Core429

Demonstration and Development Kit User's Guide

Actel®

Actel Corporation, Mountain View, CA 94043

Printed in the United States of America

Part Number: 50200042-0

Release: December 2004

Trademarks

Table of Contents

Introduction

The Core429 Demonstration kit allows users to evaluate the functionality of the Actel Core429 with a full development kit that includes Core429 FPGA programming files, ARINC (Aeronautical Radio, Inc.) 429 physical connections, example software, and full user documentation. The files included with the Core429 Development Kit CD (or available for download for use with the Platform8051 Development Kit) provide the user with the programming files necessary to program a ProASIC^{PLUS} APA600 to create a four-transmit and four-receive channel ARINC platform. The targeted FPGA (APA600) is mounted on the Platform8051 (PF8051) Development Board and ARINC 429 physical connections are included on the Core429 Daughter Board, which plugs directly onto Platform8051. With Core429 programmed into the FPGA and the Core429 Daughter Board connected to the Platform8051 Development Board (Figure 1), the customer has control of a complete ARINC 429 evaluation system.

Note: Core429 was designed per the ARINC 429-16 Specification.

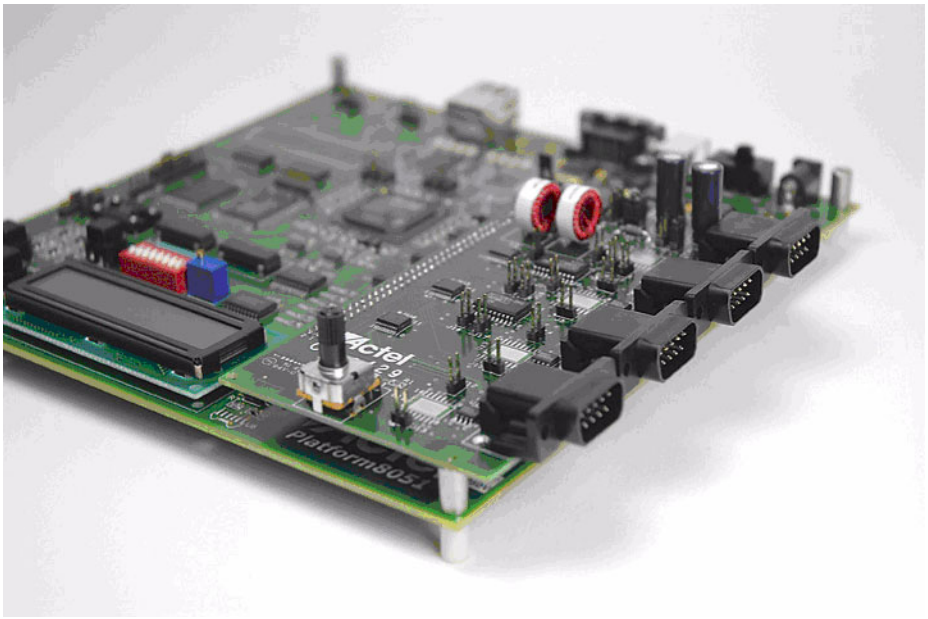

Figure 1. Platform8051 Development Board with Core429 Daughter Board Attached

This user's guide covers the following procedures:

- Setting up the Platform8051 Development Board
- Setting up the Core429 Daughter Board
- Configuring the Platform8051 Development Board to run the demonstration
- Running the demonstration
- Modifying the demonstration

Using This Document

This user's guide describes the content, architecture, and guidelines for working with the Core429 Demonstration and Development Kit. Use this document for the following:

- User information and description of the Core429 demonstration design
- Detailed reference material to be used when implementing a new design using any of the following:

  Platform8051 Development Kit (part number PF8051-DEV-KIT)

  Core429 Development Kit (Core429-DEV-KIT)

  PF8051-DEV-KIT and Core429 Daughter Board (Core429-SA)

Note: PF8051 refers to the Platform8051 Development Board; Core429-SA refers to the Core429 Standalone Daughter Card; and the Core429 Dev-Kit includes the Daughter Card.

A Core429 demonstration can be run using only a Platform8051 Development Board, provided the board is reprogrammed with the appropriate demonstration designs (see "Programming the FPGA" on page 27, "Programming the Flash Memory" on page 29, and "Operation on a Platform8051 Board" on page 31 for programming instructions). This evaluation can also be run on the Core429 Development Kit using the preprogrammed design.

