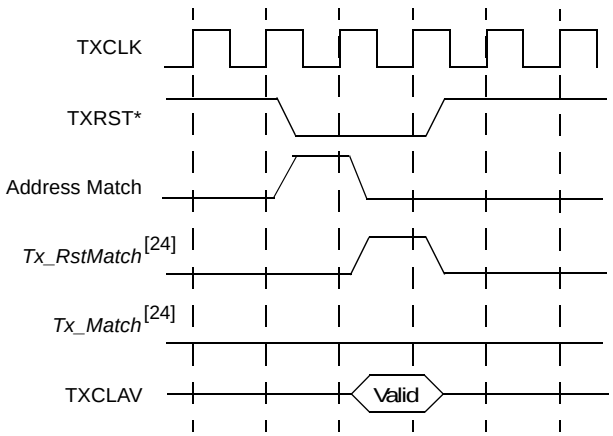


Figure 9. Transmit FIFO Reset Address Match

Note:
24. Signal names listed in italics are internal signals, shown for reference only.

FIFO Reset Sequence

On power-up, the Transmitter and Receiver FIFOs may contain random data. The Transmitter FIFO will empty automatically as the Transmitter sends the random data within the first 256 character times after power is applied. The Receiver FIFO will retain any random data stored in it at power-up, and will accumulate all the random data being received from any attached transmitter as it is powered up. Most of the incoming random data may be discarded based on the Receiver discard policy shown in *Figure 5*. This random received data can be "flushed" by reading it, or the Receive FIFO can be "reset" to remove the unwanted data.

The Transmit FIFO and Receive FIFO are reset when the Tx_RstMatch or Rx_RstMatch condition remains present for

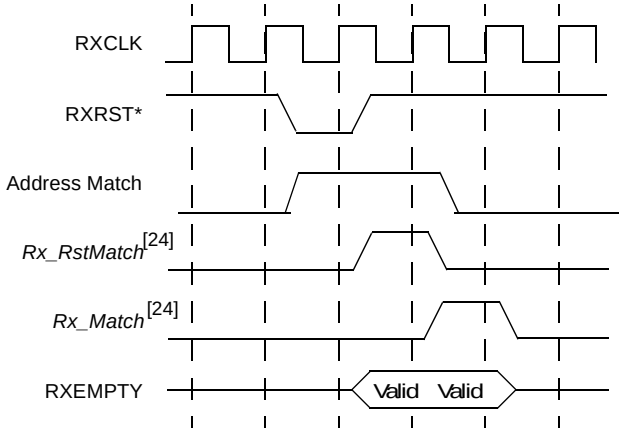


Figure 10. Receive FIFO Reset Address Match

eight consecutive clock cycles. Any disruption of the reset sequence prior to reaching the eight cycle count, either by removal of the TXADDR[2:0] or RXADDR[2:0] matches with ADDRSEL[2:0] or the respective TXRST* or RXRST*, or assertion of the associated TXEN* or RXEN*, terminates the sequence and does not reset the FIFO. Because the TXADDR[2:0] or RXADDR[2:0] matching ADDRSEL[2:0] must remain asserted during the reset sequence, the addressed FIFO flags remain driven during the entire sequence.

The FIFO Reset sequence will remove any pre-existing address match condition and TXEN* or RXEN* will need to be deasserted for one clock cycle before address match can be re-established.

Transmit FIFO Reset Sequence

The Transmit FIFO reset sequence is started when TXRST* is sampled LOW and TXADDR[2:0] matches ADDRSEL[2:0] on the rising edge of TXCLK. However, if TXEN* is asserted, the reset sequence is inhibited until it is removed (TXEN* is sampled HIGH for UTOPIA timing or LOW for Cascade timing). Because a Tx_RstMatch condition is present, the Transmit FIFO flags are asserted and can be used to track the status of any Transmit FIFO reset in progress. Once the reset sequence has reached its maximum count, the Transmit FIFO flags are forced to indicate a FULL* condition (TXCLAV is deasserted, and TXFULL* is asserted). This indicates that the Transmit FIFO reset has been recognized by the Transmit Control State Machine and that a reset has been started.

