CoreUART Handbook

v2.0

Actel®

Actel Corporation, Mountain View, CA 94043

Printed in the United States of America

Part Number: 50200095-0

Release: March 2007

Trademarks

Table of Contents

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 on page 5 illustrates the various usages of CoreUART.

Case A in Figure 1 on page 5 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.

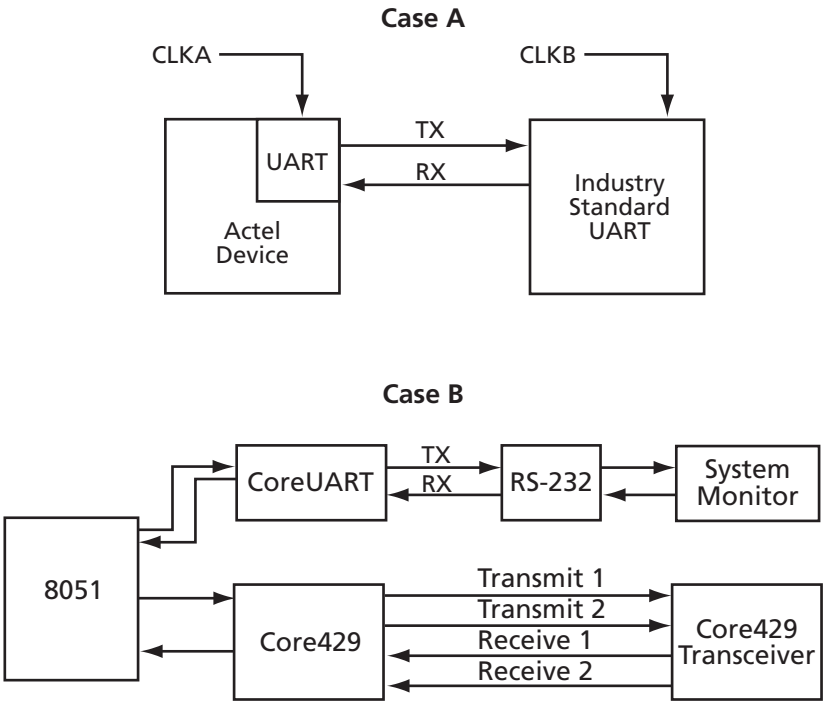


Figure 1 · System Block Diagram Depicting CoreUART Usage

Core Versions

This handbook applies to CoreUART v3.1. The release notes provided with the core list known discrepancies between this handbook and the core release associated with the release notes.

1 Functional Block Description

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.

