●新特器件应用

高速串行数据通信接收芯片CY7B933的原理及应用

海军工程大学电子工程学院　徐勤建　刘望锁　姚直象

Principle and Application of CY7B933 HOTLink Receiver for High Speed Serial Communication

Xu Qinjian　Liu Wangsuo　Yao Zhixiang

摘要：介绍了CYPRESS半导体公司推出的一种用于点对点之间的高速串行数据通信接收芯片CY7B933的原理及应用。详细说明了其管脚功能、内部组成、工作原理及工作方式。给出了一个由CY7B933构成的实际接收电路及设计方法。

关键词：串行数据通信；　CY7B933；　FIFO；　IDT7200；　基带传输；　差分PECL输出

分类号：TN914.3　　**文献标识码**：B　　**文章编号**：1006-6977(2002)01-0032-03

1　概述

CY7B933是CYPRESS半导体公司推出的一种用于点对点之间高速串行数据通信的接收芯片。与其配套的发送芯片为CY7B923。CY7B933接收芯片的内部电路主要包括两对PECL串行输入接口、PECL-TTL电平转换器、时钟同步器、成帧器、移位器、译码寄存器、译码器、输出寄存器和测试逻辑等几部分。在与CY7B923配套时，CY7B933也有三种传输速率的器件可供选择：其中标准系列的器件有CY7B933-JC、CY7B933-JI、CY7B933-SC和CY7B933-MB等四种型号，它们的传输速率为160~330Mbps；高速系列的器件有CY7B933-400JC和CY7B933-400JI，其传输速率为160~400Mbps；低速系列的器件有：CY7B933—155JC、CY7B933—155JI等，其传输速率为150~160Mbps。CY7B933的封装形式有28脚SOIC/PLCC/LCC等三种，采用0.8MBiCOMS工艺生产和单+5V电源供电，功率为650mW。

2　引脚功能及内部结构

CY7B933的引脚排列如图1所示（SOIC封装），表1所列为其引脚功能说明。图2所示是其内部结构框图，下面介绍各部分的主要功能。

●串行数据输入端口

CY7B933的两对差分线接收端可作为串行数据的输入，选用INA+还是INB+取决于A/$\overline{B}$输入端的电平；当A/$\overline{B}$为高电平时，选用INA±；当A/$\overline{B}$为低电平时，选用INB±。

●PECLL—TLL转换器

INB(INB+)输入和SI(INB-)输入的功能是由SO输出脚上的连接方式决定的，若不需要PECLL/TLL转换，则可将SO输出脚接到VCC。利用内置检测电路可以检测到这种连接方式，并可将以上两个输入脚置为INB±输入（即差分接收串行数据输入）；若需要进行PECL/TLL转换，则应在SO输出端连接一个TLL负载，此时的INB+输入即为INB(单端ECL—100K型串行数据输入)。INB-则可作为SI(单端ECK-100K型状态)输入，而SO即是SI转换成TLL电平时的输出信号。

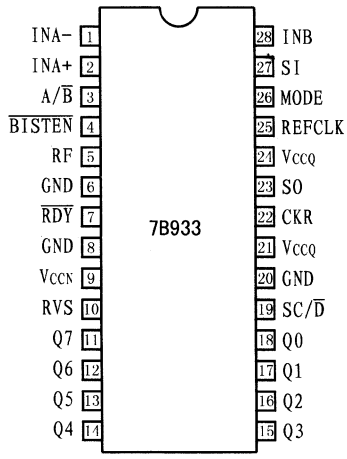


图1　CY7B933的引脚排列

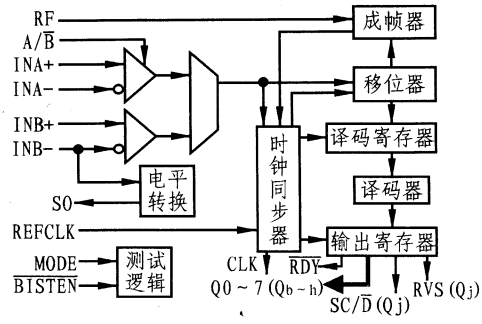


图2　CY7B933内部结构框图

●时钟同步器

时钟同步器由内部锁相环来保持与接收到的位流频率的一致，同时提供内部移、解码时钟。

●成帧器

成帧器可用于检查输入的位流和寻找字节的边界，从而实现帧同步（字符同步）。成帧器中的组合逻辑滤波器可用于寻找 X3.230 协议中定义的特殊字符（K28.5），一旦该字符找到，时钟同步单元中的位计数器将被同步复位，以开始同步接收数据，并将串行数据位流准确地重组成字符帧。

●移位器

移位器的作用是在位时钟控制下接收串行输入的位流，同时将其送到成帧器，当接收满一帧后（10位数据）便将其送到译码寄存器。

●译码寄存器

译码寄存器在接收到来自移位器中的一帧数据后将其送到译码器，但该数据在译码器译码至输出期间仍将保持在译码寄存器中。

●译码器

译码器将接收到的数据按 X3.230 协议定义的码字符重新转换成“原始数据”，然后再送到输出寄存器中。

●输出寄存器

输出寄存器用于保持译码后恢复的数据（Q0－Q7）、$\overline{SC/D}$ 和 RVS），以便在相应的输出脚输出。在 BIST 方式下，这个寄存器还可作为线性反馈移位寄

表 1　CY7B933 的引脚功能表

名称	输入/输出	引　　脚　　功　　能
Q0－Q7(Q_{b-h})	TTL 电平输出	并行数据输出。Q0－Q7 输出脚的值是刚接收到的数据。这几个脚的数据是随着 CKR 的变化而同步变化
SC/$\overline{D}$(Qa)	TLL 电平输出	特殊字符/数据指示。SC/$\overline{D}$ 表明接收的数据类型：SC/$\overline{D}$ 是高电平，表明接收的内容是控制码（特殊字符）；SC/$\overline{D}$ 是低电平，表明接收到的是数据字符
RVS(Q_J)	TLL 电平输出	接收违例字符指示。当 RVS 为高点平时，表明在接收数据流中检测到违例字符；若为低电平，表明检测到错误码；在 BIST 方式时，若 RVS 为低电平，表明由发送器、接收器以及链接线路等构成的整个系统工作正常
$\overline{RDY}$	TLL 电平输出	输出数据准备好。上的负脉冲表明已经接收到一个新数据，并且已经准备好传送。在接收到空字符时，$\overline{RDY}$不出现负脉冲。在 BIST 方式下，$\overline{RDY}$只是在每次测试循环中的最后一个字符出现时为高电平，其余时间保持低电平
CKR	TLL 输出	读数据时钟。它是一个读字节数据时钟信号，其频率和相位与输入的串行数据流保持一致
A/$\overline{B}$	PECL 电平输入	串行数据输入端选择。它是 PECL100K 类型（接 5 伏参考电压）的输入端，用于输入端的选择。若 A/$\overline{B}$ 为高电平，INA 端连接到移位寄存器，INA 上的输入信号有效并进行译码；若 A/$\overline{B}$ 为低电平，选 INB 端
INA±	差分输入	串行数据输入端（INA±）
INB(INB+)	PECL 电平输入（差分输入）	串行数据输入 B 端。这个脚即可以作为单端方式的 PECL 接收端（INB），也可以作为差分对的正半端（INB+）
SI	PECL 电平输入（差分输入）	状态输入。此脚即可以作为单端 PECL 状态监控输入（SI），也可以作为 INB 差分的负半端（INB－）
SO	TLL 输出	状态输出。SO 是将 SI 转换为 TTL 信号的输出端
RF	TLL 输入	帧同步允许
REFCLK	TLL 输入	参考时钟。RETCLK 是作为时钟/数据同步锁相环的一个参考时钟
MODE	3—电平输入	译码方式选择。在 MODE 脚上的电平可决定采用的译码方式；当该脚接地时，采用 8B/10B 译码方式；接 VCC 时，采用旁通方式；当其悬空时，内部上接电阻将该脚拉成 VCC/2 电平；此时为工厂测试方式
$\overline{BISTEN}$	TLL 输入	内置自测试允许。若要 ISTEN 置为低电平，则设置为内置测试方式
V_{CCN}		输出驱动电路供电电源
V_{CCQ}		内部电路供电电源
GND		地

存器,以产生512字节的伪随机码。

●测试逻辑

测试逻辑包括BIST工作方式的初始化及控制逻辑，以及用于测试方式时的时钟分配多工器和译码器控制逻辑等。

3　CY7B933的工作原理及操作方式

3.1 工作原理

CY7B933接收器用于接收来自差分串行输入端的串行位流，它使用内置的集成锁相环（PLL）同步时钟产生器来恢复用于数据重构所需的位同步时钟。成帧器在寻找到串行位流的字节边界后即可产生帧同步时钟。然后输入位流数据即可在移位器中实现串/并转换,同时再送到译码器中译码并检查传送错误。最后将译码后的字节数据在帧同步时钟控制下送到输出寄存器并由并行输出脚输出。

3.2 操作方式

CY7B933具有8B/10B编码方式、旁通方式、内置自测试方式和测试方式四种接收操作方式。

a. 8B/10B编码方式

在该编码方式下,串行输入的数据经译码后变成8位数据位，其中包括类型指示位及解码错误指示位。若接收后放在译码寄存器中的数据帧能够在有效数据字符表中找到，则其内容被译为正常的数据信息，并使SC/$\overline{D}$输出脚为低电平，假如在有效的特殊字符码及码系列表中找到,则译为“控制”或“协议”信息,并且将SC/$\overline{D}$变高。若检测到非法字符,则RVS将变高。

b. 旁通方式

在旁通方式下，串行输入数据不需由内部译码器对其译码,而是直接从译码寄存器到10位输出寄存器（Qa－j），然后由外加的译码电路来对其译码,译码方式由设计者确定,这种方式一般不用。

c. 内置自测试方式(BIST)

内置自测试功能有以下几种：

(1) 设置$\overline{\text{BISTEN}}$为低，允许自测试产生电路工作。若RDY变低,表明初始化码已找到。

(2) 监控RVS并检查该脚是否为高电平。若为高电平,则表明测试到失配样本(数据帧)。

若系统工作正常，则在每次测试循环中使$\overline{\text{RDY}}$出现一次正脉冲。可以对该脉冲进行计数以监控测试过程。同时Q0～Q7和SC/$\overline{D}$等脚也将出现预期的样本值,这一点对系统调试是很有用的。

(3) 当测试完成时，设置BISTEN为高电平以恢复正常工作。

BIST方式主要用于检查发送器。由链接线路和接收器构成的整个系统一般不用借助外加的信号及电路，也不用对整个电路作任何改动即可对整个系统进行严格的测试。

d. 测试方式

当Mode脚悬空时,接收器处于芯片测试方式。这种方式一般作为工厂进行芯片测试或用户新购大批器件进行测试时使用。

4　CY7B933构成的接收电路设计

由CY7B933构成的接收电路如图3所示，该电路主要由CY7B933接收芯片、IDT7200（FIFO）芯片和阻抗匹配电路等组成。FIFO芯片（IDT7200）的写信号FIFOW由CY7B933的$\overline{\text{RDY}}$信号提供。利用开关K可将CY7B933的工作设置在8B/10B译码方式

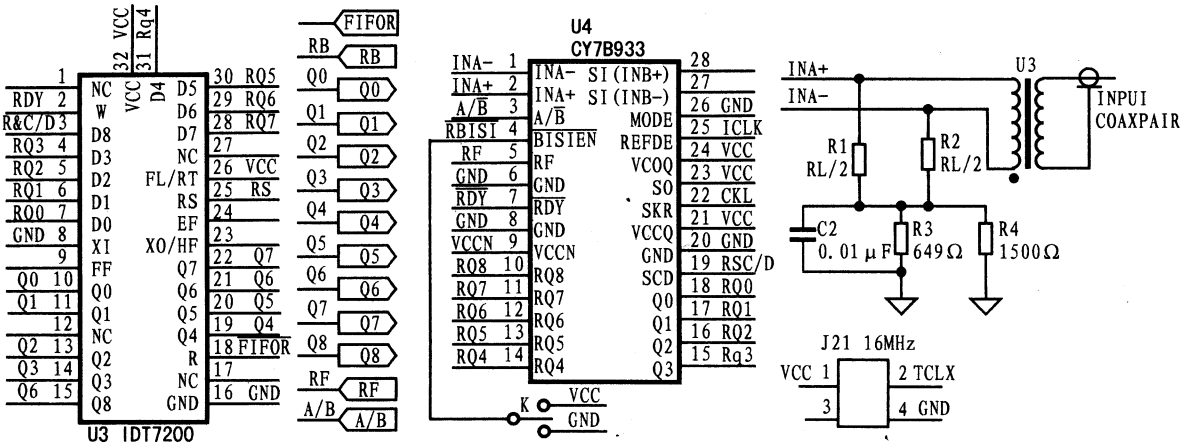


图3　CY7B933构成的接收电路

或内置自测试方式。

当 BISTEN = 0 时，CY7B933 工作在内置自测试方式。此时如果发送芯片 CY7B923 也工作在内置自测试方式，则可以通过对 CY7B923 的 RP 和 CY7B933 的 $\overline{RDY}$ 脚进行测试比较来判断整个系统是否正常工作。若这两个信号同步出现且相位相反（$\overline{RP}$ 为负脉冲，$\overline{RDY}$ 为正脉冲），则表明整个系统的发送电路、接收电路及其链接线路工作正常。

当 BISTEN = 1 时，CY7B933 工作在 8B/ 10 译码方式。当 CY7B933 工作在这种方式时，接收并译码后的数据可由 CY7B933 提供的 $\overline{RDY}$ 信号来写入 FIFO 芯片 IDT7200 中。用户可通过 FIFO 的另一端来读取该信号。注意：如按上述电路连接，则此时读取的数据在同步接收到的起始字节中有一个 K28.5 字符（其值为 05H）。若要去掉该字符，可将 $\overline{RDY}$ 信号与 SC/$\overline{D}$ 信号组合以产生 FIFO 写信号，这样，该控制字符码（05H）就不会写入 FIFO 中，从而使 FIFO 中只包含用户的数据信息。

收稿日期：2001 - 05 - 22

咨询编号：020112

CY7B923
CY7B933

HOTLink® Transmitter/Receiver

Features

- **Fibre-Channel-compliant**
- **IBM ESCON®-compliant**
- **DVB-ASI-compliant**
- **ATM-compliant**
- **8B/10B-coded or 10-bit unencoded**
- **Standard HOTLink®: 160–330 Mbps**
- **High-speed HOTLink: 160–400 Mbps for high-speed applications**
- **Low-speed HOTLink: 150–160 Mbps for low-cost fiber applications**
- **TTL synchronous I/O**
- **No external phase locked-loop (PLL) components**
- **Triple PECL 100K serial outputs**
- **Dual PECL 100K serial inputs**
- **Low power: 350 mW (Tx), 650 mW (Rx)**
- **Compatible with fiber-optic modules, coaxial cable, and twisted pair media**
- **Built-in Self-Test (BIST)**
- **Single +5V supply**
- **28-pin SOIC/PLCC/LCC**
- **0.8m BiCMOS**

Functional Description

The CY7B923 HOTLink' Transmitter and CY7B933 HOTLink Receiver are point-to-point communications building blocks that transfer data over high-speed serial links (fiber, coax, and twisted pair). Standard HOTLink data rates range from 160-330 Mbits/second. Higher speed HOTLink is also available for high-speed applications (160-400 Mbits/second), as well as for those low-Cost applications HOTLink-155 (150-160 Mbits/second operations). *Figure 1* illustrates typical connections to host systems or controllers.

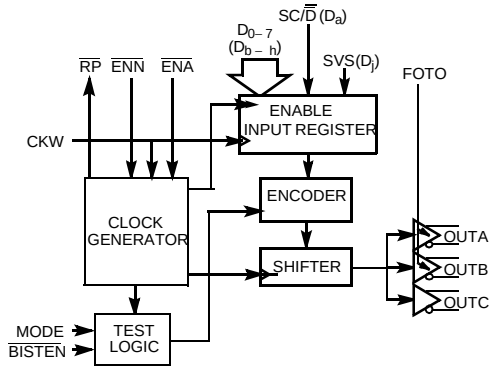
Eight bits of user data or protocol information are loaded into the HOTLink transmitter and are encoded. Serial data is shifted out of the three differential positive ECL (PECL) serial ports at the bit rate (which is ten times the byte rate).

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

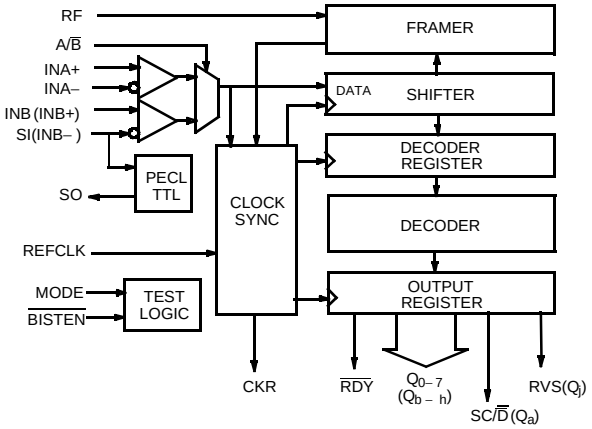
The 8B/10B encoder/decoder can be disabled in systems that already encode or scramble the transmitted data. I/O signals are available to create a seamless interface with both asynchronous FIFOs (i.e., CY7C42X) and clocked FIFOs (i.e., CY7C44X). A BIST pattern generator and checker allows testing of the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

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

CY7B923 Transmitter



CY7B933 Receiver Logic Block Diagram



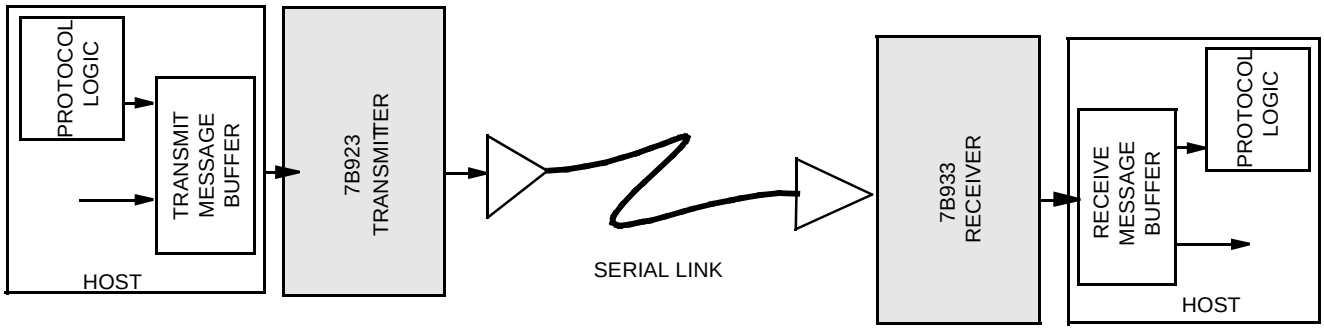
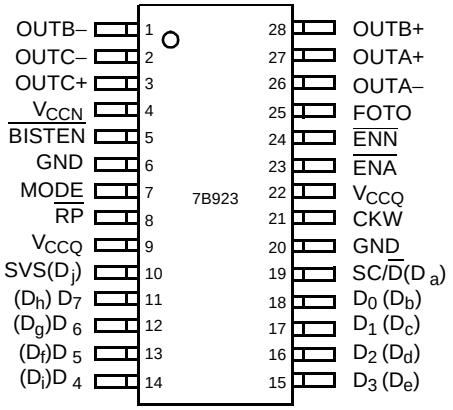