Note: The FIFO Full state forced by the reset operation is different from a Full state caused by normal FIFO data writes. For normal FIFO write operations, when Full is first asserted, the Transmit FIFO must still accept up to four additional writes of data. When a Full state is asserted due to a Transmit FIFO reset operation, the FIFO will not accept any additional data.

The Transmit FIFO reset does not complete until the external reset condition is removed. This can be removed by deassertion of either TXRST* or the address match. If the address match is deasserted to remove the reset condition, the Transmit FIFO flag's drivers are disabled, and the Transmit FIFO must be addressed at a later time to validate completion of the Transmit FIFO reset. If TXRST* is deasserted (HIGH) to remove the reset condition, the Tx_RstMatch is changed to a Tx_Match, and the Transmit FIFO status flags remain driven.

The Transmit FIFO reset operation is complete when the Transmit FIFO flags indicate an Empty state (TXEMPTY* and TXCLAV are asserted and TXFULL* is deasserted). A valid Transmit FIFO reset sequence is shown in *Figure 11*.

Here the TXADDR[2:0] matches ADDRSEL[2:0] and TXRST* is asserted (LOW) at the same time. When these signals are both sampled LOW by TXCLK, a Tx_RstMatch condition is present. With TXEN* deasserted (HIGH), the Transmit FIFO is not selected for data transfers. This Tx_RstMatch condition remains for eight TXCLK cycles to generate the Tx_FIFO_Reset. Following this the TXFULL* FIFO status flag is asserted to indicate that the Transmit FIFO reset sequence has completed and that a Transmit FIFO reset is in progress.

When the TXRST* signal is deasserted (HIGH), TXADDR[2:0] still matches ADDRSEL[2:0] to allow the FIFO status flags to be driven. This allows the completion of the reset operation to be monitored. To allow better multi-tasking on multi-PHY implementations, it is possible to deassert the address match as soon as the Full state is indicated. The FIFO reset operation will complete and the Empty state (indicating completion of the reset operation) can be detected during a separate polling operation.

For those links implemented with a single PHY, it is possible to hard wire TXADDR[2:0] to match ADDRSEL[2:0] and still perform normal accesses and reset operations. This is shown in *Figure 12*. In a single-PHY implementation with address match always TRUE, a Transmit FIFO reset can never be initiated with TXEN* asserted at the same time as TXRST*. Since the address match is always TRUE, any assertion of TXEN* causes the Transmit FIFO to be selected, preventing the reset counter from advancing.

Figure 13 shows a sequence of input signals which will not produce a FIFO reset. In this case TXEN* was asserted to select the a Transmit FIFO for data transfers. Because TXEN* remains active, the assertion of a TXADDR[2:0] matching ADDRSEL[2:0] and TXRST* does not initiate a reset operation. This is shown by the TXFULL* flag remaining HIGH (deasserted) following what would be the normal expiration of the eight-state reset counter.

Receive FIFO Reset Sequence

The Receive FIFO reset sequence operates (for the most part) the same as the Transmit FIFO reset sequence. The same requirements exist for the assertion state of RXRST* and selection of the interface. A sample Receive FIFO reset sequence is shown in *Figure 14*. Upon recognition of a Receive FIFO reset, the Receive FIFO flags are forced to indicate an Empty state to prohibit additional reads from the FIFO. Unlike the Transmit FIFO, where the internal completion of the reset operation is shown by first going Full and later going Empty when the internal reset is complete, there is no secondary indication of the completion of the internal reset of the Receive FIFO. The Receive FIFO is usable as soon as new data is placed into it by the Receive Control State Machine.

FIFO Reset and Continuous Selection

When configured for continuous selection (TXADDR[2:0] matching ADDRSEL[2:0] asserted with TXEN* always enabled, or RXADDR[2:0] matching ADDRSEL[2:0] asserted with RXEN* always enabled), it is not possible to reset the Transmit and Receive FIFOs.

