ProASIC3/E FlashROM (FROM)

Introduction

The ProASIC3/E families of Flash FPGAs offer enhanced performance, density, and features over ProASIC[PLUS]® FPGAs. The ProASIC3/E devices also provide a secure, low-power, live-at-power-up, single-chip solution. The nonvolatile, Flash-based ProASIC3/E families require no boot PROM. ProASIC3/E devices incorporate FlashLock™ technology, which provides a unique combination of reprogrammability and design security without external overhead. Only Flash FPGAs can offer these advantages. The ProASIC3/E devices have a dedicated nonvolatile FlashROM (FROM) memory of 1,024 bits, which provides a unique feature in the FPGA market. The FROM can be read, modified, and written using the JTAG (or UJTAG) interface; however, it can be read but not modified from the FPGA core. The FROM is physically organized as 8x128 bits and logically organized as 8 pages of 16 bytes. Only Flash FPGAs contain on-chip user-nonvolatile memory (NVM), and Actel's ProASIC3/E families are the only FPGAs to support this feature. Figure 1 shows the ProASIC3E device architecture.

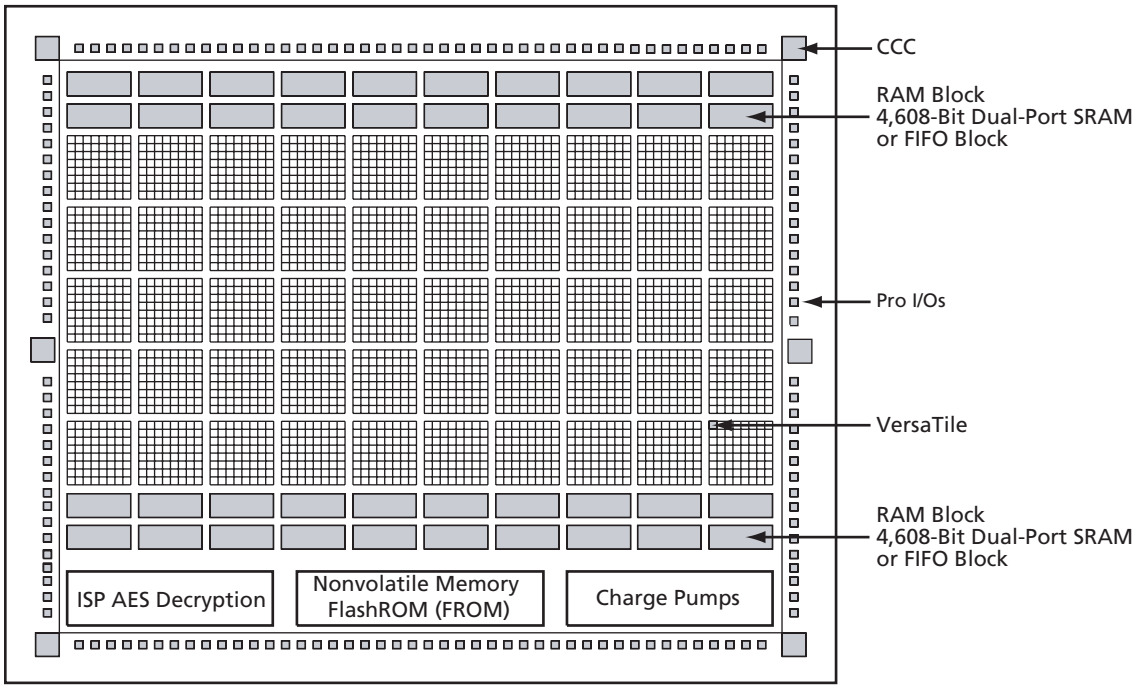

Figure 1 • **ProASIC3E Device Architecture**

FROM Architecture and Applications

Actel ProASIC3/E devices have 1 kbit of user accessible, nonvolatile FROM on-chip. The FROM is logically organized as 8 pages of 16 bytes (128 bits per page). Figure 2 shows the FROM logical level structure. The ACTgen core generator is used to configure FROM content. You can configure each page independently. ACTgen enables you to create and modify regions within a page; these regions can be from 1 to 16 bytes long.

