EB-ProASIC PLUS Evaluation Board

User Manual

Version 1.0
April 3, 2002

First Silicon Solutions, Inc.
5440 SW Westgate Dr. Suite 240
Portland, OR 97221
voice: +1-503-292-6730
fax: +1-503-292-5840
sales@fs2.com
support@fs2.com

1. Introduction

This document presents the design and usage information for the Actel EB-ProASIC^{PLUS} Evaluation Board. The board is intended to demonstrate the ability to program a ProASIC^{PLUS} device from a microprocessor.

2. Overview

The evaluation board is based on a microcontroller with a simple on-board user interface consisting of a single-line LCD and a few buttons. On-board flash memory stores one or more programming files. A device is programmed by selecting the desired file and executing the program function.

The ProASIC^{PLUS} devices are programmed via JTAG. The microcontroller drives the JTAG signals according to the programming file through a simple API. The JTAG signals are connected to general-purpose I/O pins on the microcontroller and controlled by software.

The evaluation board provides a site to accommodate either a socket or a ProASIC^{PLUS} device.

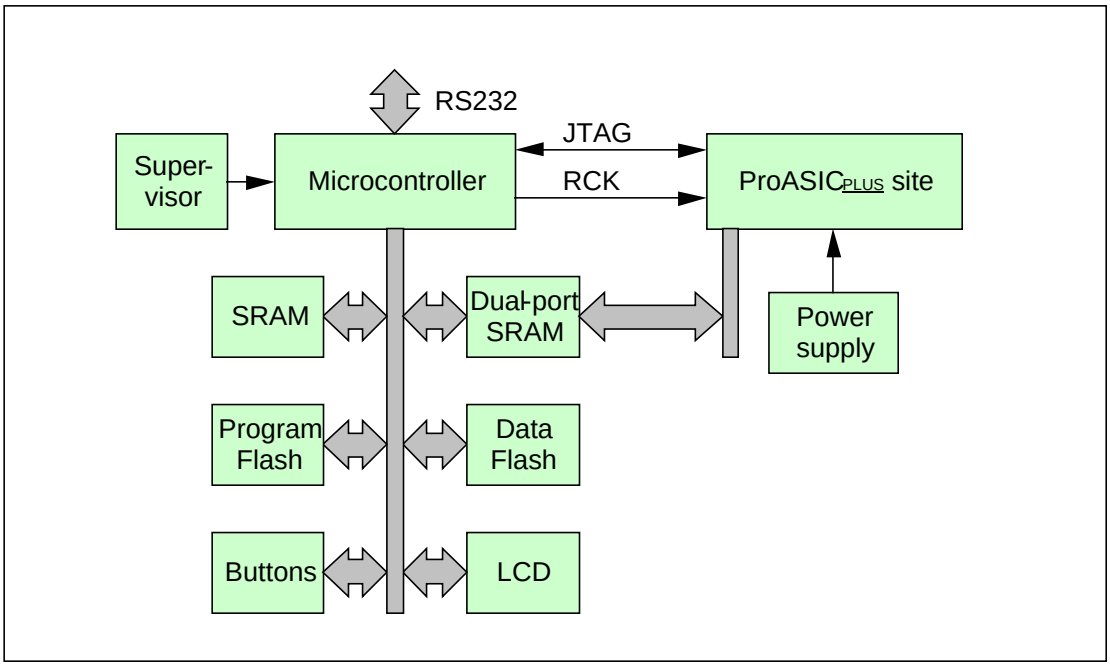

Figure 1. ProASIC^{PLUS} Evaluation Board

3. Design Details

3.1. Mechanical Design

The evaluation board is rectangular with rubber pads affixed to the bottom. It is not enclosed in a case. The board size is 5 by 8 inches.

3.2. Microcontroller

The CPU is a Zilog eZ80190. This is a single-chip integrated controller with a 16 megabyte address space, 8K on-chip SRAM, glueless interface to memory and I/O, serial ports, timers, and general-purpose I/O as well as a debug interface which will be used for development and also to download the software monitor during manufacturing.

JTAG signals are directly connected to CPU general-purpose I/O pins and toggled by software. RCK is a free-running 4 MHz clock that is generated by a dedicated oscillator that also connects to one of the FPGA's global clock inputs.

3.3. Flash

Separate flash devices on the evaluation board are used for program and data storage. Separate devices allow the CPU to execute from one device while programming the other and also provide additional security for the program memory so it is less likely to get accidentally erased.