Required Items

The following items are required to run the Core429 evaluation:

- Core429-DEV-KIT CD (or downloaded files)
- Platform8051 Development Board
- HyperTerminal or a similar serial communication program
- PC system running Windows® 2000 operating system or higher

Optional Items:

- Core429 Development Kit
- Core429 Daughter Card (not required because of loopback mode)
- FlashPro Lite
- FlashPro software, v3.2 or higher

Core429 Development Kit Contents

- Platform8051 Development Board with ProASIC^{PLUS} APA600-FG676 (Figure 1 on page 5)
- Core429 Daughter Board (Figure 1-3 on page 19)
- RS-232 cable
- Core429 Cable (DB9 connector cable modified for ARINC 429 signals)
- Universal power cable
- Core429-DEV-KIT CD
- Libero® Integrated Design Environment (IDE) Platinum Evaluation CD
- Documentation (user's guide and release notes)

Core429 Development Kit CD Contents

The Core429 Development Kit includes a CD containing Actel example files, documentation, and programming files for the example design. This CD contains folders for documentation (including a release note and this user's guide). Other folders on the CD contain printed circuit board (PCB) information (schematic, layout, and bill of material (BOM) files) and FPGA example design source files for the hardware design (minus Core429 and Core8051, which must be purchased separately). Although not all the source code for the FPGA designs is on the CD, the top-level source file integrating the various cores is provided on the CD. This is a very useful reference for understanding and evaluating the components. The FPGA folder also contains a *bitstream* subfolder with a copy of the programming files for the example FPGA designs.

The files necessary to run the Core429 demonstration are either contained on the Core429 Development Kit CD or available for download for use with the Platform8051 Development Board and the Core429 Daughter Board. The CD (or zip file) contains netlist reference documents, top-level VHDL source files, Designer files of the demonstration, and example script files.

Note: Only top-level VHDL source files are provided as an example of how to interface the core. Lower levels are not supplied as they contain run-time library (RTL) code that is subject to separate IP license agreements.

Recommended Additional Products

Actel recommends these additional products for use with this development kit:

- Actel FlashPro Lite (order code "FlashPro Lite")
- Core429 IP License (order code "Core429-EV/SN/AN/SR/AR")
- Core8051 IP License (order code "Core8051-EV/SN/AN/SR/AR")
- Actel Libero IDE Software License (order code "LIB-PL-PC-N-1YR")

Related Documents

Platform8051 Development Kit User's Guide

Core429 Bus Interface Datasheet

http://www.actel.com/ipdocs/CoreARINC429_DS.pdf

Core8051 Datasheet

http://www.actel.com/ipdocs/Core8051DS.pdf

1 The Core429 Demonstration

Core429

Core429 provides a bus interface for as many as 16 ARINC 429 receivers and transmitters. The core is supported in all recent Actel antifuse, flash, and radiation-tolerant product families. A typical system implementation using the Core429 is shown in Figure 1-1.

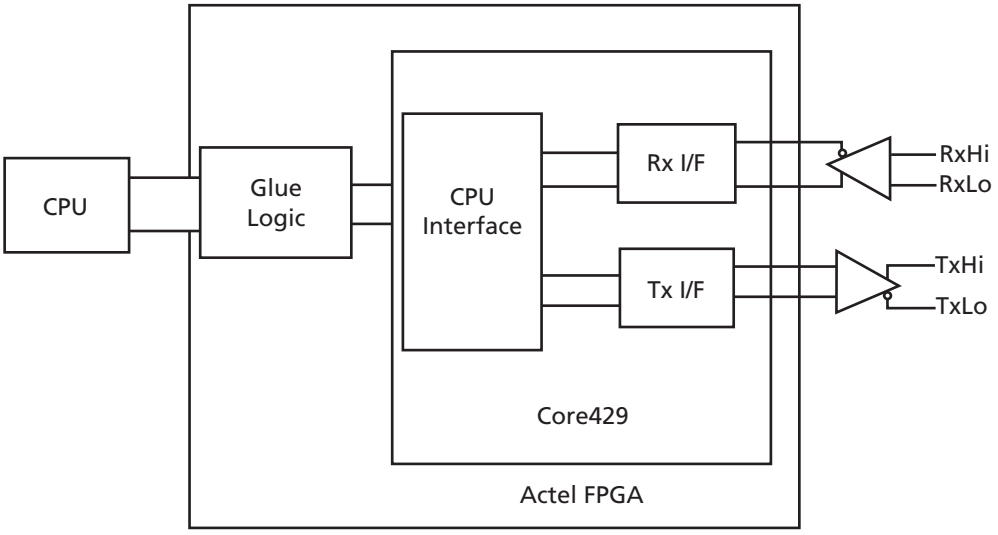

Figure 1-1. Core429 System with One Tx and One Rx

A typical ARINC 429 system requires a connection to a host CPU, which is used to set up the core and initialize the label definitions within the core. Core8051 is used as the host CPU in the Core429 demonstration design. Connection to an ARINC 429 data bus requires external line drivers and line receivers, as shown in Figure 1-1.