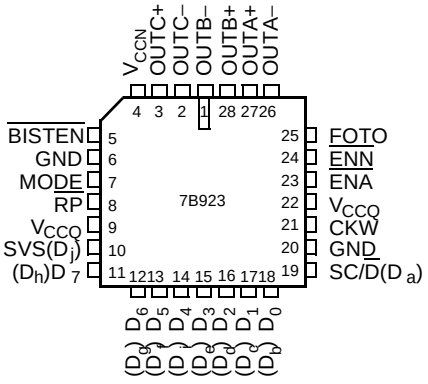
Figure 1. HOTLink System Connections

CY7B923 Transmitter Pin Configurations

SOIC
Top View

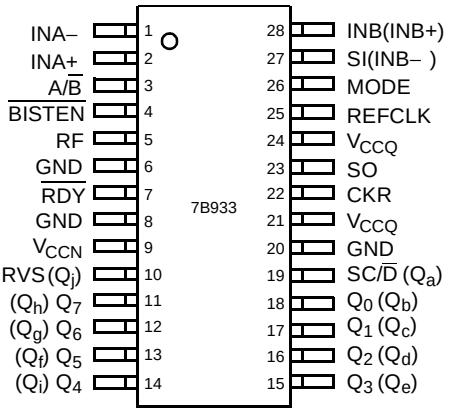


PLCC/LCC
Top View

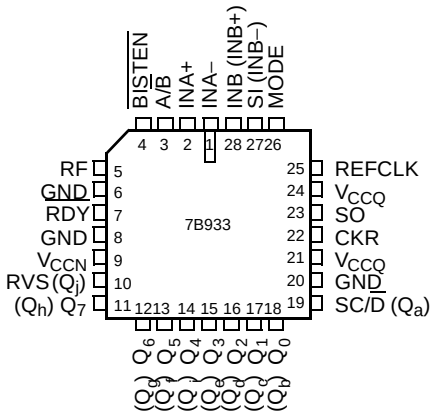


CY7B933 Receiver Pin Configurations

SOIC
Top View



PLCC/LCC
Top View



Pin Descriptions

CY7B923 HOTLink Transmitter

Name	I/O	Description
D_{0-7} (D_{b-h})	TTL In	**Parallel Data Input**. Data is clocked into the Transmitter on the rising edge of CKW if $\overline{ENA}$ is LOW (or on the next rising CKW with $\overline{ENN}$ LOW). If $\overline{ENA}$ and $\overline{ENN}$ are HIGH, a Null character (K28.5) is sent. When MODE is HIGH, $D_{0, 1, ...7}$ become $D_{b, c,...h}$, respectively.
SC/$\overline{D}$ (D_a)	TTL In	**Special Character/Data Select**. A HIGH on SC/$\overline{D}$ when CKW rises causes the transmitter to encode the pattern on D_{0-7} as a control code (Special Character), while a LOW causes the data to be coded using the 8B/10B data alphabet. When MODE is HIGH, SC/$\overline{D}$ (D_a) acts as D_a input. SC/$\overline{D}$ has the same timing as D_{0-7}.
SVS (D_j)	TTL In	**Send Violation Symbol**. If SVS is HIGH when CKW rises, a Violation symbol is encoded and sent while the data on the parallel inputs is ignored. If SVS is LOW, the state of D_{0-7} and SC/$\overline{D}$ determines the code sent. In normal or test mode, this pin overrides the BIST generator and forces the transmission of a Violation code. When MODE is HIGH (placing the transmitter in unencoded mode), SVS (D_j) acts as the D_j input. SVS has the same timing as D_{0-7}.
$\overline{ENA}$	TTL In	**Enable Parallel Data**. If $\overline{ENA}$ is LOW on the rising edge of CKW, the data is loaded, encoded, and sent. If $\overline{ENA}$ and $\overline{ENN}$ are HIGH, the data inputs are ignored and the Transmitter will insert a Null character (K28.5) to fill the space between user data. $\overline{ENA}$ may be held HIGH/LOW continuously or it may be pulsed with each data byte to be sent. If $\overline{ENA}$ is being used for data control, $\overline{ENN}$ will normally be strapped HIGH, but can be used for BIST function control.
$\overline{ENN}$	TTL In	**Enable Next Parallel Data**. If $\overline{ENN}$ is LOW, the data appearing on D_{0-7} at the next rising edge of CKW is loaded, encoded, and sent. If $\overline{ENA}$ and $\overline{ENN}$ are HIGH, the data appearing on D_{0-7} at the next rising edge of CKW will be ignored and the Transmitter will insert a Null character to fill the space between user data. $\overline{ENN}$ may be held HIGH/LOW continuously or it may be pulsed with each data byte sent. If $\overline{ENN}$ is being used for data control, $\overline{ENA}$ will normally be strapped HIGH, but can be used for BIST function control.
CKW	TTL In	**Clock Write**. CKW is both the clock frequency reference for the multiplying PLL that generates the high–speed transmit clock, and the byte rate write signal that synchronizes the parallel data input. CKW must be connected to a crystal controlled time base that runs within the specified frequency range of the Transmitter and Receiver.
FOTO	TTL In	**Fiber Optic Transmitter Off**. FOTO determines the function of two of the three PECL transmitter output pairs. If FOTO is LOW, the data encoded by the Transmitter will appear at the outputs continuously. If FOTO is HIGH, OUTA± and OUTB± are forced to their “logic zero” state (OUT+ = LOW and OUT– = HIGH), causing a fiber optic transmit module to extinguish its light output. OUTC is unaffected by the level on FOTO, and can be used as a loop-back signal source for board-level diagnostic testing.
OUTA± OUTB± OUTC±	PECL Out	**Differential Serial Data Outputs**. These PECL 100K outputs (+5V referenced) are capable of driving terminated transmission lines or commercial fiber optic transmitter modules. Unused pairs of outputs can be left open, or wired to V_{CC} to reduce power, if the output is not required. OUTA± and OUTB± are controlled by the level on FOTO, and will remain at their “logical zero” states when FOTO is asserted. OUTC± is unaffected by the level on FOTO. (OUTA+ and OUTB+ are used as a differential test clock input while in Test mode, i.e., MODE = UNCONNECTED or forced to $V_{CC/2}$.)
MODE	Three-Level In	**Encoder Mode Select**. The level on MODE determines the encoding method to be used. When wired to GND, MODE selects 8B/10B encoding. When wired to V_{CC}, data inputs bypass the encoder and the bit pattern on D_{a-j} goes directly to the shifter. When left floating (internal resistors hold the input at $V_{CC}/2$) the internal bit-clock generator is disabled and OUTA+/OUTB+ become the differential bit clock to be used for factory test. In typical applications MODE is wired to V_{CC} or GND.
$\overline{BISTEN}$	TTL In	**BIST Enable**. When $\overline{BISTEN}$ is LOW and $\overline{ENA}$ and $\overline{ENN}$ are HIGH, the transmitter sends an alternating 1–0 pattern (D10.2 or D21.5). When either $\overline{ENA}$ or $\overline{ENN}$ is set LOW and $\overline{BISTEN}$ is LOW, the transmitter begins a repeating test sequence that allows the Transmitter and Receiver to work together to test the function of the entire link. In normal use this input is held HIGH or wired to V_{CC}. The BIST generator is a free-running pattern generator that need not be initialized, but if required, the BIST sequence can be initialized by momentarily asserting SVS while $\overline{BISTEN}$ is LOW. $\overline{BISTEN}$ has the same timing as D0-7.

CY7B923 HOTLink Transmitter (continued)

Name	I/O	Description
$\overline{RP}$	TTL Out	**Read Pulse**. $\overline{RP}$ is a 60% LOW duty-cycle byte-rate pulse train suitable for the read pulse in CY7C42X FIFOs. The frequency on $\overline{RP}$ is the same as CKW when enabled by $\overline{ENA}$, and duty cycle is independent of the CKW duty cycle. Pulse widths are set by logic internal to the transmitter. In BIST mode, $\overline{RP}$ will remain HIGH for all but the last byte of a test loop. $\overline{RP}$ will pulse LOW one byte time per BIST loop.
V_{CCN}		**Power for output drivers**.
V_{CCQ}		**Power for internal circuitry**.
GND		**Ground**.

CY7B933 HOTLink Receiver

Name	I/O	Description
$Q_{0–7}$ (Q_{b-h})	TTL Out	**Q0-7 Parallel Data Output**. Q_{0-7} contain the most recently received data. These outputs change synchronously with CKR. When MODE is HIGH, $Q_{0, 1, ...7}$ become $Q_{b, c,...h}$, respectively.
SC/$\overline{D}$ (Q_a)	TTL Out	**Special Character/Data Select**. SC/$\overline{D}$ indicates the context of received data. HIGH indicates a Control (Special Character) code, LOW indicates a Data character. When MODE is HIGH (placing the receiver in Unencoded mode), SC/$\overline{D}$ acts as the Q_a output. SC/$\overline{D}$ has the same timing as $Q_{0–7}$.
RVS (Q_j)	TTL Out	**Received Violation Symbol**. A HIGH on RVS indicates that a code rule violation has been detected in the received data stream. A LOW shows that no error has been detected. In BIST mode, a LOW on RVS indicates correct operation of the Transmitter, Receiver, and link on a byte-by-byte basis. When MODE is HIGH (placing the receiver in Unencoded mode), RVS acts as the Q_j output. RVS has the same timing as $Q_{0–7}$.
$\overline{RDY}$	TTL Out	**Data Output Ready**. A LOW pulse on $\overline{RDY}$ indicates that new data has been received and is ready to be delivered. A missing pulse on $\overline{RDY}$ shows that the received data is the Null character (normally inserted by the transmitter as a pad between data inputs). In BIST mode $\overline{RDY}$ will remain LOW for all but the last byte of a test loop and will pulse HIGH one byte time per BIST loop.
CKR	TTL Out	**Clock Read**. This byte rate clock output is phase and frequency aligned to the incoming serial data stream. $\overline{RDY}$, $Q_{0–7}$, SC/$\overline{D}$, and RVS all switch synchronously with the rising edge of this output.
A/$\overline{B}$	PECL in	**Serial Data Input Select**. This PECL 100K (+5V referenced) input selects INA or INB as the active data input. If A/$\overline{B}$ is HIGH, INA is connected to the shifter and signals connected to INA will be decoded. If A/$\overline{B}$ is LOW INB is selected.
INA±	Diff In	**Serial Data Input A**. The differential signal at the receiver end of the communication link may be connected to the differential input pairs INA± or INB±. Either the INA pair or the INB pair can be used as the main data input and the other can be used as a loopback channel or as an alternative data input selected by the state of A/$\overline{B}$. One input of an intentionally unused differential-pair (INA± or INB±) should be terminated to V_{CC} through a 1–5 KΩ resistor to assure that no data transitions are accidentally created.
INB (INB+)	PECL in (Diff In)	**Serial Data Input B**. This pin is either a single-ended PECL data receiver (INB) or half of the INB differential pair. If SO is wired to V_{CC}, then INB± can be used as differential line receiver interchangeably with INA±. If SO is normally connected and loaded, INB becomes a single-ended PECL 100K (+5V referenced) serial data input. INB is used as the test clock while in Test mode.
SI (INB–)	PECL in (Diff In)	**Status Input**. This pin is either a single-ended PECL status monitor input (SI) or half of the INB differential pair. If SO is wired to V_{CC}, then INB± can be used as differential line receiver interchangeably with INA±. If SO is normally connected and loaded, SI becomes a single-ended PECL 100K (+5V referenced) status monitor input, which is translated into a TTL-level signal at the SO pin.
SO	TTL Out	**Status Out**. SO is the TTL-translated output of SI. It is typically used to translate the Carrier Detect output from a fiber-optic receiver connected to SI. When this pin is normally connected and loaded (without any external pull-up resistor), SO will assume the same logical level as SI and INB will become a single-ended PECL serial data input. If the status monitor translation is not desired, then SO may be wired to V_{CC} and the INB± pair may be used as a differential serial data input.
RF	TTL In	**Reframe Enable**. RF controls the Framer logic in the Receiver. When RF is held HIGH, each SYNC (K28.5) symbol detected in the shifter will frame the data that follows. If it is HIGH for 2,048 consecutive bytes, the internal framer switches to double-byte mode. When RF is held LOW, the reframing logic is disabled. The incoming data stream is then continuously deserialized and decoded using byte boundaries set by the internal byte counter. Bit errors in the data stream will not cause alias SYNC characters to reframe the data erroneously.

CY7B933 HOTLink Receiver (continued)

Name	I/O	Description
REFCLK	TTL In	**Reference Clock**. REFCLK is the clock frequency reference for the clock/data synchronizing PLL. REFCLK sets the approximate center frequency for the internal PLL to track the incoming bit stream. REFCLK must be connected to a crystal-controlled time base that runs within the frequency limits of the Tx/Rx pair, and the frequency must be the same as the transmitter CKW frequency (within CKW ± 0.1%).
MODE	Three-Level In	**Decoder Mode Select**. The level on the MODE pin determines the decoding method to be used. When wired to GND, MODE selects 8B/10B decoding. When wired to V_{CC}, registered shifter contents bypass the decoder and are sent to Q_{a-j} directly. When left floating (internal resistors hold the MODE pin at $V_{CC}/2$) the internal bit clock generator is disabled and INB becomes the bit rate test clock to be used for factory test. In typical applications, MODE is wired to V_{CC} or GND.
$\overline{\text{BISTEN}}$	TTL In	**Built-In Self-Test Enable**. When $\overline{\text{BISTEN}}$ is LOW the Receiver awaits a D0.0 (sent once per BIST loop) character and begins a continuous test sequence that tests the functionality of the Transmitter, the Receiver, and the link connecting them. In BIST mode the status of the test can be monitored with $\overline{\text{RDY}}$ and RVS outputs. In normal use BISTEN is held HIGH or wired to V_{CC}. $\overline{\text{BISTEN}}$ has the same timing as Q_{0-7}.
V_{CCN}		**Power for output drivers**.
V_{CCQ}		**Power for internal circuitry**.
GND		**Ground**.

CY7B923 HOTLink Transmitter Block Diagram Description

Input Register

The Input register holds the data to be processed by the HOTLink transmitter and allows the input timing to be made consistent with standard FIFOs. The Input register is clocked by CKW and loaded with information on the D_{0-7}, SC/$\overline{\text{D}}$, and SVS pins. Two enable inputs ($\overline{\text{ENA}}$ and $\overline{\text{ENN}}$) allow the user to choose when data is loaded in the register. Asserting $\overline{\text{ENA}}$ (Enable, active LOW) causes the inputs to be loaded in the register on the rising edge of CKW. If $\overline{\text{ENN}}$ (Enable Next, active LOW) is asserted when CKW rises, the data present on the inputs on the next rising edge of CKW will be loaded into the Input register. If neither $\overline{\text{ENA}}$ nor $\overline{\text{ENN}}$ are asserted LOW on the rising edge of CKW, then a SYNC (K28.5) character is sent. These two inputs allow proper timing and function for compatibility with either asynchronous FIFOs or clocked FIFOs without external logic, as shown in *Figure 4.*

In BIST mode, the Input register becomes the signature pattern generator by logically converting the parallel Input register into a Linear Feedback Shift Register (LFSR). When enabled, this LFSR will generate a 511-byte sequence that includes all Data and Special Character codes, including the explicit violation symbols. This pattern provides a predictable but pseudo-random sequence that can be matched to an identical LFSR in the Receiver.

Encoder

The Encoder transforms the input data held by the Input register into a form more suitable for transmission on a serial interface link. The code used is specified by ANSI X3.230 (Fibre Channel) and the IBM ESCON channel (code tables are at the end of this datasheet). The eight D_{0-7} data inputs are converted to either a Data symbol or a Special Character, depending upon the state of the SC/$\overline{\text{D}}$ input. If SC/$\overline{\text{D}}$ is HIGH, the data inputs represent a control code and are encoded using the Special Character code table. If SC/$\overline{\text{D}}$ is LOW, the data inputs are converted using the Data code table. If a byte time passes with the inputs disabled, the Encoder will output a Special Character Comma K28.5 (or SYNC) that will maintain link synchronization. SVS input forces the transmission of a specified Violation symbol to allow the user to check error handling system logic in the controller or for proprietary applications.

The 8B/10B coding function of the Encoder can be bypassed for systems that include an external coder or scrambler function as part of the controller. This bypass is controlled by setting the MODE select pin HIGH. When in bypass mode, D_{a-j} (note that bit order is specified in the Fibre Channel 8B/10B code) become the ten inputs to the Shifter, with D_a being the first bit to be shifted out.

Shifter

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

OutA, OutB, OutC

The serial interface PECL output buffers (ECL100K referenced to +5V) are the drivers for the serial media. They are all connected to the Shifter and contain the same serial data. Two of the output pairs (OUTA± and OUTB±) are controllable by the FOTO input and can be disabled by the system controller to force a logical zero (i.e., "light off") at the outputs. The third output pair (OUTC±) is not affected by FOTO and will supply a continuous data stream suitable for loop-back testing of the subsystem.

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

In wire-based systems, control of the outputs may not be required, and FOTO can be strapped LOW. The three outputs are intended to add system and architectural flexibility by offering identical serial bit streams with separate interfaces for

redundant connections or for multiple destinations. Unneeded outputs can be wired to VCC to disable and power down the unused output circuitry.

Clock Generator

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

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

Test Logic

Test logic includes the initialization and control for the Built-In Self-Test (BIST) generator, the multiplexer for Test mode clock distribution, and control logic to properly select the data encoding. Test logic is discussed in more detail in the CY7B923 HOTLink Transmitter Operating Mode Description.

CY7B933 HOTLink Receiver Block Diagram Description

Serial Data Inputs

Two pairs of differential line receivers are the inputs for the serial data stream. INA± or INB± can be selected with the A/B input. INA± is selected with A/B HIGH and INB± is selected with A/B LOW. The threshold of A/B is compatible with the ECL 100K signals from PECL fiber optic interface modules. TTL logic elements can be used to select the A or B inputs by adding a resistor pull-up to the TTL driver connected to A/B. The differential threshold of INA± and INB± will accommodate wire interconnect with filtering losses or transmission line attenuation greater than 20 db ($V_{DIF} \geq 50$ mv) or can be directly connected to fiber optic interface modules (any ECL logic family, not limited to ECL 100K). The common mode tolerance will accommodate a wide range of signal termination voltages. The highest HIGH input that can be tolerated is $V_{IN} = V_{CC}$, and the lowest LOW input that can be interpreted correctly is V_{IN} = GND+2.0V.

PECL-TTL Translator

The function of the INB(INB+) input and the SI(INB-) input is defined by the connections on the SO output pin. If the PECL/TTL translator function is not required, the SO output is wired to VCC. A sensor circuit will detect this connection and cause the inputs to become INB± (a differential line-receiver serial-data input). If the PECL/TTL translator function is required, the SO output is connected to its normal TTL load (typically one or more TTL inputs, but no pull-up resistor) and the INB+ input becomes INB (single-ended ECL 100K, serial data input) and the INB- input becomes SI (single-ended, ECL 100K status input).