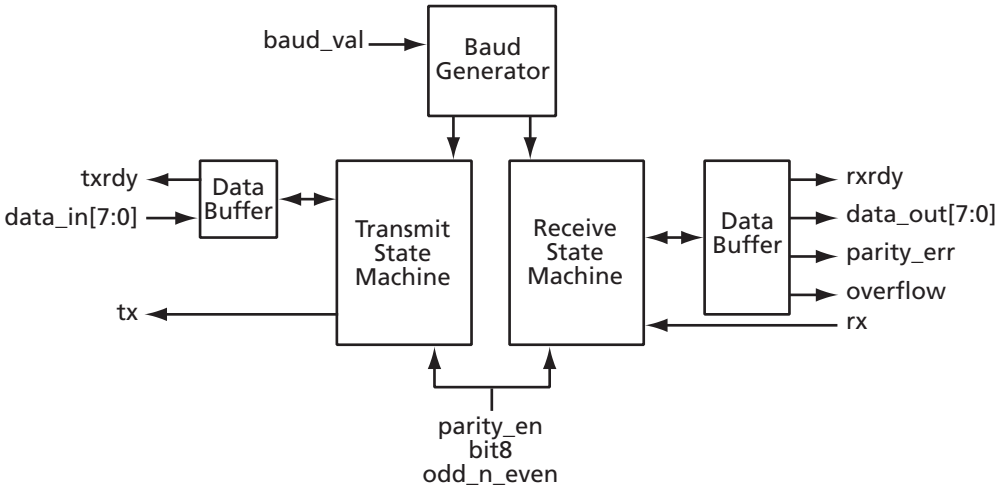


Figure 1-1 · Block Diagram of CoreUART Normal Functionality

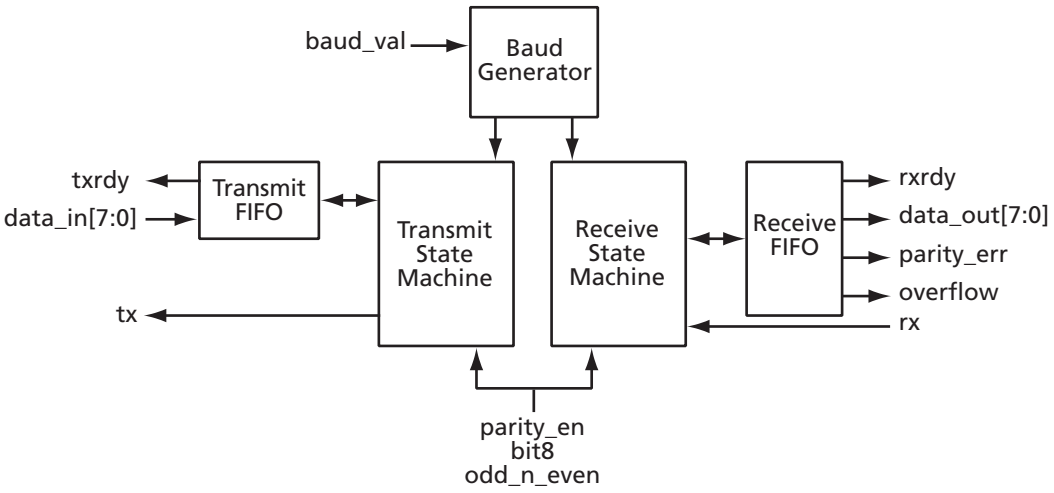


Figure 1-2 · Block Diagram of CoreUART with FIFO Functionality

Device Utilization and Performance

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

Table 1-1 · CoreUART Utilization in FIFO Mode

Family	Cells or Tiles			Memory Blocks	Utilization		Performance MHz
	Sequential	Combinatorial	Total		Device	Total	
Fusion	105	166	271	2	AFS600	2%	119
IGLOO™/e	105	166	271	2	AGL600	2%	119
ProASIC®3/E	105	166	271	2	A3P600	2%	119
ProASIC^{PLUS}®	105	238	343	2	APA150	6%	77
Axcelerator®	104	109	213	2	AX500	3%	171
RTAX-S	155	175	330	2	RTAX250S	8%	100
SX-A	430	309	739	0	A54SX16A	51%	96
RTSX-S	432	308	740	0	RT54SX32S	26%	62

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.*

Table 1-2 · CoreUART Utilization in Normal Mode

Family	Cells or Tiles			Memory Blocks	Utilization		Performance MHz
	Sequential	Combinatorial	Total		Device	Total	
Fusion	79	147	226	0	AFS600	2%	108
IGLOO/e	79	147	226	0	AGL600	2%	108
ProASIC3/E	79	147	226	0	A3P600	2%	108
ProASIC^{PLUS}	79	226	305	0	APA150	5%	91
Axcelerator	80	90	170	0	AX500	2%	190
RTAX-S	80	90	170	0	RTAX250S	4%	147
SX-A	82	93	175	0	A54SX16S	12%	138
RTSX-S	80	92	172	0	RT54SX32S	6%	87

Note: 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.

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.

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.

$$\text{baud rate} = \frac{\text{clk}}{(\text{baudval} + 1) \times 16}$$

EQ 1-1

where

clk = the frequency of the system clock in hertz
baud rate = the desired baud rate
baudval = 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_value of 214 decimal or D6 hex. So, to get the desired baud rate, the user should assign 16#D6 to BAUD_VAL input.

2 Tool Flows

Licenses

CoreUART is licensed in three ways, depending on your license. Tool flow functionality may be limited.

Evaluation

Precompiled simulation libraries are provided, allowing the core to be instantiated in CoreConsole and simulated within Actel Libero® Integrated Design Environment (IDE), as described in the "CoreConsole" section. Using the Evaluation version of the core, it is possible to create and simulate the complete design in which the core is being included. The design may not be synthesized, as source code is not provided.

Obfuscated

Complete RTL code is provided for the core, enabling the core to be instantiated with CoreConsole. Simulation, Synthesis, and Layout can be performed with Libero IDE. The RTL code for the core is obfuscated, and some of the testbench source files are not provided. They are precompiled into the compiled simulation library instead.

RTL

Complete RTL source code is provided for the core and testbenches.

