# Table of Contents

**Introduction** .............................................. 5  
General Description ........................................ 5  
Core Versions .................................................. 5  

1 **Functional Block Description** ............................. 7  
Device Utilization and Performance .......................... 8  
Programmable Options ....................................... 9  

2 **Tool Flows** ............................................... 11  
SmartDesign ............................................... 11  

3 **Core Interfaces** .......................................... 13  
Core Parameters ........................................... 14  

4 **Timing Diagrams** ......................................... 15  

5 **Testbench Operation** ..................................... 19  
User Testbench ............................................. 19  

6 **Ordering Information** .................................... 21  
Ordering Codes ............................................ 21  

A **List of Document Changes** ............................... 23  

B **Product Support** .......................................... 25  
Customer Service ........................................... 25  
Actel Customer Technical Support Center .................. 25  
Actel Technical Support ..................................... 25  
Website ....................................................... 25  
Contacting the Customer Technical Support Center .... 25  

Index .......................................................... 27
Introduction

General Description

CoreUART is a serial communication controller with a flexible serial data interface that is intended primarily for embedded systems. CoreUART can be used to interface directly to industry standard UARTs. CoreUART is intentionally a subset of full UART capability to make the function cost-effective in a programmable device. Figure 1 illustrates the various usages of CoreUART.

Case A in Figure 1 represents the interface to an industry standard UART, such as an 8251 or a 16550. In Case B, CoreUART is transferring data from the 8051 to the system monitor through the RS-232 interface and vice versa.

Core Versions

This handbook applies to CoreUART v4.0. The release notes provided with the core list known discrepancies between this handbook and the core release associated with the release notes.
Supported Families

- IGLOO®
- IGLOOe
- IGLOO PLUS
- ProASIC®3
- ProASIC3L
- ProASIC®PLUS
- Fusion
- Accelerator®
- RTAX-S
- SX-A
- RTSX-S
Figure 1-1 shows the block diagram of the CoreUART normal mode functionality. Figure 1-2 on page 8 shows the block diagram of CoreUART with FIFO mode functionality. The baud generator creates a divided down clock enable that correctly paces the transmit and receive state machines.

The function of the receive and transmit state machines is affected by the control inputs BIT8, PARITY_EN, and ODD_N_EVEN. These signals indicate to the state machines how many bits should be transmitted. In addition, the signals suggest the type of parity and whether parity should be generated or checked. The activity of the state machines is paced by the outputs of the baud generator.

To transmit data, it is first loaded into the transmit data buffer in normal mode, and into the transmit FIFO in FIFO mode. Data can be loaded into the buffer until the TXRDY signal is driven inactive. The transmit state machine will immediately begin to transmit data and will continue transmission until the data buffer is empty in normal mode, and until the transmit FIFO is empty in FIFO mode. The state machine first transmits a START bit, followed by the data (LSB first), then the parity (optional), and finally the STOP bit. The data buffer is double-buffered in normal mode, so there is no loading latency.

The receive state machine monitors the activity of the RX signal. Once a START bit is detected, the receive state machine begins to store the data in the receive buffer in normal mode and the receive FIFO in FIFO mode. When the transaction is complete, the RXRDY signal indicates that valid data is available. Parity errors are reported on the PARITY_ERR signal (if enabled), and data overrun conditions are reported on the OVERFLOW signal. Framing errors are reported on the FRAMING_ERR signal. A framing error is defined as a missing stop bit detected by the UART receiver.
Device Utilization and Performance

Utilization statistics for targeted devices are listed in Table 1-1 and Table 1-2 on page 9.