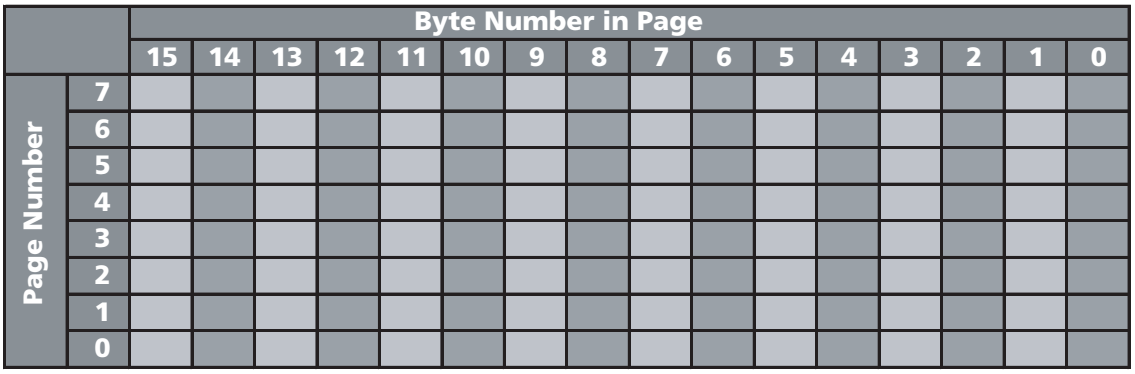

Figure 2 • **FROM Configuration**

The FROM content may be changed independently of the FPGA core content. It may be easily accessed and programmed via JTAG, depending on the security settings of the device. The ACTgen core generator enables each region to be independently updated (described in the "Programming and Accessing FROM" section on page 4). This enables you to change the FROM content on a per-part basis while keeping some regions "constant" for all parts. These features allow the FROM to be used in diverse system applications. Consider the following possible uses of FROM:

- Internet protocol (IP) addressing (wireless or fixed)
- System-calibration settings
- Restoring configuration after unpredictable system power down
- Device serialization and/or inventory control
- Subscription-based business models (e.g., set-top boxes)
- Secure key storage
- Asset management tracking
- Date stamping
- Version management

FROM Security

ProASIC3/E devices have an on-chip Advanced Encryption Standard (AES) decryption core, combined with an enhanced version of the Actel Flash-based lock technology (FlashLock). Together, they provide unmatched levels of security in a programmable logic device. This security applies to both the FPGA core and FROM content. ProASIC3/E devices use the 128-bit AES (Rijndael) algorithm to encrypt programming files for secure transmission to the on-chip AES decryption core. The same algorithm is then used to decrypt the programming file. This key size provides approximately 3.4×10^{38} possible 128-bit keys. A computing system that could find a DES key in a second would take approximately 149 trillion years to crack a 128-bit AES key. The 128-bit FlashLock feature in ProASIC3/E works via a FlashLock security Pass Key mechanism, where the user locks or unlocks the device with a user-defined key.

If the device is locked with certain security settings, functions such as device read, write, and erase are disabled. This unique feature helps to protect against invasive and noninvasive attacks. Without the correct Pass Key, access to the FPGA is denied. In order to gain access to the FPGA, the device first must be unlocked using the correct Pass Key. During programming of the FROM and/or the FPGA core, you can generate the security header programming file, which is used to program the AES key and/or FlashLock Pass Key. The security header programming file can also be generated independently of the FROM and FPGA core content. The FlashLock Pass Key is not stored in the FROM.

ProASIC3/E devices with AES-based security allow for secure remote field updates over public networks such as the Internet, and ensure that valuable intellectual property (IP) remains out of the hands of IP thieves. Figure 3 shows this flow diagram. See the *ProASIC3/E Security* application note for details on security features and settings in ProASIC3/E devices.