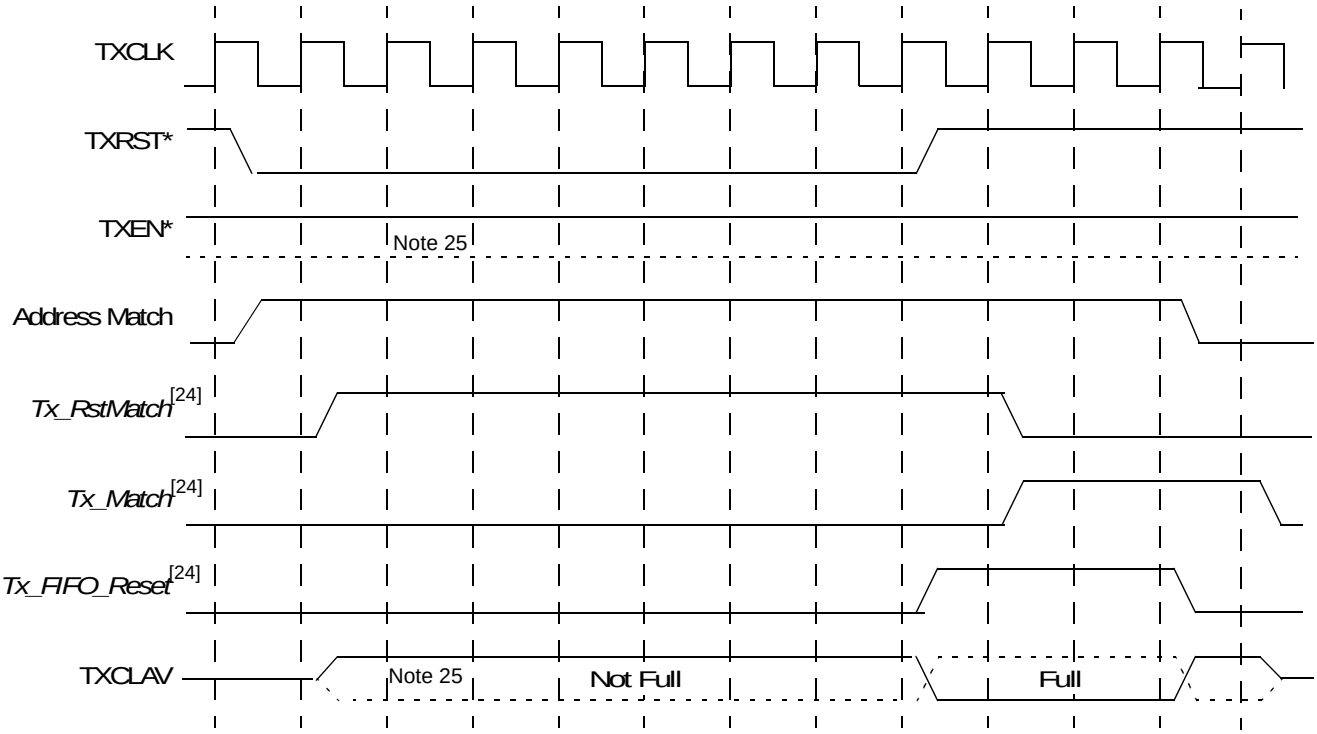


Figure 11. Transmit FIFO Reset Sequence

Note:
25. Signals shown as dotted lines indicate timing and levels when configured for external FIFOs (EXTFIFO is HIGH).