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

Clock Synchronization

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

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

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

Framer

Framer logic checks the incoming bit stream for the pattern that defines the byte boundaries. This combinatorial logic filter looks for the X3.230 symbol defined as a Special Character Comma (K28.5). When it is found, the free-running bit counter in the Clock Synchronization block is synchronously reset to its initial state, thus framing the data correctly on the correct byte boundaries.

Random errors that occur in the serial data can corrupt some data patterns into a bit pattern identical to a K28.5, and thus cause an erroneous data-framing error. The RF input prevents this by inhibiting reframing during times when normal message data is present. When RF is held LOW, the HOTLink receiver will deserialize the incoming data without trying to reframe the data to incoming patterns. When RF rises, $\overline{RDY}$ will be inhibited until a K28.5 has been detected, after which $\overline{RDY}$ will resume its normal function. While RF is HIGH, it is possible that an error could cause misframing, after which all data will be corrupted. Likewise, a K28.7 followed by D11.x, D20.x, or an SVS (C0.7) followed by D11.x will create alias K28.5 characters and cause erroneous framing. These sequences must be avoided while RF is HIGH.

If RF remains HIGH for greater than 2048 bytes, the framer converts to double-byte framing, requiring two K28.5 characters aligned on the same byte boundary within 5 bytes in order to reframe. Double-byte framing greatly reduces the possibility of erroneously reframing to an aliased K28.5 character.

Shifter

The Shifter accepts serial inputs from the Serial Data inputs one bit at a time, as clocked by the Clock Synchronization

logic. Data is transferred to the Framer on each bit, and to the Decode register once per byte.

Decode Register

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

Decoder

Parallel data is transformed from ANSI-specified X3.230 8B/10B codes back to “raw data” in the Decoder. This block uses the standard decoder patterns shown in the Valid Data Characters and Valid Special Character Codes and Sequences sections of this datasheet. Data patterns are signaled by a LOW on the SC/$\overline{D}$ output and Special Character patterns are signaled by a HIGH on the SC/$\overline{D}$ output. Unused patterns or disparity errors are signaled as errors by a HIGH on the RVS output and by specific Special Character codes.

Output Register

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

In BIST mode, this register becomes the signature pattern generator and checker by logically converting itself into a Linear Feedback Shift Register (LFSR) pattern generator. When enabled, this LFSR will generate a 511-byte sequence that includes all Data and Special Character codes, including the explicit violation symbols. This pattern provides a predictable but pseudo-random sequence that can be matched to an identical LFSR in the Transmitter. When synchronized, it checks each byte in the Decoder with each byte generated by the LFSR and shows errors at RVS. Patterns generated by the LFSR are compared after being buffered to the output pins and then fed back to the comparators, allowing test of the entire receive function.

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

Test Logic

Test logic includes the initialization and control for the Built-In Self-Test (BIST) generator, the multiplexer for Test mode clock distribution, and control logic for the decoder. Test logic is discussed in more detail in the CY7B933 HOTLink Receiver Operating Mode Description.

HOTLink CY7B923 Transmitter and CY7B933 Receiver Operation

The CY7B923 Transmitter operating with the CY7B933 Receiver form a general purpose data communications subsystem capable of transporting user data at up to 33 Mbytes per second (40 Mbytes per second for –400 devices) over several types of serial interface media. *Figure 7* illustrates the flow of data through the HOTLink CY7B923 transmitter pipeline. Data is latched into the transmitter on the rising edge of CKW when enabled by $\overline{ENA}$ or $\overline{ENN}$. $\overline{RP}$ is asserted LOW with a 60% LOW/40% HIGH duty cycle when $\overline{ENA}$ is LOW. $\overline{RP}$ may be used as a read strobe for accessing data stored in a FIFO. The parallel data flows through the encoder and is then shifted out of the OUTx± PECL drivers. The bit-rate clock is generated internally from a multiply-by-ten PLL clock generator. The latency through the transmitter is approximately 21tB – 10 ns over the operating range. A more complete description is found in the section CY7B923 HOTLink Transmitter Operating Mode Description.

Figure 2 illustrates the data flow through the HOTLink CY7B933 receiver pipeline. Serial data is sampled by the receiver on the INx± inputs. The receiver PLL locks onto the serial bit stream and generates an internal bit rate clock. The bit stream is deserialized, decoded and then presented at the parallel output pins. A byte rate clock (bit clock ÷ 10) synchronous with the parallel data is presented at the CKR pin. The $\overline{RDY}$ pin will be asserted to LOW to indicate that data or control characters are present on the outputs. $\overline{RDY}$ will not be asserted LOW in a field of K28.5s except for any single K28.5 or the last one in a continuous series of K28.5’s. The latency through the receiver is approximately $24t_B$ + 10 ns over the operating range. A more complete description of the receiver is in the section CY7B933 HOTLink Receiver Operating Mode Description.

The HOTLink Receiver has a built-in byte framer that synchronizes the Receiver pipeline with incoming SYNC (K28.5) characters. *Figure 3* illustrates the HOTLink CY7B933 Receiver framing operation. The Framer is enabled when the RF pin is asserted HIGH. RF is latched into the receiver on the falling edge of CKR. The framer looks for K28.5 characters embedded in the serial data stream. When a K28.5 is found, the framer sets the parallel byte boundary for subsequent data to the K28.5 boundary. While the framer is enabled, the $\overline{RDY}$ pin indicates the status of the framing operation.

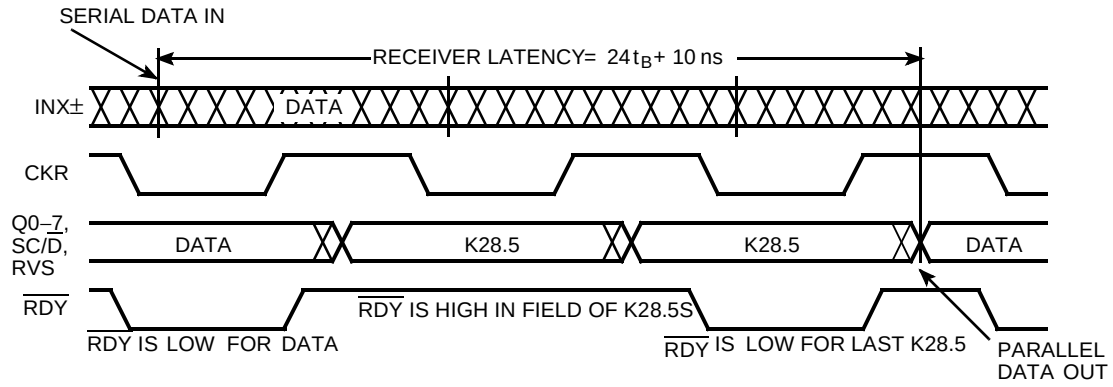


Figure 2. CY7B933 Receiver Data Pipeline in Encoded Mode

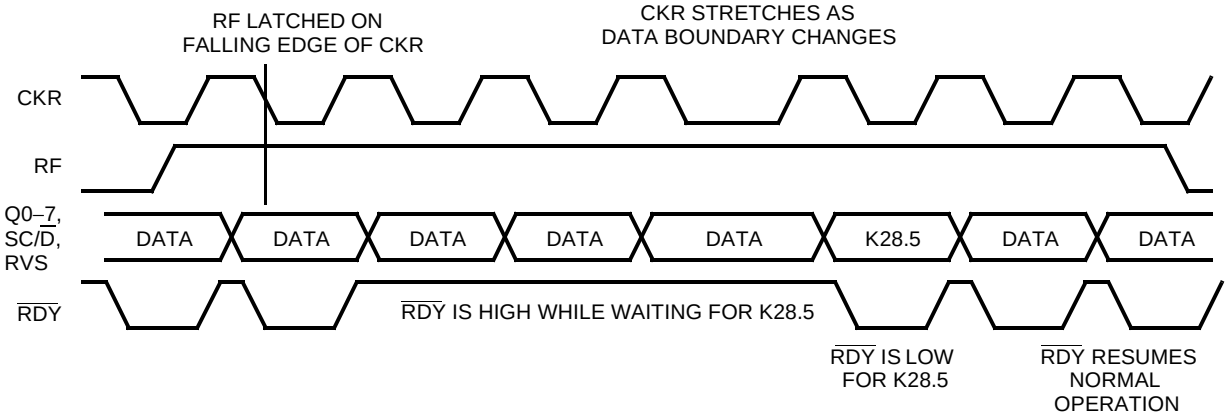


Figure 3. CY7B933 Framing Operation in Encoded Mode

When the RF pin is asserted HIGH, $\overline{RDY}$ leaves it normal mode of operation and is asserted HIGH while the framer searches the data stream for a K28.5 character. After the framer has synchronized to a K28.5 character, the Receiver will assert the $\overline{RDY}$ pin LOW when the K28.5 character is present at the parallel output. The $\overline{RDY}$ pin will then resume its normal operation as dictated by the $\overline{MODE}$ and $\overline{BISTEN}$ pins.

The normal operation of the $\overline{RDY}$ pin in encoded mode is to signal when parallel data is present at the output pins by pulsing LOW with a 60% LOW/40% HIGH duty cycle. $\overline{RDY}$ does not pulse LOW in a field of K28.5 characters; however, $\overline{RDY}$ does pulse LOW for the last K28.5 character in the field or for any single K28.5. In unencoded mode, the normal operation of the $\overline{RDY}$ pin is to signal when any K28.5 is at the parallel output pins.

The Transmitter and Receiver parallel interface timing and functionality can be made to match the timing and functionality of either an asynchronous FIFO or a clocked FIFO by appropriately connecting signals (See *Figure 4*). Proper operation of the FIFO interface depends upon various FIFO-specific access and response specifications.

The HOTLink Transmitter and Receiver serial interface provides a seamless interface to various types of media. A minimal number of external components are needed to properly terminate transmission lines and provide PECL loads. For proper power supply decoupling, a single 0.01 mF for each device is all that is required to bypass the VCC and GND pins. *Figure 5* illustrates a HOTLink Transmitter and Receiver interface to fiber optic and copper media. More information on interfacing HOTLink to various media can be found in the *HOTLink Design Considerations* application note.

CY7B923 HOTLink Transmitter Operating Mode Description

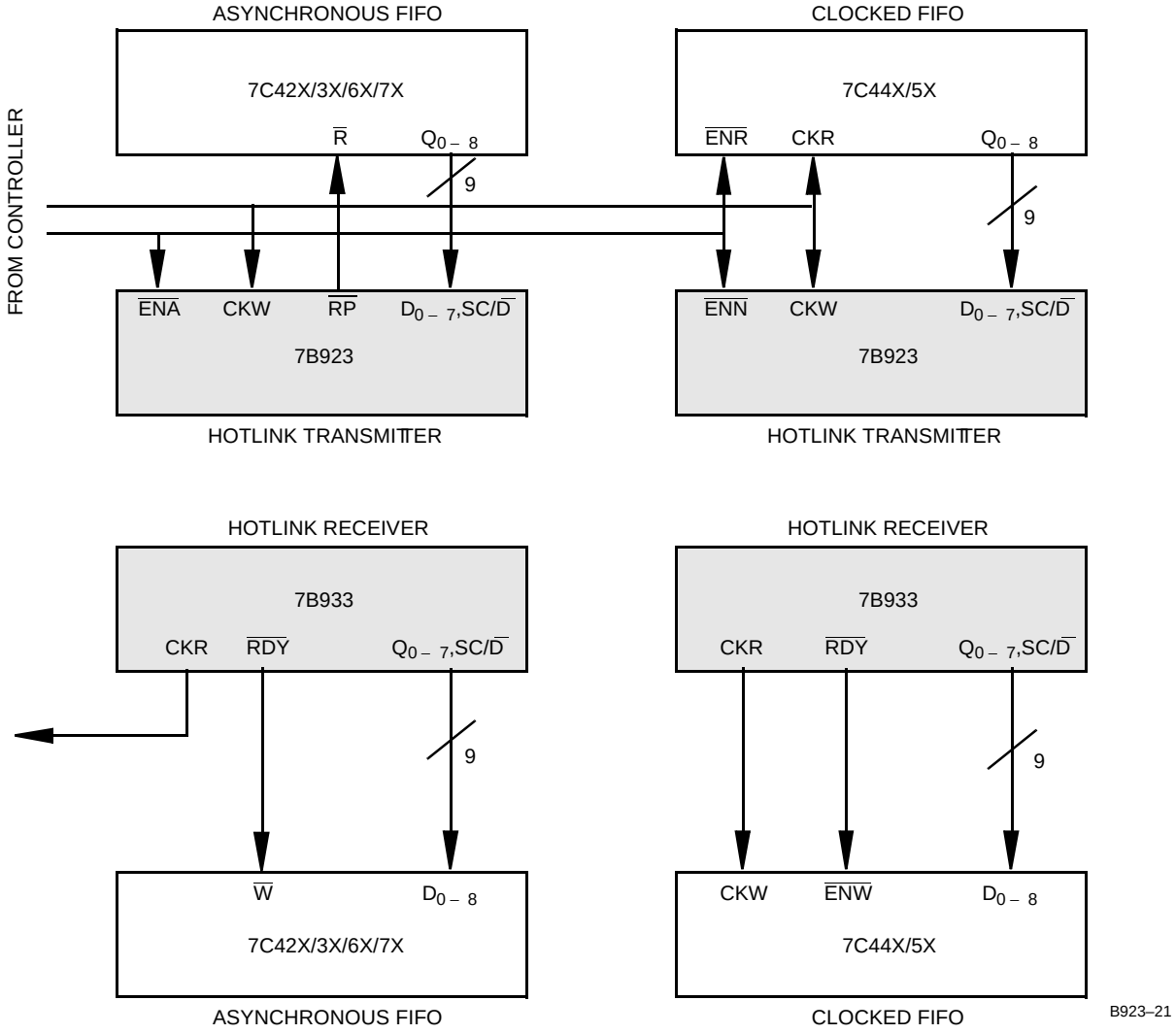
In normal operation, the Transmitter can operate in either of two modes. The Encoded mode allows a user to send and receive eight-bit data and control information without first converting it to transmission characters. The Bypass mode is used for systems in which the encoding and decoding is performed in an external protocol controller.

In either mode, data is loaded into the Input register of the Transmitter on the rising edge of CKW. The input timing and functional response of the Transmitter input can be made to match the timing and functionality of either an asynchronous FIFO or a clocked FIFO by an appropriate connection of input signals (see *Figure 4*). Proper operation of the FIFO interface depends upon various FIFO-specific access and response specifications.

Encoded Mode Operation

In Encoded mode the input data is interpreted as eight bits of data (D0 – D7), a context control bit (SC/$\overline{D}$), and a system diagnostic input bit (SVS). If the context of the data is to be normal message data, the SC/$\overline{D}$ input should be LOW, and the data should be encoded using the valid data character set described in the Valid Data Characters section of this datasheet. If the context of the data is to be control or protocol information, the SC/$\overline{D}$ input will be HIGH, and the data will be encoded using the valid special character set described in the Valid Special Character Codes and Sequences section. Special characters include all protocol characters necessary to encode packets for Fibre Channel, ESCON, proprietary systems, and diagnostic purposes.

The diagnostic characters and sequences available as Special Characters include those for Fibre Channel link testing, as well as codes to be used for testing system response to link errors and timing. A Violation symbol can be explicitly sent as part of a user data packet (i.e., send C0.7; D_{7-0} = 111 00000 and SC/$\overline{D}$ = 1), or it can be sent in response to an external system using the SVS input. This will allow system diagnostic logic to evaluate the errors in an unambiguous manner, and will not require any modification to the transmission interface to force transmission errors for testing purposes.



B923–21

Figure 4. Seamless FIFO Interface

Bypass Mode Operation

In Bypass mode the input data is interpreted as ten (10) bits (D_{b-h}), SC/$\overline{D}$ (Da), and SVS (Dj) of pre-encoded transmission data to be serialized and sent over the link. This data can use any encoding method suitable to the designer. The only restrictions upon the data encoding method is that it contain suitable transition density for the Receiver PLL data synchronizer (one per 10 bit byte), and that it be compatible with the transmission media.

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

While in either the Encoded mode or Bypass mode, if a CKW edge arrives when the inputs are not enabled (ENA and ENN both HIGH), the Encoder will insert a pad character K28.5 (e.g., C5.0) to maintain proper link synchronization (in Bypass

mode the proper sense of running disparity cannot be guaranteed for the first pad character, but is correct for all pad characters that follow). This automatic insertion of pad characters can be inhibited by insuring that the Transmitter is always enabled (i.e., ENA or ENN is hard-wired LOW).

PECL Output Functional and Connection Options

The three pairs of PECL outputs all contain the same information and are intended for use in systems with multiple connections. Each output pair may be connected to a different serial media, each of which may be a different length, link type, or interface technology. For systems that do not require all three output pairs, the unused pairs should be wired to V_{CC} to minimize the power dissipated by the output circuit, and to minimize unwanted noise generation. An internal voltage comparator detects when an output differential pair is wired to V_{CC}, causing the current source for that pair to be disabled. This results in a power savings of around 5 mA for each unused pair.

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

Transmitter Serial Data Characteristics

The CY7B923 HOTLink Transmitter serial output conforms to the requirements of the Fibre Channel specification. The serial data output is controlled by an internal Phase-Locked Loop that multiplies the frequency of CKW by ten (10) to maintain the proper bit clock frequency. The jitter characteristics (including both PLL and logic components) are shown below:

- Deterministic Jitter (D_j) < 35 ps (peak-peak). Typically measured while sending a continuous K28.5 (C5.0).
- Random Jitter (R_j) < 175 ps (peak-peak). Typically measured while sending a continuous K28.7 (C7.0).

Transmitter Test Mode Description

The CY7B923 Transmitter offers two types of test mode operation, BIST mode and Test mode. In a normal system application, the Built-In Self-Test (BIST) mode can be used to check the functionality of the Transmitter, the Receiver, and the link connecting them. This mode is available with minimal impact on user system logic, and can be used as part of the normal system diagnostics. Typical connections and timing are shown in *Figure 6*.

