# 8b10b Macro

## Product Summary

- Gigabit Ethernet 8b10b Function
- 125 MHz Operation
- Transmit and Receive Function
- Disparity and Illegal Code Error Checking
- Connects directly to industry-standard Gigabit Ethernet Transceiver devices.
- Supports either single or dual channel transceiver in a single device.

### Version

This data sheet defines the functionality of Version 1.0 of the 8b10b macro.

## General Description

The 8b10b macro implements the function for the physical coding sublayer for Gigabit Ethernet as defined in the IEEE 802.3z specification. The 8b10b is a marriage of two sub-blocks, the 5b6b and the 3b4b encoder/decoders (ENDECs). The purpose of the ENDEC is to convert 8-bit data into a 10-bit code that contains an equal number of 0's and 1's. In addition, the code is built so that no more than five consecutive 0's or 1's are ever transmitted. The 8b10b macro is designed to work with a variety of standard transceiver devices. A set of generic signals provides a data and command interface with the system logic. A system-level block diagram describing the use of the 8b10b macro is shown in Figure 1.

The 8b10b macro provides a user interface and a transceiver interface. The user interface consists of transmit data, receive data, and several control and status signals used to qualify the data. To simplify the timing of the user interface, the data transmission is word-wide (16-bits) and operates at 62.5 MHz. This strategy provides a simplified timing interface for system logic yet still meets the 125 megabyte per second requirements for Gigabit Ethernet.

The transceiver interface is designed to connect directly to most commercially-available Gigabit Ethernet transceiver devices. The transceiver is responsible for serializing transmit data and deserializing receive data. In addition, the receiver is designed to resynchronize the serial stream whenever an external device detects illegal coding errors.

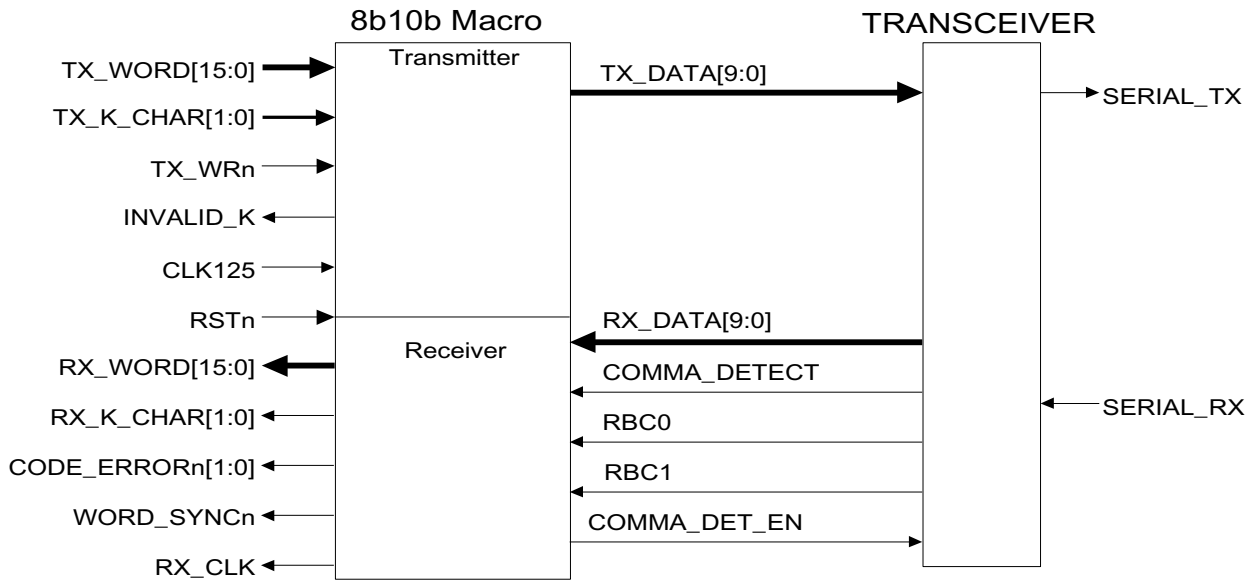


***Figure 1 •*** *System Block Diagram Depicting 8b10b Macro Usage*

## 8b10b Device Requirements

Performance requirements of the 8b10b macro drives device selection. Table 1 defines the minimum device requirements for the A54SXA family.