The eZ80 executes the software monitor from the program flash starting at reset. Program flash is protected and requires a jumper to be installed in order to erase or program the device.

Programming files downloaded from a PC are stored in the data flash using a simple sector-based file system. The data flash is allocated to programming files in 64KB sectors. Each file may occupy any number of sectors and the sectors that make up a file need not be contiguous. One sector is reserved for the directory. A 2 megabyte JEDEC-compatible device is installed but the site can accommodate devices from 1 to 8 megabytes. A 1 megabyte flash contains 15 data sectors while an 8 megabyte device contains 127 data sectors.

Both the program flash and the data flash are in 48-pin TSOP packages.

3.4. RS232

The eZ80 CPU includes an integrated UART. An interface chip in the Maxim MAX232 family is used to translate the TTL signals to RS232 levels. A DB-9 connector is provided so the board may be hooked up to a PC with a 9-pin serial cable. A crossover cable is required and is provided with the board.

3.5. SRAM

Four SRAM devices are populated for a total memory storage of 2 megabytes. The SRAMs are in 44-pin TSOP packages.

3.6. Dual-Port SRAM

A small (8K byte) dual-ported SRAM is installed on the board and may be addressed from either the eZ80 CPU or the FPGA. After programming the FPGA, this SRAM is used to pass messages to and from the FPGA using the buttons and LCD display. The dual-port SRAM is in a 64-pin TQFP package.

3.7. LCD

An integrated high-contrast liquid-crystal character display will be used to display messages to the user allowing selection of a programming file and initiation of the programming sequence.

The LCD is 16 characters by 2 lines. The content is determined by firmware and is shown in the document section describing the software monitor.

3.8. Buttons

A set of four buttons is provided as the user interface to the software monitor. The software monitor is menu-based. The buttons are labeled ">", "<", "Enter", and "Cancel". The user interface is described in the document section describing the software monitor.

The buttons are directly read through general-purpose I/O signals. Each button is an SPST switch connecting the I/O signal to ground. A pullup resistor causes the signal to be read high when the switch is not depressed. Debouncing is done in software.

3.9. ProASIC^{PLUS} Device

One site is provided for a ProASIC^{PLUS} device. The site is a BGA that can be populated by either a BGA socket or a ProASIC^{PLUS} chip. A number of unassigned I/O pins of the device are made available at 0.1" double-row header sites.

A 38-pin Mictor connector is directly connected to the ProASIC^{PLUS} to be used by a logic analyzer or the FS2 Configurable Logic Analysis Monitor (CLAM) instantiated in the device.

Finally, a ProASIC^{PLUS} external programming header is provided to support programming the device from an external programmer.

3.10. Power Supplies

Power enters the board from a +24V DC universal desktop power supply. The supply voltages are created with a combination of DC/DC converters and linear regulators.