Three versions of the core are available:

- An evaluation version that allows core simulation with the Actel Libero IDE toolset or with Model*Sim*
- A netlist version that provides VHDL and Verilog netlists and pre-compiled testbenches, which include top-level RTL code so the ports can be understood
- An RTL version with full access to the source code

See the *Core429 Bus Interface datasheet* for more information.

The Core429 Demonstration Design

The Core429 demonstration design (Figure 1-2 on page 11) allows the user to evaluate the ARINC 429 Bus Interface by implementing four ARINC 429 transmitters and four ARINC 429 receivers in a single APA600 FPGA.

FPGA Architecture

The Core429 demonstration design consists of the following blocks:

- ARINC 429 Receiver Interface
- ARINC 429 Transmitter Interface
- Core429 CPU Interface
- Core8051 CPU (refer to the *Core8051 Datasheet*, page 3, Figure 1)
- Core8051 Special Function Register (SFR) Interface
- Core8051 Serial Interface

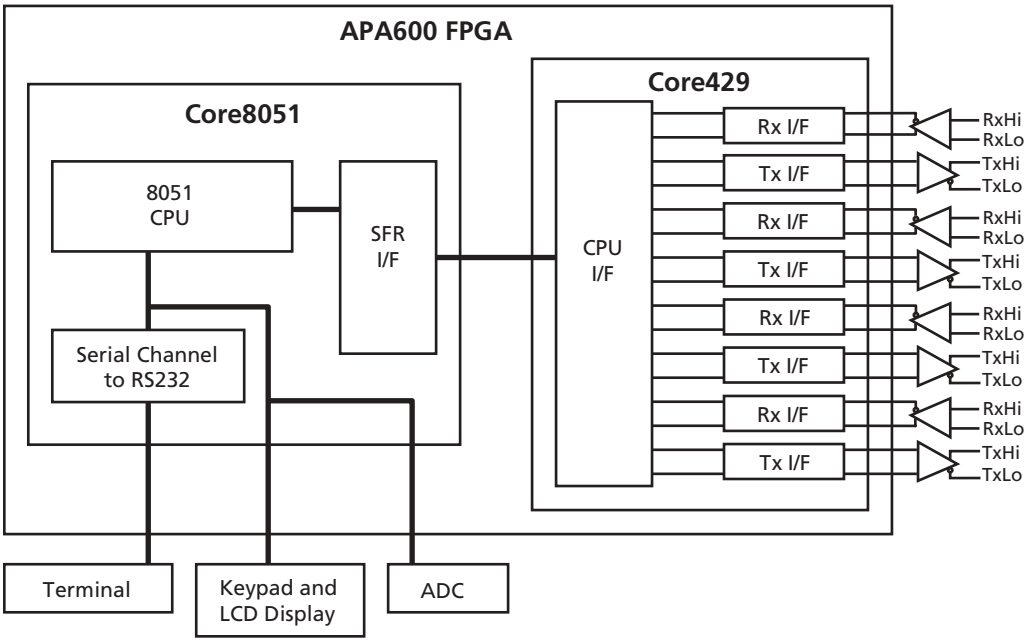

Figure 1-2. Core429 Demonstration Design Architecture

ARINC 429 Receiver Interface

Each Rx block is responsible for assembling the incoming serial data into 32-bit ARINC 429 words. The CPU can configure the Rx block to perform label comparisons and gap/parity checks on the incoming data. The default demonstration set-up contains four Rx channels, configured to run at a data rate of 100 kbits/s.

ARINC 429 Transmitter Interface

Each Tx block is responsible for transmitting ARINC data out of the internal FIFO as soon as a complete ARINC word is written to the FIFO by the CPU Interface. This block can be configured to operate in loopback mode so that transmitted ARINC data does not leave the FPGA, but instead is received by the corresponding Rx block. The default demonstration configuration contains four Tx channels that run at a data rate of 100 kbits/s and have loopback disabled (messages are sent through the connectors, not internally).