Table 1-1 · CoreUART Utilization in FIFO Mode

<table>
<thead>
<tr>
<th>Family</th>
<th>Cells or Tiles</th>
<th>Memory Blocks</th>
<th>Utilization</th>
<th>Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sequential</td>
<td>Combinatorial</td>
<td>Total</td>
<td>Device</td>
</tr>
<tr>
<td>IGLOO®</td>
<td>116</td>
<td>192</td>
<td>308</td>
<td>2</td>
</tr>
<tr>
<td>IGLOOe</td>
<td>116</td>
<td>192</td>
<td>308</td>
<td>2</td>
</tr>
<tr>
<td>IGLOO PLUS</td>
<td>116</td>
<td>192</td>
<td>308</td>
<td>2</td>
</tr>
<tr>
<td>ProASIC®3</td>
<td>116</td>
<td>192</td>
<td>308</td>
<td>2</td>
</tr>
<tr>
<td>ProASIC3E</td>
<td>116</td>
<td>192</td>
<td>308</td>
<td>2</td>
</tr>
<tr>
<td>ProASIC3L</td>
<td>116</td>
<td>192</td>
<td>308</td>
<td>2</td>
</tr>
<tr>
<td>ProASICPLUS®</td>
<td>118</td>
<td>281</td>
<td>399</td>
<td>2</td>
</tr>
<tr>
<td>Fusion</td>
<td>116</td>
<td>192</td>
<td>308</td>
<td>2</td>
</tr>
<tr>
<td>Axcelerator®</td>
<td>171</td>
<td>215</td>
<td>386</td>
<td>2</td>
</tr>
<tr>
<td>RTAX-S</td>
<td>195</td>
<td>199</td>
<td>394</td>
<td>2</td>
</tr>
<tr>
<td>SX-A</td>
<td>430</td>
<td>309</td>
<td>739</td>
<td>0</td>
</tr>
<tr>
<td>RTSX-S</td>
<td>432</td>
<td>308</td>
<td>740</td>
<td>0</td>
</tr>
</tbody>
</table>

Notes:
1. CoreUART supports all standard baud rates, including 110, 300, 1,200, 2,400, 4,800, 9,600, 19,200, 38,400, 57,600, 115,200, 230,400, 460,800, and 921,600 baud.
2. The depth of the FIFO for SX-A and RTSX-S is 16. For the other families, the depth of the FIFO is 256.
Programmable Options

There are four programmable inputs to CoreUART: BAUD_VAL (baud rate), BIT8 (number of data bits), PARITY_EN (parity enable), and ODD_N_EVEN (odd or even parity).

Number of Data Bits

The input BIT8 is used to define the number of valid data bits in the serial bitstream. The most significant bit is a "don't care" for the seven-bit case.

Parity

Parity is enabled/disabled with the input PARITY_EN. When parity is enabled, the ODD_N_EVEN input defines the type of parity.

Table 1-2 - CoreUART Utilization in Normal Mode

<table>
<thead>
<tr>
<th>Family</th>
<th>Cells or Tiles</th>
<th>Memory Blocks</th>
<th>Utilization</th>
<th>Performance MHz</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sequential</td>
<td>Combinatorial</td>
<td>Total</td>
<td>Device</td>
</tr>
<tr>
<td>IGLOO®</td>
<td>84</td>
<td>167</td>
<td>251</td>
<td>0</td>
</tr>
<tr>
<td>IGLOOe</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IGLOO PLUS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ProASIC®3</td>
<td>84</td>
<td>167</td>
<td>251</td>
<td>0</td>
</tr>
<tr>
<td>ProASIC3E</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ProASIC3L</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ProASICPLUS</td>
<td>85</td>
<td>254</td>
<td>339</td>
<td>0</td>
</tr>
<tr>
<td>Fusion</td>
<td>84</td>
<td>167</td>
<td>251</td>
<td>0</td>
</tr>
<tr>
<td>Acelerator</td>
<td>86</td>
<td>108</td>
<td>194</td>
<td>0</td>
</tr>
<tr>
<td>RTAX-S</td>
<td>86</td>
<td>108</td>
<td>194</td>
<td>0</td>
</tr>
<tr>
<td>SX-A</td>
<td>82</td>
<td>93</td>
<td>750</td>
<td>0</td>
</tr>
<tr>
<td>RTSX-S</td>
<td>80</td>
<td>92</td>
<td>172</td>
<td>0</td>
</tr>
</tbody>
</table>

Note: CoreUART supports all standard baud rates, including 110, 300, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200, 230400, 460800, and 921600 baud.
Baud Rate

This baud value is a function of the system clock and the desired baud rate. The value should be set according to \( EQ\ 1-1 \).

\[
\text{baud rate} = \frac{\text{clk}}{(\text{baud val} + 1) \times 16}
\]

where

- \( \text{clk} \) = the frequency of the system clock in hertz
- \( \text{baud rate} \) = the desired baud rate
- \( \text{baud val} \) = \( \text{BAUD\_VAL} \) input

The term baudval must be rounded to the nearest integer. For example, a system with a 33 MHz system clock and a desired baud rate of 9,600 should have a baud_val of 214 decimal or D6 hex. So, to get the desired baud rate, the user should assign \( 16\#D6 \) to \( \text{BAUD\_VAL} \) input.
SmartDesign

CoreUART is available for download to the SmartDesign IP Catalog via the Libero® Integrated Design Environment (IDE) web repository. For information on using SmartDesign to instantiate, configure, connect, and generate cores, please refer to the Libero IDE online help.