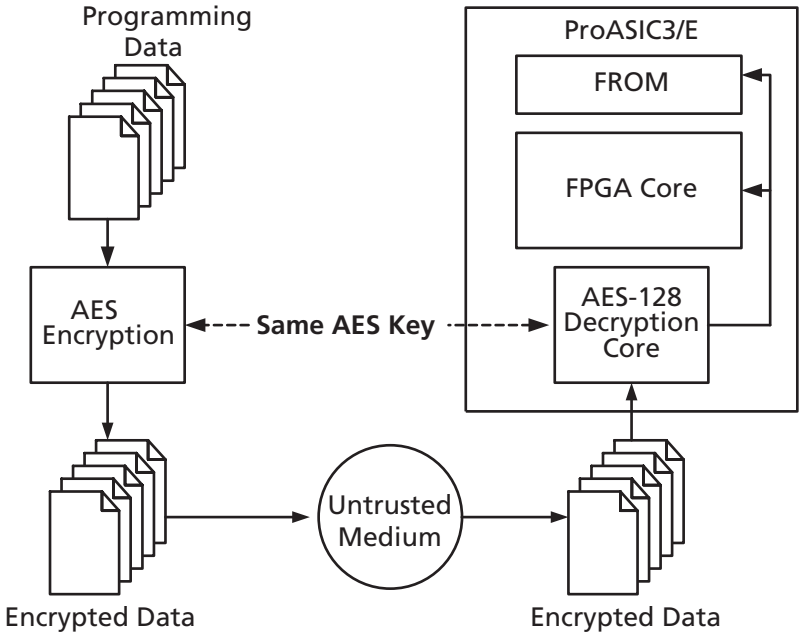

Figure 3 • **Programming FROM Using AES**

Programming and Accessing FROM

The FROM content can only be programmed via JTAG, but it can be read back selectively through the JTAG programming interface, the UJTAG interface, or via direct FPGA core addressing. The pages of the FROM can be made secure to prevent read back via JTAG. In that case, read back on these secured pages is only possible by the FPGA core fabric or via UJTAG.

A 7-bit address from the FPGA core defines which of the 8 pages (three MSBs) is being read and which of the 16 bytes within the selected page (4 LSBs) are being read. The FROM content can be read on a random basis; the access time is 10 ns for a device supporting commercial specifications. The FPGA core will be powered down during writing of the FROM content. FPGA power-down during FROM programming is managed on-chip, and FPGA core functionality is not available during programming of the FROM. Table 1 summarizes various FROM accessing scenarios.

Figure 4 shows the accessing of the FROM using the UJTAG macro. This is similar to FPGA core access, where 7-bit address defines which of the 8 pages (three MSBs) is being read and which of the 16 bytes within the selected page (four LSBs) are being read. Refer to the *UJTAG Applications in ProASIC3/E Devices* for details on using the UJTAG macro to read the FROM.

Figure 5 and Figure 6 on page 5 show the FROM access from the JTAG port. The FROM content can be read on a random basis. The three-bit address defines which page is being read or updated.