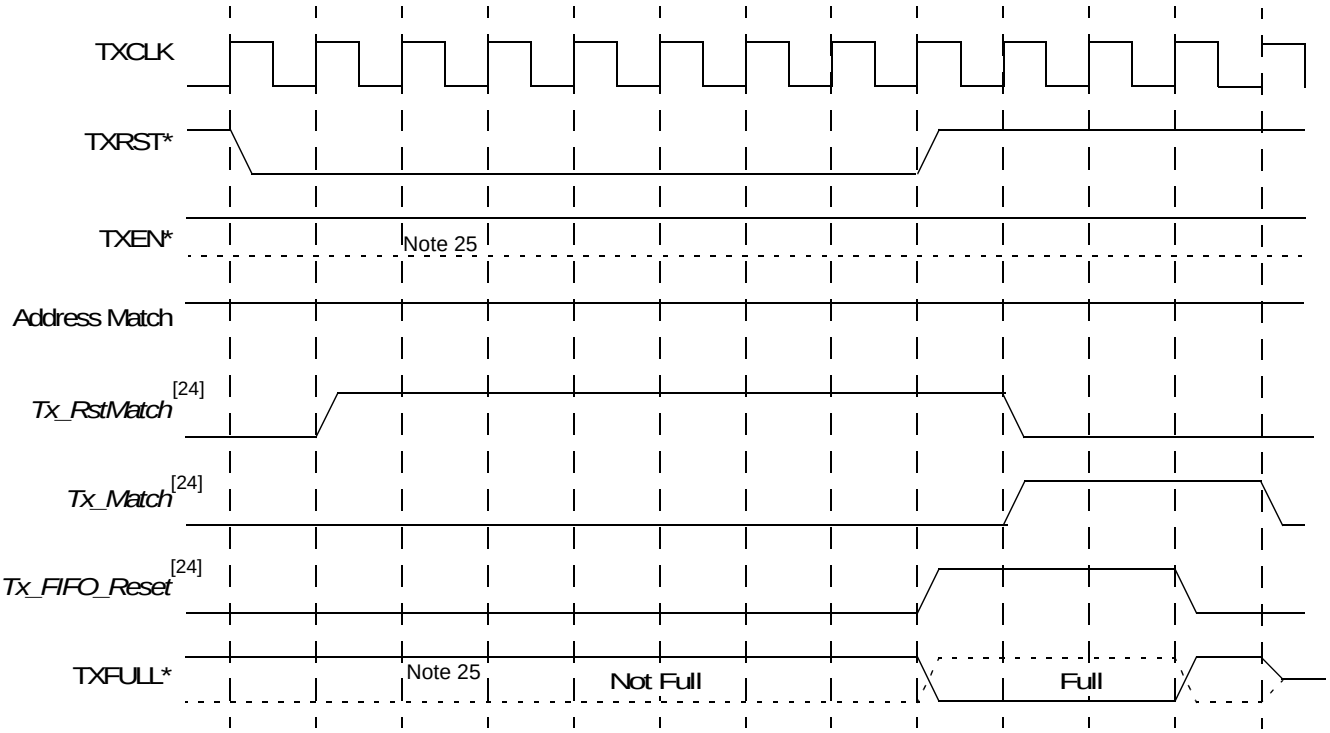


Figure 12. Transmit FIFO Reset Sequence with Constant Address Match

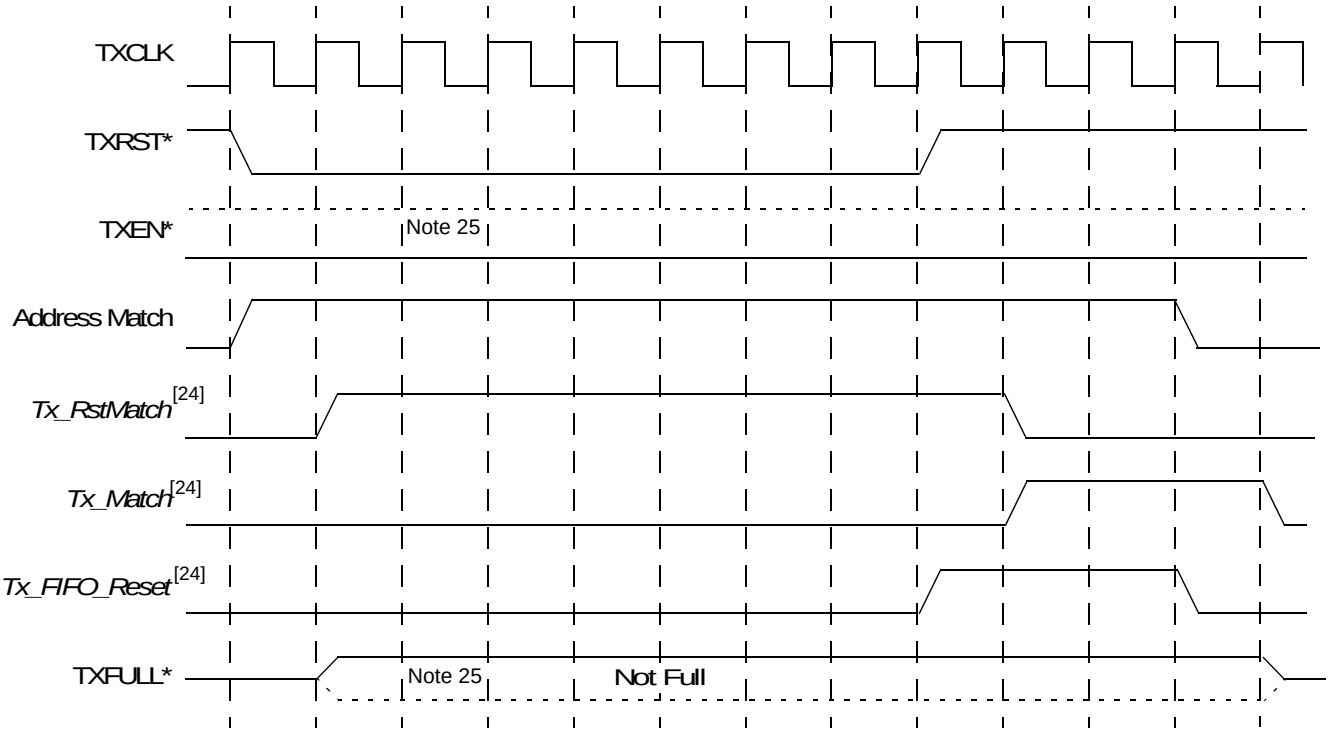


Figure 13. Invalid Transmit FIFO Reset Sequence with TXEN* Asserted

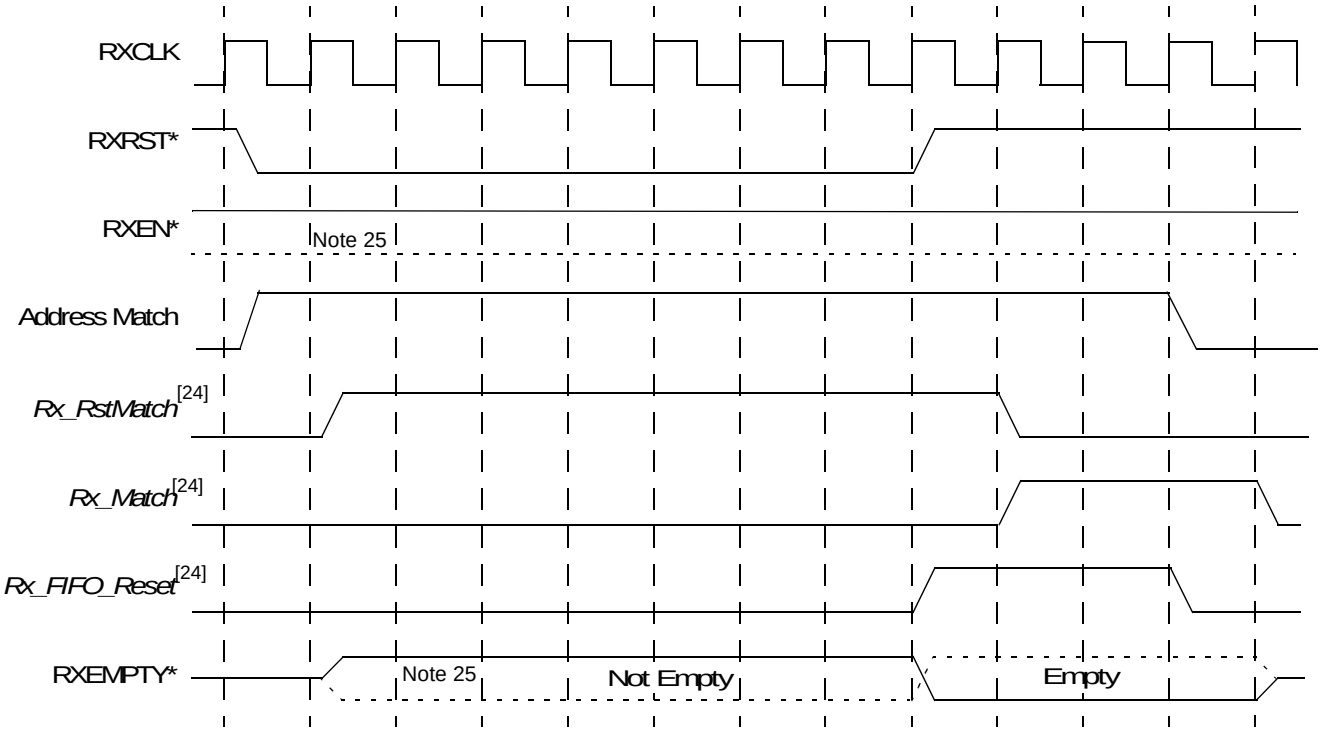


Figure 14. Receive FIFO Reset Sequence

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, 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:	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 the SC/D* pin is LOW) or a Special Character (c is set to K, and the SC/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 