# 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.

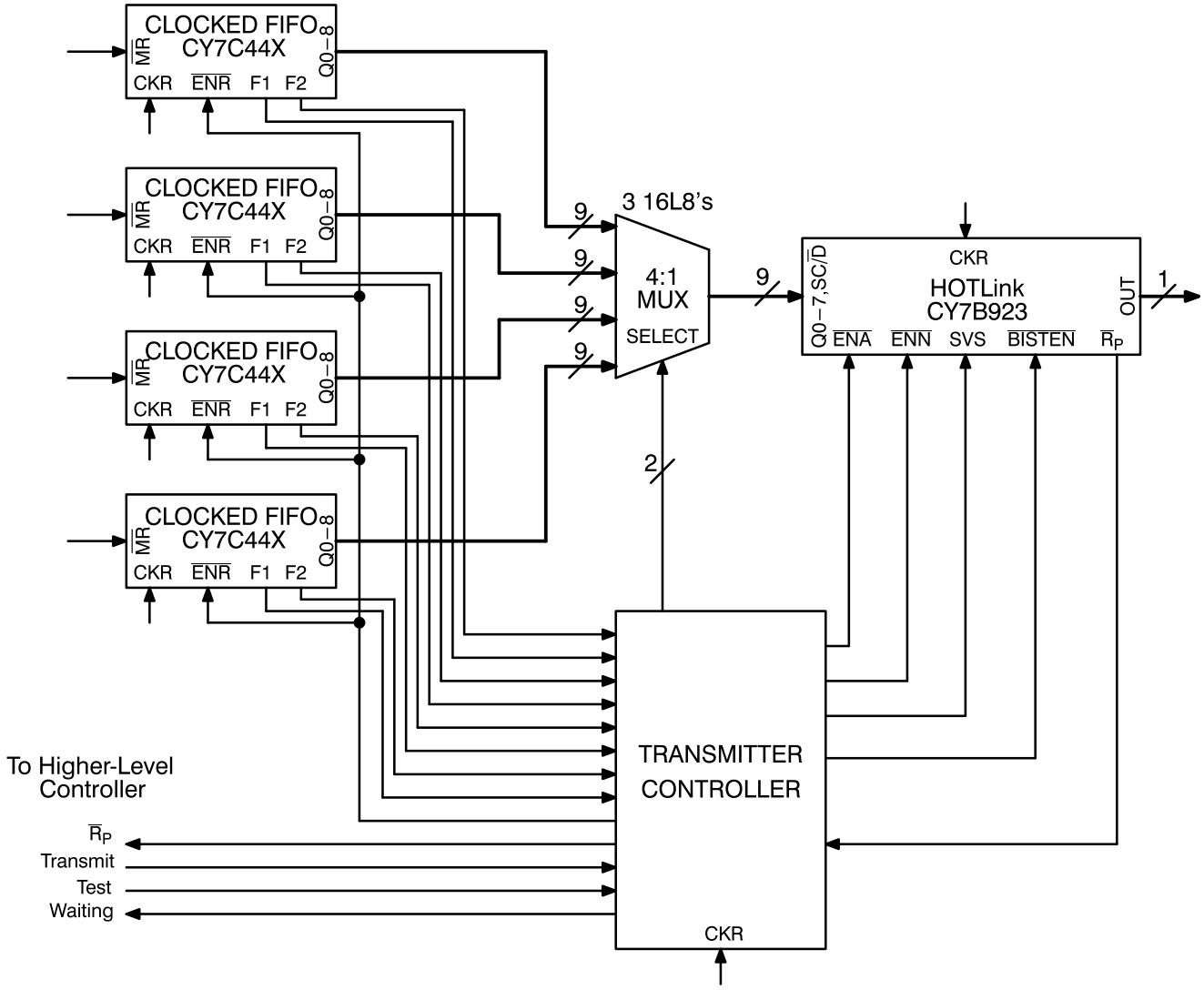



**Figure 1. Transmitter Interface Block Diagram**

**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 $\overline{MR}$ (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 $\overline{MR}$. $\overline{MR}$ 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, HOT-