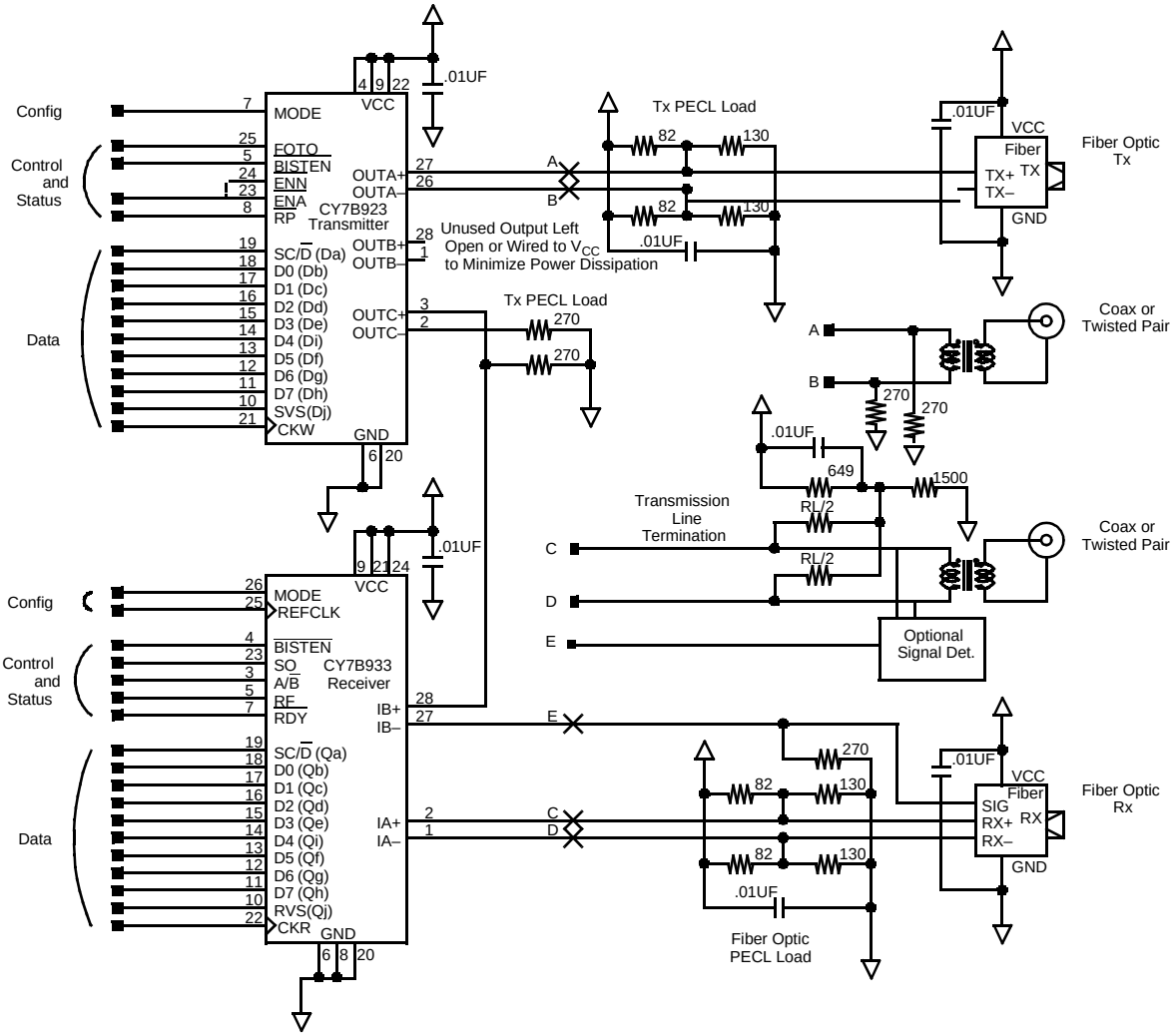


Figure 5. HOTLink Connection Diagram

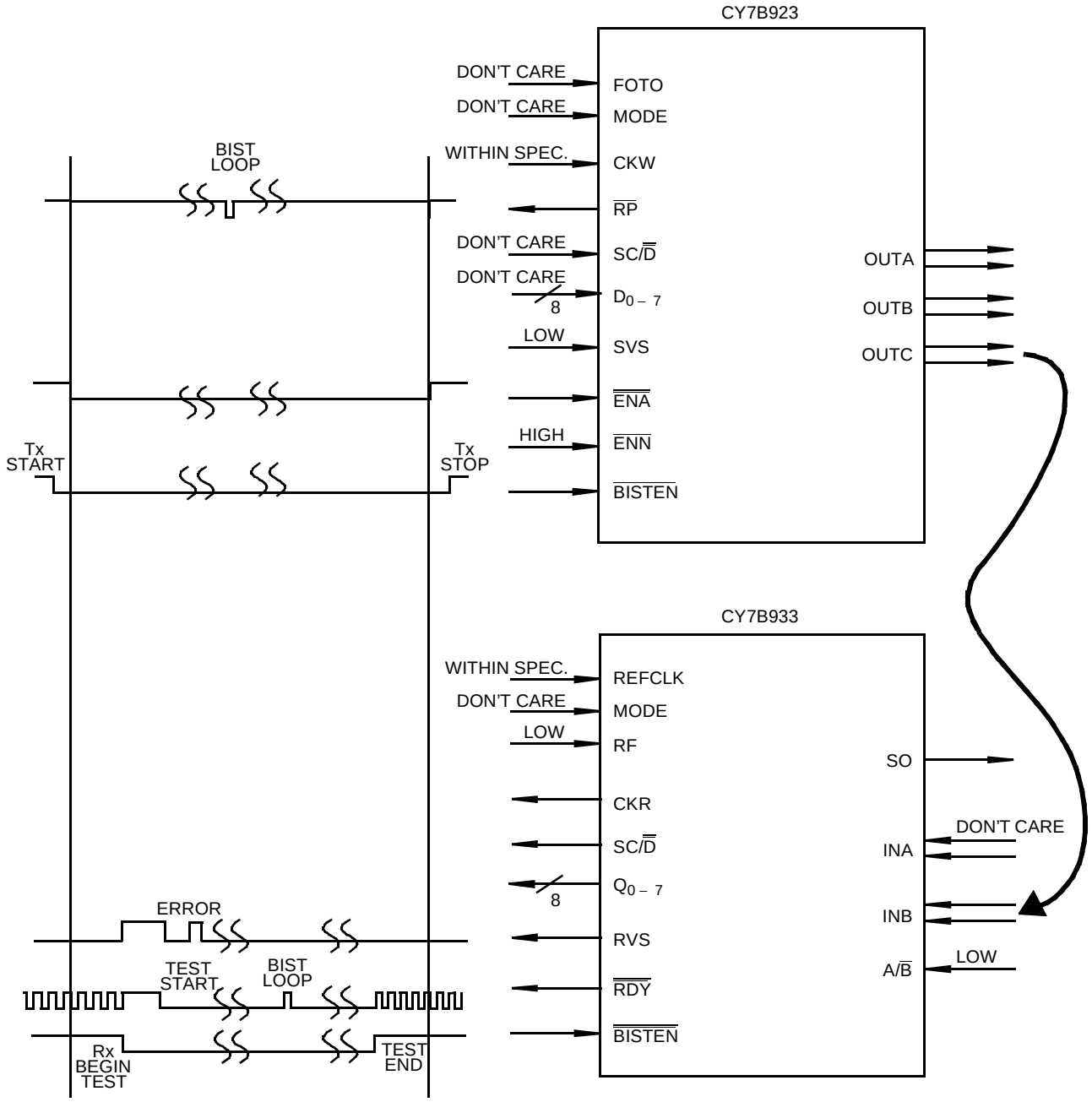


Figure 6. BIST Illustration

BIST Mode

BIST mode functions as follows:

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

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

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

While in Bypass mode, the BIST logic will function in the same way as in the Encoded mode. MODE = HIGH and $\overline{\text{BISTEN}}$ = LOW causes the Transmitter to switch to Encoded mode and begin sending the BIST pattern, as if MODE = LOW. When $\overline{\text{BISTEN}}$ returns to HIGH, the Transmitter resumes normal Bypass operation. In Test mode the BIST function works as in the Normal mode. For more information on BIST, consult the "HOTLink Built-In Self-Test" application note.

Test Mode

The MODE input pin selects between three transmitter functional modes. When wired to VCC, the $D_{(a-j)}$ inputs bypass the Encoder and load directly from the Input register into the Shifter. When wired to GND, the inputs D_{0-7}, SVS, and SC/$\overline{\text{D}}$ are encoded using the Fibre Channel 8B/10B codes and sequences (shown at the end of this datasheet). Since the Transmitter is usually hard wired to Encoded or Bypass mode and not switched between them, a third function is provided for the MODE pin. Test mode is selected by floating the MODE pin (internal resistors hold the MODE pin at $V_{CC}/2$). Test mode is used for factory or incoming device test.

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

1. With the MODE pin either HIGH or LOW, stop CKW and bit-clock.
2. Force the MODE pin to MID (open or $V_{CC}/2$) while the clocks are stopped.
3. Start the bit-clock and let it run for at least two cycles.
4. Start the CKW clock at the bit-clock/10 rate.

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

CY7B933 HOTLink Receiver Operating Mode Description

In normal user operation, the Receiver can operate in either of two modes. The Encoded mode allows a user system to send and receive eight-bit data and control information without first converting it to transmission characters. The Bypass mode is used for systems in which the encoding and decoding is performed by an external protocol controller.

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

To properly align the incoming bit stream to the intended byte boundaries, the bit counter in the Clock Synchronizer must be initialized. The Framer logic block checks the incoming bit stream for the unique pattern that defines the byte boundaries. This combinatorial logic filter looks for the X3.230 symbol defined as "Special Character Comma" (K28.5). Once K28.5 is found, the free running bit counter in the Clock Synchronizer block is synchronously reset to its initial state, thus "framing" the data to the correct byte boundaries.

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

Encoded Mode Operation

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

The Violation symbol that can be explicitly sent as part of a user data packet (i.e., Transmitter sending C0.7; D_{7-0} = 111 00000 and SC/$\overline{\text{D}}$ = 1; or SVS = 1) will be decoded and indicated in exactly the same way as a noise-induced error in the transmission link. This function will allow system diagnostics to evaluate the error in an unambiguous manner, and will not require any modification to the receiver data interface for error-testing purposes.

Bypass Mode Operation

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

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

Parallel Output Function

The 10 outputs (Q_{0-7}, SC/$\overline{D}$, and RVS) all transition simultaneously, and are aligned with $\overline{RDY}$ and CKR with timing allowances to interface directly with either an asynchronous FIFO or a clocked FIFO. Typical FIFO connections are shown in *Figure 4*.

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

When the Transmitter is disabled it will continuously send pad characters (K28.5). To assure that the receive FIFO will not be overfilled with these dummy bytes, the $\overline{RDY}$ pulse output is inhibited during fill strings. Data at the Q_{0-7} outputs will reflect the correct received data, but will not appear to change, since a string of K28.5s all are decoded as Q7-0 =000 00101 and SC/$\overline{D}$ = 1 (C5.0). When new data appears (not K28.5), the $\overline{RDY}$ output will resume normal function. The “last” K28.5 will be accompanied by a normal $\overline{RDY}$ pulse.

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

As noted above, $\overline{RDY}$ can also be used as an indication of correct framing of received data. While the Receiver is awaiting receipt of a K28.5 with RF HIGH, the $\overline{RDY}$ outputs will be inhibited. When $\overline{RDY}$ resumes, the received data will be properly framed and will be decoded correctly. In Bypass mode with RF HIGH, $\overline{RDY}$ will pulse once for each K28.5 received. For more information on the $\overline{RDY}$ pin, consult the “HOTLink CY7B933 $\overline{RDY}$ Pin Description” application note.

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

	RVS	SC/$\overline{D}$	Qouts	Name
1. Good Data code received with good running disparity (RD)	0	0	00-FF	D0.0-31.7
2. Good Special Character code received with good RD	0	1	00-0B	C0.0-11.0
3. K28.7 immediately following K28.1 (ESCON Connect_SOF)	0	1	27	C7.1
4. K28.7 immediately following K28.5 (ESCON Passive_SOF)	0	1	47	C7.2
5. Unassigned code received	1	1	E0	C0.7
6. -K28.5+ received when RD was +	1	1	E1	C1.7
7. +K28.5- received when RD was -	1	1	E2	C2.7
8. Good code received with wrong RD	1	1	E4	C4.7

Receiver Serial Data Requirements

The CY7B933 HOTLink Receiver serial input capability conforms to the requirements of the Fibre Channel specification. The serial data input is tracked by an internal PLL that is used to recover the clock phase and to extract the data from the serial bit stream. Jitter tolerance characteristics (including both PLL and logic component requirements) are shown below:

- Deterministic Jitter Tolerance (Dj) > 40% of tB. Typically measured while receiving data carried by a bandwidth-limited channel (e.g., a coaxial transmission line) while maintaining a Bit Error Rate (BER) < 10–12.
- Random Jitter Tolerance (Rj) > 90% of tB. Typically measured while receiving data carried by a random-noise-limited channel (e.g., a fiber-optic transmission system with low light levels) while maintaining a Bit Error Rate (BER) < 10–12.
- Total Jitter Tolerance > 90% of tB. Total of Dj + Rj.
- PLL-Acquisition Time < 500-bit times from worst-case phase or frequency change in the serial input data stream, to receiving data within BER objective of 10-12. Stable power supplies within specifications, stable REFCLK input frequency and normal data framing protocols are assumed. Note: Acquisition time is measured from worst-case phase or frequency change to zero phase and frequency error. As a result of the receiver’s wide jitter tolerance, valid data will appear at the receiver’s outputs a few byte times after a worst-case phase change.

Receiver Test Mode Description

The CY7B933 Receiver offers two types of test mode operation, BIST mode and Test mode. In a normal system application, the Built-In Self-Test (BIST) mode can be used to check the functionality of the Transmitter, the Receiver and the link connecting them. This mode is available with minimal impact on user system logic, and can be used as part of the normal system diagnostics. Typical connections and timing are shown in *Figure 6*.

BIST Mode

BIST Mode function is as follows:

1. Set $\overline{\text{BISTEN}}$ LOW to enable self-test generation and await $\overline{\text{RDY}}$ LOW indicating that the initialization code has been received.
2. Monitor RVS and check for any byte time with the pin HIGH to detect pattern mismatches. $\overline{\text{RDY}}$ will pulse HIGH once per BIST loop, and can be used by an external counter to monitor test pattern progress. Q_{0-7} and SC/$\overline{\text{D}}$ will show the expected pattern and may be useful for debug purposes.
3. When testing is completed, set $\overline{\text{BISTEN}}$ HIGH and resume normal function.

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

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

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

Test Mode

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

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

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

(While in Test mode and in BIST mode with RF HIGH, the Q_{0-7}, RVS, and SC/$\overline{\text{D}}$ outputs reflect various internal logic states and not the received data.)

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

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:ABCDEFGH
10100 010

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

Bits: abcdeifghj
1010010101

Each valid Transmission Character of the 8B/10B Transmission Code has been given a name using the following convention: cxx.y, where c is used to show whether the Transmission Character is a Data Character (c is set to D, and the SC/$\overline{D}$ pin is LOW) or a Special Character (c is set to K, and the SC/$\overline{D}$ pin is HIGH). When c is set to D, xx is the decimal value of the binary number composed of the bits E, D, C, B, and A in that order, and the y is the decimal 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 (dpANS X3.230-199X 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 1* 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 1. 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 2* shows an example of this behavior.

Table 2. 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	+

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

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

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

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

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

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

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

Data Byte Name	Bits		Current RD–		Current RD+	
	HGF	EDC-BA	abcdei	fghj	abcdei	fghj
D7.3	011	00111	111000	1100	000111	0011
D8.3	011	01000	111001	0011	000110	1100
D9.3	011	01001	100101	1100	100101	0011
D10.3	011	01010	010101	1100	010101	0011
D11.3	011	01011	110100	1100	110100	0011
D12.3	011	01100	001101	1100	001101	0011
D13.3	011	01101	101100	1100	101100	0011
D14.3	011	01110	011100	1100	011100	0011
D15.3	011	01111	010111	0011	101000	1100
D16.3	011	10000	011011	0011	100100	1100
D17.3	011	10001	100011	1100	100011	0011
D18.3	011	10010	010011	1100	010011	0011
D19.3	011	10011	110010	1100	110010	0011
D20.3	011	10100	001011	1100	001011	0011
D21.3	011	10101	101010	1100	101010	0011
D22.3	011	10110	011010	1100	011010	0011
D23.3	011	10111	111010	0011	000101	1100
D24.3	011	11000	110011	0011	001100	1100
D25.3	011	11001	100110	1100	100110	0011
D26.3	011	11010	010110	1100	010110	0011
D27.3	011	11011	110110	0011	001001	1100
D28.3	011	11100	001110	1100	001110	0011
D29.3	011	11101	101110	0011	010001	1100
D30.3	011	11110	011110	0011	100001	1100
D31.3	011	11111	101011	0011	010100	1100
D0.4	100	00000	100111	0010	011000	1101
D1.4	100	00001	011101	0010	100010	1101
D2.4	100	00010	101101	0010	010010	1101
D3.4	100	00011	110001	1101	110001	0010
D4.4	100	00100	110101	0010	001010	1101
D5.4	100	00101	101001	1101	101001	0010
D6.4	100	00110	011001	1101	011001	0010
D7.4	100	00111	111000	1101	000111	0010
D8.4	100	01000	111001	0010	000110	1101
D9.4	100	01001	100101	1101	100101	0010

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

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

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

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

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

Data Byte Name	Bits		Current RD–		Current RD+	
	HGF	EDC-BA	abcdei	fghj	abcdei	fghj
D16.6	110	10000	011011	0110	100100	0110
D17.6	110	10001	100011	0110	100011	0110
D18.6	110	10010	010011	0110	010011	0110
D19.6	110	10011	110010	0110	110010	0110
D20.6	110	10100	001011	0110	001011	0110
D21.6	110	10101	101010	0110	101010	0110
D22.6	110	10110	011010	0110	011010	0110
D23.6	110	10111	111010	0110	000101	0110
D24.6	110	11000	110011	0110	001100	0110
D25.6	110	11001	100110	0110	100110	0110
D26.6	110	11010	010110	0110	010110	0110
D27.6	110	11011	110110	0110	001001	0110
D28.6	110	11100	001110	0110	001110	0110
D29.6	110	11101	101110	0110	010001	0110
D30.6	110	11110	011110	0110	100001	0110
D31.6	110	11111	101011	0110	010100	0110
D0.7	111	00000	100111	0001	011000	1110
D1.7	111	00001	011101	0001	100010	1110
D2.7	111	00010	101101	0001	010010	1110
D3.7	111	00011	110001	1110	110001	0001
D4.7	111	00100	110101	0001	001010	1110
D5.7	111	00101	101001	1110	101001	0001
D6.7	111	00110	011001	1110	011001	0001
D7.7	111	00111	111000	1110	000111	0001
D8.7	111	01000	111001	0001	000110	1110
D9.7	111	01001	100101	1110	100101	0001
D10.7	111	01010	010101	1110	010101	0001
D11.7	111	01011	110100	1110	110100	1000
D12.7	111	01100	001101	1110	001101	0001
D13.7	111	01101	101100	1110	101100	1000
D14.7	111	01110	011100	1110	011100	1000
D15.7	111	01111	010111	0001	101000	1110
D16.7	111	10000	011011	0001	100100	1110
D17.7	111	10001	100011	0111	100011	0001
D18.7	111	10010	010011	0111	010011	0001

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

Data Byte Name	Bits		Current RD–		Current RD+	
	HGF	EDC-BA	abcdei	fghj	abcdei	fghj
D19.7	111	10011	110010	1110	110010	0001
D20.7	111	10100	001011	0111	001011	0001
D21.7	111	10101	101010	1110	101010	0001
D22.7	111	10110	011010	1110	011010	0001
D23.7	111	10111	111010	0001	000101	1110
D24.7	111	11000	110011	0001	001100	1110
D25.7	111	11001	100110	1110	100110	0001
D26.7	111	11010	010110	1110	010110	0001
D27.7	111	11011	110110	0001	001001	1110
D28.7	111	11100	001110	1110	001110	0001
D29.7	111	11101	101110	0001	010001	1110
D30.7	111	11110	011110	0001	100001	1110
D31.7	111	11111	101011	0001	010100	1110