Core ARINC 429 CPU Interface

The CPU interface allows the system CPU to access the Rx and Tx blocks. Access is necessary to configure control registers of both the Rx and Tx blocks, monitor the status registers, write to Rx label memory, and read/write ARINC data.

Core8051 CPU

The Core8051 is an 8-bit microprocessor used to control the ARINC system for the demonstration design. It interfaces with the serial interface via the Special Function Register Bus. For a more detailed description of Core8051, refer to the *Core8051 Datasheet*.

Core8051 SFR Interface

The Core8051 SFR Interface provides access to Core8051 memory-mapped special function registers that reside either internally or externally. The demonstration design implements external SFRs to communicate between the system CPU (Core8051) and the Core429. The SFR Interface is controlled by Core8051 and communicates with the Core429 CPU interface to address the external SFRs and provide the data necessary for the Core429 system in the demonstration design to operate.

Core8051 Serial Interface

The Core8051 serial interface allows Core8051 to communicate with the off-chip RS-232 transceiver via a standard 9-pin D-type connector (J24). Core8051 serial communication utilizes Core8051's internal serial channel and Interrupt Services.

Demonstration Configuration

On power-up, default configuration information is written to the control register of each Rx and Tx block.

The default Rx block configuration is as follows:

- Data Rate = 100 kbits/s
- Label Recognition is enabled
- Parity is disabled
- Decoder is disabled
- Bit Matching is disabled
- The Rx block label memory is initialized with nine active labels to display nine Platform8051 ADC values.

The default Tx block configuration is as follows:

- Data Rate = 100 kbits/s
- Loopback is disabled
- Parity is disabled

The Rx and Tx configuration can be changed by using the terminal interface to write the appropriate values to the Rx and Tx control registers. Refer to the *Core429 Datasheet* for the appropriate register values for each available configuration.

Demonstration Design Operation

User Interfaces

Upon power-up, the demonstration design enters the Automatic Tx/Rx mode, in which data from the Platform8051 Development Board Analog-to-Digital Converter (ADC) is transmitted on all four Tx channels. Connecting two of the channels together using the supplied DB9 cables enables the ARINC 429 Rx blocks to receive the transmitted data. There are two interfaces that allow the user to examine the received data:

- The on-board keypad and LCD (limited user interaction)
- The serial interface via the RS-232 port and HyperTerminal

Keypad and LCD Screen

The on-board keypad (SW3 to SW7) allows the user to view the data received by the ARINC 429 receiver. The demonstration design will, by default, display the Core429 Daughter Board Knob voltage level. This level is intended to represent a typical airplane system such as fuel level or flap position. The keypad can be used to cycle through the other active ARINC labels on each ARINC channel:

- UP (SW3) and DOWN (SW7) will cycle through the nine active default labels.
- LEFT (SW4) and RIGHT (SW6) will cycle through the four default channels.
- MIDDLE (SW6) will return the LCD to the default Volume Knob readout.

HyperTerminal and RS-232

Setting up HyperTerminal

While disconnected, HyperTerminal should be configured (**File** > **Properties** > **Configure**) to interface with the serial port of the demonstration board:

- 2,400 bits per second
- 8 data bits
- Parity set to none
- 1 stop bit
- Flow control set to none

Commands

Table 1-1 lists the available commands for controlling the demonstration design via HyperTerminal.

Table 1-1. Core429 Terminal Interface Commands

Terminal Commands	Function	Format
GLOBAL COMMANDS		
auto	Return to auto Tx/Rx mode	–
ch	Set current channel	ch <X> (0–3)
esc	Exit auto Tx/Rx mode and enter command mode (whether or not loopback is enabled)	–
intlbk	Forces all channels into internal loopback and enters auto Tx/Rx mode with data looped back	–
version	Displays version of terminal interface	–
CHANNEL COMMANDS		
adlbl	Add a label value to Rx label memory	adlbl <01 [02][03] ... [FF]> (max 80 characters per line)
config	Current channel's Rx and Tx settings	–
conrx	Read current channel's Rx control register	conrx
	Write to Rx control register	conrx <XX>

Table 1-1. Core429 Terminal Interface Commands (Continued)