Table 1 • **FROM Read/Write Capabilities by Access Mode**

Access Mode	FROM Read	FROM Write
JTAG	Yes	Yes
UJTAG	Yes	No
FPGA core	Yes	No

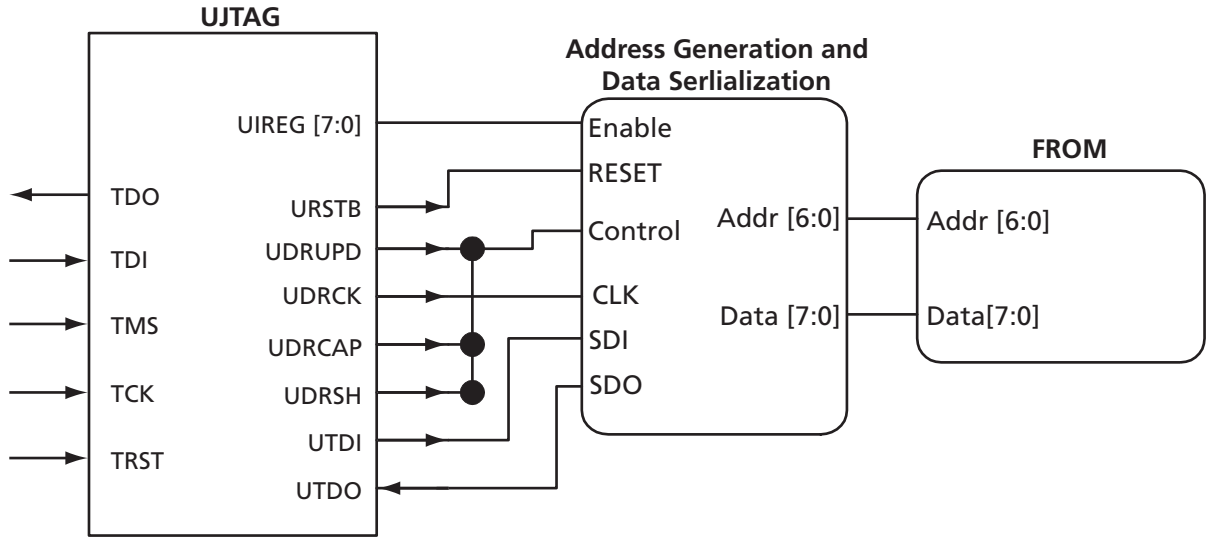

Figure 4 • **Block Diagram of Using UJTAG to Read FROM Contents**

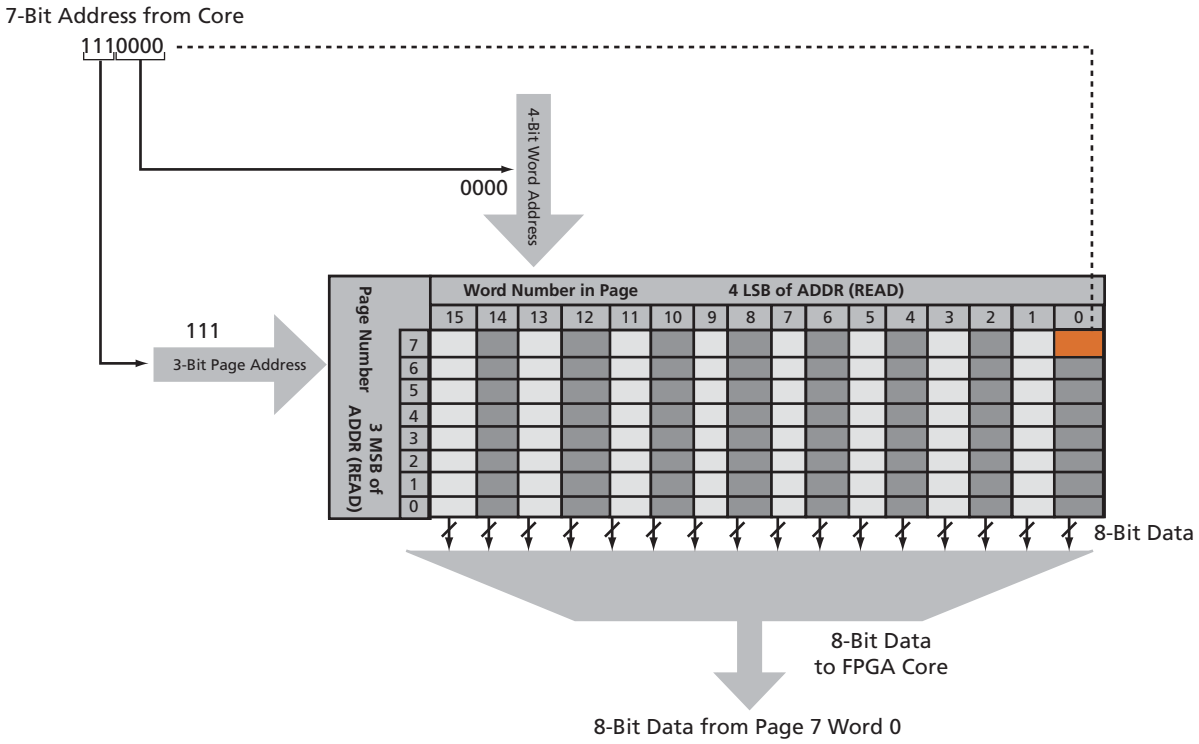

Figure 5 • **Accessing FROM Using FPGA Core**

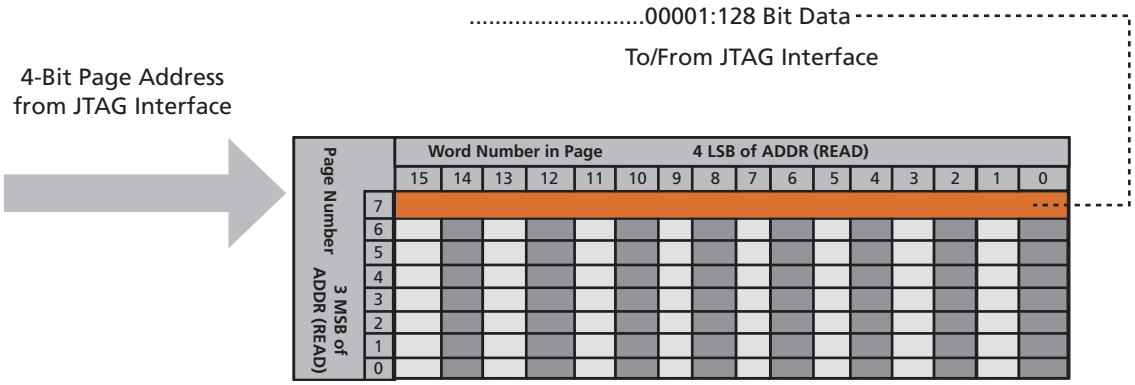

Figure 6 • **Accessing FROM Using JTAG Port**

FROM Design Flow

The Actel Libero® Integrated Design Environment (IDE) software has extensive FROM support, including FROM generation, instantiation, simulation, and programming. Figure 7 shows the user flow diagram. In the design flow, there are three main steps:

1. FROM generation and instantiation in the design
2. Simulation of FROM design
3. Programming file generation for FROM design