Valid Special Character Codes and Sequences (SC/$\overline{D}$ = HIGH)[1, 2]

S.C. Byte Name	S.C. Code Name		Bits HGF	Bits EDCBA	Current RD– abcdei	Current RD– fghj	Current RD+ abcdei	Current RD+ fghj
K28.0	C0.0	(C00)	000	00000	001111	0100	110000	1011
K28.1	C1.0	(C01)	000	00001	001111	1001	110000	0110
K28.2	C2.0	(C02)	000	00010	001111	0101	110000	1010
K28.3	C3.0	(C03)	000	00011	001111	0011	110000	1100
K28.4	C4.0	(C04)	000	00100	001111	0010	110000	1101
K28.5	C5.0	(C05)	000	00101	001111	1010	110000	0101
K28.6	C6.0	(C06)	000	00110	001111	0110	110000	1001
K28.7	C7.0	(C07)	000	00111	001111	1000	110000	0111
K23.7	C8.0	(C08)	000	01000	111010	1000	000101	0111
K27.7	C9.0	(C09)	000	01001	110110	1000	001001	0111
K29.7	C10.0	(C0A)	000	01010	101110	1000	010001	0111
K30.7	C11.0	(C0B)	000	01011	011110	1000	100001	0111
Idle	C0.1	(C20)	001	00000	–K28.5+,D21.4,D21.5,D21.5,repeat[3]			
R_RDY	C1.1	(C21)	001	00001	–K28.5+,D21.4,D10.2,D10.2,repeat[4]			
EOFxx	C2.1	(C22)	001	00010	–K28.5,Dn.xxx0[5]+K28.5,Dn.xxx1[5]			
	Follows K28.1 for ESCON Connect–SOF (Rx indication only)							
C–SOF	C7.1	(C27)	001	00111	001111	1000	110000	0111
	Follows K28.5 for ESCON Passive–SOF (Rx indication only)							
P–SOF	C7.2	(C47)	010	00111	001111	1000	110000	0111
					Code Rule Violation and SVS Tx Pattern			
Exception	C0.7	(CE0)	111	00000	100111	1000[6]	011000	0111[6]
–K28.5	C1.7	(CE1)	111	00001	001111	1010[29]	001111	1010[29]
+K28.5	C2.7	(CE2)	111	00010	110000	0101[30]	110000	0101[30]
					Running Disparity Violation Pattern			
Exception	C4.7	(CE4)	111	00100	110111	0101[31]	001000	1010[31]

Notes:

1. All codes not shown are reserved.
2. Notation for Special Character Byte Name is consistent with Fibre Channel and ESCON naming conventions. Special Character Code Name is intended to describe binary information present on I/O pins. Common usage for the name can either be in the form used for describing Data patterns (i.e., C0.0 through C31.7), or in hex notation (i.e., Cnn where nn = the specified value between 00 and FF).
3. C0.1 = Transmit Negative K28.5 (–K28.5+) disregarding Current RD when input is held for only one byte time. If held longer, transmitter begins sending the repeating transmit sequence –K28.5+, D21.4, D21.5, D21.5, (repeat all four bytes)... defined in X3.230 as the primitive signal “Idle word.” This Special Character input must be held for four (4) byte times or multiples of four bytes or it will be truncated by the new data. The receiver will never output this Special Character, since K28.5 is decoded as C5.0, C1.7, or C2.7, and the subsequent bytes are decoded as data.
4. C1.1 = Transmit Negative K28.5 (–K28.5+) disregarding Current RD when input is held for only one byte time. If held longer, transmitter begins sending the repeating transmit sequence –K28.5+, D21.4, D10.2, D10.2,(repeat all four bytes)... defined in X3.230 as the primitive signal “Receiver_Ready (R_RDY).” This Special Character input must be held for four (4) byte times or multiples of four bytes or it will be truncated by the new data.
   The receiver will never output this Special Character, since K28.5 is decoded as C5.0, C1.7, or C2.7 and the subsequent bytes are decoded as data.
5. C2.1 = Transmit either –K28.5+ or +K28.5– as determined by Current RD and modify the Transmission Character that follows, by setting its least significant bit to 1 or 0. If Current RD at the start of the following character is plus (+) the LSB is set to 0, and if Current RD is minus (–) the LSB becomes 1. This modification allows construction of X3.230 “EOF” frame delimiters wherein the second data byte is determined by the Current RD.
   For example, to send “EOFdt” the controller could issue the sequence C2.1–D21.4– D21.4–D21.4, and the HOTLink Transmitter will send either K28.5–D21.4–D21.4–D21.4 or K28.5–D21.5– D21.4–D21.4 based on Current RD. Likewise to send “EOFdti” the controller could issue the sequence C2.1–D10.4–D21.4–D21.4, and the HOTLink Transmitter will send either K28.5–D10.4–D21.4– D21.4 or K28.5–D10.5–D21.4– D21.4 based on Current RD.
   The receiver will never output this Special Character, since K28.5 is decoded as C5.0, C1.7, or C2.7, and the subsequent bytes are decoded as data.
6. 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 SVS = HIGH.
   The receiver will only output this Special Character if the Transmission Character being decoded is not found in the tables.

Maximum Ratings

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

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

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

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

DC Input Voltage.....–0.5V to +7.0V

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

Output Current into PECL Outputs (HIGH) .....–50 mA

Static Discharge Voltage.....> 4001V (per MIL-STD-883, Method 3015)

Latch-up Current.....> 200 mA

Operating Range

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

CY7B923/CY7B933 Electrical Characteristics Over the Operating Range[7]

Parameter	Description	Test Conditions		Min.	Max.	Unit
TTL OUTs, CY7B923: RP; CY7B933: Q_{0-7}, SC/D, RVS, RDY, CKR, SO						
V_{OHT}	Output HIGH Voltage	I_{OH} = - 2 mA		2.4		V
V_{OLT}	Output LOW Voltage	I_{OL} = 4 mA			0.45	V
I_{OST}	Output Short Circuit Current	V_{OUT} = 0V[8]		–15	–90	mA
TTL INs, CY7B923: D_{0-7}, SC/D, SVS, ENA, ENN, CKW, FOTO, BISTEN; CY7B933: RF, REFCLK, BISTEN						
V_{IHT}	Input HIGH Voltage		Com'l, Ind'l, and Mil	2.0	V_{CC}	V
			Ind'l and Mil (CKW and FOTO, only)	2.2	V_{CC}	V
V_{ILT}	Input LOW Voltage			–0.5	0.8	V
I_{IHT}	Input HIGH Current	$V_{IN} = V_{CC}$		–10	+10	µA
I_{ILT}	Input LOW Current	V_{IN} = 0.0V			–500	µA
Transmitter PECL-Compatible Output Pins: OUTA+, OUTA–, OUTB+, OUTB–, OUTC+, OUTC–						
V_{OHE}	Output HIGH Voltage (V_{CC} referenced)	Load = 50Ω to V_{CC} – 2V	Com'l	V_{CC} – 1.03	V_{CC} – 0.83	V
			Ind'l and Mil	V_{CC} – 1.05	V_{CC} – 0.83	V
V_{OLE}	Output LOW Voltage (V_{CC} referenced)	Load = 50Ω to V_{CC} – 2V	Com'l	V_{CC} – 1.86	V_{CC} – 1.62	V
			Ind'l and Mil	V_{CC} – 1.96	V_{CC} – 1.62	V
V_{ODIF}	Output Differential Voltage |(OUT+) – (OUT–)|	Load = 50Ω to V_{CC} – 2V		0.6		V
Receiver PECL-Compatible Input Pins: A/B, SI, INB						
V_{IHE}	Input HIGH Voltage		Com'l	V_{CC} – 1.165	V_{CC}	V
			Ind'l and Mil	V_{CC} – 1.14	V_{CC}	V
V_{ILE}	Input LOW Voltage		Com'l	2.0	V_{CC} – 1.475	V
			Ind'l and Mil	2.0	V_{CC} – 1.50	V
I_{IHE}[9]	Input HIGH Current	$V_{IN} = V_{IHE}$ Max.			+500	µA
I_{ILE}[9]	Input LOW Current	$V_{IN} = V_{ILE}$ Min.		+0.5		µA
Differential Line Receiver Input Pins: INA+, INA–, INB+, INB–						
V_{DIFF}	Input Differential Voltage |(IN+) – (IN–)|			50		mV
V_{IHH}	Highest Input HIGH Voltage				V_{CC}	V
V_{ILL}	Lowest Input LOW Voltage			2.0		V
I_{IHH}	Input HIGH Current	$V_{IN} = V_{IHH}$ Max.			750	µA
I_{ILL}[10]	Input LOW Current	$V_{IN} = V_{ILL}$ Min.		–200		µA

Notes:

7. See the last page of this specification for Group A subgroup testing information.
8. Tested one output at a time, output shorted for less than one second, less than 10% duty cycle.
9. Applies to A/B only.
10. Input currents are always positive at all voltages above V_{CC}/2.

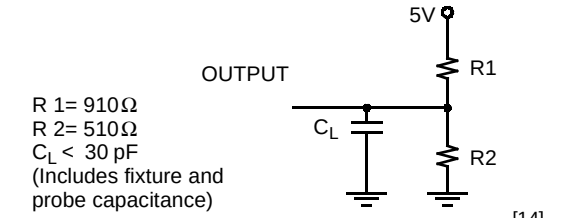
CY7B923/CY7B933 Electrical Characteristics Over the Operating Range[7] (continued)

Parameter	Description	Test Conditions		Min.	Max.	Unit
Miscellaneous				**Typ.**	**Max.**	
I_{CCT}[11]	Transmitter Power Supply Current	Freq. = Max.	Com'l	65	85	mA
			Ind'l and Mil	75	95	mA
I_{CCR}[12]	Receiver Power Supply Current	Freq. = Max.	Com'l	120	155	mA
			Ind'l and Mil	135	160	mA

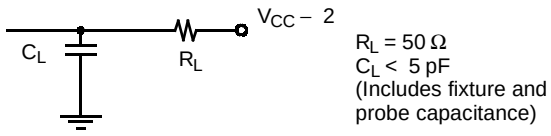
Capacitance[13]

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

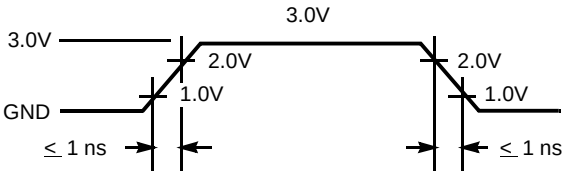
AC Test Loads and Waveforms



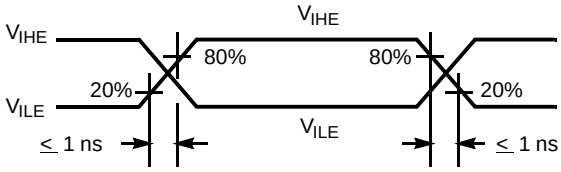
(a) TTL AC Test Load[14]



(b) PECL AC Test Load [14]



(c) TTL Input Test Waveform



(d) PECL Input Test Waveform

Notes:

11. Maximum I_{CCT} is measured with V_{CC} = Max., one PECL output pair loaded with 50 ohms to V_{CC} – 2.0V, and other PECL outputs tied to V_{CC}. Typical I_{CCT} is measured with V_{CC} = 5.0V, T_A = 25°C, one output pair loaded with 50 ohms to V_{CC} – 2.0V, others tied to V_{CC}, $\overline{\text{BISTEN}}$ = LOW. I_{CCT} includes current into V_{CCQ} (pin 9 and pin 22) only. Current into V_{CCN} is determined by PECL load currents, typically 30 mA with 50 ohms to V_{CC} – 2.0V. Each additional enabled PECL pair adds 5 mA to I_{CCT} and an additional load current to V_{CCN} as described. When calculating the contribution of PECL load currents to chip power dissipation, the output load current should be multiplied by 1V instead of V_{CC}.
12. Maximum I_{CCR} is measured with V_{CC} = Max., RF = LOW, and outputs unloaded. Typical I_{CCR} is measured with V_{CC} = 5.0V, T_A = 25°C, RF = LOW, $\overline{\text{BISTEN}}$ = LOW, and outputs unloaded. I_{CCR} includes current into V_{CCQ} (pins 21 and 24). Current into V_{CCN} (pin 9) is determined by the total TTL output buffer quiescent current plus the sum of all the load currents for each output pin. The total buffer quiescent current is 10mA max., and max. TTL load current for each output pin can be calculated as follows: Where R_L = equivalent

$$\frac{I_{I\,CCN}}{TTLPin} = \left[\frac{0.95 + (V_{CCN} - 5) * 0.3}{R_L} + C_L * \left[\frac{V_{CCN}}{2} + 1.5\right] * F_{pin}\right] * 1.1$$

load resistance, C_L = capacitive load, and F_{pin} = frequency in MHz of data on pin. A derating factor of 1.1 has been included to account for worst process corner and temperature condition.
13. Tested initially and after any design or process changes that may affect these parameters, but not 100% tested.
14. Cypress uses constant current (ATE) load configurations and forcing functions. This figure is for reference only.

Transmitter Switching Characteristics Over the Operating Range[7]

Parameter	Description	7B923-155		7B923		7B923-400		Unit
		Min.	Max	Min.	Max	Min.	Max	
t_{CKW}	Write Clock Cycle	62.5	66.7	30.3	62.5	25	62.5	ns
t_B	Bit Time[15]	6.25	6.67	3.03	6.25	2.5	6.25	ns
t_{CPWH}	CKW Pulse Width HIGH	6.5		6.5		6.5		ns
t_{CPWL}	CKW Pulse Width LOW	6.5		6.5		6.5		ns
t_{SD}	Data Set-Up Time[16]	5		5		5		ns
t_{HD}	Data Hold Time[16]	0		0		0		ns
t_{SENP}	Enable Set-Up Time (to insure correct $\overline{RP}$)[17]	$6t_B + 8$		$6t_B + 8$		$6t_B + 8$		ns
t_{HENP}	Enable Hold Time (to insure correct $\overline{RP}$)[17]	0		0		0		ns
t_{PDR}	Read Pulse Rise Alignment[18]	–4	2	–4	2	–4	2	ns
t_{PPWH}	Read Pulse HIGH[18]	$4t_B$–3		$4t_B$–3		$4t_B$–3		ns
t_{PDF}	Read Pulse Fall Alignment[18]	$6t_B$–3		$6t_B$–3		$6t_B$–3		ns
t_{RISE}	PECL Output Rise Time 20–80% (PECL Test Load)[13]		1.2		1.2		1.2	ns
t_{FALL}	PECL Output Fall Time 80–20% (PECL Test Load)[13]		1.2		1.2		1.2	ns
t_{DJ}	Deterministic Jitter (peak-peak)[13, 19]		35		35		35	ps
t_{RJ}	Random Jitter (peak-peak)[13, 20]		175		175		175	ps
t_{RJ}	Random Jitter (σ)[13,20]		20		20		20	ps

Receiver Switching Characteristics Over the Operating Range [7]

Parameter	Description	7B933-155		7B933		7B933-400		Unit
		Min.	Max	Min.	Max.	Min.	Max.	
t_{CKR}	Read Clock Period (No Serial Data Input), REFCLK as Reference[21]	–1	+1	–1	+1	–1	+1	%
t_B	Bit Time[22]	6.25	6.67	3.03	6.25	2.5	6.25	ns
t_{CPRH}	Read Clock Pulse HIGH	$5t_B$–3		$5t_B$–3		$5t_B$–3		ns
t_{CPRL}	Read Clock Pulse LOW	$5t_B$–3		$5t_B$–3		$5t_B$–3		ns
t_{RH}	$\overline{RDY}$ Hold Time	t_B–2.5		t_B–2.5		t_B–2.5		ns
t_{PRF}	$\overline{RDY}$ Pulse Fall to CKR Rise	$5t_B$–3		$5t_B$–3		$5t_B$–3		ns
t_{PRH}	$\overline{RDY}$ Pulse Width HIGH	$4t_B$–3		$4t_B$–3		$4t_B$–3		ns
t_A	Data Access Time[23, 24]	$2t_B$–2	$2t_B$+4	$2t_B$–2	$2t_B$+4	$2t_B$–2	$2t_B$+4	ns
t_{ROH}	Data Hold Time[23, 24]	t_B–2.5		t_B–2.5		t_B–2.5		ns
t_H	Data Hold Time from CKR Rise [23, 24]	$2t_B$–3		$2t_B$–3		$2t_B$–3		ns
t_{CKX}	REFCLK Clock Period Referenced to CKW of Transmitter[25]	–0.1	+0.1	–0.1	+0.1	–0.1	+0.1	%

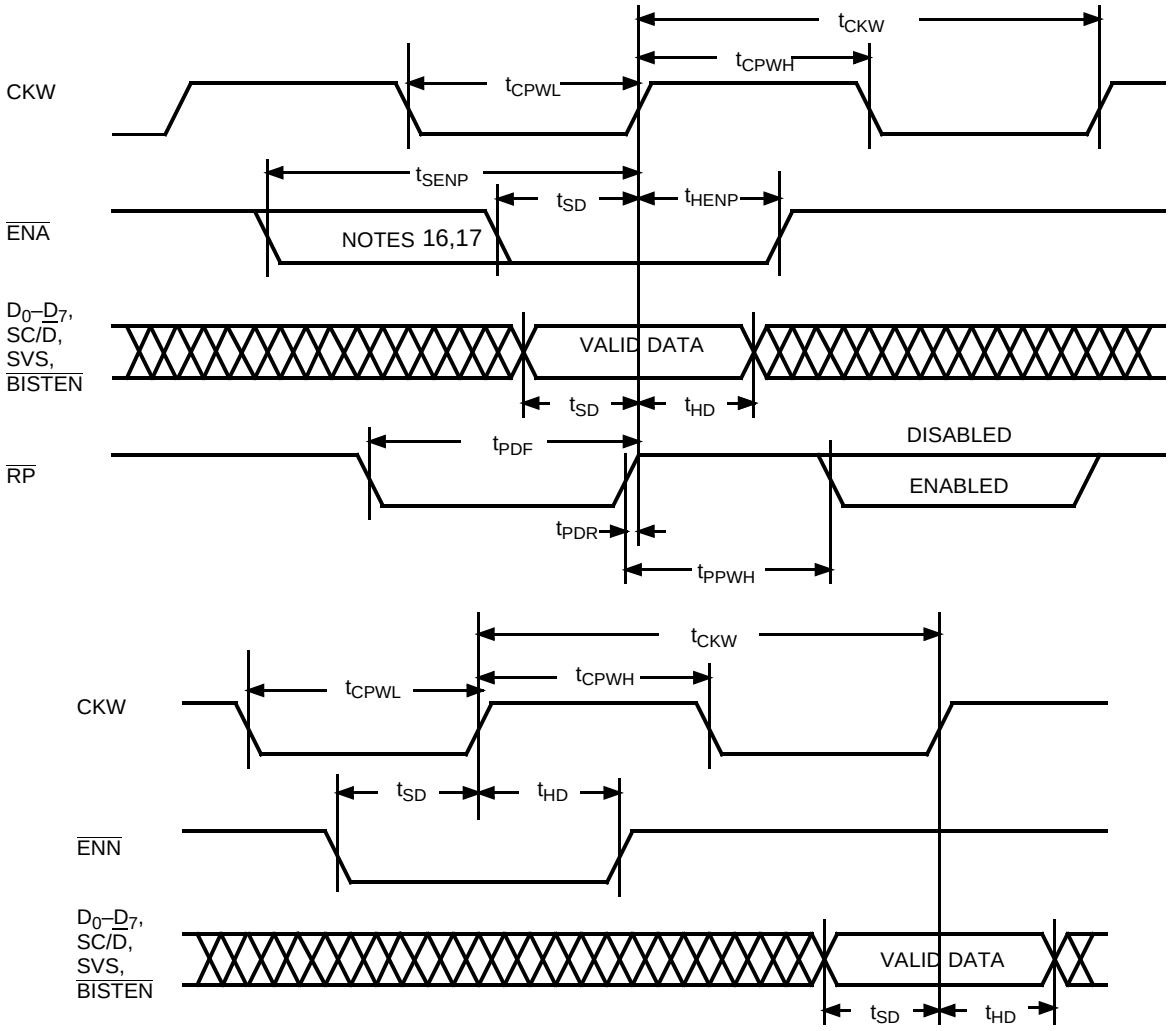
Notes:

15. Transmitter t_B is calculated as t_{CKW}/10. The byte rate is one tenth of the bit rate.
16. Data includes D_{0-7}, SC/$\overline{D}$, SVS, $\overline{ENA}$, $\overline{ENN}$, and $\overline{BISTEN}$. t_{SD} and t_{HD} minimum timing assures correct data load on rising edge of CKW, but not $\overline{RP}$ function or timing.
17. t_{SENP} and t_{HENP} timing insures correct $\overline{RP}$ function and correct data load on the rising edge of CKW.
18. Loading on $\overline{RP}$ is the standard TTL test load shown in part (a) of AC Test Loads and Waveforms except C_L = 15 pF.
19. While sending continuous K28.5s, RP unloaded, outputs loaded to 50Ω to V_{CC}–2.0V, over the operating range.
20. While sending continuous K28.7s, after 100,000 samples measured at the cross point of differential outputs, time referenced to CKW input, over the operating range.
21. The period of t_{CKR} will match the period of the transmitter CKW when the receiver is receiving serial data. When data is interrupted, CKR may drift to one of the range limits above.
22. Receiver t_B is calculated as t_{CKR}/10 if no data is being received, or t_{CKW}/10 if data is being received. See note.
23. Data includes Q_{0-7}, SC/$\overline{D}$, and RVS.
24. t_A, t_{ROH}, and t_H specifications are only valid if all outputs (CKR, $\overline{RDY}$, Q_{0-7}, SC/$\overline{D}$, and RVS) are loaded with similar DC and AC loads.
25. REFCLK has no phase or frequency relationship with CKR and only acts as a centering reference to reduce clock synchronization time. REFCLK must be within 0.1% of the transmitter CKW frequency, necessitating a ±500-PPM crystal.
26. The PECL switching threshold is the midpoint between the PECL– V_{OH} and V_{OL} specification (approximately V_{CC} – 1.35V). The TTL switching threshold is 1.5V.
27. Static alignment is a measure of the alignment of the Receiver sampling point to the center of a bit. Static alignment is measured by sliding one bit edge in 3,000 nominal transitions until a byte error occurs.
28. Error Free Window is a measure of the time window between bit centers where a transition may occur without causing a bit sampling error. EFW is measured over the operating range, input jitter < 50% Dj.

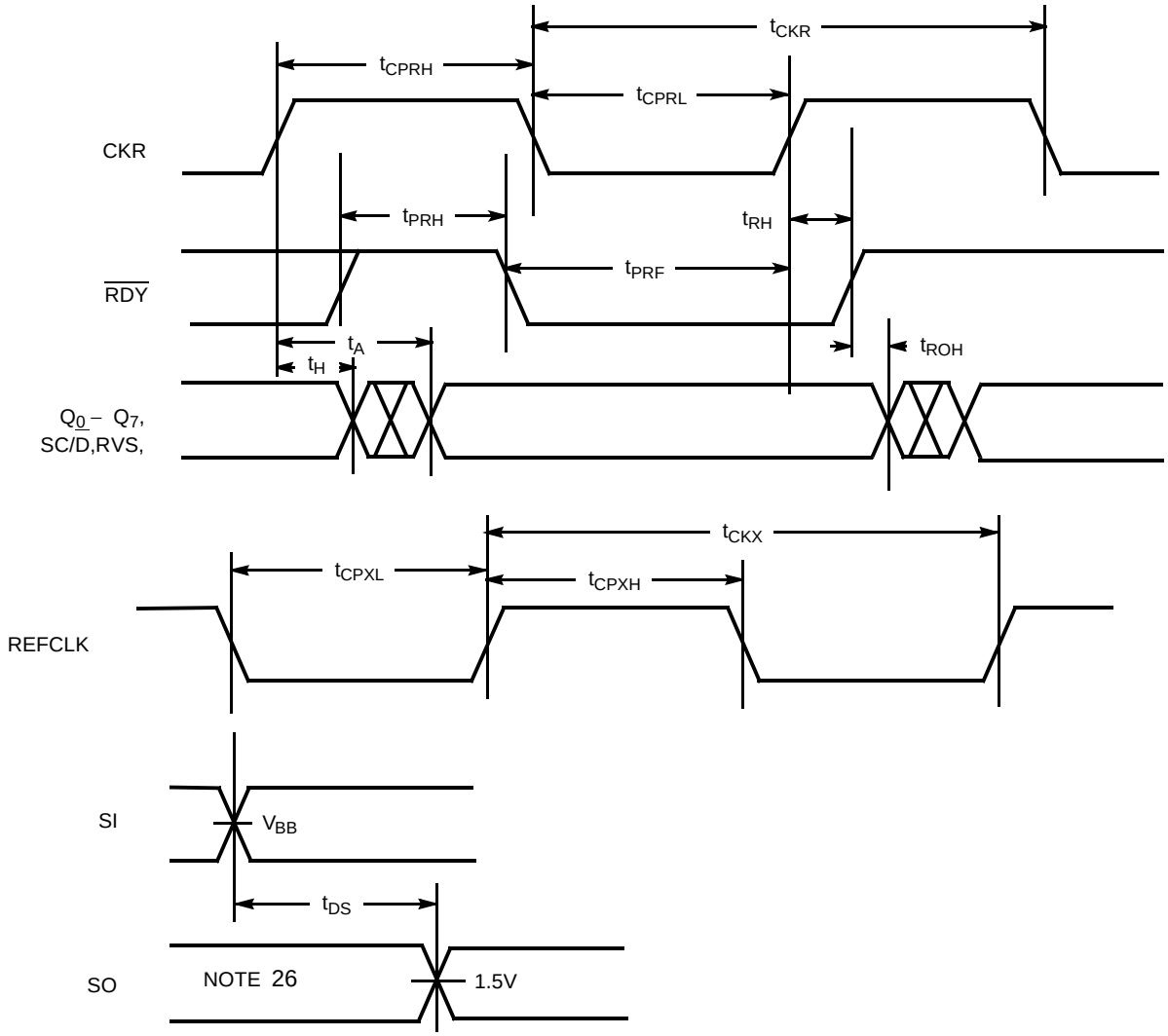
Receiver Switching Characteristics Over the Operating Range (continued)[7]

Parameter	Description	7B933-155		7B933		7B933-400		Unit
		Min.	Max	Min.	Max.	Min.	Max.	
t_{CPXH}	REFCLK Clock Pulse HIGH	6.5		6.5		6.5		ns
t_{CPXL}	REFCLK Clock Pulse LOW	6.5		6.5		6.5		ns
t_{DS}	Propagation Delay SI to SO (note PECL and TTL thresholds)[26]		20		20		20	ns
t_{SA}	Static Alignment[13, 27]		100		100		100	ps
t_{EFW}	Error Free Window[13, 28]	$0.9t_B$		$0.9t_B$		$0.9t_B$		

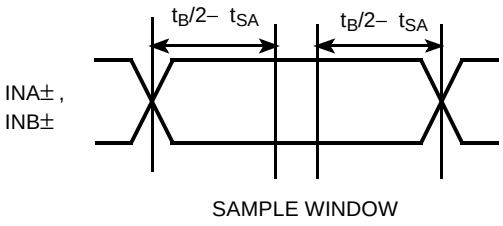
Switching Waveforms for the CY7B923 HOTLink Transmitter



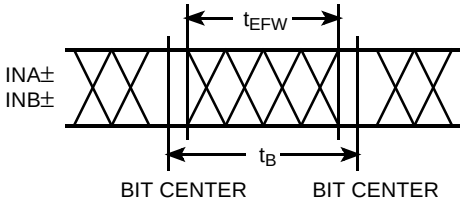
Switching W0aveforms for the CY7B933 HOTLink Receiver



Static Alignment



Error-free Window



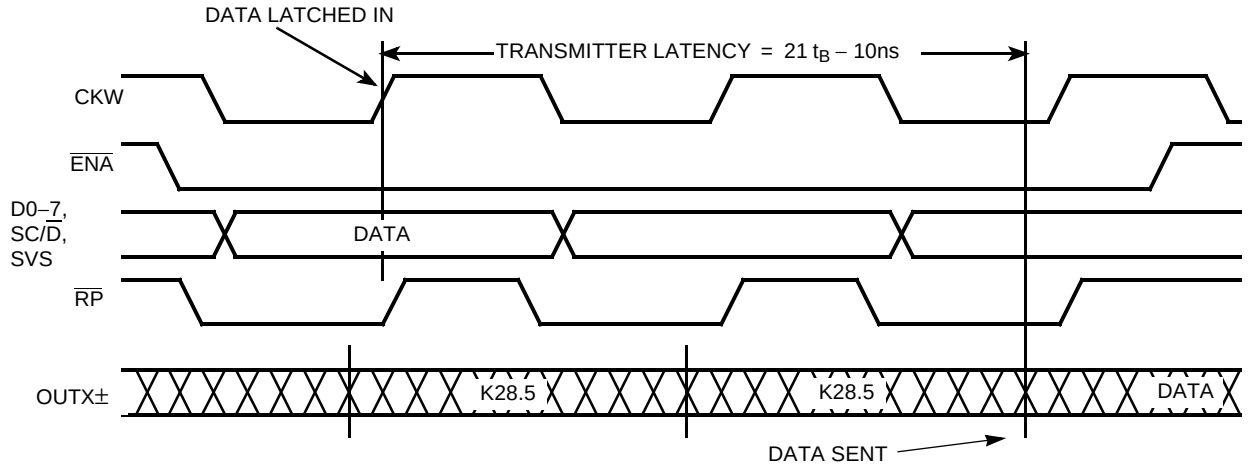


Figure 7. CY7B923 Transmitter Data Pipeline

Ordering Information

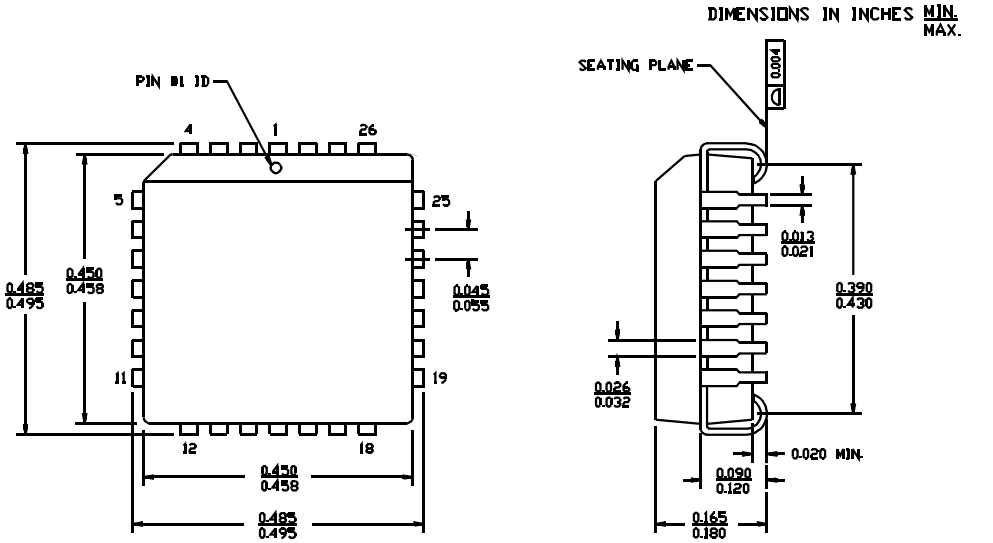
Speed	Ordering Code	Package Name	Package Type	Operating Range
Standard	CY7B923-JC	J64	28-Lead Plastic Leaded Chip Carrier	Commercial
	CY7B923-JI	J64	28-Lead Plastic Leaded Chip Carrier	Industrial
	CY7B923-SC	S21	28-Lead Small Outline IC	Commercial
400	CY7B923-400JC	J64	28-Lead Plastic Leaded Chip Carrier	Commercial
	CY7B923-400JI	J64	28-Lead Plastic Leaded Chip Carrier	Industrial
155	CY7B923-155JC	J64	28-Lead Plastic Leaded Chip Carrier	Commercial
	CY7B923-155JI	J64	28-Lead Plastic Leaded Chip Carrier	Industrial
Standard	CY7B933-JC	J64	28-Lead Plastic Leaded Chip Carrier	Commercial
	CY7B933-JI	J64	28-Lead Plastic Leaded Chip Carrier	Industrial
	CY7B933-SC	S21	28-Lead Small Outline IC	Commercial
400	CY7B933-400JC	J64	28-Lead Plastic Leaded Chip Carrier	Commercial
	CY7B933-400JI	J64	28-Lead plastic Leaded Chip Carrier	Industrial
155	CY7B933-155JC	J64	28-Lead Plastic Leaded Chip Carrier	Commercial
	CY7B933-155JI	J64	28-Lead Plastic Leaded Chip Carrier	Industrial

Notes:

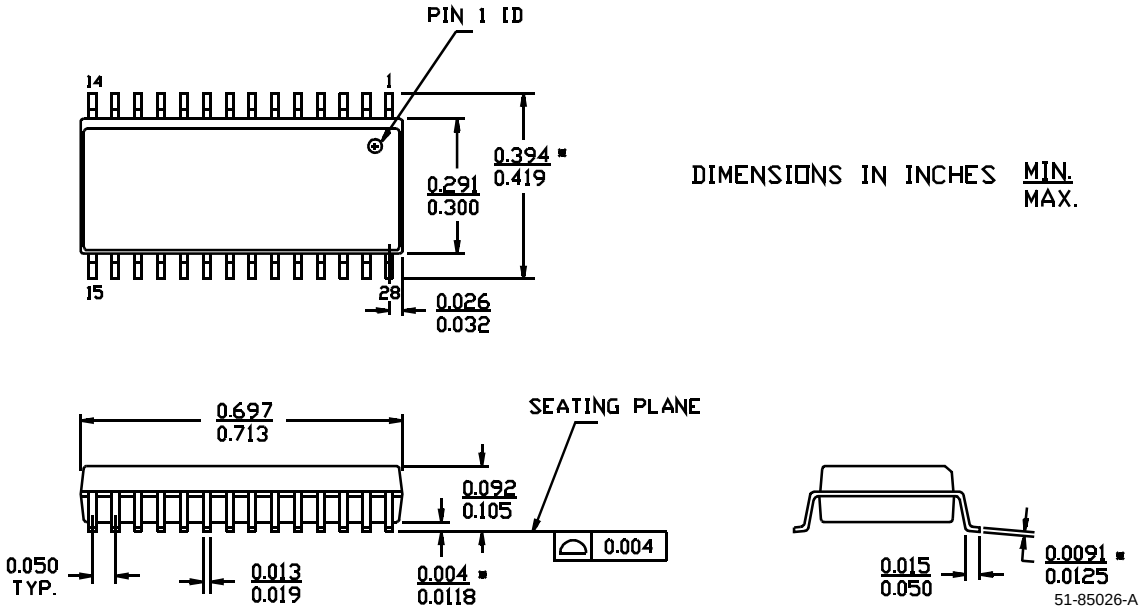
29. C1.7 = Transmit Negative K28.5 (–K28.5+) disregarding Current RD.
The receiver will only output this Special Character if K28.5 is received with the wrong running disparity. The receiver will output C1.7 if –K28.5 is received with RD+, otherwise K28.5 is decoded as C5.0 or C2.7.
30. C2.7 = Transmit Positive K28.5 (+K28.5–) disregarding Current RD.
The receiver will only output this Special Character if K28.5 is received with the wrong running disparity. The receiver will output C2.7 if +K28.5 is received with RD–, otherwise K28.5 is decoded as C5.0 or C1.7.
31. 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.

Package Diagrams

28-lead Plastic Leaded Chip Carrier J64



28-lead (300-mil) Molded SOIC S21



Document History Page

Document Title: CY7B923/CY7B933 HOTLink® Transmitter/Receiver Document Number: 38-02017				
REV	**ECN NO.**	**Issue Date**	**Orig. of Change**	**Description of Change**
**	105855	03/28/01	SZV	Changed from Spec number: 38-00189 to 38-02017
*A	112164	03/25/02	REV	Changed OUTA± pin description to improve consistency with diagram. Changed INA± pin description to include what to do with unused pairs of inputs. Changed Equation in note 6–old one made no sense.
*B	114562	03/27/02	BSS	Changed Hotlink™ Transmitter/Receiver to Hotlink® Transmitter/Receiver.
*C	125525	04/01/03	OOR	Removed all references to Military parts (Obsolete): CY7B923-LMB, CY7B933-LMB

Interfacing the CY7B923 and CY7B933 (HOTLink™) to a Wide Data Clocked FIFO

This application note considers general interfacing issues between the Cypress CY7B923/CY7B933 (HOTLink™) Transmitter/Receiver and Cypress clocked FIFOs. The focus is on applications with a 36-bit data bus requiring high data transfer rates. A parallel FIFO solution is recommended for applications requiring large data bandwidth. Four FIFOs can achieve parallel data transfers on and off a 36-bit bus at rates of up to 280 Mbytes/s. The HOTLink serial link can transfer data at a serial rate of 330 Mbits/s. The FIFOs act as asynchronous storage buffers between the data bus and the serial link.

Transmitter Interface

This section describes the design considerations of a high-speed transmitter interface with FIFO (First In First Out) data buffers. The design implements basic data transmission and serial link testing capabilities. The transmitter is intended to interface to a higher-level controller responsible for coordinating bus transactions and handling the various protocol layers. The design considerations are easily extended to handle specific design requirements.

The transmitter interface consists of four Cypress CY7C441/3-14 clocked FIFOs buffering data between a 36-bit data bus and a Cypress HOTLink Transmitter. A 4:1 multiplexer (9 bits wide) funnels the wide FIFO data into the HOTLink parallel port. A local state machine controller coordinates the flow of data between the FIFOs and HOTLink. The FIFO-data bus interface and local controller architecture are left unspecified for generality. A block diagram of the FIFO-HOTLink interface is shown in *Figure 1*.