Terminal Commands	Function	Format
contx	Read current channel's Tx control register	contx
	Write to Tx control register	contx <XX>
label	Reset label memory and set new labels. Predefined labels are shown in Table 1-3 on page 18.	label <01 [02][03] ... [FF]> (max 80 characters per line)
loop	Enable/disable internal loopback mode for current channel	loop <X> (0 – Disable, 1 – Enable)
rate	To alter the rate of data transfer between consecutive ARINC words transmitted from the terminal interface to Core429.	rate <N> (0–1023 ms approximate time)
reg	Write data to a register on current channel	reg <N> <data> (N is 9-bit CPU address in hex)
	Read data from a register	reg <N> (N is 9-bit CPU address in hex)
rx	Empty current channel's Rx FIFO to screen	–
statrx	Read current channel's Rx status register	–
stattx	Read current channel's Tx status register	–
tx	Transmit data on current channel	tx <XXXXXXXX> (8 hex characters max)

Note: The terminal interface specifies all ARINC data and labels as hexadecimal values.

Examples

Table 1-2 gives examples of terminal interface commands.

Table 1-2. Core429 Terminal Interface Command Examples

Keystrokes	Description
>ch 0 [return]	Select channel 0.
>reg 000 [return]	Read Rx register 0 on channel 0 (see CPU address bit positions in Table 10 of the *Core429 Bus Interface datasheet*).
>reg 064 0 [return]	Write 0 to Rx register 1 on channel 3 (see CPU address bit positions in Table 10 of the *Core429 Bus Interface datasheet*).
>tx ABCDEF12 [return]	Transmits ABCDEF12 from Core429 transmitter on current channel. Same effect as the following sequence (assume ch0): >reg 010 ABCDEF12 [return]
>rx [return]	Reads Rx FIFO on current channel until the FIFO is empty. Same effect as repeatedly typing (while mode = 0): >reg 000 [return]
>label 12 45 4 9 [return]	Resets channel 3's Rx label memory and writes hexadecimal values 12, 45, 4, and 9. Same effect as the following: >reg 064 80 [return] (to tell the Rx block to reset label memory) >reg 06C 12 [return] >reg 06C 45 [return] >reg 06C 4 [return] >reg 06C 9 [return]
>adlbl 5 8 7 1 3 [return]	Adds 5, 8, 7, 1, and 3 to existing values in Rx label memory without resetting the label memory.
>contx 02 [return]	Writes 2 (00000010) to the current channel's Tx control register (loopback mode enabled).

Note: A full description of the Core429 Rx and Tx registers can be found in the *Core429 Datasheet*.

The Core429 terminal interface is initialized in auto Tx/Rx mode. Pressing the **Esc** key puts the terminal interface into Command Mode. The Command Mode user interface is initialized with the following defaults:

- Loop is 0, internal loopback mode is disabled, data transmissions go off-chip to the Core429 Line Drivers on the Core429 Daughter Board
- The channel must be selected before channel-specific commands will take effect.

Furthermore, each time the ch <X> command is issued, channel X is restored to the default Rx and Tx block configurations, described in "Demonstration Configuration" on page 12, with the exception that label recognition is disabled in the Rx block. Label recognition can be enabled by writing 00000010 (bit 1 is high) to the Rx Control Register. For example, the following command would enable label recognition on the current channel's Rx block:

```
>conrx 2[return]
```

Platform8051 Development Board

The Platform8051 Development Board is intended as a demonstration and evaluation medium for a variety of Actel intellectual property, including Core8051, Core10/100, CoreSPI, CoreI2C, CoreSDLC, Core16X50, CoreUART, and the Utopia family cores. With the daughter card capability, it is a versatile board that can demonstrate a variety of other cores, including Core429.

The Platform8051 board contains the following items:

- ProASIC^{PLUS} APA600-FG676
- 4 MB SRAM memory
- 32 MB Flash memory
- 10/100 Ethernet physical layer device (PHY)
- RS-232 serial interface
- LCD display
- 10-bit ADC

All subsystems can be addressed by the APA600. In addition, there are several unpopulated component footprints on the Platform8051 board including locations for a second Ethernet PHY, a 1553B Transceiver, and USB circuitry.

The Platform8051 Development Board contained in the Core429 Development Kit is delivered pre-programmed with the Core429 Design in the FPGA and the Core429 demonstration design in the on-board Flash Memory.

The Platform8051 Development Kit is delivered with a simple web server demonstration project running over the on-board Ethernet connection. See *Platform8051 Development Kit User's Guide* for additional details.

On-Board ADC

The ADC on the development board monitors voltage rails, temperature, FPGA power consumption, and the Core429 Daughter Board Volume Input. In auto mode, the Core429 demonstration transmits scaled ADC data with the labels shown in Table 1-3.