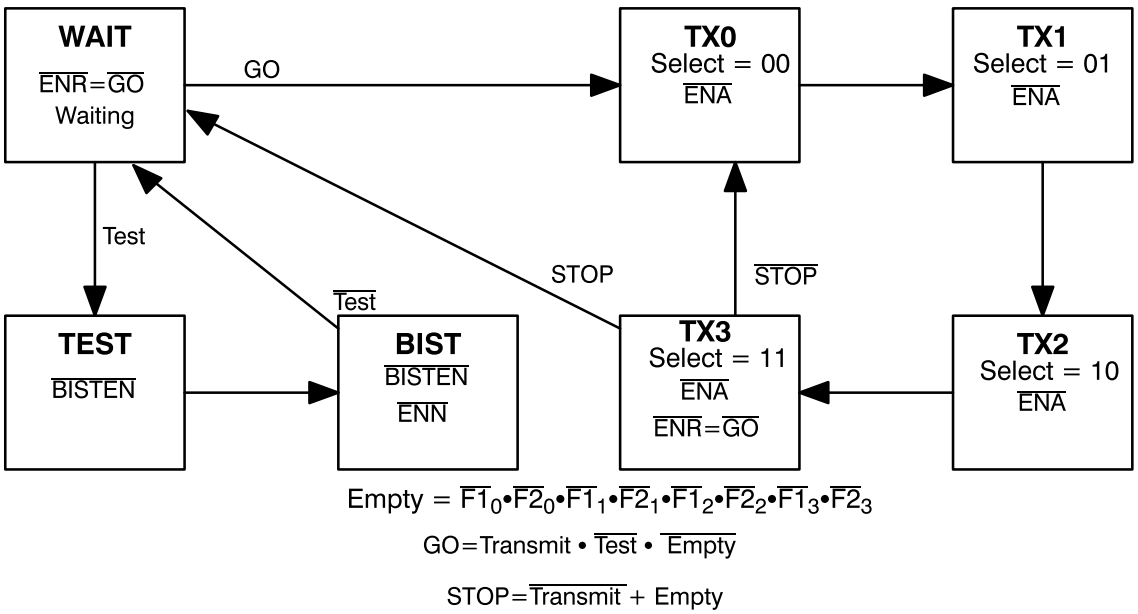



**Figure 2. Transmitter Controller State Diagram**

Link will transmit Idle special characters (K28.5). When the higher-level controller asserts the Transmit signal, the local transmitter controller issues a read ($\overline{ENR}$ 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 $\overline{ENR}$ 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 $\overline{RP}$ for BIST loop counting. $\overline{RP}$ will pulse LOW one time per BIST loop. *Figure 2* illustrates the controller state diagram.