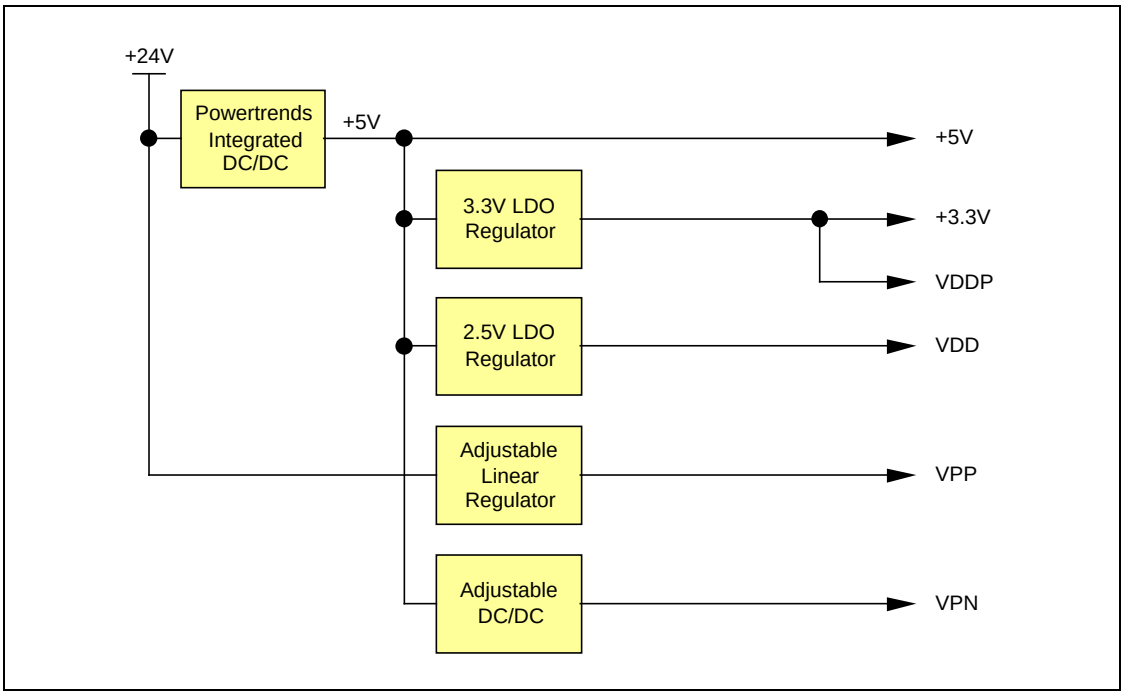

Figure 2. Power Supply Tree

There are four power supplies required for the ProASIC^{PLUS} and two for other board circuitry. All of these are fixed supply voltages as follows:

Supply	Purpose	Voltage
VDDP	ProASIC^{PLUS} I/O supply	+3.3V
VDD	ProASIC^{PLUS} logic supply	+2.5V
VPP	ProASIC^{PLUS} programming voltage	+16.2V ± 0.3V
VPN	ProASIC^{PLUS} programming voltage	-13.6V ± 0.2V
+5V	Board logic supply	+5V
+3.3V	Board logic supply	+3.3V

3.11. Digital Signals

Digital signals between the CPU and ProASICPLUS are all +3.3V level.

4. Software Overview

There are 4 major software levels--a Windows-based downloader GUI, the on-board user-interface, the on-board STAPL player, and the low-level board control or ABI.

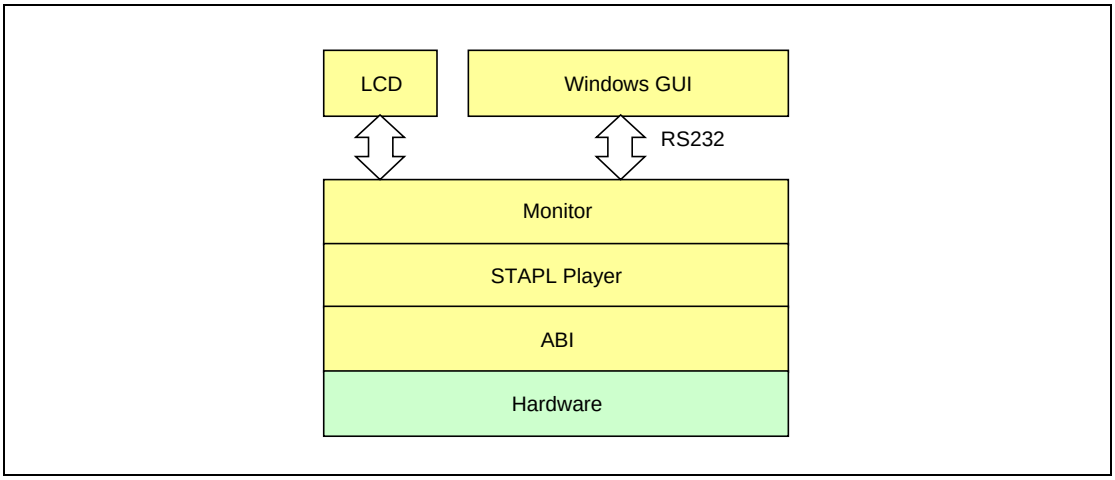

Figure 3. Software Structure

5. Low-Level Control Software ABI

Low-level control software is a collection of C functions described in ebactel.h.

5.1. Initialization

AbiInit initializes the ABI layer. It is called when programming begins. AbiCleanup is called when programming completes.

5.2. JTAG Communication

There are several primitive functions available to allow the application to fully control the JTAG port. These functions execute by toggling TCK, TDI, and TMS in a sequence that executes the desired operation. TDO is sampled and is available for reading if desired.

6. Software Monitor

On reset, firmware in the program flash is executed. After initialization to set up chip selects, waitstates, etc, the monitor displays the menu on the LCD and sends a message to the RS232 port.

