CoreUARTapb Handbook

v2.0

Actel®

Actel Corporation, Mountain View, CA 94043

© 2007 Actel Corporation. All rights reserved.

Printed in the United States of America

Part Number: 50200101-0

Release: May 2007

No part of this document may be copied or reproduced in any form or by any means without prior written consent of Actel.

Actel makes no warranties with respect to this documentation and disclaims any implied warranties of merchantability or fitness for a particular purpose. Information in this document is subject to change without notice. Actel assumes no responsibility for any errors that may appear in this document.

This document contains confidential proprietary information that is not to be disclosed to any unauthorized person without prior written consent of Actel Corporation.

Trademarks

Actel and the Actel logo are registered trademarks of Actel Corporation.

Adobe and Acrobat Reader are registered trademarks of Adobe Systems, Inc.

All other products or brand names mentioned are trademarks or registered trademarks of their respective holders.

Table of Contents

Introduction

General Description

CoreUARTapb is a serial communication controller with a flexible serial data interface that is intended primarily for embedded systems. CoreUARTapb can be used to interface directly to industry standard UARTs. CoreUARTapb is intentionally a subset of full UART capability to make the function cost-effective in a programmable device. Figure 1 on page 5 illustrates system-level usages of CoreUARTapb.

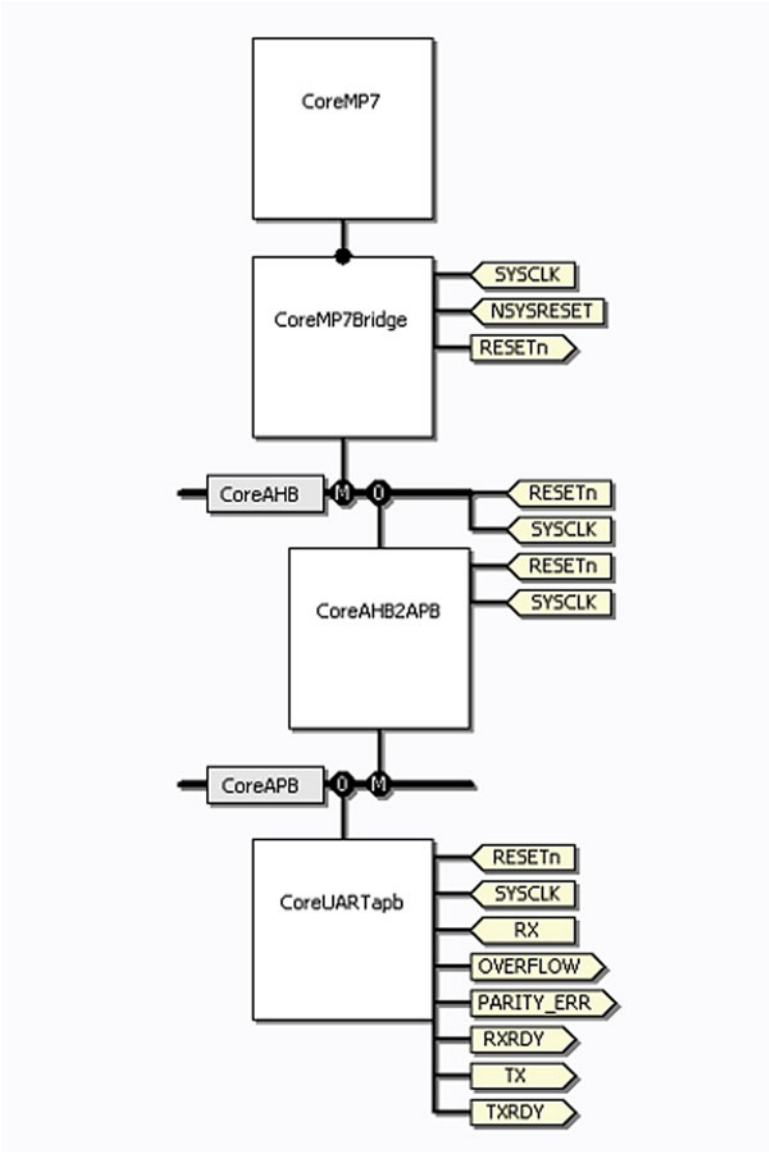

Figure 1 · System Block Diagram Depicting CoreUARTapb Usage

Core Versions

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

1 Functional Block Description

Figure 1-1 shows the block diagram of the CoreUARTapb normal mode functionality. Figure 1-2 on page 8 shows the block diagram of CoreUARTapb 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 or received. 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 transmit 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.