CoreConsole

CoreUART is pre-installed in the CoreConsole IP Deployment Platform (IDP). To use the core, click and drag it from the IP core list into the main window. The CoreConsole project can be exported to Libero IDE at this point, providing access to the core only. Alternatively, IP blocks can be interconnected, allowing the complete system to be exported from CoreConsole to Libero IDE.

The core can be configured using the configuration GUI within CoreConsole, as shown in Figure 2-1.

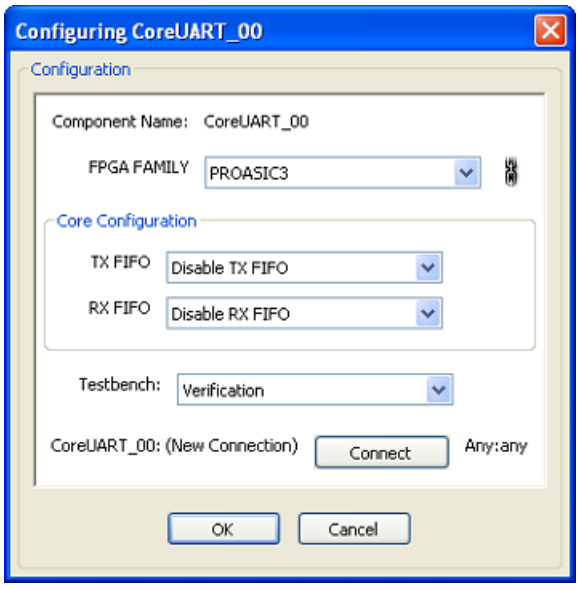


Figure 2-1 · CoreUART Configuration within CoreConsole

After configuring the core, Actel recommends that you use the top-level Auto Stitch function to connect all the core interface signals to the top level of the CoreConsole project.

Once the core is configured, invoke the **Generate** function in CoreConsole. This will export all the required files to the project directory in the *LiberoExport* directory. This is in the CoreConsole installation directory by default.

Importing into Libero IDE

After generating and exporting the core from CoreConsole, it can be imported into Libero IDE. Create a new project in Libero IDE, and import the CoreConsole project from the *LiberoExport* directory. Libero IDE will then install the core and the selected testbenches, along with constraints and documentation, into its project.

Note: If two or more DirectCores are required, they can both be included in the same CoreConsole project and imported into Libero IDE at the same time.

Simulation Flows

To run simulations, the required testbench flow must be selected within CoreConsole, and Save & Generate must be run from the Generate pane. The required testbench is selected through the core configuration GUI in CoreConsole. The following simulation environments are supported:

- Full CoreUART verification environment (Verilog and VHDL)

When CoreConsole generates the Libero IDE project, it will install the appropriate testbench files.

To run the testbenches, simply set the design root to the CoreUART instantiation in the Libero IDE file manager, and click the **Simulation** icon in Libero IDE. This will invoke Model*Sim* and automatically run the simulation.

Synthesis in Libero IDE

To run Synthesis on the core with parameters set in CoreConsole, set the design root to the top of the project imported from CoreConsole. This is a wrapper around the core that sets all the generics appropriately.

Make sure the required timing constraint files are associated with the synthesis tool.

Click the **Synthesis** icon in Libero IDE. The synthesis window appears, displaying the Synplicity® project. To run Synthesis, click the **Run** icon.

Place-and-Route in Libero IDE

Having set the design route appropriately and run Synthesis, click the **Layout** icon in Libero IDE to invoke Designer. CoreUART requires no special place-and-route settings.

Core Parameters

CoreUART Configurable Options

There are a number of configurable options that apply to CoreUART, as shown in Table 2-1. 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 2-1 · CoreUART Configurable Options

Configurable Options	Default Setting	Description
TX_FIFO	FIFO	Enables or disables transmit FIFO
RX_FIFO	FIFO	Enables or disables receive FIFO
Device Family	ProASIC3	Selects target family. Must be set to match the supported FPGA family. 8 – 54SXA 9 – RTSXS 11 – Axcelerator 12 – RTAX-S 14 – ProASIC PLUS 15 – ProASIC3 16 – ProASIC3E 17 – Fusion 20 – IGLOO 21 – IGLOOe

3 Core Interfaces

Signal descriptions for CoreUART are defined in Table 3-1. The signals are broken down into the following classes: system signals, parallel data transfer signals, serial control and status signals, and serial data signals. System signals consist of the CLK and reset_n signals. Parallel data transfer signals include data_in[7:0], data_out[7:0], wen, oen, and csn. Control signals are bit8, parity_en, odd_n_even, and baud_val. Status signals are txrdy, rxrdy, parity_err, and overflow. The serial data signals consist of rx and tx.