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 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 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 6* 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 6. 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	000101	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 7* shows an example of this behavior.

Table 7. 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 8. Valid Data Characters (TXSC/D* = LOW, RXSC/D* = LOW)

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

Table 8. Valid Data Characters (TXSC/D* = LOW, RXSC/D* = LOW) (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
D0.2	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
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 8. Valid Data Characters (TXSC/D* = LOW, RXSC/D* = LOW) (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

Table 8. Valid Data Characters (TXSC/D* = LOW, RXSC/D* = LOW) (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
D0.6	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
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

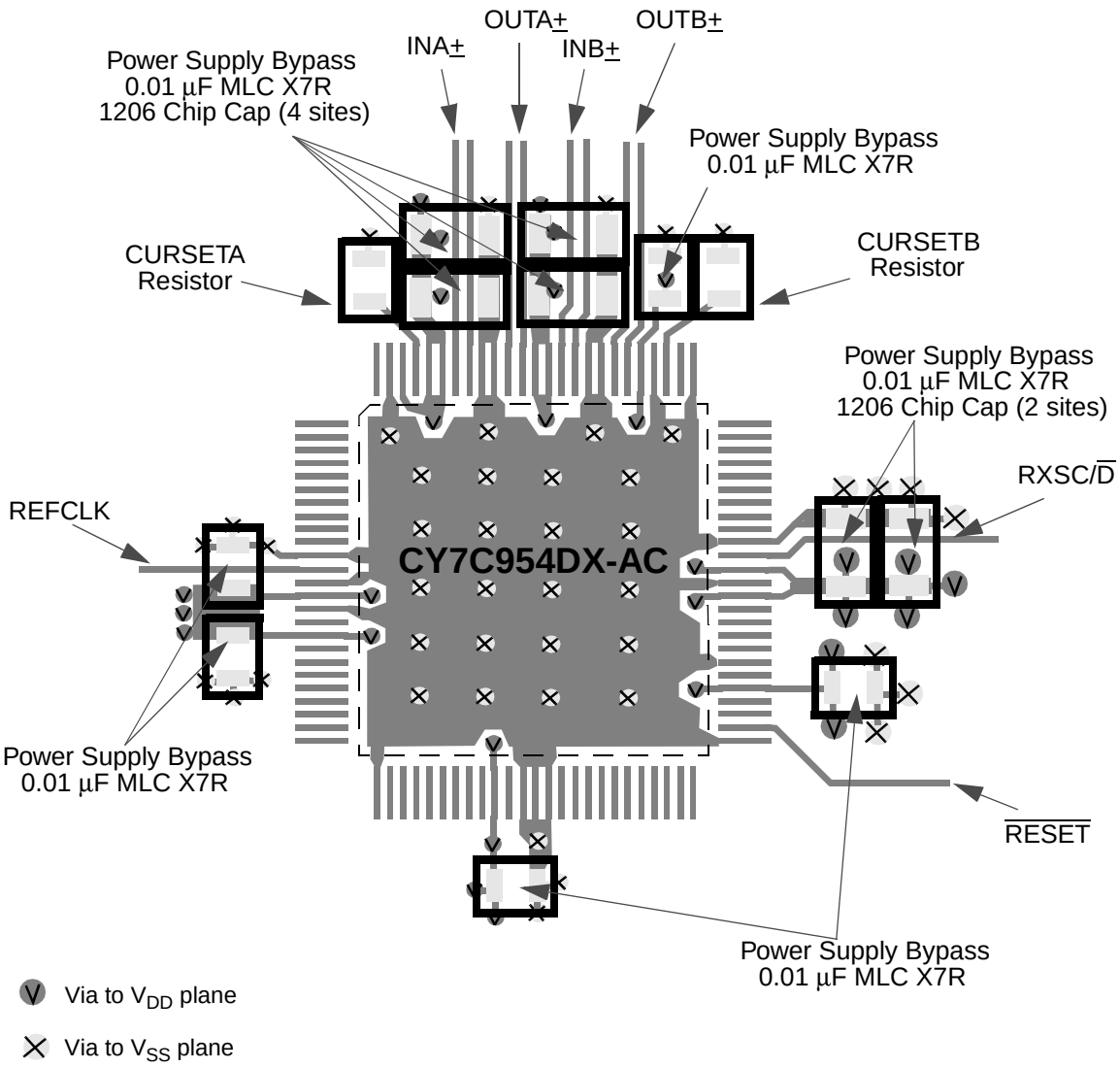
Table 9. Valid Special Character Codes and Sequences (TXSC/D* = HIGH or RXSC/D* = HIGH)[26, 27]