Data Multiplexers

The 4:1 multiplexers are part of the critical data path timing. These multiplexers can be implemented in several ways. Standard high-speed 153 dual 4:1 multiplexers can be used. Five of these devices are needed to accommodate 9-bit data. 74ACT153s with a maximum t_{SZ} of 11.5 ns and t_{DZ} of 9.5 ns are sufficient.

The 4:1 multiplexers can also be implemented with three Cypress 16L8-10s. Each 16L8 can accommodate three 4:1 multiplexers. This solution provides a smaller footprint and improves the critical timing margins. Critical timing margins are discussed in the Critical Timing Analysis section of this application note.

Built-In Self-Test

The transmitter interface is capable of checking the functionality of the serial link by exercising the Built-In Self-Test (BIST) mode of HOTLink. To initiate BIST, the $\overline{\text{BISTEN}}$ pin is held LOW, resulting in the transmission of the sequence ...1 0 1 0... . The $\overline{\text{ENN}}$ (Enable Next Parallel Data) pin is then pulled LOW to enable transmission of the BIST test pattern. HOTLink will assert the $\overline{\text{RP}}$ (Read Pulse) pin LOW at the beginning of BIST and will pulse it HIGH once per BIST loop. $\overline{\text{RP}}$ can be used to count the number of BIST loops sent. During BIST, HOTLink ignores data at its parallel port and the FIFOs do not perform any reads.

Resetting the FIFOs

The higher-level controller should reset the FIFOs at power-up, before a new block of data is transmitted, or if an error is detected. Resetting or clearing the FIFOs is accomplished by pulsing the $\overline{\text{MR}}$ (Master Reset) pin on the FIFOs LOW. Neither a read nor a write can occur on the cycles immediately preceding, during, or following the assertion of $\overline{\text{MR}}$. $\overline{\text{MR}}$ must be glitch free. During the FIFO Master Reset cycle, the local transmitter controller should be in the WAIT state (see *Figure 2*). The higher-level controller is responsible for insuring that these conditions are met while performing the Master Reset cycle.

Transmitter Controller State Description

The local transmitter controller is responsible for reading data from the parallel FIFOs via the mux select lines and initiating the HOTLink BIST feature. The controller can be synthesized into a PLD or FPGA. Timing requirements of the controller are considered in the next section.

The local controller waits in the WAIT state while data is loaded into the FIFO. Meanwhile, HOTLink will transmit Idle special characters (K28.5). When the higher-level controller asserts the Transmit signal, the local transmitter controller issues a read ($\overline{\text{ENR}}$ LOW) to all the FIFOs and transitions to the TX0 state.

The transmit states (TX0–3) select data from the FIFOs in an ordered sequence. The TX0 state selects the byte out of FIFO0 for transmission and then transitions to the TX1 state. The TX1 state selects a byte out of FIFO1 and then transitions to the TX2 state. The TX3 state is responsible for checking the flags to determine if all of the FIFOs are empty, and then asserts $\overline{\text{ENR}}$ if they are not. (The controller can be designed to report an error if not all FIFOs are empty at the same time.) The transmit loop continues until all the FIFOs are empty or until Transmit is deasserted. Control then returns to the WAIT state. The Waiting signal should be monitored to determine when data transmission has ceased.

The state diagram of the local transmitter controller includes states for exercising the Built-In Self-Test capabilities of HOTLink. The local state machine enters the BIST state from the WAIT state when the higher-level controller asserts the Test signal. BIST is exited when Test is deasserted. The higher-level controller monitors $\overline{\text{RP}}$ for BIST loop counting. $\overline{\text{RP}}$ will pulse LOW one time per BIST loop. *Figure 2* illustrates the controller state diagram.

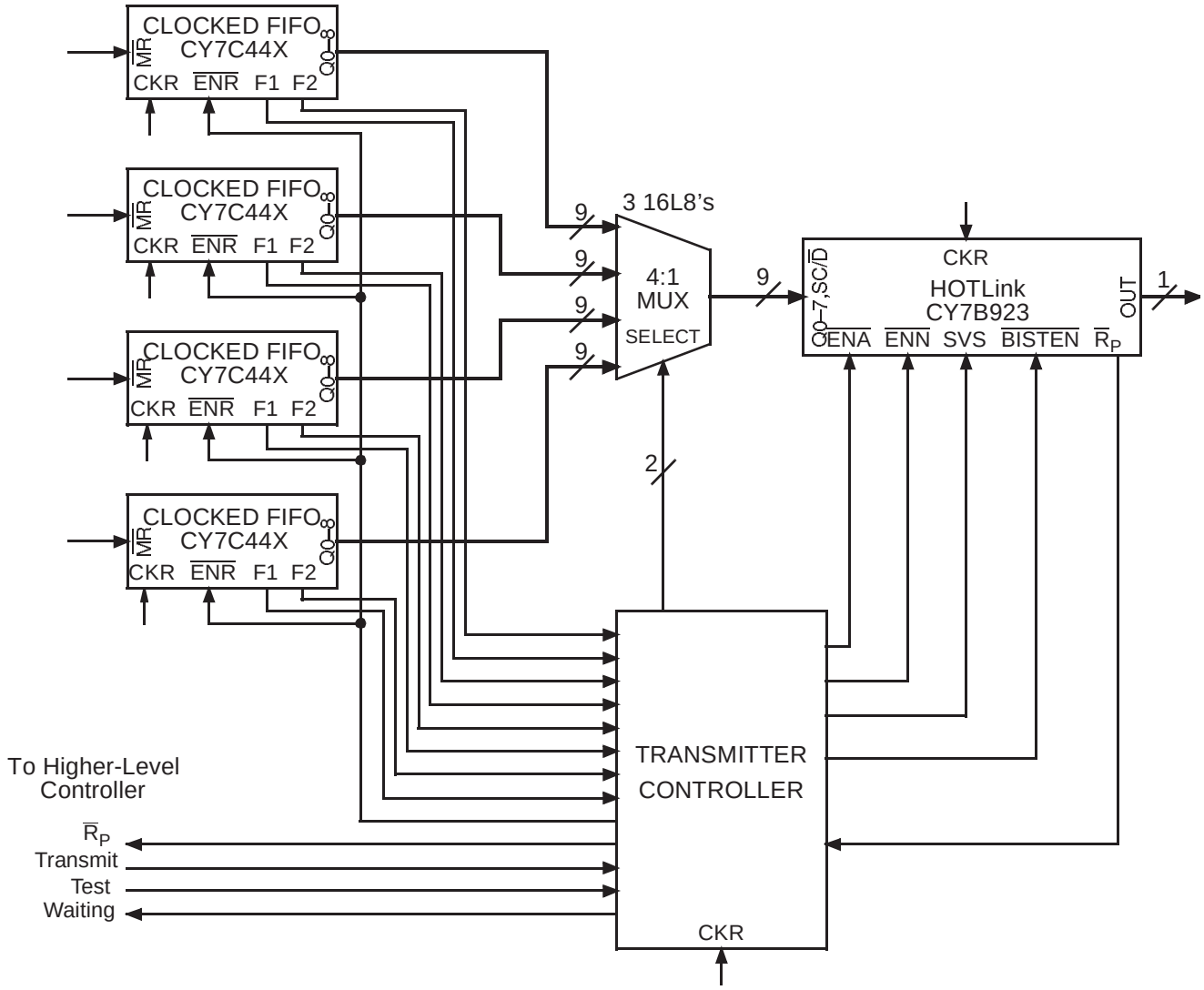


Figure 1. Transmitter Interface Block Diagram

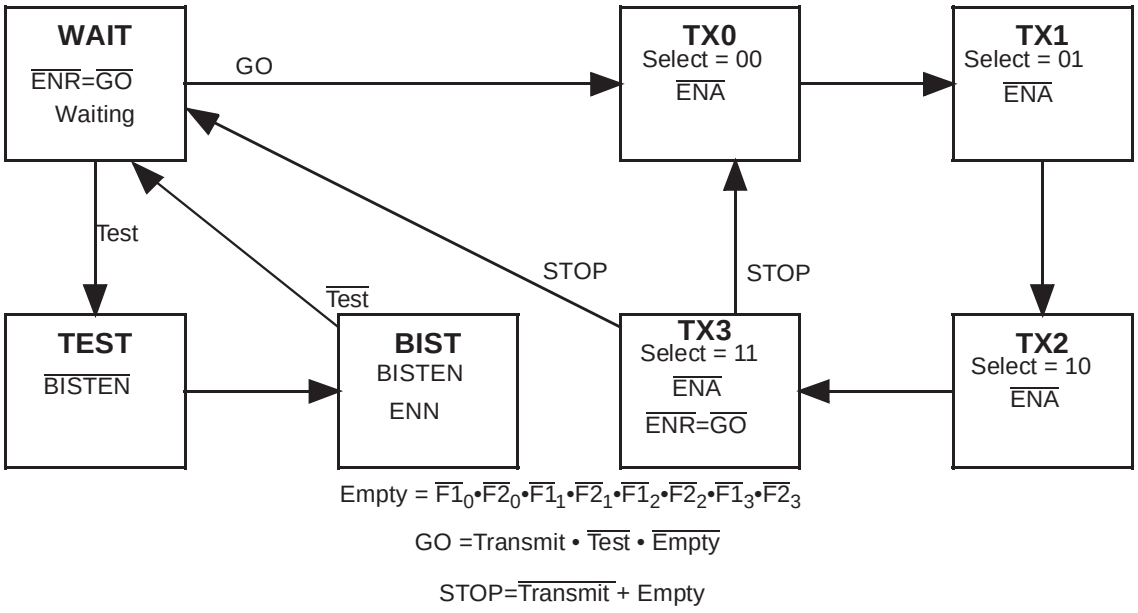


Figure 2. Transmitter Controller State Diagram

Critical Timing Analysis

The timing analysis in *Figure 3* highlights three critical data timing paths. The first critical path arises in the WAIT or TX3 states from the delay associated with decoding the flags and generating the read enable for the clocked FIFOs. The FIFO delay for generating the flags, t_{FD}, is 10 ns. The delay due to the controller decoding the flags and generating the enable is represented as t_{PD}. The read enable set-up time for FIFOs, t_{SEN}, is 7 ns ($t_{SEN} > t_{SD}$). The constraint imposed upon the controller is

$$t_{PD} \le t_{CKW} - t_{FD} - t_{SEN}$$

With a 30-ns clock period, the signal propagation delay through the controller must be $t_{PD} \le 13$ ns excluding trace delays and clock skew. This timing analysis assumes that the state register outputs are fed back to the controller before the flags signals are valid ($t_{CO} < t_{FD}$).

The second critical timing case assumes that data is available at the mux before the data selector signals ($t_{SEL} > t_A$, where the delay from a clock edge to the arrival of the data selectors at the muxes is t_{SEL}). The delay from the selector pins to valid output data is t_{SZ}. The data set-up time to HOTLink, t_{SD}, is 5 ns. The critical timing associated with this path is

$$t_{SEL} + t_{SZ} + t_{SD} \le t_{CKW}$$

The time to generate the data selectors from the controller is minimized by using the low-order bits of the state machine as the selectors and assigning TX0–3 to these states. This decreases the hardware required for the controller and reduces the selector signal-generation time to the clock-to-output time (t_{CO}) of the state registers. Assuming a 30-ns clock and t_{CO}=10 ns, the mux delay must be $t_{SZ} \le 15$ ns.

The delay through the mux from valid input data to valid output is t_{DZ}. Assuming that the data selectors arrive before the data ($t_A > t_{SEL}$), the critical timing of this path is given by

$$t_A + t_{DZ} + t_{SD} \le t_{CKW}$$

The data access time of the FIFOs, t_A, is 10 ns. With a 30-ns clock period, the constraint imposed upon the mux is $t_{DZ} \le 15$ ns, assuming no trace delays or clock skew.

Receiver Interface

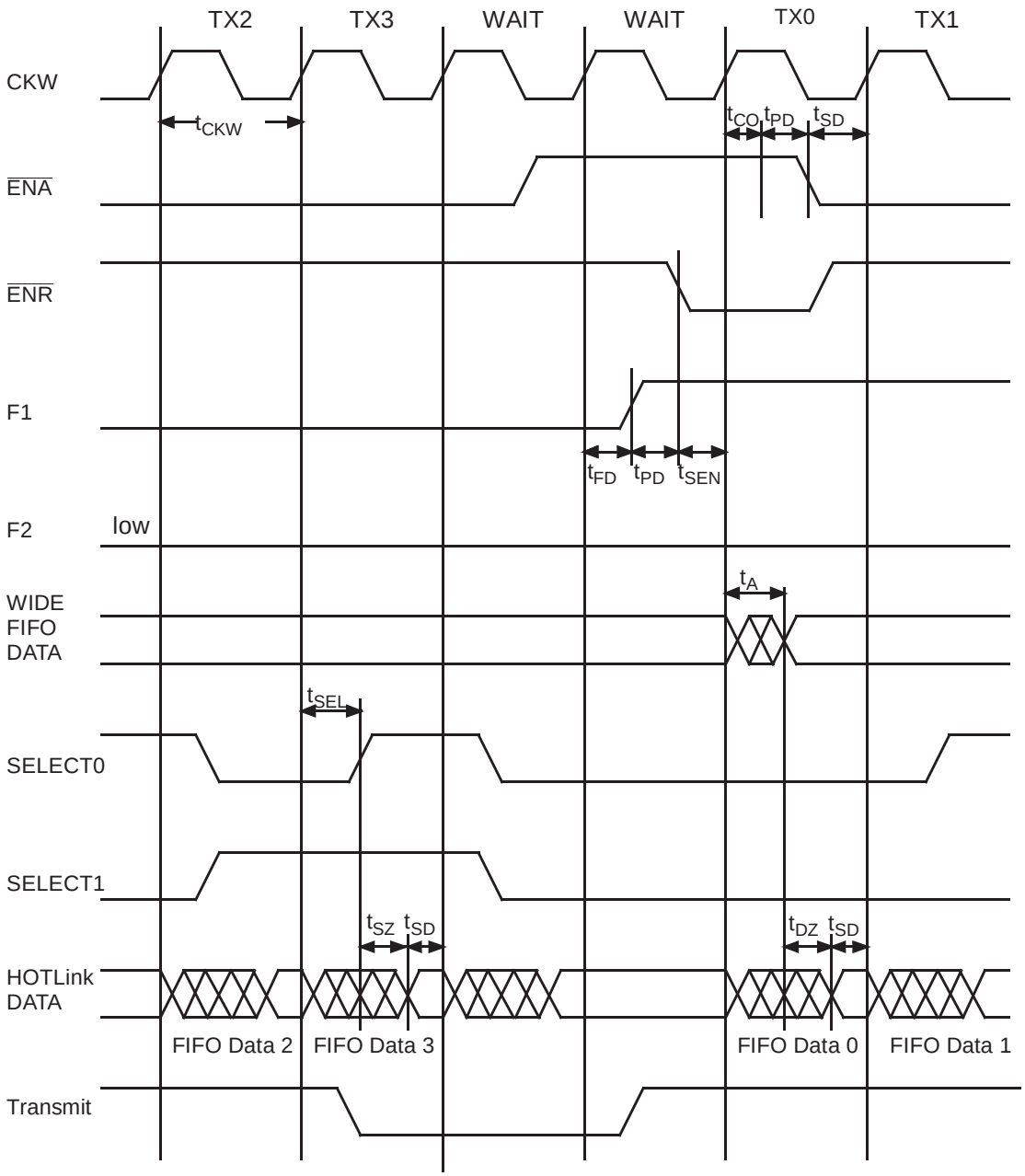
In this section a solution is presented for interfacing a HOTLink receiver to a 36-bit data bus. Control of the interface is simple and is easily adapted to system requirements. The four parallel CY7C451/3-14 FIFOs provide a high-speed interface to the data bus, allowing parallel transfer at rates up to 280 Mbytes/s. The serial link can receive data at serial rates up to 330 Mbits/s. The receiver interface is designed to provide proper word alignment in the FIFOs after synchronization to the data stream has been achieved. *Figure 4* shows a block diagram of the HOTLink-FIFO receiver interface.

Reframe

The receiver interface must synchronize itself to the incoming data and then store the data in the FIFOs with proper word alignment. The HOTLink RF (Reframe) input is used to synchronize the receiver to the transmitted data. Assertion of RF forces HOTLink to synchronize its internal bit counter with the boundary of a K28.5 character. HOTLink will respond by asserting $\overline{RDY}$ LOW when the first K28.5 is received. Reframing may be performed before data storage in order to synchronize HOTLink to the incoming serial data stream.

Idle Decoder

The Idle Decoder decodes the three types of idle characters: K28.5 (C5.0), –K28.5 (C1.7), +K28.5 (C2.7). These idle characters are used to signal the boundary of data words to be read into the FIFOs. A logic equation for the Idle Decoder is contained in *Figure 5*. A–H refer to HOTLink output pins Q0–Q7. When the Receive1 signal is asserted by the higher-level controller to the local controller, reception of any of these idle characters will trigger received data to be continually stored in the FIFOs starting with FIFO0 (*Figure 5*). The combinatorial delay through the decoder is modeled as t_{ID}.