*Table 1 • Device Statistics for the 8b10b Macro*

| Device | Speed Grade | Utilization |
|---|---|---|
| A54SX08A | Standard | 96% |
| A54SX16A | Standard | 48% |
| A54SX32A | -1 | 24% |

## I/O Signal Descriptions

The 8b10b macro signals are defined in Tables 2 and 3.

*Table 2 • Transceiver Interface Signals*

| Name[1] | Type | Description |
|---|---|---|
| COMMA_DETECT | Input | Active high pulse from the transceiver indicating that a comma character has been detected and the received data is aligned with the rising edge of the RBC1 clock. |
| RBC0, RBC1 | Input | Clock signals recovered from the received data stream. These clocks are 180 degrees out of phase and the rising edge of each clock qualifies receive data. |
| RX_DATA[9:0] | Input | 10-bit encoded input data from the transceiver qualified by the rising edges of RBC0 and RBC1. |
| COMMA_DET_EN | Output | Active high signal indicating that the transceiver should align the data stream with the rising edge of RBC1. This output is asserted when the 8b10b detects multiple consecutive encoding errors. |
| TX_DATA[9:0] | Output | 10-bit encoded output data to the transceiver. |

***Note:***

1. *Active LOW signals are designated with a trailing lower-case n.*

## 8b10b Transmitter Detailed Operation

The 8b10b transmitter is a pipelined structure that converts 16-bit command or data information into two 10-bit encoded values. Command and data information are qualified by the TX_K_CHAR[1:0] bus. TX_K_CHAR[1] corresponds to the upper data byte on TX_WORD[15:0] and TX_K_CHAR[0] is for the lower byte. The data on the TX_WORD bus is continuously registered into the transmitter; however, this data is only transferred to the encoder when the TX_WRn signal is driven low for a single cycle. The transmitter will encode and send the upper byte first followed by the lower byte. Because of the pipelined nature of the transmitter, the first encoded data will be driven on the TX_DATA bus several cycles after the TX_WRn pulse. All data input information is valid, though command possibilities are limited. If the transmitter detects a bad command, then it will assert the INVALID_K signal. When the TX_WRn input is inactive, the transmitter will continuously send an IDLE2 (K28.5/D16.2) command defined in the 802.3 specification. Figure 2 on page 3 illustrates the implementation of the transmitter function.

The core of the transmitter consists of a data encoder, a command encoder, and a disparity calculator. Each encoder calculates a 4B and 6B code for the input data. The correct code, command or data, is then selected based on the original input value of TX_K_CHAR. The disparity calculator determines whether encoded values need to be inverted to maintain the correct running disparity. Finally, the code is registered and sent to the transceiver on the TX_DATA bus.

## 8b10b Receiver Detailed Operation

The 8b10 receiver is also a pipelined structure that converts two 10-bit encoded values qualified by the clocks RBC0 and RBC1 and converts them to 16-bit command or data information. Command information is indicated by the RX_K_CHAR[1:0] bus signals asserted high. The data on the upper byte of the RX_WORD bus is the first decoded value in the sequence.

Several signals qualify the validity of the information on RX_WORD. RX_WORD contains good information whenever CODE_ERRORn is inactive (high) and WORD_SYNCn is active (low). If WORD_SYNCn is high or CODE_ERRORn is low, this indicates some problem in transmission. Whenever the receiver loses sync (WORD_SYNCn is high), it asserts the COMMA_DET_EN output so that the transceiver resynchronizes the data on subsequent K28.5 commands. When sync is reestablished, the WORD_SYNCn will again be driven low after the pipeline has been flushed of potentially bad data. Figure 3 on page 4 illustrates the implementation of the 8b10b receive function.