The Figure 2-1 shows the CoreUART configuration window, as well as cross references to the corresponding top-level parameters.

Figure 2-1 · CoreUART Configuration Window
## Core Interfaces

Signal descriptions for CoreUART are defined in Table 3-1.

### Table 3-1 · CoreUART Signals

<table>
<thead>
<tr>
<th>Name*</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLK</td>
<td>Input</td>
<td>Main system clock</td>
</tr>
<tr>
<td>RESET_N</td>
<td>Input</td>
<td>Active low asynchronous reset</td>
</tr>
<tr>
<td>DATA_IN[7:0]</td>
<td>Input</td>
<td>Transmit write data bus</td>
</tr>
<tr>
<td>DATA_OUT[7:0]</td>
<td>Output</td>
<td>Receive read data bus</td>
</tr>
<tr>
<td>WEN</td>
<td>Input</td>
<td>Active low write enable. This signal indicates that the data presented on the DATA_IN[7:0] bus should be registered by the transmit buffer/FIFO logic. This signal should only be active for a single clock cycle per transaction and should only be active when the TXRDY signal is active.</td>
</tr>
<tr>
<td>OEN</td>
<td>Input</td>
<td>Active low read enable. This signal is used to indicate that the data on DATA_OUT[7:0] has been read and will reset the RXRDY bit and any error conditions (OVERFLOW or PARITY_ERR).</td>
</tr>
<tr>
<td>CSN</td>
<td>Input</td>
<td>Active low chip select. The CSN signal qualifies both the WEN and OEN signals. For embedded applications, this signal should be tied to logic 0.</td>
</tr>
<tr>
<td>BIT8</td>
<td>Input</td>
<td>Control bit for data bit width for both receive and transmit functions. When BIT8 is logic 1, the data width is 8 bits; otherwise, the data width is 7 bits, data defined by DATA_IN[7] is ignored, and DATA_OUT[7] is “don’t care.”</td>
</tr>
<tr>
<td>PARITY_EN</td>
<td>Input</td>
<td>Control bit to enable parity for both receive and transmit functions. Parity is enabled when the bit is set to logic 1.</td>
</tr>
<tr>
<td>ODD_N_EVEN</td>
<td>Input</td>
<td>Control bit to define odd or even parity for both receive and transmit functions. When the PARITY_EN control bit is set, a ’1’ on this bit indicates odd parity and a ’0’ indicates even parity.</td>
</tr>
<tr>
<td>BAUD_VAL[12:0]</td>
<td>Input</td>
<td>13-bit control bus used to define the baud rate</td>
</tr>
<tr>
<td>TXRDY</td>
<td>Output</td>
<td>Status bit; when set to logic 0, indicates that the transmit data buffer/FIFO is not available for additional transmit data.</td>
</tr>
<tr>
<td>RXRDY</td>
<td>Output</td>
<td>Status bit; when set to logic 1, indicates that data is available in the receive data buffer/FIFO to be read by the system logic. The data buffer/FIFO controller must be notified of the receipt by simultaneous activation of the OEN and CSN signals to prevent erroneous overflow conditions.</td>
</tr>
<tr>
<td>PARITY_ERR</td>
<td>Output</td>
<td>Status bit; when set to logic 1, indicates a parity error during a receive transaction. This bit is synchronously cleared by simultaneous activation of the OEN and CSN signals.</td>
</tr>
<tr>
<td>OVERFLOW</td>
<td>Output</td>
<td>Status bit; when set to logic 1, indicates that a receive overflow has occurred. This bit is synchronously cleared by simultaneous activation of the OEN and CSN signals.</td>
</tr>
<tr>
<td>RX</td>
<td>Input</td>
<td>Serial receive data</td>
</tr>
</tbody>
</table>

**Notes:**

1. *Active low signals are designated with a trailing uppercase N.*
2. *When RX_FIFO is enabled, PARITY_ERR is asserted when a parity error occurs, but deasserted before CoreUART receives the next byte. It is the user’s responsibility to monitor the PARITY_ERR signal (e.g., treat it as an interrupt signal), as it is non-persistent when RX_FIFO = 1.*
There are a number of configurable options that apply to CoreUART, as shown in Table 3-2. If a configuration other than the default is required, the user should use the configuration dialog box in CoreConsole to select appropriate values for the configurable options.