The monitor may be controlled from either the on-board buttons or via the RS232 port from a remote PC. The on-board buttons and LCD display together form a simple user interface with a hierarchical menu.

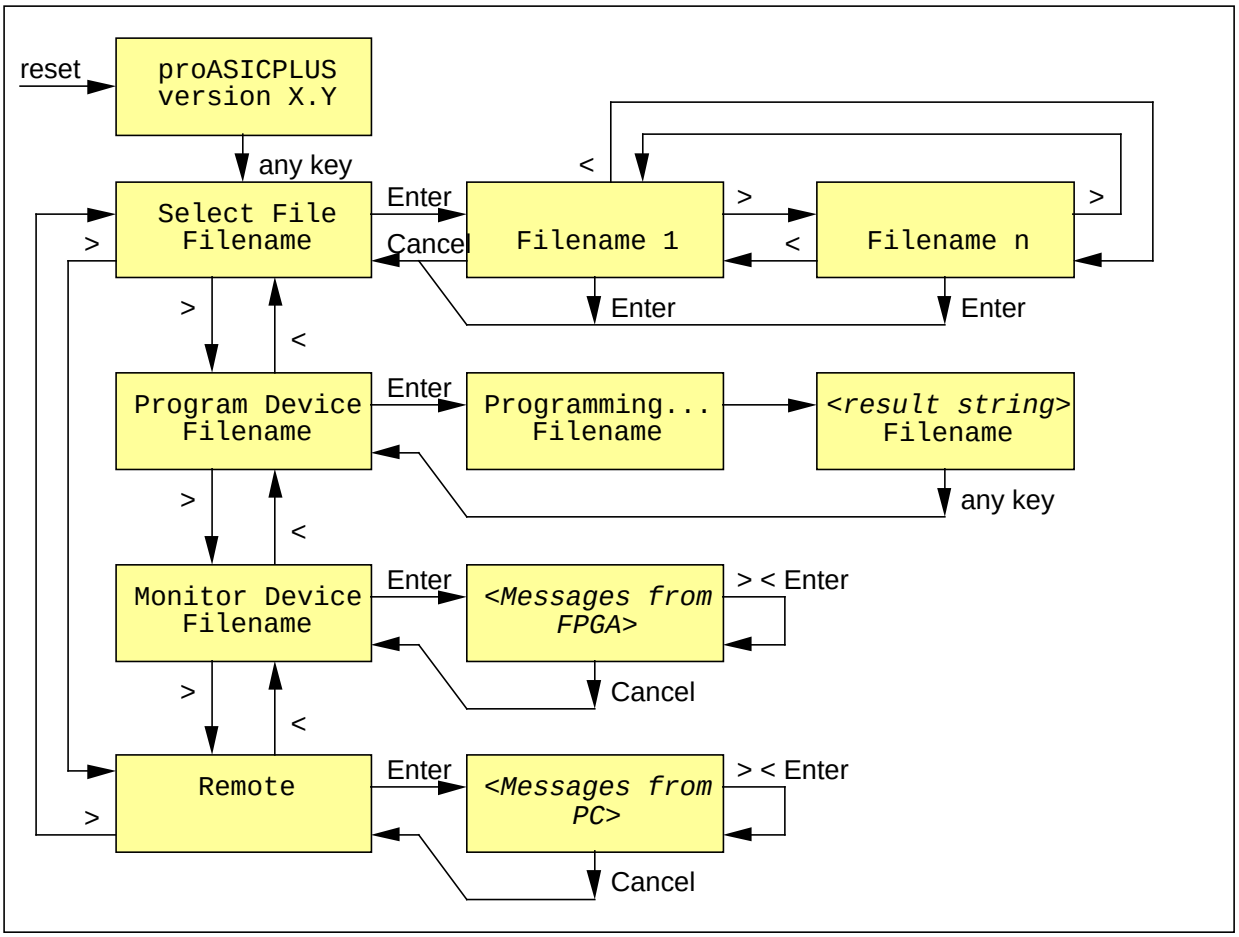

Figure 4. Menu Structure

To program a device, first one or more files are downloaded from a PC (see next section). The Select File menu is used to cycle through the filenames until the desired one is found. Press Enter to return to the main menu. Use the arrow keys to cycle to Program Device, then Enter to begin the programming process. When programming is complete, a message is displayed until any key is pressed.