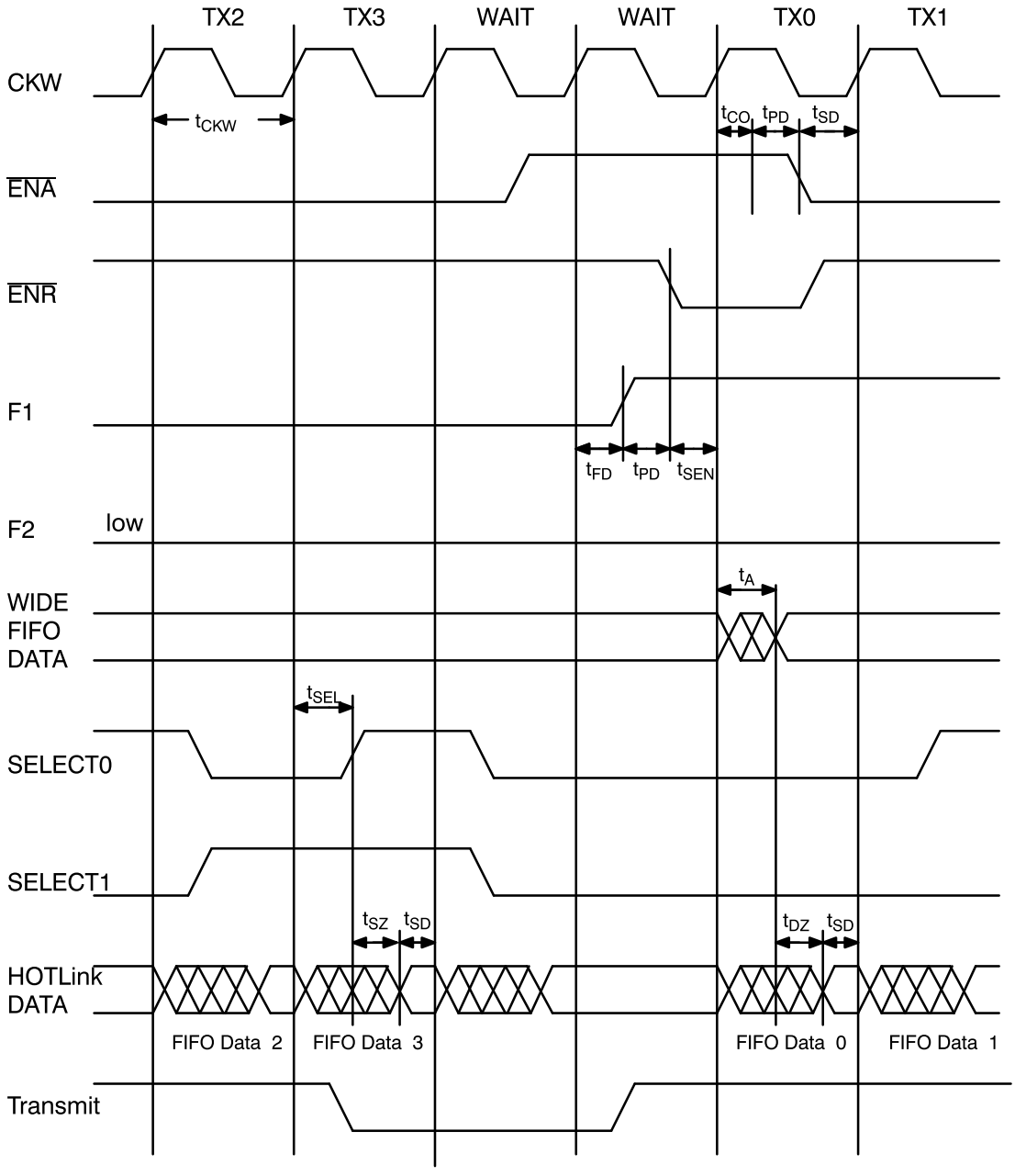
## Critical Timing Analysis

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

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

With a-30 ns clock period, the signal propagation delay through the controller must be $t_{PD} \leq 13$ 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




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**

($t_{SEL} > t_A$, 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

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

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 $t_{SZ} \leq 15$ 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

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

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 $t_{DZ} \leq 15$ 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 $t_{ID}$.