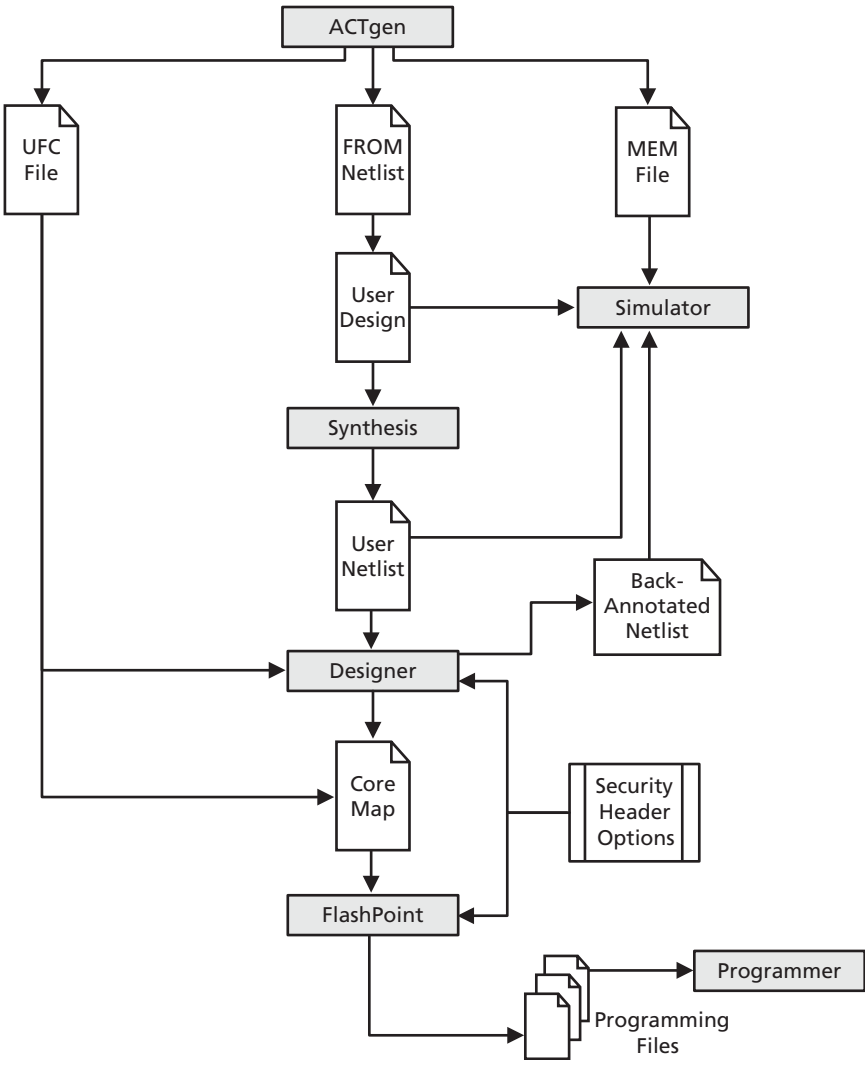

Figure 7 • **FROM Design Flow**

1. FROM Generation and Instantiation in the Design

The ACTgen core generator, available in Libero IDE and Designer software, is the only tool that can be used to generate the FROM content. ACTgen has several user-friendly features to help generate the FROM contents. Instead of selecting each byte and assigning values, you can create a region within a page, modify the region, and assign properties to that region. The FROM user interface, shown in Figure 8, includes the configuration grid, existing regions list, and properties field. The Properties field specifies the region-specific information and defines the data used for that region. You can assign values to the following properties:

1. Static Fixed Data – Enables you to fix the data so that it cannot be changed during programming time. This option is useful when the you have fixed data stored in this region, which is required for the operation of the design in the FPGA. Key storage is one example.
2. Static Modifiable Data – Select this option when the data in a particular region is expected to be static data (such as a version number, which remains the same for a long duration but could conceivably change in future). This option enables you to avoid changing the value every time you enter new data.
3. Read from File – This provides the full flexibility of FROM usage to the customer. If you have a customized algorithm for generating the FROM data, you can specify this setting. You can then generate a text file with data for as many devices you wish to program and load that into the FlashPoint programming file generation software to get programming files that include all the data. ACTgen will optionally pass the location of the file where the data is stored, if the file is specified in ACTgen. Each text file has only one type of data format (binary, decimal, Hex or ASCII text). The length of each data file must be shorter or equal to the selected region length. If the data is shorter then the selected region length, the most significant bits shall be padded with 0s. For multiple text files for multiple regions, the first lines are for the first device. In ACTgen, the **Load Sim. Value From File** allows you to load the first device data in the mem file for simulation.
4. Auto Increment/Decrement – This scenario is useful when you specify the contents of FROM for a large number of devices in a series. You can specify the step value for the serial number and a maximum value for inventory control. During programming file generation, the actual number of devices to be programmed is specified and a start value is input to the software.