Receive data is first loaded into two parallel registers. The first register is active on the rising edge of RBC0 and the second on the rising edge of RBC1. The RBC0 data is then

***Table 3** • System Interface Signals*

| Name[1] | Type | Description |
|---|---|---|
| CLK125 | Input | Primary 125MHz clock signal for the transmit block of the 8b10b macro. |
| RSTn | Input | Asynchronous reset signal for the macro. |
| TX_K_CHAR[1:0] | Input | Active high signal indicating that the TX_WORD[15:0] contains command information. Bit 0 corresponds to the lower byte (bits 7:0) and bit 1 corresponds to the upper byte (bits 15:8) of TX_WORD. |
| TX_WORD[15:0] | Input | 16-bit input data to the transmitter. Byte 1 is transmitted first followed by byte 0. |
| TX_WRn | Input | Active low signal that qualifies the TX_WORD data. When this signal is asserted, the data defined on TX_WORD will be registered into the 8b10b macro, encoded, and sent to the transceiver in two consecutive 10-bit transfers. |
| CODE_ERRORn[1:0] | Output | Active low signal indicating that the ENDEC has detected an error in the received data stream. Bit 0 corresponds to the lower byte (bits 7:0) and bit 1 corresponds to the upper byte (bits 15:8) of RX_WORD. |
| INVALID_K | Output | Active high signal indicating that the upstream device requested the ENDEC to transmit an invalid command character. This signal is asserted when either of the TX_K_CHAR[1:0] is active, but the associated don on the byte lane of TX_WORD[15:0] does not correspond to a valid command character. |
| RX_CLK | Output | The receive clock. The rising edge of this clock qualifies RX_WORD[15:0], RX_K_CHAR[1:0], WORD_SYNCn, and CODE_ERRORn[1:0]. |
| RX_K_CHAR[1:0] | Output | Output from the ENDEC to the transceiver indicating that the received data is a command code. |
| RX_WORD[15:0] | Output | 16-bit decoded receive data. The upper byte was received first and the lower byte was received second in the data sequence. |
| WORD_SYNCn | Output | Active low signal indicating that the received data is correctly aligned. |

***Note:***

*1. Active LOW signals are designated with a trailing lower-case n.*

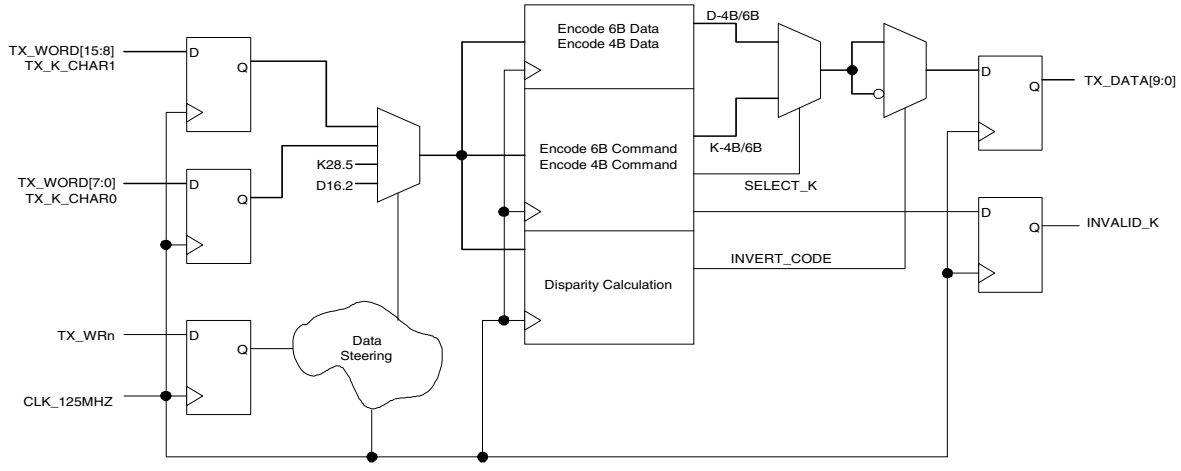


***Figure 2** • 8b10b Transmitter Block Diagram*

resynchronized with RBC1 on the next RBC1 rising clock edge. From this point, the two codes are decoded in parallel and move from stage to stage based on the RBC1 clock input.