Table 1-3. ADC Data and Corresponding ARINC Labels

ADC Channel	Label	Description	ADC Label Abbreviations
0	1	2.5 V regulator output (in mV)*	RO2
1	2	2.5 V input to FPGA (in mV)	VIN
2	3	3.3 V regulator output (in mV)	RO3
3	4	Input voltage from PSU (in mV)	PSU
4	5	5.0 V regulator output (in mV)	RO5
5	6	Daughter board volume input (in mV)	VOL
6	Not assigned	Unused	N/A
7	7	Temperature °C	TMP
8	8	FPGA core power consumption (in mW)	PWR

Note: The voltage difference between Channel 0 and 1 indicates the current being taken by the FPGA.

Core429 Daughter Board

The Core429 Daughter Board contains the external line drivers and receivers needed to transmit and receive ARINC 429 data on an ARINC 429 data bus. The daughter board has four male DB9 connectors, one for each channel in the Core429 demonstration design. Refer to Figure 1-3 on page 19 for Core429 channel assignment. In addition, the daughter board has a volume control that is monitored as an input to the Core8051 ADC. The Core429 Daughter Board connects to the Platform8051 Development Board via the 90-pin connector J23.

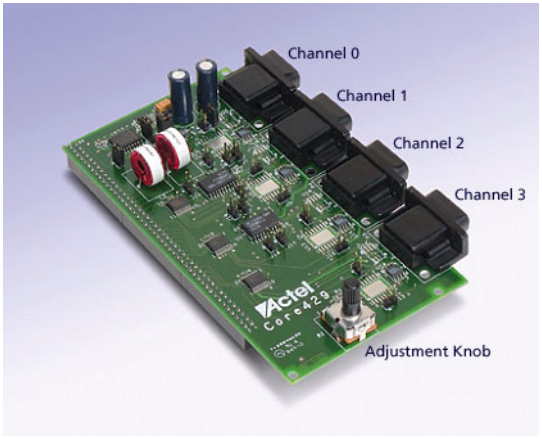

Figure 1-3. Core429 Daughter Board

2

Running the Demonstration

Preparing to Run the Demonstration

The demonstration can be run from the Platform8051 Development Kit by configuring it as a Core429 Development Kit.

To configure the Platform8051 Development Kit as a Core429 Development Kit:

1. **Connect the Core429 Daughter Board to the 90-pin connector (J23) on the Platform8051 Board.**

   Note: Ensure that pin 1 of the daughter board connects to pin 1 of the Platform8051 connector (J23).

2. **Program the FPGA with the Core429 Design (see "Programming the FPGA" on page 27).**
3. **Program the Flash memory with the demonstration design (see "Programming the Flash Memory" on page 29).**
4. **Continue to "Setting Up the Core429 Development Kit".**

Setting Up the Core429 Development Kit

The Core429 Development Kit comes with the Platform8051 Development Board pre-programmed with the Core429 Hardware Design and the Core429 terminal interface. The user has the option of running a minimal demonstration by using only the on-board keypad and LCD. The user can run the full demonstration by using the terminal interface via serial RS-232 connection.

To set up the Core429 Development Kit:

1. **Connect the supplied demonstration cable with female DB9 connectors to any two of the four male DB9 connectors on the Core429 Daughter Board.**

   Note: Only the two channels connected by the cable will receive ARINC 429 data.

2. **Ensure that the DIP switch located at S1 has all switches closed, corresponding to a value of 00000000.**

   Note: These switches are wired to the inputs of the FPGA so that *open* will correspond to a logic **1** (4.6 k pull-up to 3.3 V) and *closed* will correspond to logic **0** (GND).

3. **If running the demonstration via the terminal interface, connect the serial cable between the RS-232 connector (J24) on the board and the serial port on the PC.**
4. **Ensure that the power switch at SW1 is set to *OFF* and connect the power supply to the mains and to the development board. Power up the board using SW1.**

Note: After a few seconds, the LCD display will show the following:

```
C0:LVOL= XXXX
```

The output is formatted to display

```
C<channel #>:L<label name>= <value>.
```

If the LCD displays

```
C0:= NO DATA
```

then use the LEFT/RIGHT switches to select the channel corresponding to the channel connected via the external cable.

Alternatively, the user can ensure that there is data received on channel 0 by connecting the supplied cable to the channel 0 DB9 connector on the Core429 Daughter Board.