Critical Timing Analysis

1. Read enable set-up time:

   $t_{FD} + t_{PD} + t_{SEN} \leq t_{CKW}$

2. HOTLink data set-up time from MUX data select:

   $t_{SEL} + t_{SZ} + t_{SD} \leq t_{CKW}$

3. HOTLink data set-up time from FIFO data access:

   $t_{A} + t_{DZ} + t_{SD} \leq t_{CKW}$

Figure 3. Transmitter Timing Diagram

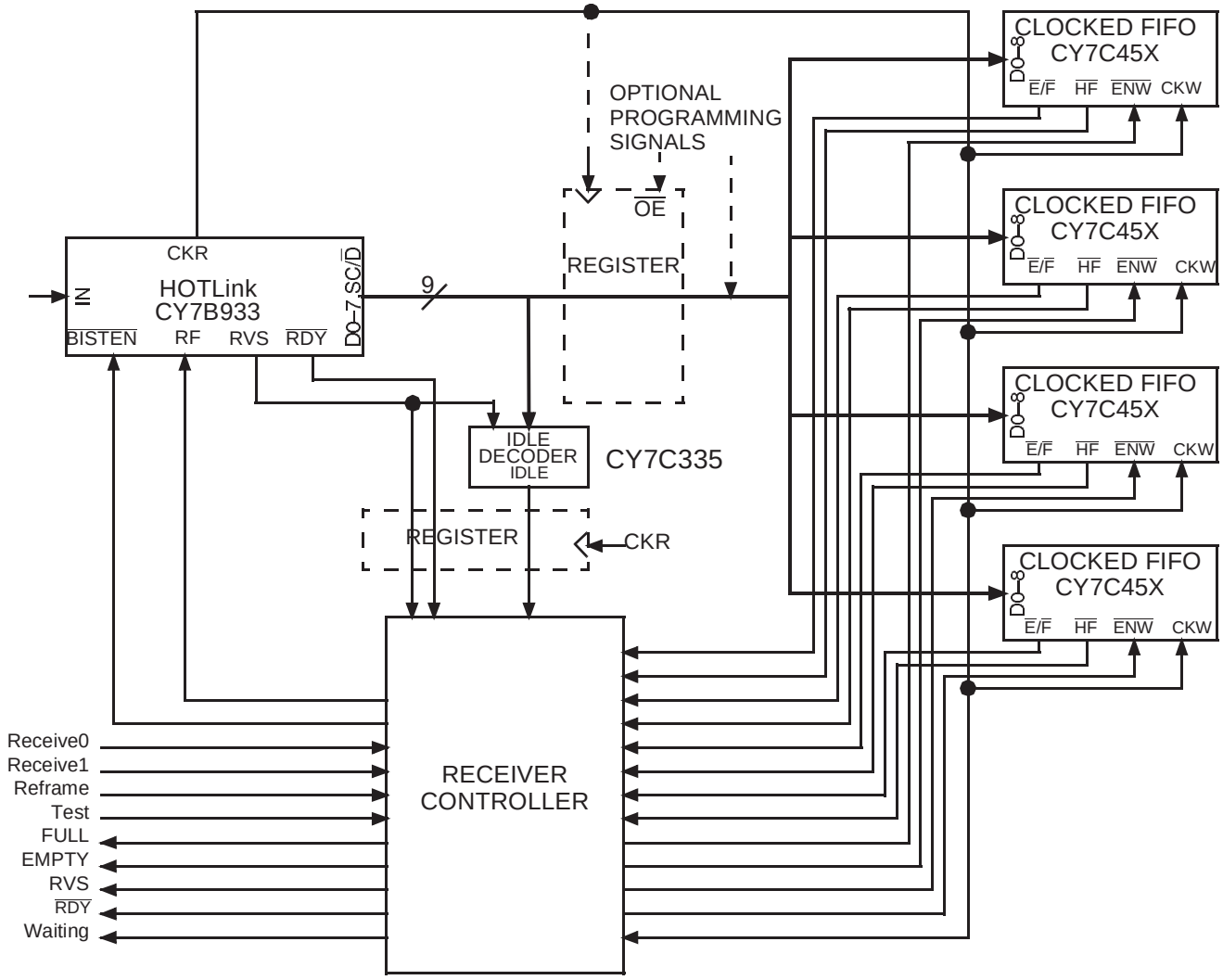


Figure 4. Receiver Interface Block Diagram

Data Path and Controller

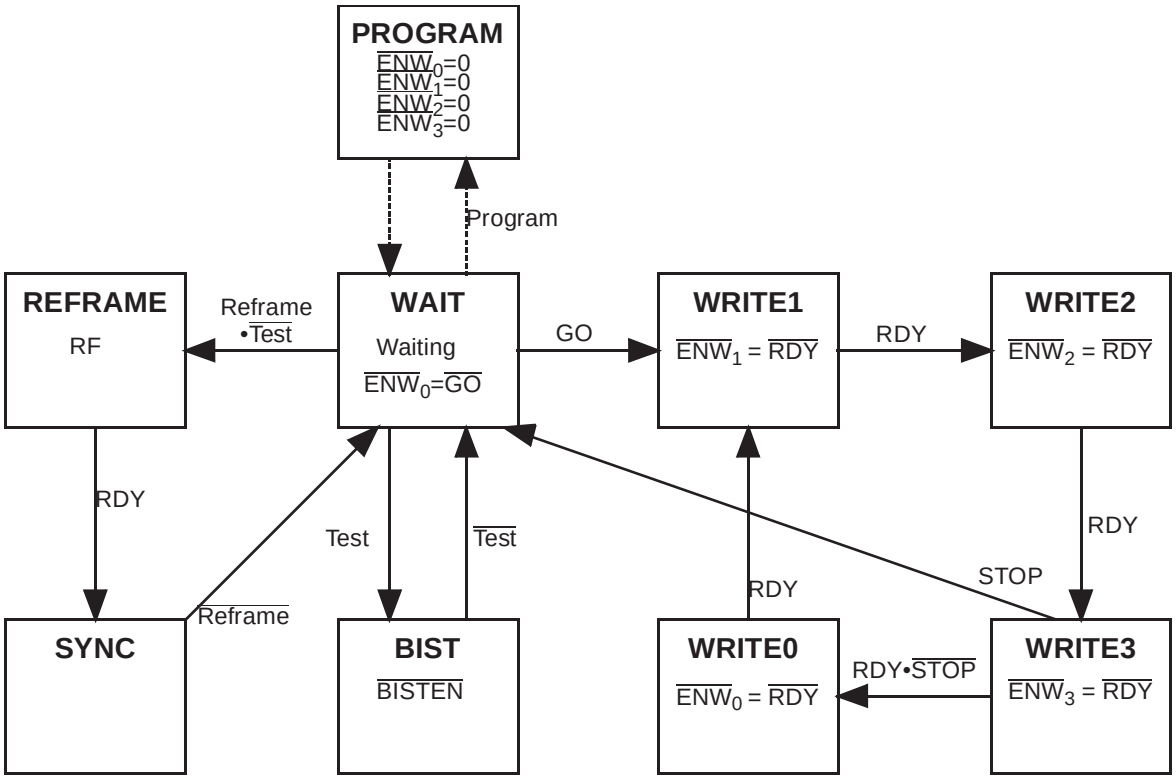
The HOTLink receiver parallel port interfaces directly to the FIFOs' write ports. A pipeline register may be inserted to improve timing margins or allow the FIFOs to be programmed. A local receiver controller coordinates the data flow and enables the HOTLink receiver BIST feature. The local receiver controller interfaces to a higher-level controller that coordinates all of the protocol layers of the link and the data bus transactions.

The higher-level controller instructs the local controller when to start data reception. A K28.5 character delimits the start of a data transmission. When this character is detected by either HOTLink or the Idle Decoder, the local controller writes the incoming data into the 45X FIFOs. The writing process continues until the higher-level controller signals the local receiver controller to stop.

The FIFO flags are decoded to signal when the FIFOs are empty or are full. A full FIFO will ignore attempted writes. The 45X FIFO features programmable Almost Full and Almost Empty flags that can assist in signaling when the FIFO is becoming too full. Programmable flag signals are left out of the design for clarity.

The Cypress 45X family of clocked FIFOs feature three-state data output drivers for direct interfacing to a data bus. The higher-level controller is responsible for reading words from the FIFOs' read port to the data bus.

The architecture of the local receiver controller is unspecified, but can be implemented with a PLD or FPGA. State machine descriptions and a timing analysis of the data path and local receiver controller are provided in the next sections.



$$IDLE = \overline{H}\bullet\overline{G}\bullet\overline{F}\bullet\overline{E}\bullet\overline{D}\bullet C\bullet\overline{B}\bullet A\bullet SC/\overline{D} + H\bullet G\bullet F\bullet\overline{E}\bullet\overline{D}\bullet\overline{C}\bullet\overline{B}\bullet A\bullet SC/\overline{D} + H\bullet G\bullet F\bullet\overline{E}\bullet\overline{D}\bullet\overline{C}\bullet B\bullet\overline{A}\bullet SC/\overline{D}$$

$$GO = (Receive0\bullet RDY + Receive1\bullet IDLE)\bullet\overline{Reframe}\bullet\overline{Test}$$

$$STOP = \overline{Receive0 + Receive1}$$

$$Empty = \overline{E}/\overline{F}_0\bullet\overline{HF}_0\bullet\overline{E}/\overline{F}_1\bullet\overline{HF}_1\bullet\overline{E}/\overline{F}_2\bullet\overline{HF}_2\bullet\overline{E}/\overline{F}_3\bullet\overline{HF}_3$$

$$Full = \overline{E}/\overline{F}_0\bullet\overline{HF}_0\bullet\overline{E}/\overline{F}_1\bullet\overline{HF}_1\bullet\overline{E}/\overline{F}_2\bullet\overline{HF}_2\bullet\overline{E}/\overline{F}_3\bullet\overline{HF}_3$$

Figure 5. Receiver Controller State Diagram

Optional Pipeline Registers

The optional pipeline registers increase the interface speed by capturing the $\overline{RDY}$ pulse and easing timing constraints on the controller. $\overline{RDY}$ is a 60% LOW duty cycle signal shaped for interfacing to generic asynchronous FIFOs. The LOW phase of $\overline{RDY}$ leaves less than $^1/_2t_{CKR}$ – 10 ns to generate the FIFO write enable and meet the FIFOs' set up time. A 40-ns clock period (250 Mbit/s) allows 10 ns for the local controller to generate a FIFO enable. This time shrinks to 5 ns when a clock period of 30 ns (330 Mbit/s) is used. The optional pipeline register captures the delayed $\overline{RDY}$ pulse and allows it to be processed earlier during the next clock cycle. The data and control signals must also be delayed by one clock cycle to ensure proper data alignment. A single CY7C335 PLD can be used to accommodate the data pipeline registers, the Idle Decoder, and the control signal delay registers. The timing implications of the registers are considered in the section on critical timing analysis.

The pipeline registers also isolate the HOTLink parallel port from the FIFO write ports while programming the FIFOs. A data pipeline register with three-state output drivers should be used so that data from an external source can be used to program the FIFOs. Additional states and control signals must be added to the controller. Programming is performed during the FIFO master reset cycle.

Built-In Self-Test

The Built-In Self-Test mode is exercised by asserting $\overline{BISTEN}$. Upon entering BIST, HOTLink will await the BIST initialization code and then assert $\overline{RDY}$ LOW when the code has been received. $\overline{RDY}$ will pulse HIGH once per received BIST loop. RVS will pulse HIGH if a byte pattern mismatch occurs. $\overline{RDY}$ and RVS can be monitored by the high-level controller to characterize the error rate.

Resetting and Programming the FIFOs

The higher-level controller should reset the FIFOs after power-up, before a new block of data is received, if an error occurs or in order to program the FIFOs. Resetting or program-

ming the FIFOs is accomplished by pulsing the $\overline{MR}$ pin on the FIFOs LOW. Neither a read nor a write can occur on the cycles immediately preceding, during, or following the assertion of $\overline{MR}$ unless the FIFOs are being programmed. FIFO programming information is contained in the CY7C451/3 data sheet. $\overline{MR}$ must be glitch free. The receiver controller should only be in the WAIT or PROGRAM states during a master reset. The higher-level controller is responsible for insuring that these conditions are met.

Controller State Description

A state diagram for the receiver interface controller is shown in *Figure 5*. Five simple signals control the interface. The Receive0 and Receive1 signals are used to initiate and stop the reception of data. Reframe is used to synchronize the receiver to the serial data stream. Test causes HOTLink to perform BIST. Waiting is an output signal that indicates that the receiver is in the WAIT state.

Full and Empty signals are decoded for use by the higher-level controller to assist in managing data out of the FIFOs. The programmable flags may also be decoded but are not shown. A full FIFO ignores attempted writes resulting in lost data. Monitoring the state of the FIFOs is the responsibility of the higher-level controller. Resetting the FIFOs by pulsing $\overline{MR}$ LOW is also the responsibility of the higher-level controller.

The REFRAME state is used to synchronize the receiver to the incoming serial data stream. The REFRAME state asserts RF to the HOTLink receiver, signaling it to synchronize its internal bit counter with the first-received K28.5 character. $\overline{RDY}$ will pulse LOW when a synchronized K28.5 character is available. The controller will transition back to the WAIT state when synchronization is achieved and the Reframe signal is deasserted.

Receive0 and Receive1 initiate the storing of data in the FIFOs from the WAIT state. The assertion of Receive0 causes the controller to look for the assertion of $\overline{RDY}$ in order to begin data storage. The assertion of Receive1 causes the controller to look for the assertion of IDLE in order to begin data storage. The received K28.5 is written into FIFO0 and then the write loop is entered. The choice of which receive mode to use depends on the serial link protocol.

The write loop continually writes valid characters into the FIFOs. $\overline{ENW}0–3$ are cycled in order as the data is received. The fullness of the FIFOs is ignored by the controller. The higher-level controller monitors the Full flag signal and takes corrective action if the FIFOs become too full. The deassertion of both receive signals will end the writing process and return control back to the WAIT state on the next word boundary. The higher-level controller should monitor the Waiting signal to determine when receiver controller has returned to the WAIT state.

The BIST state is included for handling the Built-In Self-Test. During BIST, writing to the FIFOs is disabled. HOTLink signals are passed on to the higher-level controller for error analysis. RVS will signal character reception errors. $\overline{RDY}$ will pulse HIGH once per BIST loop and should be used to count the number of completed BIST loops.

A single PROGRAM state that writes an external program word to all of the FIFOs in parallel can be added to the state machine. This state is entered and exited during a FIFO master reset cycle. The higher-level controller should assert $\overline{MR}$ LOW, put the data pipeline register in the high-impedance state, and then drive the external program word to the FIFO write ports. The higher-level controller then puts the local controller in the PROGRAM state. The program word is written into the FIFOs' internal program registers when the local controller exits the PROGRAM state.

Critical Timing Analysis

A critical timing analysis of both the pipelined and unpipelined receiver interfaces is presented in this section. A timing diagram with critical timing equations is provided in *Figure 6* for the receiver interface that does not include the optional pipeline registers. Timing for the pipelined case is very similar. The analysis assumes that the state register bits are valid before any critical signals are available to the controller.

The critical timing path constrains the propagation delays associated with the local receiver controller and Idle Decoder. The combinatorial timing delay through the controller is modeled as t_{PD}. The combinatorial delay through the Idle Decoder is modeled as t_{ID}.

Unpipelined Timing

The timing for the unpipelined configuration is as follows. Assuming t_{CKR}=30 ns and t_{SEN}=7 ns, the propagation delays are

Write enable generation time from $\overline{RDY}$ LOW:

$$t_{PD} \le 1/2\ t_{CKR} - t_{SEN} - 3ns = 5\ ns$$

IDLE generation time from data:

$$t_{ID} \le 4/5\ t_{CKR} - t_{SEN} - t_{PD} - 3\ ns = 9\ ns$$

These constraints require (approximately) $t_{PD} \le 5$ ns and $t_{ID} \le 9$ ns. With a 40 ns clock cycle, these timing constraints are relaxed to $t_{PD} \le 10$ ns and $t_{ID} \le 12$ ns.

Pipelined Timing

With the optional pipeline registers inserted the timing margins of the control logic are eased. Assuming the register access time is t_{AR}=10 ns and the register set up time is t_{SU}=5 ns

Write enable generation time from clock:

$$t_{PD} \le t_{CKR} - t_{SEN} - t_{AR} = 13\ ns$$

IDLE generation time from data:

$$t_{ID} \le 4/5\ t_{CKR} - t_{SU} - 3\ ns = 14ns$$

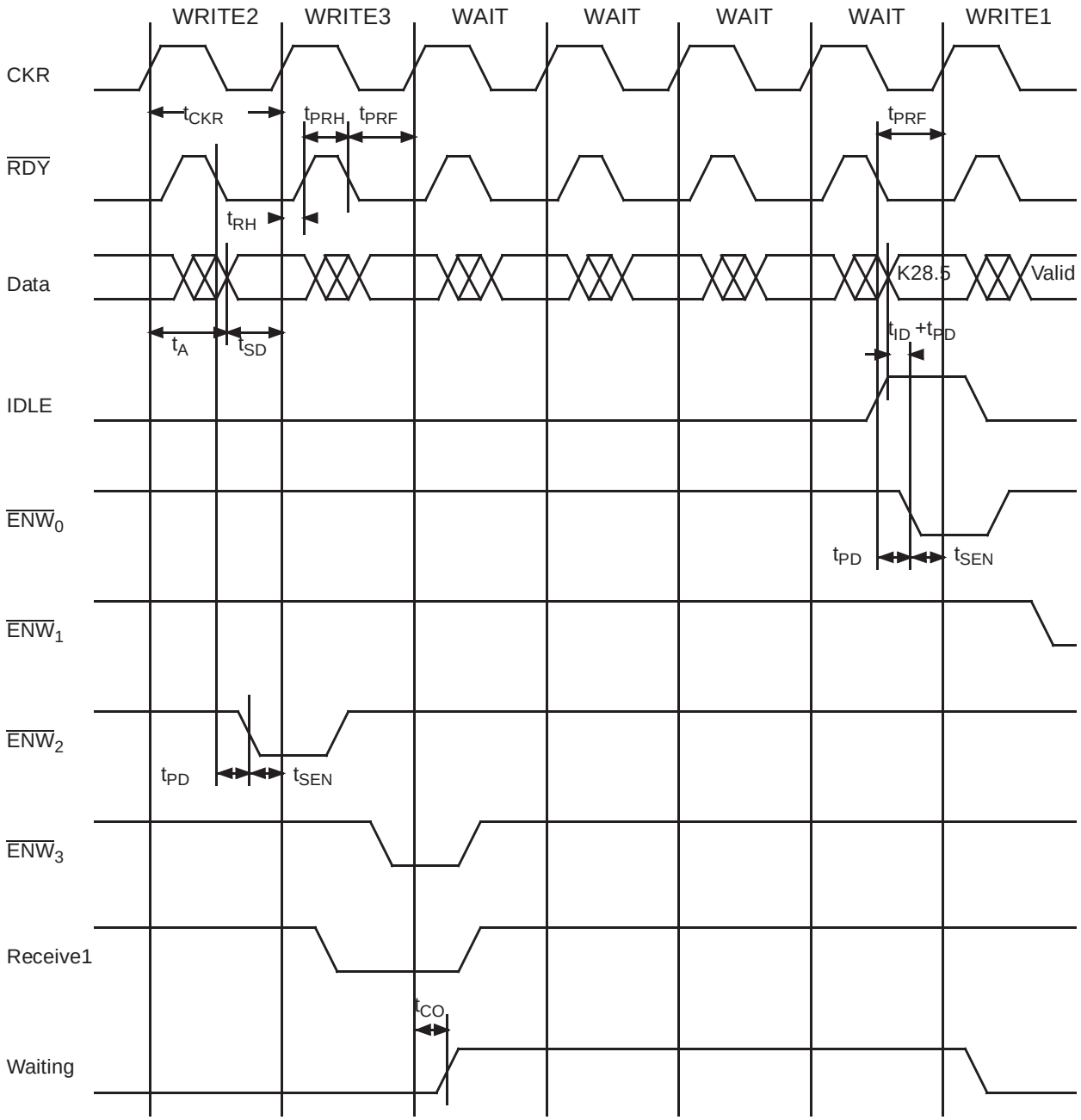
$\overline{RDY}$ capture timing:

$$t_{SU} \le 1/2t_{CKR} - 3ns = 12\ ns$$

The pipeline registers ease the receiver control logic timing margins to (approximately) 13 ns. The entire pipeline circuitry, including the Idle Decoder, can be synthesized into a single CY7C335-83 PLD while meeting these timing constraints.

Conclusion

The HOTLink Transmitter/Receiver interfaces to wide data FIFOs can operate at speeds of up to 330 Mbits/s with minimal interface logic. State machine controllers ensure proper word alignment during data transfers over the HOTLink serial link and provide Built-In Self-Test capability. Critical timing equations are provided. The interface designs are easily modified to meet specific demands.



Critical Timing Analysis:

1. Data set-up time

   $t_A + t_{SD} \le t_{CKR}$

2. Write enable set-up time from $\overline{RDY}$ LOW

   $t_{PD} + t_{SEN} \le t_{PRF}$

3. Write enable set-up time from idle HIGH:

   $t_A + t_{ID} + t_{PD} + t_{SEN} \le t_{CKR}$

Figure 6. Receiver Timing Diagram