### CoreUART Configurable Options

There are a number of configurable options that apply to CoreUART, as shown in Table 3-2. If a configuration other than the default is required, the user should use the configuration dialog box in CoreConsole to select appropriate values for the configurable options.

<table>
<thead>
<tr>
<th>Configurable Options</th>
<th>Default Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TX_FIFO</td>
<td>Disabled</td>
<td>Enables or disables transmit FIFO</td>
</tr>
<tr>
<td>RX_FIFO</td>
<td>Disabled</td>
<td>Enables or disables receive FIFO</td>
</tr>
<tr>
<td>FAMILY</td>
<td>ProASIC3</td>
<td>Selects target family. Must be set to match the supported FPGA family. 8 – SX-A 9 – RTSXS 11 – Axcelerator 12 – RTAX-S 14 – ProASICPLUS 15 – ProASIC3 16 – ProASIC3E 17 – Fusion 20 – IGLOO 21 – IGLOOe 22 – ProASIC3L 23 – IGLOO PLUS</td>
</tr>
<tr>
<td>RX_LEGACY_MODE</td>
<td>Disabled</td>
<td>When disabled, the RXRDY signal is synchronized with the FRAMING_ERR output, which occurs after the stop bit. When enabled (Legacy mode), the RXRDY signal is asserted after all data bits have been received, but before the stop bit.</td>
</tr>
<tr>
<td>USE_SOFT_FIFO</td>
<td>Disabled</td>
<td>When disabled, the FIFO is implemented using a device-specific hard macro. When enabled, a 16-byte FIFO is implemented in FPGA logic instead. RTAX and RTSX-S devices use this soft FIFO by default.</td>
</tr>
</tbody>
</table>
Timing Diagrams

The UART waveforms can be broken down into a few basic functions: transmit data, receive data, and errors. Figure 4-1 shows serial transmit signals, and Figure 4-2 on page 16 shows serial receive signals. Figure 4-3 on page 16 and Figure 4-4 on page 17 show the parity and overflow error cycles, respectively. The number of clock cycles required is equal to the clock frequency divided by the baud rate. All waveforms assume that eight bits of data and parity are enabled. All waveforms, except the ‘Framing Error’ (Figure 4-5 on page 17), assume Legacy mode is ENABLED.

![Figure 4-1 - Serial Transmit](image)

Notes:

1. A serial transmit is initiated by writing data into CoreUART. This is accomplished by providing valid data and asserting the WEN and CSN signals. The TXRDY signal will become inactive for one cycle while the data is being transferred from the transmit hold register to the transmit register that begins the serial transfer.

2. The transmission begins with a START bit, followed by data bits 0 through 6, the optional seventh bit, the optional parity bit, and finally the STOP bit.

3. Because the UART is double-buffered, data can be queued in the transmit hold register (cycle 7). The TXRDY line, when LOW, indicates that no more data can be transferred to the UART.

Once the previous serial transfer is complete, the data in the transmit hold register is passed to the transmit register, and the transfer begins. The TXRDY line is also asserted, indicating that the next data byte can be loaded.
Notes:

1. CoreUART continuously monitors the RX line, polling for a START bit. Once the START bit is detected, CoreUART registers the data stream. The optional parity bit is also registered and checked. A START bit is defined as logic 0-bit value on the RX line when the core is idle.

2. The data is then loaded into the receive hold buffer, and the RXRDY signal is asserted. The RXRDY signal will remain asserted until the data is read externally, indicated by the simultaneous assertion of CSN and OEN.

Notes:

1. When a parity error occurs, the PARITY_ERR signal is asserted.

2. The error is cleared by the same method used to read the data, simultaneous assertion of CSN and OEN.
Notes:

1. When a data overflow error occurs, the OVERFLOW signal is asserted.
2. The previous data is held and the new data is lost. The error is cleared by the same method used to read the data, simultaneous assertion of CSN and OEN.