### 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.

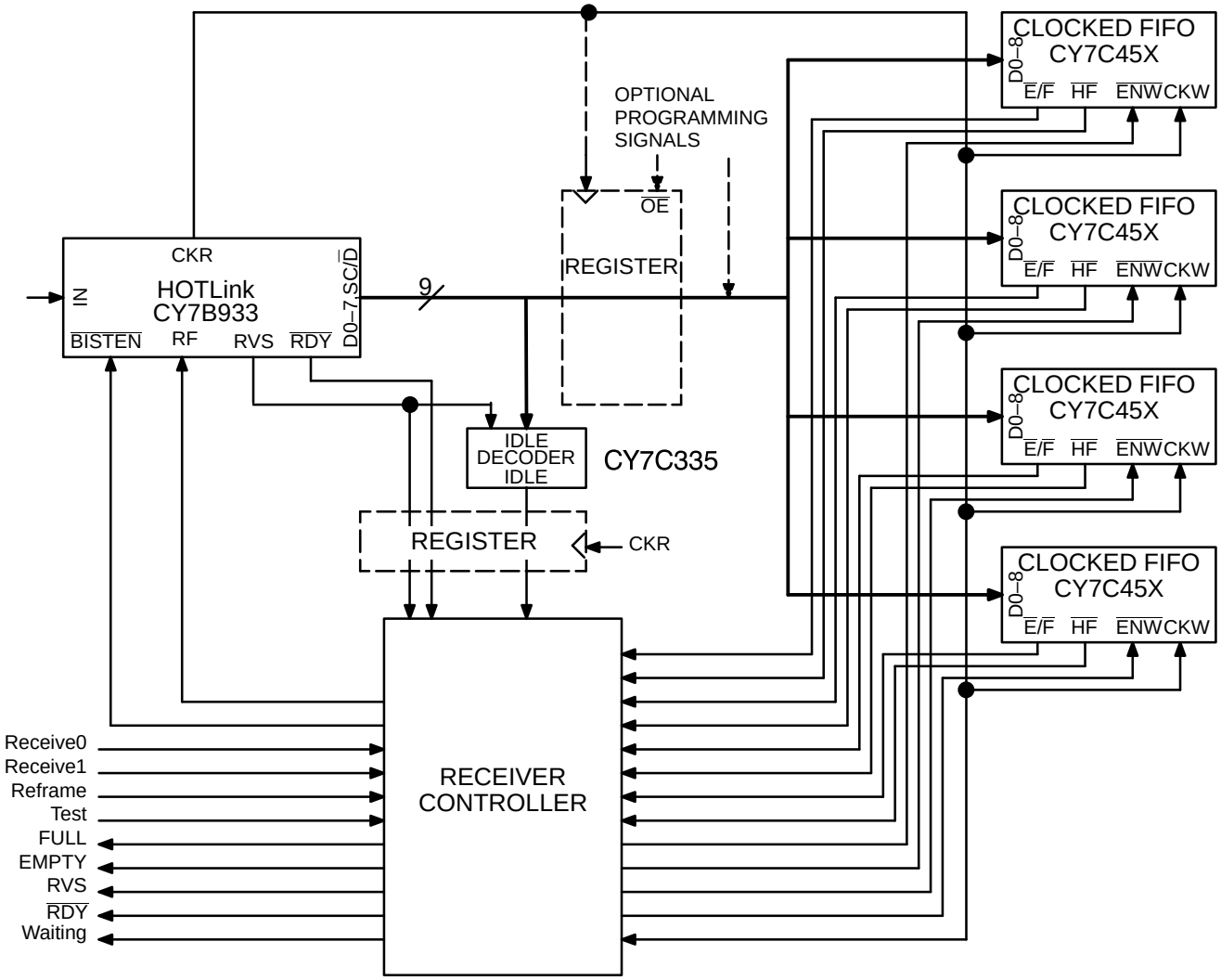



**Figure 4. Receiver Interface Block Diagram**

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.

**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 $\frac{1}{2}t_{CKR} - 10$ 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 programming 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

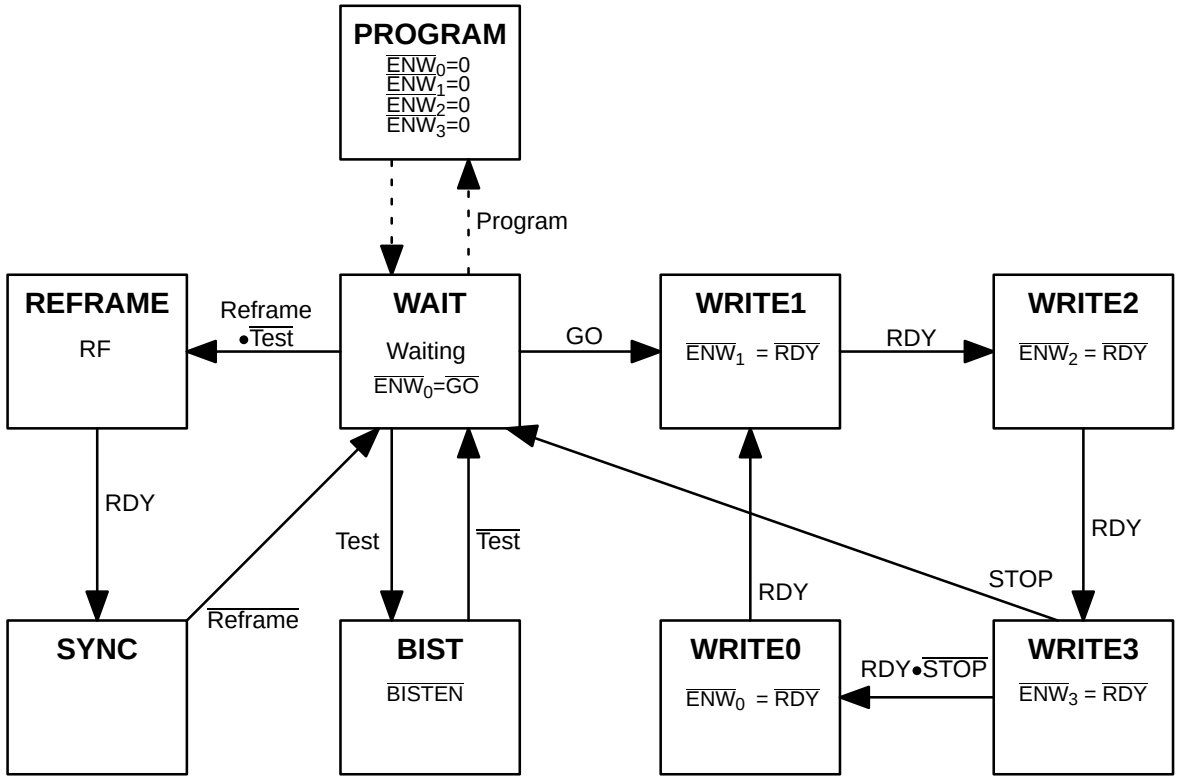
A state diagram for the receiver interface controller is shown in *Figure 5*. Five simple signals control the interface. The Receive0 and Receive1 signals are used to initiate and stop the reception of data. Reframe is used to synchronize the receiver to the serial data stream. Test causes HOTLink to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