The error check block monitors the incoming codes and checks for illegal codes and/or bad running disparity. When an error in the 8b10b code is detected, the CODE_ERRORn is asserted. If several codes in a row are received with errors, then the 8b10b will assume that synchronization with the transceiver has been lost and will deactivate WORD_SYNCn and assert the COMMA_DET_EN signal. The number of

consecutive errors required to force a resynchronization is programmable, from 2 to 16 (default is 6). The transceiver then resynchronizes the data on the rising edge of RBC1 using K28.5 codes. A pulse on the COMMA_DETECT input indicates that the transceiver has reacquired sync. The 8b10b responds by deasserting COMMA_DET_EN and asserting WORD_SYNCn.

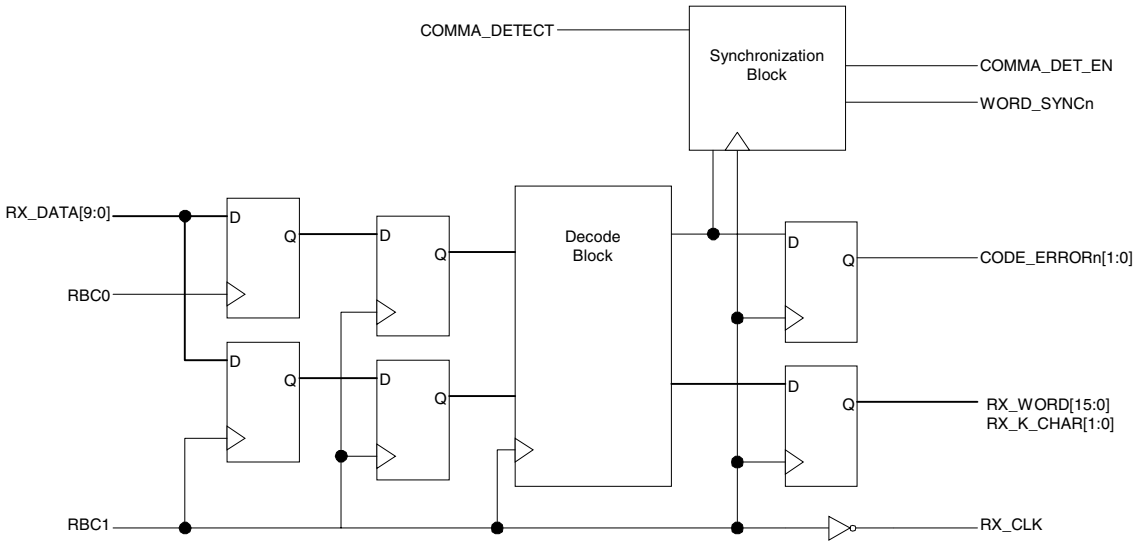


***Figure 3 •*** *8b10b Receiver Block Diagram*

## Hierarchy of the 8b10b Model

The hierarchy of the 8b10b model is shown in Figure 4. The transmitter is the encoder. The encoder is subdivided into the data encoder (enc_d), command encoder (enc_k), and the running disparity calculator (enc_flip). The enc_d is composed of the mux32x6, mux32x1, and mux4x1 modules that create a ROM for data encoding. The receiver is the decoder, which is subdivided into the data decoder (dec_data), running disparity decoder (dec_rd), and the synchronization state machine (sync_fsm).