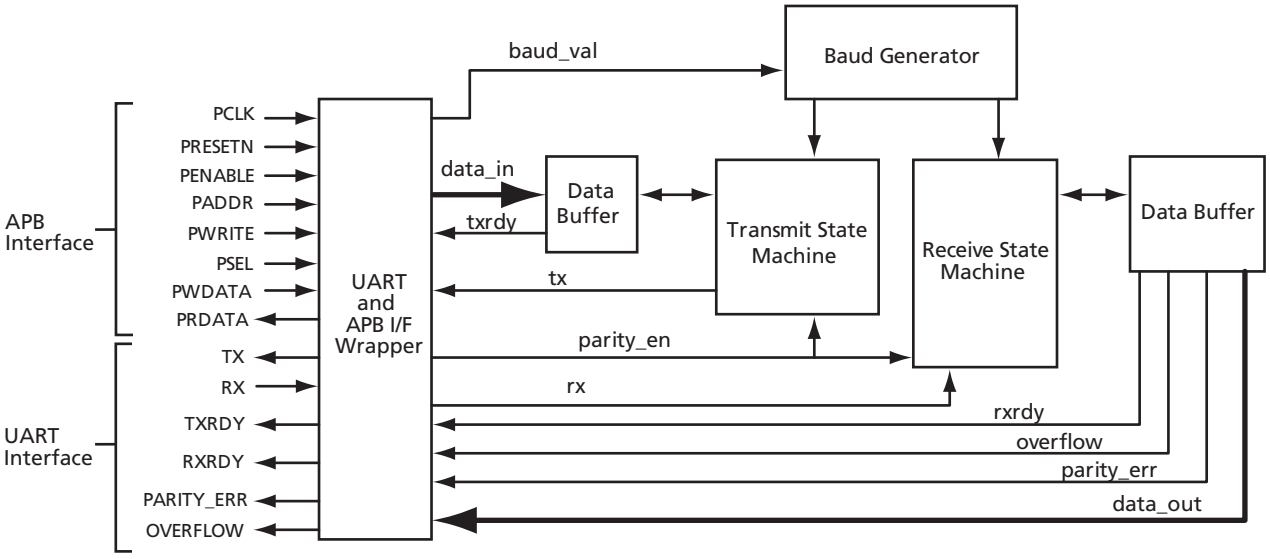

Figure 1-1 · Block Diagram of CoreUARTapb Normal Functionality

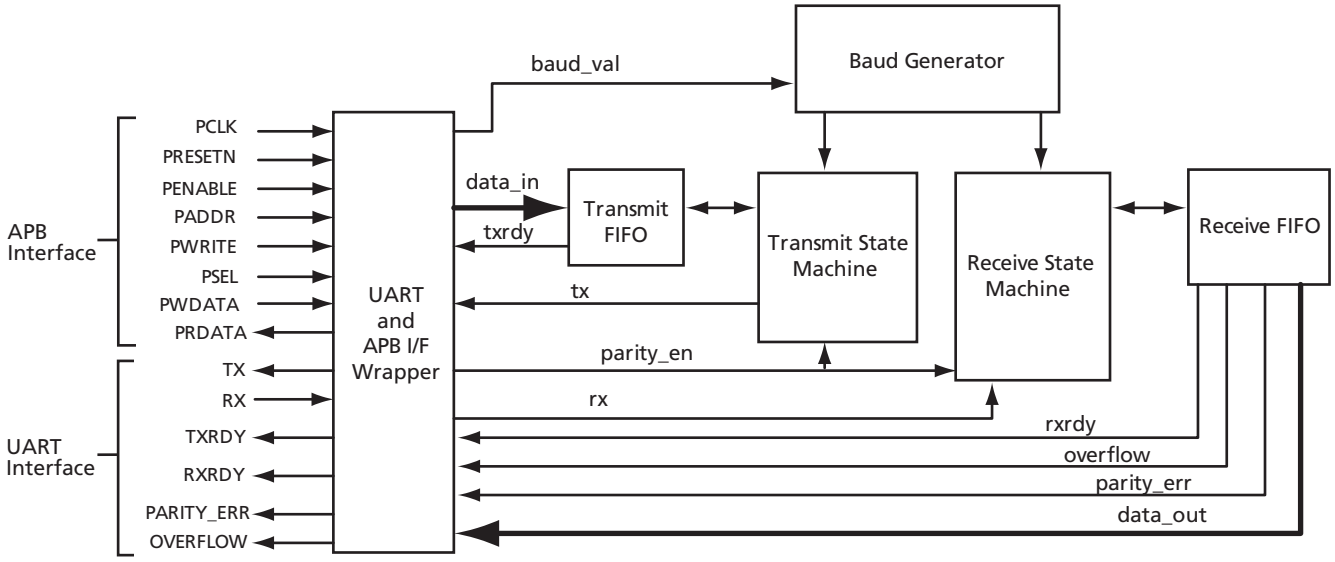

Figure 1-2 · Block Diagram of CoreUARTapb with FIFO Functionality

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 · CoreUARTapb 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. *CoreUARTapb 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 · CoreUARTapb 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: CoreUARTapb 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.