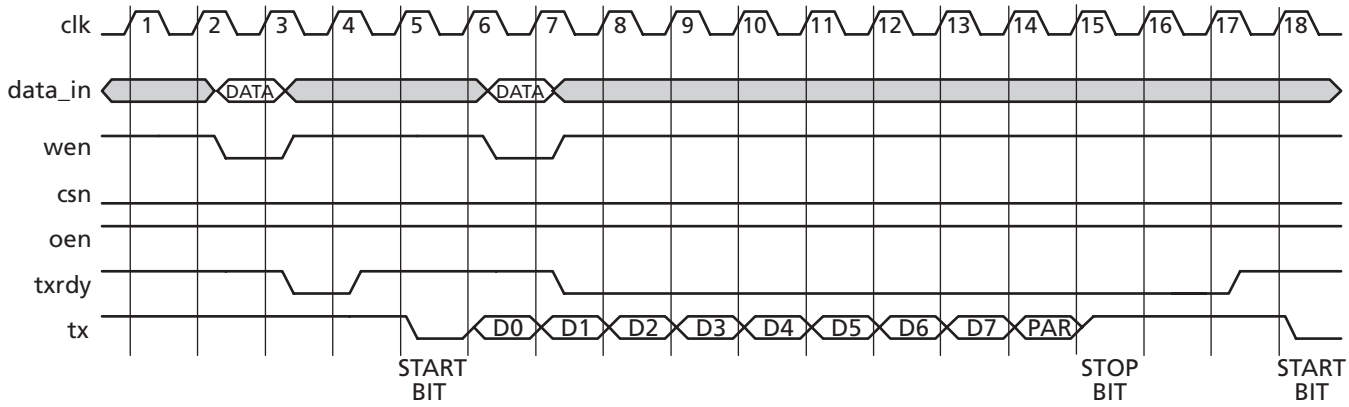
Table 3-1 · CoreUART Signals

Name*	Type	Description
clk	Input	Main system clock
reset_n	Input	Active low asynchronous reset
data_in[7:0]	Input	Transmit write data bus
data_out[7:0]	Output	Receive read data bus
wen	Input	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.
oen	Input	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).
csn	Input	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.
bit8	Input	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.”
parity_en	Input	Control bit to enable parity for both receive and transmit functions. Parity is enabled when the bit is set to logic 1.
odd_n_even	Input	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.
baud_val	Input	8-bit control bus used to define the baud rate
txrdy	Output	Status bit; when set to logic 0, indicates that the transmit data buffer/FIFO is not available for additional transmit data.
rxrdy	Output	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.
parity_err	Output	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.
overflow	Output	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.
rx	Input	Serial receive data
tx	Output	Serial transmit data

Note: **Active low signals are designated with a trailing lowercase* n.