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

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

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

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

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

**Figure 5. Receiver Controller State Diagram**

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 equa-

tions 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:

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

IDLE generation time from data:

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

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

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

IDLE generation time from data:

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

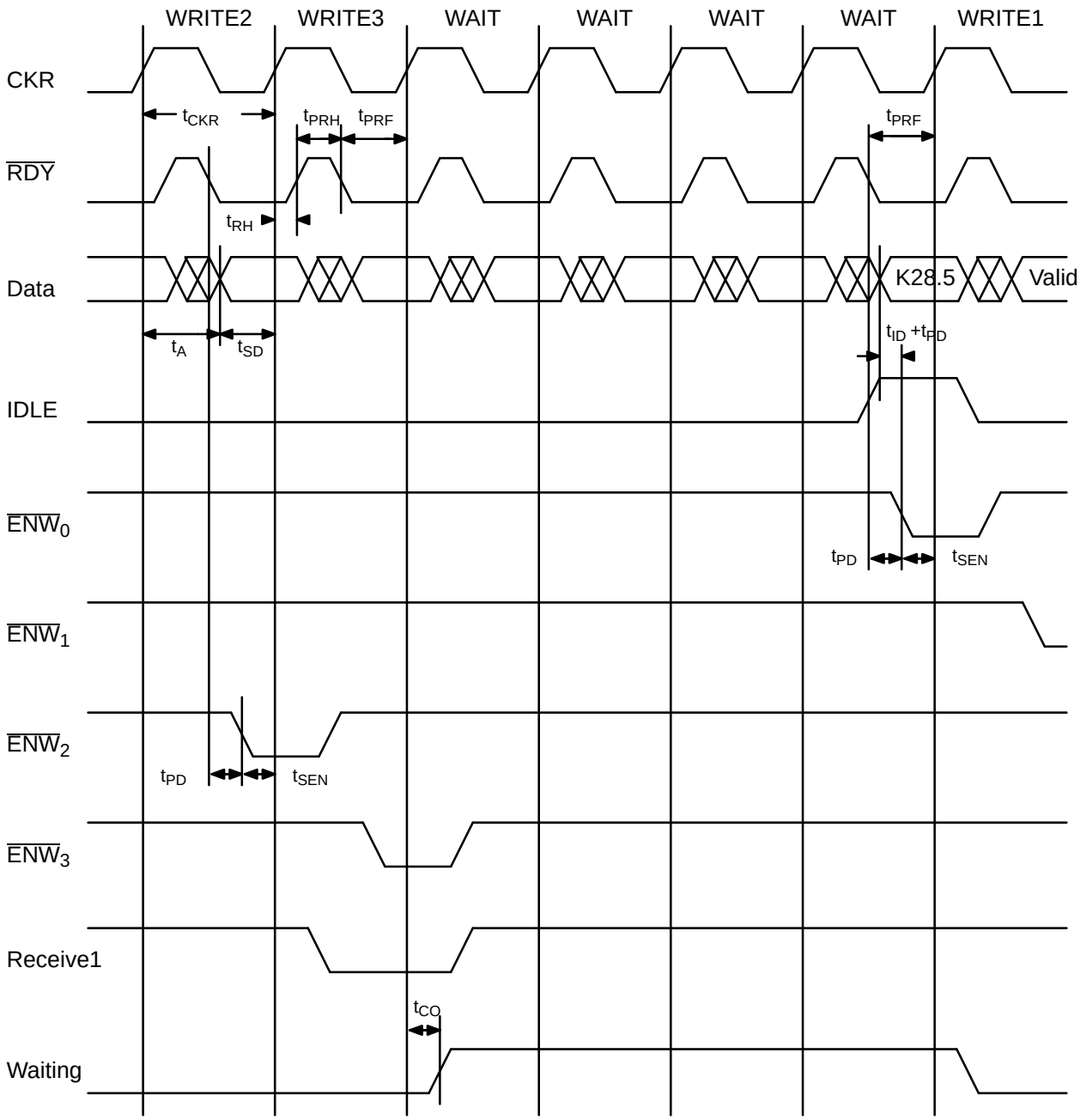
$\overline{RDY}$ capture timing:

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

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
   $t_A + t_{SD} \leq t_{CKR}$
2. Write enable set-up time from $\overline{RDY}$ LOW
   $t_{PD} + t_{SEN} \leq t_{PRF}$
3. Write enable set-up time from idle HIGH:
   $t_A + t_{ID} + t_{PD} + t_{SEN} \leq t_{CKR}$

**Figure 6. Receiver Timing Diagram**