S.C. Byte Name	S.C. Code Name		Bits		Current RD–		Current RD+	
			HGF	EDCBA	abcdei	fghj	abcdei	fghj
K28.0	C0.0 [28]	(C00)	000	00000	001111	0100	110000	1011
K28.1	C1.0 [29]	(C01)	000	00001	001111	1001	110000	0110
K28.2	C2.0 [29]	(C02)	000	00010	001111	0101	110000	1010
K28.3	C3.0 [28]	(C03)	000	00011	001111	0011	110000	1100
K28.4	C4.0 [29]	(C04)	000	00100	001111	0010	110000	1101
K28.5	C5.0 [29, 30]	(C05)	000	00101	001111	1010	110000	0101
K28.6	C6.0 [29]	(C06)	000	00110	001111	0110	110000	1001
K28.7	C7.0 [29, 31]	(C07)	000	00111	001111	1000	110000	0111
K23.7	C8.0 [28]	(C08)	000	01000	111010	1000	000101	0111
K27.7	C9.0 [28]	(C09)	000	01001	110110	1000	001001	0111
K29.7	C10.0 [28]	(C0A)	000	01010	101110	1000	010001	0111
K30.7	C11.0	(C0B)	000	01011	011110	1000	100001	0111
					End of Frame Sequence			
EOFxx	C2.1	(C22)	001	00010	–K28.5,Dn.xxx0[32]		+K28.5,Dn.xxx1[32]	
					Code Rule Violation and SVS Tx Pattern			
Exception	C0.7 [31]	(CE0)	111	00000	100111	1000[33]	011000	0111[33]
–K28.5	C1.7	(CE1)	111	00001	001111	1010[34]	001111	1010[34]
+K28.5	C2.7	(CE2)	111	00010	110000	0101[35]	110000	0101[35]
					Running Disparity Violation Pattern			
Exception	C4.7	(CE4)	111	00100	110111	0101[36]	001000	1010[36]

Notes:

26. All codes not shown are reserved.
27. 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 00h and FFh).
28. When received, these characters are discarded, but cause the RXSOC to be set HIGH, along with various combinations of RXRVS and RXSC/D*. They can be explicitly sent by a Transmitter using the TXSC/D* and the appropriate data pattern, or by setting TXSOC and various combinations of TXSVS and TXSC/D*.
29. 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.
30. 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 and is discarded prior to loading into the Receive FIFO.
31. 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, and alias K28.5 sync character is created. These sequences can cause erroneous framing and should be avoided while RFEN is HIGH.
32. 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 special data sequences wherein the second data byte is determined by the Current RD.
For example, to send a Fibre-Channel “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 (which is discarded), C1.7, or C2.7 (both of which are accompanied by an RXRVS error indication), and the subsequent bytes are decoded as data.
33. 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.
34. 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 discarded or decoded as C2.7.
35. 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 discarded or decoded as C1.7.
36. 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.

Printed Circuit Board Layout Suggestions



This is a typical printed circuit board layout showing example placement of power supply bypass components and other components mounted on the same side as the CY7C954DX.