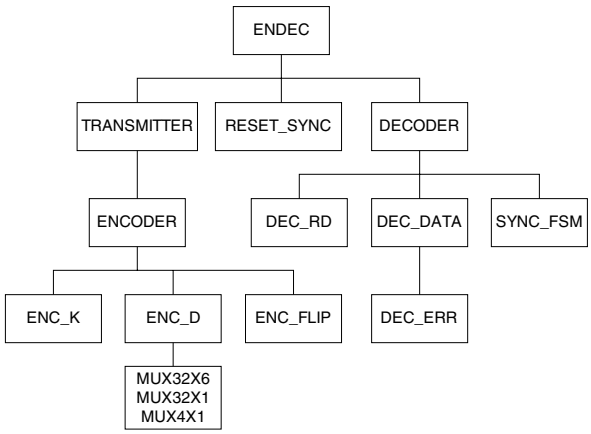


***Figure 4 •*** *Hierarchy of the 8b10b ENDEC.*

## Utilization Statistics

The 8b10b macro uses approximately 250 sequential modules and 350 combinatorial modules in the A54SX-A devices. The macro also uses approximately 70 I/Os and requires two clock networks for the CLK125 and RBC1 inputs. Because of the light loading, the clock input RBC0 can use a regular input. It is possible to implement a dual channel 8b10b in either the A54SX16A or A54SX32A device.

## System Timing

The 8b10b macro is divided into two functions, the transmitter and the receiver. The transmitter is designed to operate at 125 MHz, the receiver at 62.5 MHz. The input setup time for transmitter signals (TX_WORD, TX_K_CHAR, and TX_WRn) are measured with respect to the rising edge of CLK125. The input setup time for the receiver signal RX_DATA is measured with respect to the rising edge of both RBC0 and RBC1. The input setup time for the COMMA_DETECT signal is measured with respect to RBC1 only. Receiver output timing is defined with respect to the rising edge of RX_CLK, an inverted version of RBC1 (refer to Figure 5 and Figure 6).

Table 4 defines the internal register-to-register delays for the CLK125MHZ domain (transmitter) and the RBC1 domain (receiver). RBC0 domain to RBC1 domain timing is provided for reference. Input setup requirements are defined in Table 5. Output valid times are defined in Table 6 and Table 7.

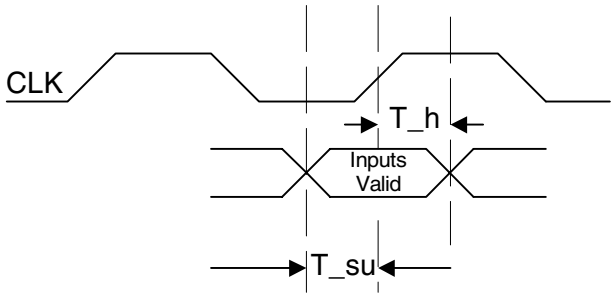


***Figure 5 •*** *Input Timing for 8b10b Signals*

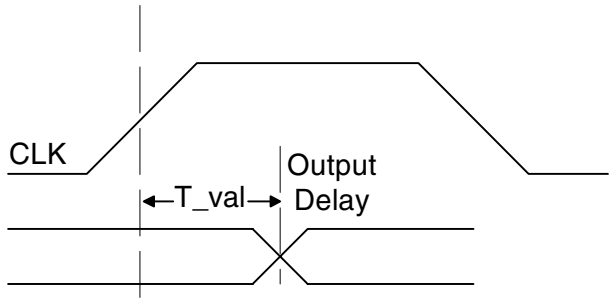


***Figure 6 •*** *Output Timing for 8b10b Signals*

***Table 4 •*** *Internal Reg-Reg Delays (ns max)*

| Name | SX08A | SX16A | SX32A-1 |
|---|---|---|---|
| CLK125MHZ | 7.5 | 7.5 | 7.0 |
| RBC1 | 13.5 | 15.0 | 12.5 |
| RBC0 -> RBC1 | 2.0 | 2.0 | 2 |

***Notes:***

1. *All timing is for worst-case commercial conditions.*
2. *Expected values from commercially available synthesis tools using standard design practices.*

## 8B10B Waveforms

The operation of the 8b10b macro is illustrated in the following waveforms. The function of the 8b10b can be illustrated using 5 different waveforms:

- Normal transmission
- Transmission with an invalid K command
- Normal receive
- Loss of synchronization with the transceiver
- Synchronization with the transceiver