Fixed Mode Options

There are three options in Fixed mode CoreUARTapb operation:

1. Character size
2. Parity
3. Baud rate

These values are hardwired and cannot be changed during runtime.

Character Size

The default value for the number of data bits is 7. The option PRG_bit8 sets the serial bitstream to 8-bit data mode.

Parity

PRG_PARITY sets the parity enabled/disabled. It also sets parity Even/Odd.

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.

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

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

CoreUARTapb 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

CoreUARTapb 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.

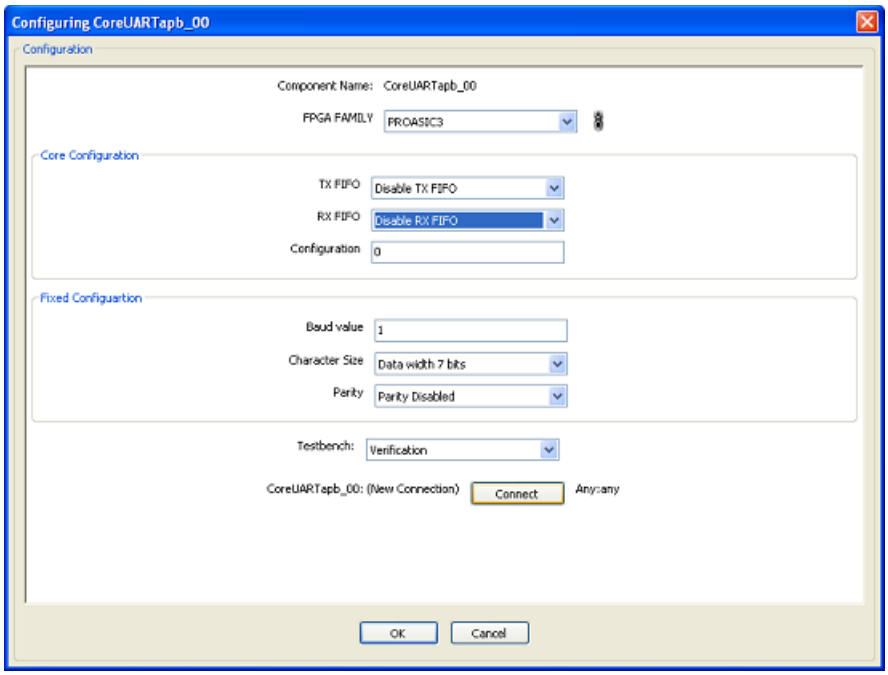

Figure 2-1 · CoreUARTapb Configuration within CoreConsole

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 CoreUARTapb 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 CoreUARTapb 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. CoreUARTapb requires no special place-and-route settings.

Core Parameters

CoreUARTapb Configurable Options

There are a number of configurable options that apply to CoreUARTapb, 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 · CoreUARTapb Configurable Options

Configurable Options	Default Setting	Description
TX_FIFO	Disable TX_FIFO	Enables or disables transmit FIFO.
RX_FIFO	Disable RX_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
Configuration	Programmable	0 – Programmable 1 – Fixed Fixed or Programmable mode. In Fixed mode, the parameters BAUD_VALUE, Character Size, and Parity are hardwired. In Programmable mode they are programmed by the control registers.

Table 2-1 · CoreUARTapb Configurable Options

Configurable Options	Default Setting	Description
BAUD_VALUE	1	Baud value is set only when configuration is set to fixed mode.
Character Size	7 bits	This option can only be set when configuration mode is set to fixed mode. This option defines the number of valid data bits in the serial bitstream. Character size can be 8 bits or 7 bits.
Parity	Parity disabled	This option can only be set when configuration mode is set to Fixed mode. The options for parity are as follows: Parity Disable, Even Parity, or Odd Parity.

3 Core Interfaces

Signal descriptions for CoreUARTapb are defined in Table 3-1. The APB interface allows access to the CoreUARTapb internal registers, FIFO, and internal memory. This interface is synchronous to the clock.