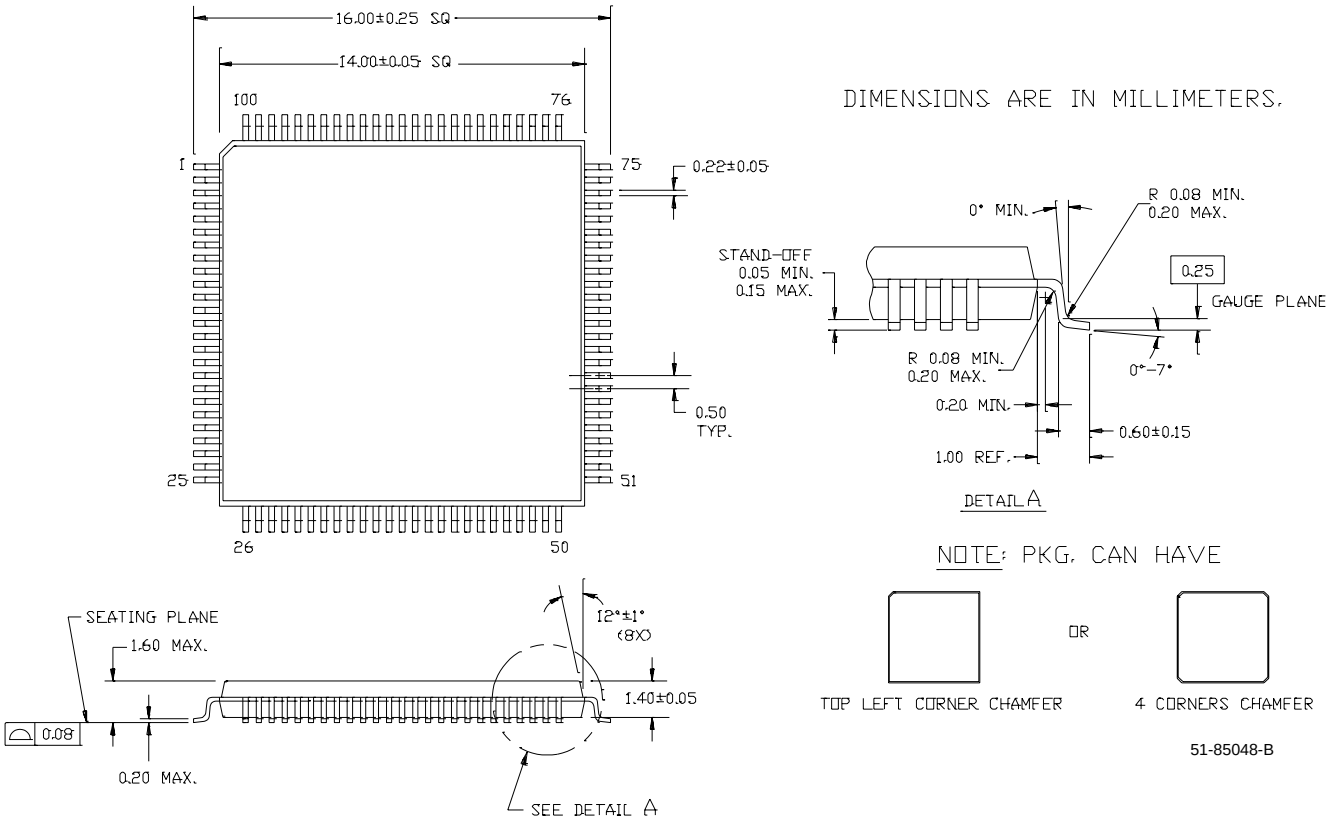
Other layouts, including cases with components mounted on the reverse side would work as well.

Ordering Information

Ordering Code	Package Name	Package Type	Operating Range
CY7C954DX-AC	A100	100-Lead Thin Quad Flat Pack	Commercial

Package Diagram

100-Pin Thin Plastic Quad Flat Pack (TQFP) A100



Document Title:CY7C954DX ATM HOTLink® Transceiver Document Number: 38-02007				
REV.	**ECN NO.**	**Issue Date**	**Orig. of Change**	**Description of Change**
**	105845	03/26/01	SZV	Change from Sped number: 38-00812 to 38-02007