The XXXX corresponds to the ADC reading (in mV) for the Core429 Daughter Board Volume Input. When turning the volume knob, the user should see the LCD display updated every few seconds. The demonstration is transmitting data between the ARINC Tx and Rx blocks via the external cable.

Demonstration with Keypad and LCD only

Once initialized, the LCD will, by default, continuously show updates of the Volume Knob input monitored by the ADC. The user can then use the UP/DOWN switches to cycle through the active ARINC 429 Rx labels. The user can cycle through the four channels by using the LEFT/RIGHT switches as described in "Demonstration Design Operation" on page 13.

To run the demonstration with the terminal interface via RS-232:

1. **Connect to the demonstration design using HyperTerminal and ensure that it is set to a 2400-baud rate with 8 data bits, no parity, 1 stop bit, and no flow control.** See "HyperTerminal and RS-232" on page 13 for set-up details.
2. **Press RESET (SW2) and you should see the following message:**

```
------------------------------------------------------------
Welcome to the monitor program interface
© Actel 2004
------------------------------------------------------------
Initializing...
```

After a brief initialization period, the LCD will display the Volume Knob voltage level, and the HyperTerminal will echo the LCD content. The active labels correspond to the ADC values discussed in "On-Board ADC" on page 18.

3. **Press *Esc* to enter the command mode of the user interface.** Notice the following display:

```
ESC key pressed.
Command mode: Enter Command

>
```

4. **Select one of the channels connected to the daughter board cable by using the *ch* command.**
5. **Type the following, where <*data*> is replaced with an eight-digit hexadecimal value:**

```
tx <data> [return]
```

6. **Use the *ch* command to change to the other channel connected to the daughter board cable.**
7. **Type *rx* to empty the current channel's Rx FIFO and compare the transmitted and received data.**

The terminal interface can be used to configure the various modes of operation available in Core429. Refer to "Demonstration Design Operation" on page 13 and the *Core429 Datasheet* for more information about exercising the various configurations of the Core429 and the terminal interface.

3 Modifying the Demonstration

Operating in Internal Loopback Mode via the Terminal Interface

In this section the demonstration design will be run while the Core429 is operating in internal loopback mode on a per channel basis. To run the auto Tx/Rx mode with data looped back see "Operation on a Platform8051 Board" on page 31. The daughter board and cable are not required for this mode of operation since the transmitted data does not leave the FPGA, but is looped to the Rx block on the same channel as the transmitter.

Note: Ensure the board is powered off and SW1 is in the **OFF** position during connection or removal of the FlashPro Lite Programmer, the power supply connector, or the Core429 Daughter Board.

To run the demonstration in internal loopback mode (on a per channel basis):

1. **Unplug the cable connecting the daughter board channels together.**
2. **Disconnect the Core429 daughter board (optional).**
3. **Connect the serial cable between the RS-232 Connector (J24) and the PC.**
4. **Connect to the demonstration design using HyperTerminal.** See "HyperTerminal and RS-232" on page 13 for setup details.
5. **Ensure that all eight dipswitches (S1) are closed.**
6. **Power-up the Platform8051 Development Board.**
7. **Press the *Esc* key to enter command mode.**
8. **Select a channel using the *ch* command.**
9. **Enable loopback on that channel by typing**

   ```
   loop 1 [return]
   ```

10. **Transmit 32-bit ARINC data by typing**

    ```
    tx <data> [return]
    ```

    where <data> represents eight hexadecimal values to comprise a 32-bit ARINC data word.

11. **Empty the Rx FIFO by typing**

    ```
    rx [return]
    ```

    and compare the received data with the transmitted data.

Note: Issuing another **ch** command will disable loopback, and restore the default channel configuration described in "Demonstration Configuration" on page 12. To run the entire demonstration in loopback mode, including the auto Tx/Rx mode, see "Operation on a Platform8051 Board" on page 31.