Notes:

1. Legacy mode is disabled in this timing diagram.
2. In Normal (non-Legacy) mode, RXRDY and FRAMING_ERR are synchronized. They are asserted in the same clock cycle. The error is cleared using a read operation - simultaneous assertion of OEN and CSN.
Notes:

1. Legacy mode is enabled in this timing diagram.
2. In Legacy mode, RXRDY is asserted one cycle before FRAMING_ERR is asserted. The error is cleared using a read operation: simultaneous assertion of OEN and CSN.
Testbench Operation

Two testbenches are provided with CoreUART: Verilog and VHDL verification testbenches. These are complex testbenches that verify core operation. These testbenches exercise all the features of the core. Actel recommends not modifying these testbenches.

User Testbench

CoreUART is provided with a user testbench (Figure 5-1) to demonstrate sample UART operation. The testbenches are available in both Verilog and VHDL and contain two instances of CoreUART connected to each other. The source code is made available with Obfuscated and RTL licenses of the core.

![Figure 5-1 · Verification Testbench](image)

The testbench contains the tests listed in Table 5-1.

<table>
<thead>
<tr>
<th>No.</th>
<th>Bit</th>
<th>Parity</th>
<th>Parity Setting</th>
<th>Parity Error</th>
<th>Overflow Error</th>
<th>Framing Error</th>
<th>Procedure Call</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8</td>
<td>Enabled</td>
<td>Even</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>txrxtest</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>Enabled</td>
<td>Odd</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>txrxtest</td>
</tr>
<tr>
<td>3</td>
<td>7</td>
<td>Enabled</td>
<td>Even</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>txrxtest</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td>Enabled</td>
<td>Odd</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>txrxtest</td>
</tr>
<tr>
<td>5</td>
<td>8</td>
<td>Disabled</td>
<td>N/A</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>txrxtest</td>
</tr>
<tr>
<td>6</td>
<td>8</td>
<td>Disabled</td>
<td>N/A</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>txrxtest</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>Disabled</td>
<td>N/A</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>txrxtest</td>
</tr>
<tr>
<td>8</td>
<td>7</td>
<td>Disabled</td>
<td>N/A</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>txrxtest</td>
</tr>
<tr>
<td>9</td>
<td>8</td>
<td>Enabled</td>
<td>Even</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>paritytest</td>
</tr>
<tr>
<td>10</td>
<td>8</td>
<td>Enabled</td>
<td>Odd</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>paritytest</td>
</tr>
<tr>
<td>11</td>
<td>7</td>
<td>Enabled</td>
<td>Even</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>paritytest</td>
</tr>
<tr>
<td>12</td>
<td>7</td>
<td>Enabled</td>
<td>Odd</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>paritytest</td>
</tr>
<tr>
<td>13</td>
<td>8</td>
<td>Enabled</td>
<td>Odd</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>testoverflow</td>
</tr>
<tr>
<td>14</td>
<td>8</td>
<td>Enabled</td>
<td>Odd</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>N/A</td>
</tr>
</tbody>
</table>

The procedure calls txrxtest, paritytest, and testoverflow are defined in the file tbpack.vhd. The top-level testbench, testbench.vhd, utilizes these procedures to perform the corresponding tests listed in Table 5-1.

Refer to the source directory on the release CD for source code for the testbench.
Ordering Information

Ordering Codes

CoreUART can be ordered through your local Actel sales representative. It should be ordered using the following number scheme: CoreUART-XX, where XX is listed in Table 6-1.

<table>
<thead>
<tr>
<th>XX</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OM</td>
<td>RTL for Obfuscated RTL – multiple-use license</td>
</tr>
<tr>
<td>RM</td>
<td>RTL for RTL source – multiple-use license</td>
</tr>
</tbody>
</table>

*Note:* CoreUART-OM is included free with a Libero IDE license
# List of Document Changes

The following table lists critical changes that were made in the current version of the document.

<table>
<thead>
<tr>
<th>Previous Version</th>
<th>Changes in Current Version (v2.1)</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>December 08</td>
<td>Added framing error (FRAMING_ERR signal) support</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>Updated tool flow section with SmartDesign flow</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>Updated Table 3-1 to include FRAMING_ERR signal</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>Updated Table 3-2 with new configurable options</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>Added Framing Error and Framing Error in Legacy Mode timing diagrams</td>
<td>17–18</td>
</tr>
</tbody>
</table>
Product Support

Actel backs its products with various support services including Customer Service, a Customer Technical Support Center, a web site, an FTP site, electronic mail, and worldwide sales offices. This appendix contains information about contacting Actel and using these support services.