Table 3-1 · CoreUARTapb Signals

Name*	Type	Description
PCLK	In	Master clock input
PRESETN	In	Active low asynchronous reset
PWRITE	In	APB write/read enable, active high
PADDR[8:0]	In	APB address
PSEL	In	APB select
PENABLE	In	APB enable
PWDATA[APB_DATA_WIDTH–1:0]	In	APB data input
PRDATA[APB_DATA_WIDTH–1:0]	Out	APB data output
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 17 shows serial receive signals. Figure 4-3 on page 17 and Figure 4-4 on page 18 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.

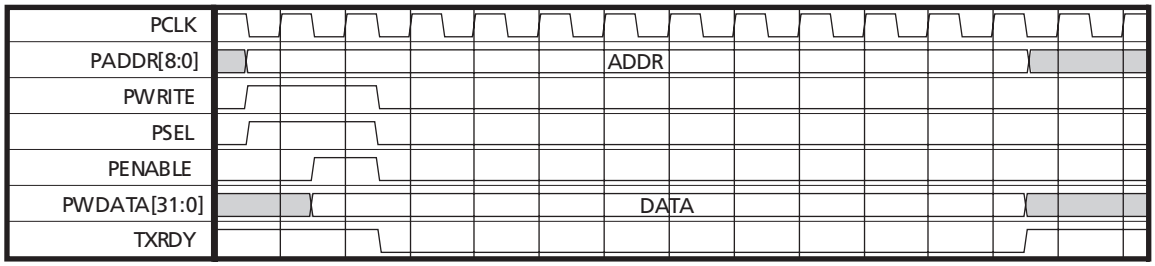

Figure 4-1 · Serial Transmit

Note: A serial transmit is initiated by writing data into CoreUARTapb. This is accomplished by providing valid data and asserting the PWRITE, PSEL, and PENABLE signals.

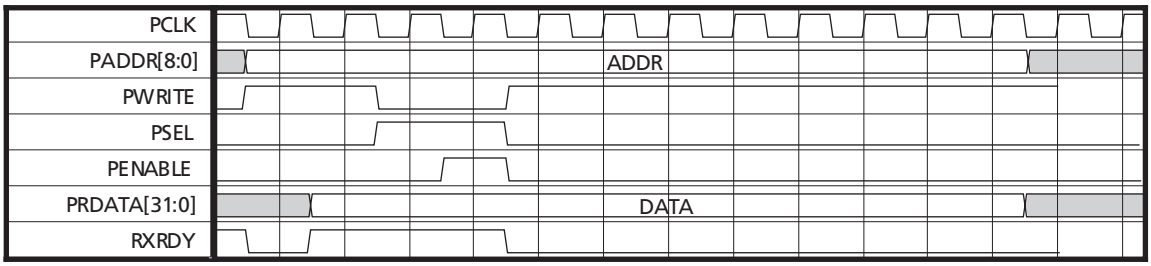

Figure 4-2 · Serial Receive

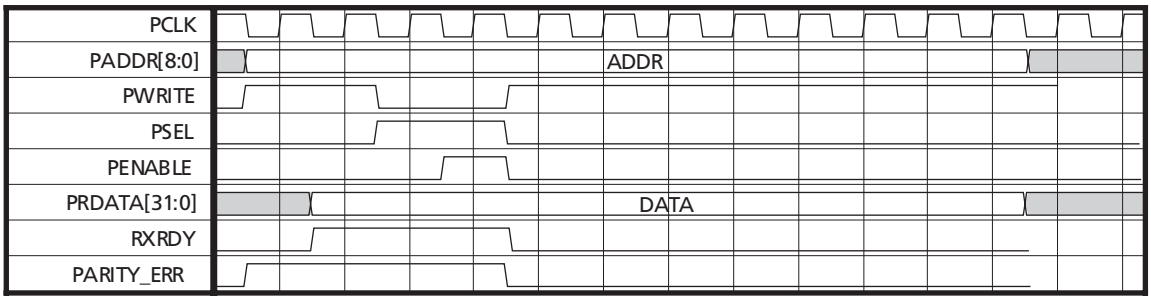

Figure 4-3 · Parity Error

Note: When a parity error occurs, the parity_err signal is asserted.