***Table 5 •*** *Input Required Set-Up Times (ns max)*

| Name | SX08A | SX16A | SX32A-1 |
|---|---|---|---|
| COMMA_DETECT | 0.5 | 0.5 | 0.5 |
| RX_DATA[9:0] | 1.0 | 1.0 | 0.5 |
| TX_WRn | 0.5 | 0.5 | 0.5 |
| TX_K_CHAR | 1.0 | 1.0 | 0.5 |
| TX_WORD[15:0] | 1.0 | 1.0 | 0.5 |

***Notes:***

1. *All timing is for worst-case commercial conditions.*
2. *Expected values from commercially available synthesis tools using standard design practices.*

***Table 6 •*** *Output Valid Times (ns max)*

| Name | SX08A | SX16A | SX32A-1 |
|---|---|---|---|
| COMMA_DET_EN | 6.0 | 6.0 | 6.0 |
| INVALID_K | 6.0 | 6.0 | 6.0 |
| TX_DATA[9:0] | 5.5 | 5.5 | 5.5 |
| RX_CLK | 6.0 | 6.0 | 6.0 |

***Notes:***

1. *All timing is for worst-case commercial conditions.*
2. *Expected values from commercially available synthesis tools using standard design practices.*

A normal transmission begins by placing valid data/command information on TX_WORD and TX_K_CHAR while simultaneously asserting the TX_WRn signal. After several cycles, the encoded data is driven onto the TX_DATA bus. If command information is illegal, the INVALID_K signal will assert for one cycle. Normal 8b10b transmission is illustrated in Figure 7 and an invalid command waveform is shown in Figure 8 on page 7.

A normal receive assumes that the 8b10b encoded data on RX_DATA is aligned with RBC1. Encoded data is registered into the 8b10b off the rising edge of both RBC1 and RBC0.

***Table 7 •*** *Receive Data Valid Prior to RX_CLK (ns max)*

| Name | SX08A | SX16A | SX32A-1 |
|---|---|---|---|
| CODE_ERRORn | 6.0 | 6.0 | 6.0 |
| RX_K_CHAR | 6.0 | 6.0 | 6.0 |
| RX_WORD[15:0] | 6.0 | 6.0 | 6.0 |
| WORD_SYNCn | 6.0 | 6.0 | 6.0 |

***Notes:***

1. *All timing is for worst-case commercial conditions.*
2. *Expected values from commercially available synthesis tools using standard design practices.*
3. *Hold times for all signals is at least 2ns after the rising edge of RX_CLK.*

After several cycles, the unencoded data or command is driven onto the RX_WORD and RX_K_CHAR buses. This information is qualified by the rising edge of RX_CLK. The normal 8b10b receive is depicted in Figure 9 on page 7.

In some cases, the 8b10b detects an error condition on the incoming data stream. When this occurs, the output CODE_ERRORn is asserted. If several consecutive errors are found, the 8b10b will assume that it has lost synchronization with the transceiver and will attempt to resynchronize by asserting the COMMA_DET_EN output as shown in Figure 10 on page 7.

When the COMMA_DET_EN output is asserted, the transceiver will scan the incoming data stream for a K28.5 command code and will resynchronize RX_DATA on the rising edge of RBC1. If the data stream is synchronized and a K28.5 command is detected, then the transceiver will indicate synchronization by asserting the COMMA_DETECT signal. After two pulses, the 8b10b will again be synchronized as indicated by the WORD_SYNCn signal in Figure 11 on page 8.