Customer Service
Contact Customer Service for non-technical product support, such as product pricing, product upgrades, update information, order status, and authorization.
From Northeast and North Central U.S.A., call 650.318.4480
From Southeast and Southwest U.S.A., call 650.318.4480
From South Central U.S.A., call 650.318.4434
From Northwest U.S.A., call 650.318.4434
From Canada, call 650.318.4480
From Europe, call 650.318.4252 or +44 (0) 1276 401 500
From Japan, call 650.318.4743
From the rest of the world, call 650.318.4743
Fax, from anywhere in the world 650.318.8044

Actel Customer Technical Support Center
Actel staffs its Customer Technical Support Center with highly skilled engineers who can help answer your hardware, software, and design questions. The Customer Technical Support Center spends a great deal of time creating application notes and answers to FAQs. So, before you contact us, please visit our online resources. It is very likely we have already answered your questions.

Actel Technical Support
Visit the Actel Customer Support website (www.actel.com/custsup/search.html) for more information and support. Many answers available on the searchable web resource include diagrams, illustrations, and links to other resources on the Actel web site.

Website
You can browse a variety of technical and non-technical information on Actel’s home page, at www.actel.com.

Contacting the Customer Technical Support Center
Highly skilled engineers staff the Technical Support Center from 7:00 A.M. to 6:00 P.M., Pacific Time, Monday through Friday. Several ways of contacting the Center follow:

Email
You can communicate your technical questions to our email address and receive answers back by email, fax, or phone. Also, if you have design problems, you can email your design files to receive assistance. We constantly monitor the email account throughout the day. When sending your request to us, please be sure to include your full name, company name, and your contact information for efficient processing of your request.
The technical support email address is tech@actel.com.
Phone

Our Technical Support Center answers all calls. The center retrieves information, such as your name, company name, phone number and your question, and then issues a case number. The Center then forwards the information to a queue where the first available application engineer receives the data and returns your call. The phone hours are from 7:00 A.M. to 6:00 P.M., Pacific Time, Monday through Friday. The Technical Support numbers are:

650.318.4460
800.262.1060

Customers needing assistance outside the US time zones can either contact technical support via email (tech@actel.com) or contact a local sales office. Sales office listings can be found at www.actel.com/contact/offices/index.html.
Index

A
Actel
  electronic mail 25
telephone 26
  web-based technical support 25
website 25

B
baud generator 7
block diagram 7
  FIFO mode 8
  normal mode 7
buffers
  receive 7
  transmit 7

C
configurable options 14
contacting Actel
  customer service 25
electronic mail 25
telephone 26
  web-based technical support 25
control inputs 7
core versions 5
CoreUART
  configuration window, 11
customer service 25

D
description, general 5
device
  utilization and performance 8
double-buffering 7

F
FIFO mode 7
FIFOs
  receive 7
  transmit 7
functional description 7

G
general description 5

I
I/O signals 13
interfaces 13

N
normal mode 7

O
ordering code 21

P
parity
  errors 7
product support 25–26
  customer service 25
electronic mail 25
technical support 25
telephone 26
website 25

S
signals, I/O 13
SmartDesign 11
state machines
  receive 7
  transmit 7

T
technical support 25
testbenches 19
  operation 19, 21
  verification 19
timing diagrams 15
  framing error 17
  framing error in legacy mode 18
  overflow error 17
  parity error 16
  serial receive 16
  serial transmit 15

U
use cases 5

V
verification testbench 19
versions, core 5
W
web-based technical support 25
Actel Corporation • 2061 Stierlin Court • Mountain View, CA 94043 • USA
Phone 650.318.4200 • Fax 650.318.4600 • Customer Service: 650.318.1010 • Customer Applications Center: 800.262.1060

Actel Europe Ltd. • River Court, Meadows Business Park • Station Approach, Blackwater • Camberley Surrey GU17 9AB • United Kingdom
Phone +44 (0) 1276 609 300 • Fax +44 (0) 1276 607 540

Actel Japan • EXOS Ebisu Building 4F • 1-24-14 Ebisu Shibuya-ku • Tokyo 150 • Japan
Phone +81.03.3445.7671 • Fax +81.03.3445.7668 • http://jp.actel.com

Actel Hong Kong • Room 2107, China Resources Building • 26 Harbour Road • Wanchai • Hong Kong
Phone +852 2185 6460 • Fax +852 2185 6488 • www.actel.com.cn

Actel is the leader in low-power and mixed-signal FPGAs and offers the most comprehensive portfolio of system and power management solutions. Power Matters. Learn more at www.actel.com.