4 Timing Diagrams

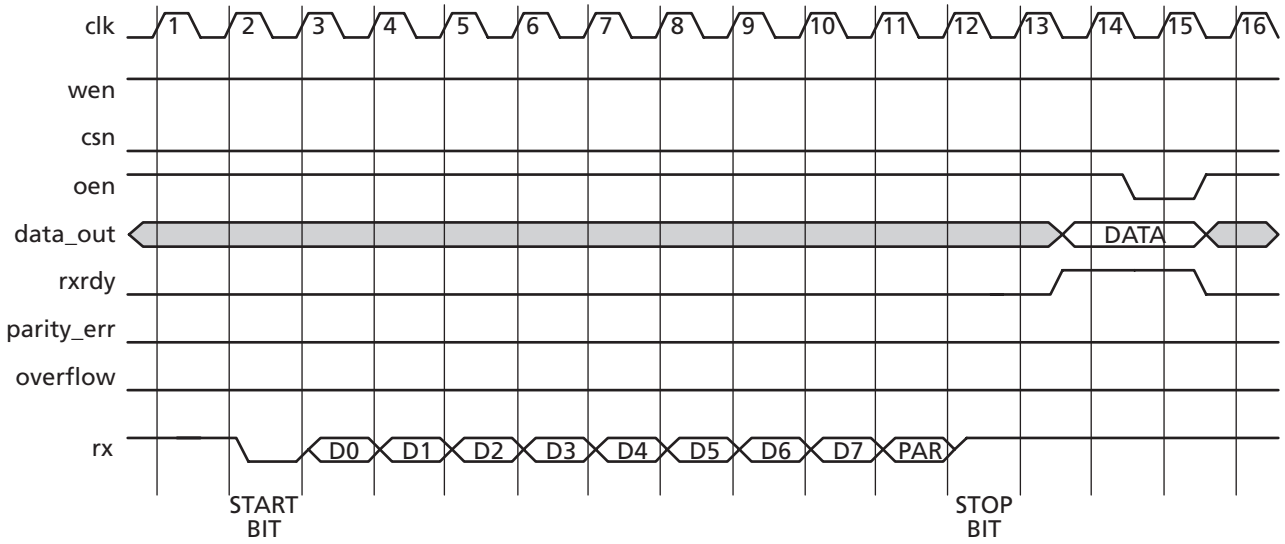
The UART waveforms can be broken down into a few basic functions: transmit data, receive data, and errors. Figure 4-1 shows serial transmit signals, and Figure 4-2 on page 18 shows serial receive signals. Figure 4-3 on page 18 and Figure 4-4 on page 19 show the parity and overflow error cycles, respectively. 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.



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.*
4. *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.*

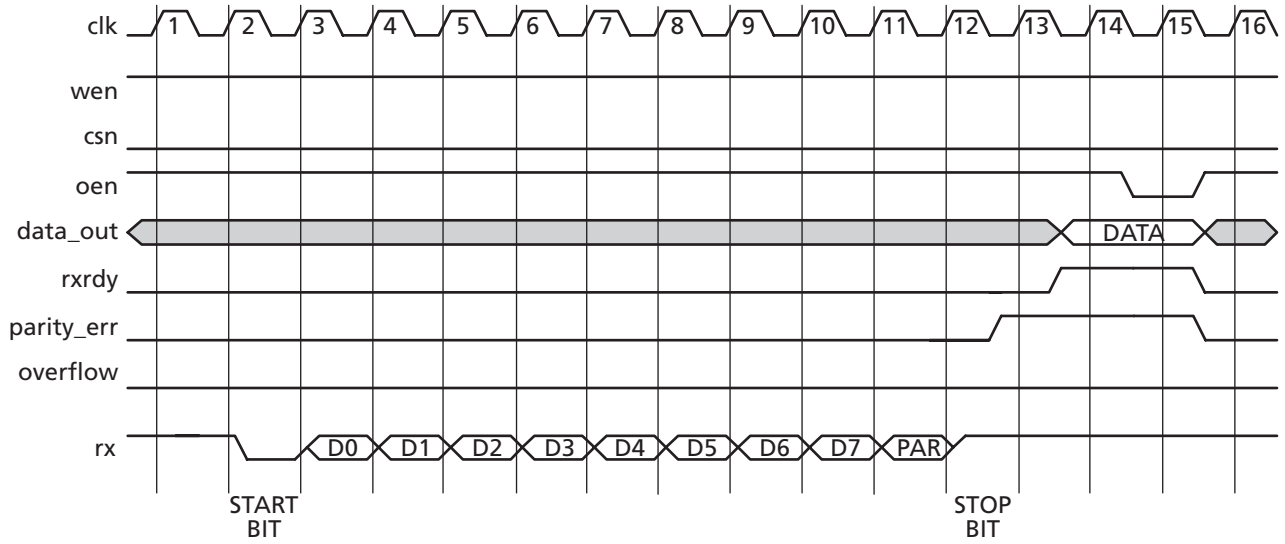
Figure 4-1 · Serial Transmit



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.*
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.*