A

Programming the FPGA

Requirements

- PC with at least one parallel port
- FlashPro Lite
- Parallel port printer cable

Prepare the PC

To prepare the pc:

1. **Install the Actel FlashPro v3.2 programming software.**
2. **Install the FS2® Debugger.**
3. **Place the Core429 Development Kit CD in the CDROM drive.**

Program the FPGA

To program the FPGA:

1. **Connect the FlashPro Lite to the Platform8051 Development Board (J3) and to the PC parallel port.**
2. **Connect the Platform8051 Board to the power supply and slide SW1 to the *ON* position.**
3. **Start the Actel FlashPro programming software on the PC.**
4. **From the *File* menu, select *Connect*.**
5. **In the pop-up window titled *FlashPro: Connect to Programmer*, ensure that the *Configuration* is set to *ProASIC*^{PLUS}, and click *Connect*.**
6. **From the *File* menu, select *Analyze Chain*. Verify that the APA600 device is identified.**
7. **From the *File* menu, select *Open STAPL File*. From the Core429 CD, browse to <CD drive>:/core429_devkit_2.0/Core429_Demonstration/FPGA_design/bitstream/core429.stp**

   Click **Open**.
8. **In the main FlashPro window, set the *Action* field to *Program*.**
9. **Click the *Execute* button; or from the File menu, select *Execute* to begin programming.** When the FlashPro Main Window says **PASSED**, programming is complete. Exit the FlashPro software and turn off the Platform8051 Board.

B

Programming the Flash Memory

Requirements

- PC with at least one serial port
- FlashPro Lite

Prepare the PC

To prepare the pc:

1. **Install the FS2 ISA-Actel51 Debugger software.**
2. **Place the Core429 Development Kit CD in the CDROM drive.**
3. **Connect the FlashPro Lite to the Platform8051 Development Board (J3) and to the PC parallel port.**

Program the Flash Memory

To program the Flash memory:

1. **Set DIP switch (S1) so that Switch[1:8] = 00000011.**

   Note: These switches are wired to the inputs of the FPGA so that *open* will correspond to a logic **1** (4.6 k pull-up to 3.3 V) and *closed* will correspond to a logic **0** (GND).

2. **Connect the Platform8051 Board to the power supply and slide SW1 to the *ON* position.**
3. **Start the FS2 Debugger software.**
4. **From the *Tools* menu, select *Load Hex…***
5. **Set the *Address to load at* field to *0x0000x* and the *Filename* to:**

   <CD drive>:/core429_devkit_2.0/Core429_Demonstration/demo_software/demo.ihx

   Click ***OK***.

6. **Click *Go*.**

   The demonstration should commence when the LCD displays the Volume Knob Input Voltage.

7. **Change DIP switch (S1) by moving switch 1 to the *open* position, corresponding to a value of Switch[1:8] = 10000011.**
8. **Press the RESET switch (SW2) on the Platform8051 Development Board.**

9. **Wait for at least 10 seconds; then turn off the board power.** Exit the FS2 Debugger and disconnect the FlashPro Lite from the Platform8051.

   Note: The on-board Flash memory is now programmed.

10. **Set DIP switch (S1) to Switch[1:8] = 00000000, corresponding to all switches *closed*.**

C

Operation on a Platform8051 Board

The Core429 demonstration design can be run without the Core429 physical interface daughter board. This mode of operation requires that the Platform8051 Development Board be programmed with the Core429 Hardware Design (see "Programming the FPGA" on page 27) and the terminal interface in the flash memory (see "Programming the Flash Memory" on page 29).

In this mode of operation, ARINC 429 data does not leave the Actel FPGA because the demonstration configures all Core429 Tx blocks to function in internal loopback mode. This demonstration can also be run using the keypad and LCD screen, or with HyperTerminal via RS-232. However, the terminal interface is initially required to configure the demonstration for internal loopback operation.

Running the Demonstration

To run the demonstration on a Platform8051 board:

1. **After following the steps in "Programming the FPGA" on page 27 and "Programming the Flash Memory" on page 29, connect the Platform8051 Board to the PC with the null modem cable.**
2. **Connect to the demonstration design using HyperTerminal and ensure that it is set to a 2400-baud rate with 8 data bits, no parity, 1 stop bit, and no flow control.** See "HyperTerminal and RS-232" on page 13 for set-up details.
3. **Power-up the board.**
4. **Type *Esc* to enter Command Mode.**
5. **Type *intlbk* and press *Enter*.**
6. **The demonstration will commence in internal loopback mode and the keypad and LCD screen will function as described in "Demonstration Design Operation" on page 13.** The terminal interface will also function as described previously.

D

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*. • Dunlop House, Riverside Way • Camberley, Surrey GU15 3YL • United Kingdom*
Phone +44 (0) 1276 401 450 • Fax +44 (0) 1276 401 490

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

Actel Hong Kong *• 39th Floor, One Pacific Place • 88 Queensway, Admiralty Hong Kong*
Phone +852.227.35712 Fax +852.227.35999

50200042-0