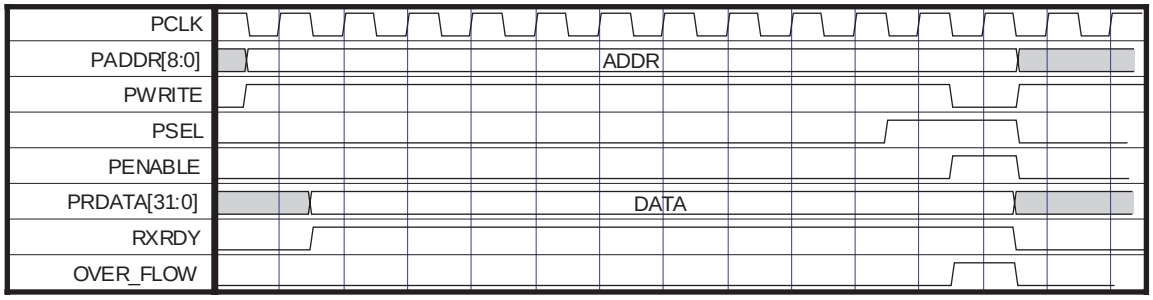

Figure 4-4 · Overflow Error

Note: When a data overflow error occurs, the overflow signal is asserted.

5

CoreUARTapb Configuration Registers

CoreUARTapb Programmer's Model

Table 5-1 lists the registers for CoreUARTapb.

Table 5-1 · CoreUARTapb Registers

Address	Type	Width	Reset Value	Name	Description
base + 0x000	Write	32	0	TxData	Transmit Data Register
base + 0x004	Read	32	0	RxData	Receive Data Register
base + 0x008	Read/Write	32	0	Ctrl1	Control Register 1
base + 0x00C	Read/Write	32	0	Ctrl2	Control Register 2
base + 0x010	Read	32	0	Status	Status Register

Transmit Data Register

The Transmit Data Register contains the 7- or 8-bit transmit data.

Receive Data Register

The Receive Data Register contains the 7- or 8-bit receive data.

Control Register 1

Control Register 1 contains a single field, baud value, used to set the baud rate for CoreUARTapb. The baud value should be set according to EQ 5-1:

$$\text{baud value (decimal)} = \frac{\text{clock}}{(16 \times \text{baud rate}) - 1}$$

EQ 5-1

where clock is the system clock frequency in hertz.

The result of this calculation must be rounded to the nearest integer and converted to hexadecimal to obtain the value that should be written to Control Register 1, shown in Table 5-2. For example, when the clock frequency is 10 MHz and a baud rate of 9.600 is desired, 0x41 should be written to Control Register 1.

Table 5-2 · Control Register 1

Bit(s)	Name	Type	Function
7:0	Baud value	Read/write	8-bit value setting the baud rate

Control Register 2

Table 5-3 shows Control Register 2, which is used to assign values to the configuration inputs available on CoreUARTapb.

Table 5-3 · Control Register 2

Bit(s)	Name	Type	Function
0	bit8	Read/write	Data width setting: bit8 = 0: 7-bit data bit8 = 1: 8-bit data
1	parity_en	Read/write	Parity is enabled when this bit is set to 1.
2	odd_n_even	Read/write	Parity is set as follows: odd_n_even = 0: even odd_n_even = 1: odd
7:3			Unused

Status Register

Table 5-3 shows the Status Register, which provides information on the status of CoreUARTapb.

Table 5-4 · Status Register

Bit(s)	Name	Type	Function
0	txrdy	Read only	When LOW, the transmit data buffer/FIFO is not available for additional transmit data.
1	rxrdy	Read only	When HIGH, data is available in the receive data buffer/FIFO. This bit is cleared by reading the Receive Data Register.
2	parity-en	Read only	When HIGH, a parity error has occurred during a receive transaction. This bit is cleared by reading the Receive Data Register.
3	overflow	Read only	When HIGH, a receive overflow occurs. This bit is cleared by reading the Receive Data Register.
7:4			Unused

6 Testbench Operation

Two testbenches are provided with CoreUARTapb: 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 6-1) that you can use to verify core performance. The testbenches are available in both Verilog and VHDL and contain two instances of CoreUARTapb connected to each other. The source code is made available with Obfuscated and RTL licenses of the core.

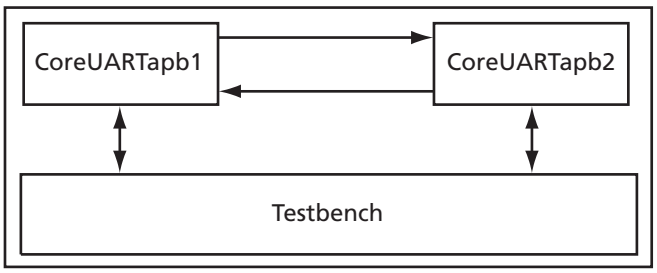

Figure 6-1 · Verification Testbench

The testbench contains the tests listed in Table 6-1.

Table 6-1 · 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 6-1 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

50200101-0 /5.07