To allow interaction with the FPGA after programming, select Monitor Device. In this state, the CPU continually updates the display by reading 32 bytes from the dual-ported memory and writing those bytes to the 32 characters in the display. Line 1 of the display is offset 0x00..0x0F and line 2 is offset 0x10..0x1F in the dual-port RAM. During programming, these memory regions are filled with blanks which will be displayed if the programmed FPGA image does not initialize the memory region.

In FPGA Monitor mode, dual-port memory location 0x20 bits 0, 1, and 2 reflect the states of the Enter, >, and < keys, respectively. 1 indicates the key is depressed. The key states are debounced in the CPU. This feature provides basic user interaction with the programmed FPGA image.

The remaining space in the dual-port RAM may be used by the FPGA. Host software can read and write locations in this memory so two-way communication to/from the PC can be accomplished if desired.

7. RS232 Communication Protocol

The on-board monitor can operate on files once they are in the data flash. The files are downloaded from the PC to the data flash via RS232. The RS232 port is fixed at 38400 baud and uses a line-based command/response protocol. Each command and response line ends with a newline.

Command	Response	Action
F	index name length >	Select next file. Response is new selected file.
Del	index name length >	Delete file. Response is new selected file.
Din	>	Initialize data flash (erase all files).
C <length>	index name length >	Create new file of length <length> bytes and name "newNNN" where NNN is a 3-digit decimal number. Response is new selected file.
N <filename>	index name length >	Rename file. Response is selected file.
S…	>	Load S record into download buffer. The S record is checked for parity.
K	index name length >	Write download buffer to selected file.
Xe	>	Erase program flash.
Xp	>	Write download buffer to program flash.
I	>	Initialize download buffer to zero.
P	index name length >	Program FPGA from current file.
W <addr> <data>	>	Write data to addr in eZ80 memory space.
R <addr>	data >	Read addr in eZ80 memory space.
Z	>	Display 32 bytes at 0x7FFFE0 on LCD.
V	product name version number >	Retrieve firmware version information.
JA	>	AbiInit

JZ	>	AbiCleanup
JR	>	AbiJtagReset
JI	>	AbiJtagIrEnter
JD	>	AbiJtagDrEnter
JT	>	AbiJtagIdle
JW <clocks>	>	AbiWait (clocks)
JS <data>	tdodata >	AbiJtagShift (data, 8) AbiJtagRead (tdodata, 8)

8. Windows Host Software

On the Windows PC, a simple program allows a user to download files and otherwise control the evaluation board via RS232. The program is a command window based on Tcl/tk and is fully extensible using Tcl/tk scripts. The following table lists the available commands apart from the built-in Tcl commands.

Command	Syntax	Action
help	help	List commands and syntax.
openport	openport <port>	Open communication port. <port> must be **com1:** to **com9:**.
firmware	firmware	Display firmware version information
version	version	Display software version information
download	download <filename> [<remote filename>]	Copy file from PC to data flash, then select it. If not specified, the remote filename is the same as the PC filename. The remote filename is limited to 12 characters and must contain only letters, numbers, underscore, and period.
nextfile	nextfile	Select next file. Display new file's index, name, and length (if any).
deletefile	deletefile	Delete selected file. Display new file's index, name, and length (if any).
renamefile	renamefile <filename>	Rename selected file to <filename>. Display file's index, name, and length.
programfpga	programfpga	Program FPGA from selected file.
monitorfpga	monitorfpga	Shows a popup window continuously updated with the message from the FPGA, read from dual-port RAM locations 0x00-0x1F.
initdataflash	initdataflash	Initialize data flash (erase all files).
updateprogramflash	updateprogramflash [<filename>]	Download and reprogram firmware image in the board. Filename defaults to ebfw.bin.
LCD	LCD <string>	Display <string> on LCD.

The installer sets up an icon with the following command line:

```
c:\Program Files\fs2\ebactel\bin\cliebactel.exe ebstartup.tcl
```

ebstartup.tcl is a 2-line script that sets the default communication port, then calls the main initialization script initebactel.tcl.

```
set defaultport "com1:"
source initebactel.tcl
```

If you attach to the unit via a port other than com1:, you can change the port in ebstartup.tcl to avoid having to issue an openport command every time you start the software.

If a user.tcl exists in the startup directory, it is executed after the general initialization script has completed. This file could be used to install a different command set, create alternate display windows, etc.