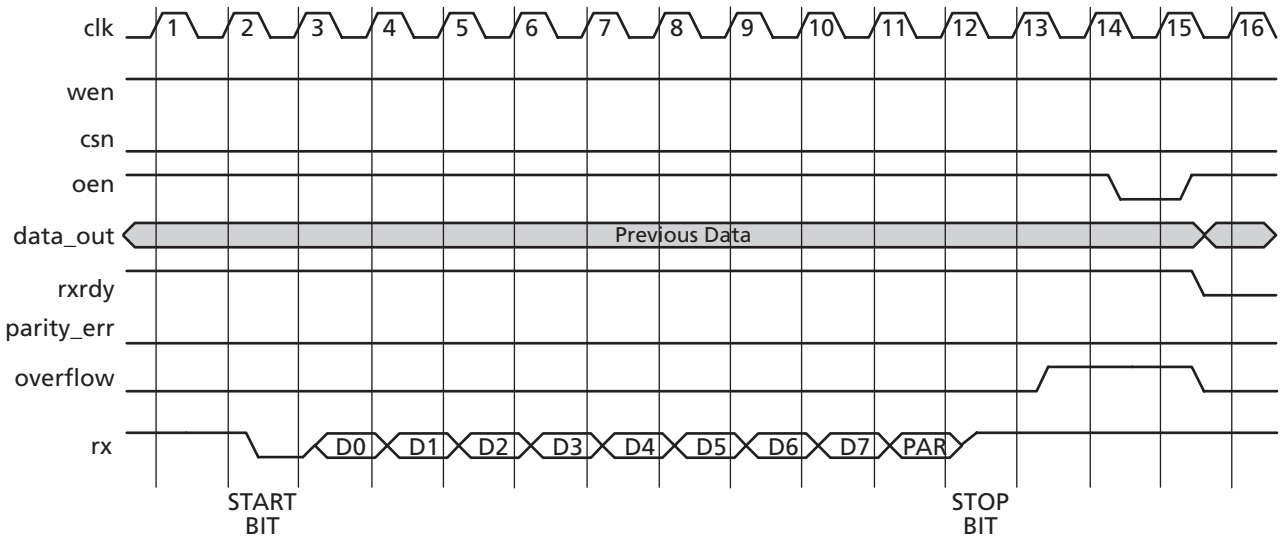
Figure 4-2 · Serial Receive



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.*

Figure 4-3 · Parity Error



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.*
3. *The error is cleared by the same method used to read the data, simultaneous assertion of csn and oen.*

Figure 4-4 · Overflow Error

5 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.

Verification Testbench

Actel has developed a verification testbench (Figure 5-1) that you can use to verify core performance. 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.

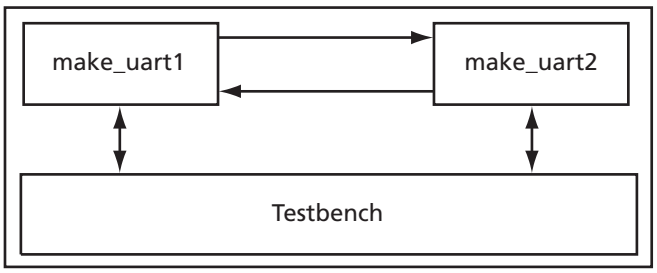


Figure 5-1 · Verification Testbench

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

Table 5-1 · Verification Tests

No.	Bit	Parity	Parity Setting	Parity Error	Overflow Error	Procedure Call
1	8	Enabled	Even	No	No	txrxtest
2	8	Enabled	Odd	No	No	txrxtest
3	7	Enabled	Even	No	No	txrxtest
4	7	Enabled	Odd	No	No	txrxtest
5	8	Disabled	N/A	No	No	txrxtest
6	8	Disabled	N/A	No	No	txrxtest
7	7	Disabled	N/A	No	No	txrxtest
8	7	Disabled	N/A	No	No	txrxtest
9	8	Enabled	Even	Yes	No	paritytest
10	8	Enabled	Odd	Yes	No	paritytest
11	7	Enabled	Even	Yes	No	paritytest
12	7	Enabled	Odd	Yes	No	paritytest
13	8	Enabled	Odd	No	Yes	testoverflow

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 on page 21.

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

A

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

For more information about Actel's products, visit our website at http://www.actel.com

Actel Corporation *• 2061 Stierlin Court • Mountain View, CA 94043 USA*
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 Bldg. 4F • 1-24-14 Ebisu Shibuya-ku • Tokyo 150 • Japan*
Phone +81.03.3445.7671 • Fax +81.03.3445.7668 • www.jp.actel.com

Actel Hong Kong *• Suite 2114, Two Pacific Place • 88 Queensway, Admiralty Hong Kong*
Phone +852 2185 6460 • Fax +852 2185 6488 • www.actel.com.cn

50200095-0 /3.07