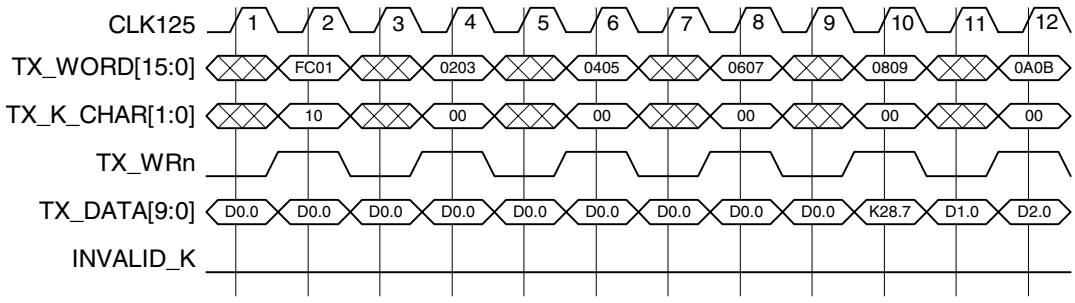


***Figure 7 •*** *Normal Transmit*

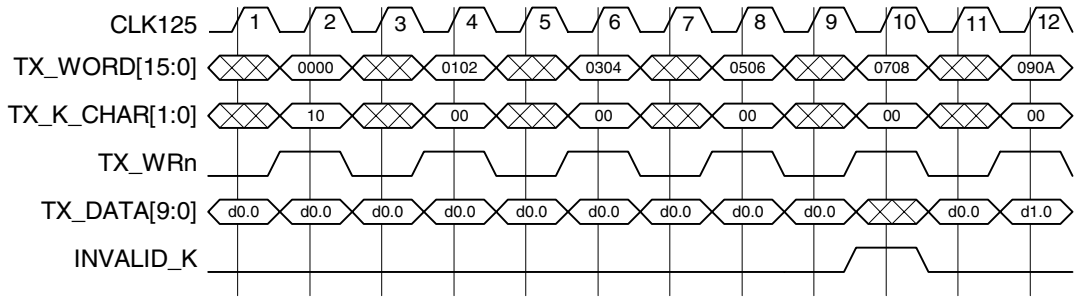


***Figure 8** • Invalid Command*

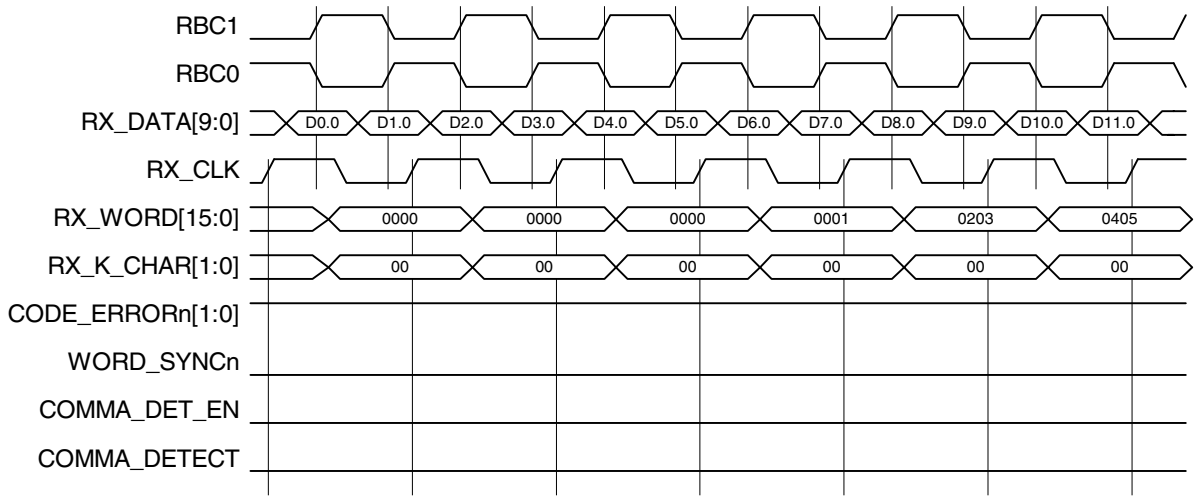


***Figure 9** • Normal Receive*

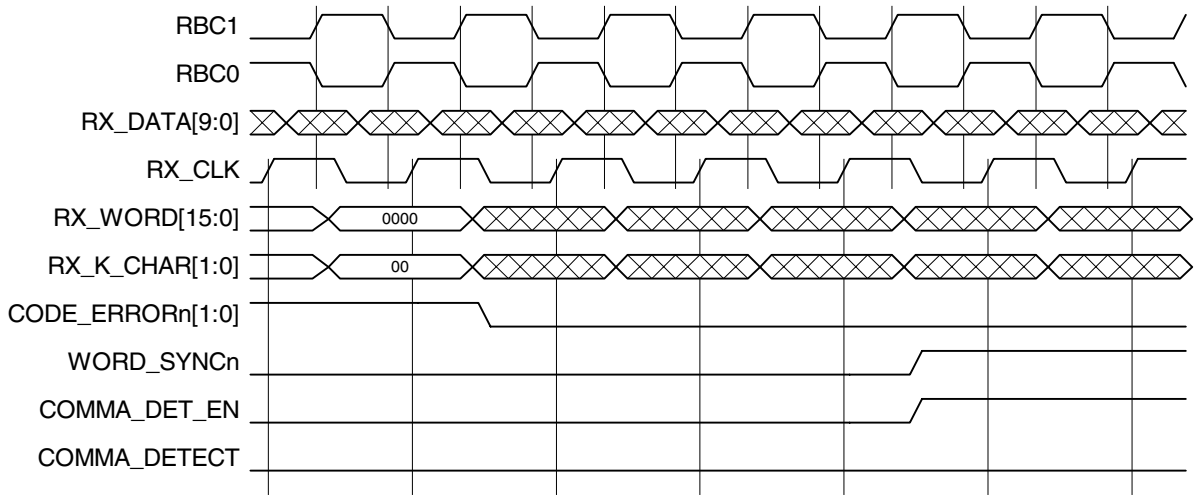


***Figure 10** • Receive Error*

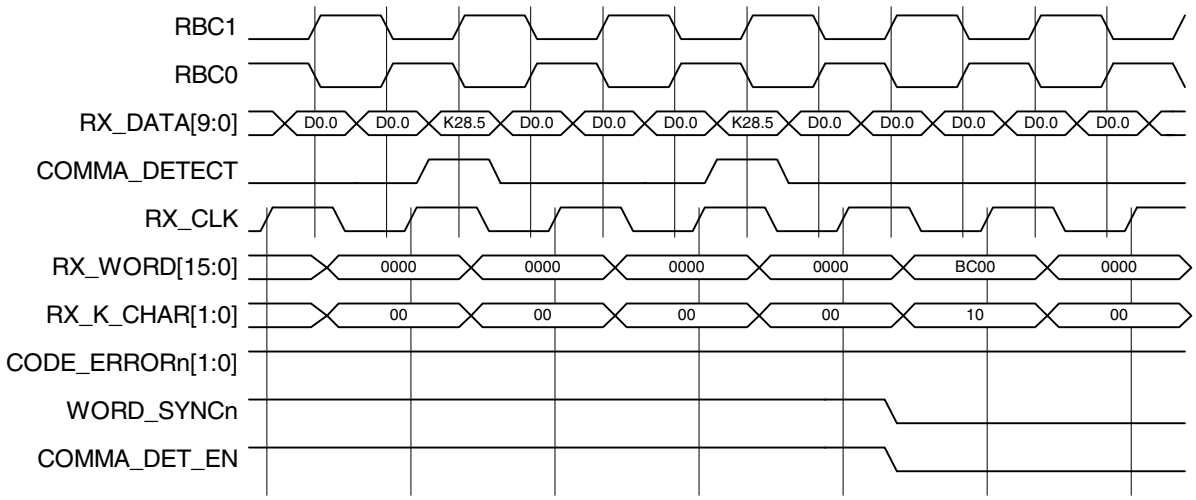


***Figure 11 •*** *Synchronization with the Transceiver*

http://www.actel.com

**Actel Europe Ltd.**
Daneshill House, Lutyens Close
Basingstoke, Hampshire RG24 8AG
United Kingdom
**Tel:** +44-(0)125-630-5600
**Fax:** +44-(0)125-635-5420

**Actel Corporation**
955 East Arques Avenue
Sunnyvale, California 94086
USA
**Tel:** (408) 739-1010
**Fax:** (408) 739-1540

**Actel Asia-Pacific**
EXOS Ebisu Bldg. 4F
1-24-14 Ebisu Shibuya-ku
Tokyo 150 Japan
**Tel:** +81-(0)3-3445-7671
**Fax:** +81-(0)3-3445-7668

5172155-0/5.00