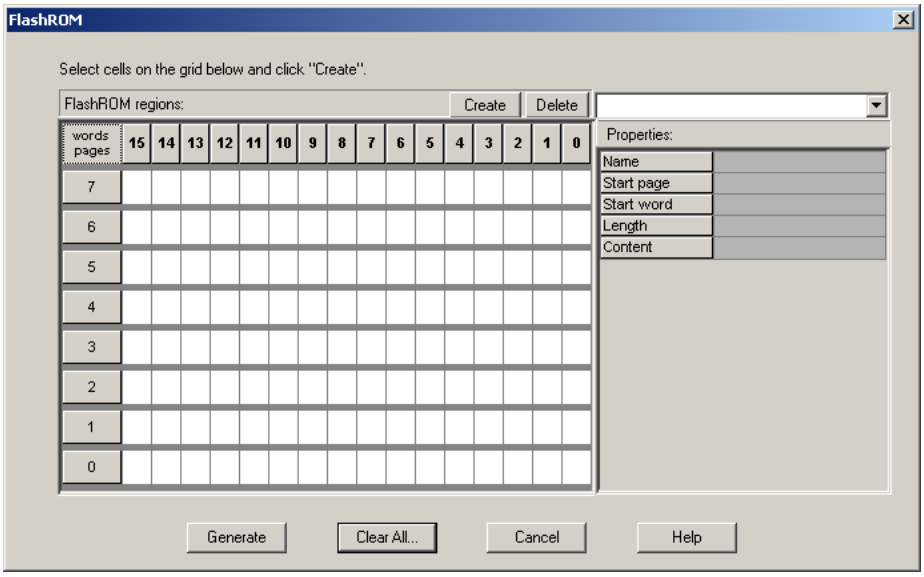

Figure 8 • **ACTgen GUI for the FROM**

ACTgen allows you to generate the FROM netlist in VHDL, Verilog, or EDIF formats. After the FROM netlist is generated, the core can be instantiated in the main design like other ACTgen cores. Note that the macro library name for FROM is UFROM. The following is a sample FROM VHDL netlist that can be instantiated in the main design.

```
library ieee;
use ieee.std_logic_1164.all;
library proasic3e;

entity FROM_a is
    port( ADDR : in std_logic_vector(6 downto 0); DOUT : out
        std_logic_vector(7 downto 0)) ;
end FROM_a;



architecture DEF_ARCH of  FROM_a is

    component UFROM
        generic (MEMORYFILE:string);
        port(DO0, DO1, DO2, DO3, DO4, DO5, DO6, DO7 : out
        std_logic;  ADDR0, ADDR1, ADDR2, ADDR3, ADDR4, ADDR5,
        ADDR6 : in std_logic := 'U') ;
    end component;

    component GND
        port( Y : out std_logic);
    end component;

    signal U_7_PIN2 : std_logic ;
    begin

    GND_1_net : GND port map(Y => U_7_PIN2);
    UFROM0 : UFROM
      generic map(MEMORYFILE => "FROM_a.mem")
      port map(DO0 => DOUT(0), DO1 => DOUT(1), DO2 => DOUT(2),
        DO3 => DOUT(3), DO4 => DOUT(4), DO5 => DOUT(5), DO6 =>
        DOUT(6), DO7 => DOUT(7), ADDR0 => ADDR(0), ADDR1 =>
        ADDR(1), ADDR2 => ADDR(2), ADDR3 => ADDR(3), ADDR4 =>
        ADDR(4), ADDR5 => ADDR(5), ADDR6 => ADDR(6));
end DEF_ARCH;
```

ACTgen generates the following files along with the netlist and are located in the actgen folder for Libero IDE project.

1. MEM (Memory Initialization) file
2. UFC (User Flash Configuration) file
3. Log file

The MEM file is used for simulation and explained in the next section. The UFC file, generated by ACTgen, has the FROM configuration for single or multiple devices and is used during STAPL generation. It contains the region properties and simulation value. Note that any changes on the mem file will not be reflected on the UFC file. Do not modify the UFC for changing FROM content. Instead, use the ACTgen GUI to modify the FROM content. See the "3. Programming File Generation for FROM Design"section for a description of how the UFC file is used during the programming file generation. The Log file has information regarding the file type and file location.

2. Simulation of FROM Design

The MEM file has 128 rows of 8 bits, each representing the contents of the FROM that is used for simulation. For example, the first row represents page 0, byte 0; the next row is page 0, byte 1 and the pattern continues. Note that the three MSBs of the address define the page number, and the four LSBs define byte number. So, if you send address 0000100 to FROM, this corresponds to the page 0 and byte 4 location, which is the fifth row in MEM file. ACTgen defaults to 0s for any unspecified locations of the FROM memory. Besides using the MEM file generated by ACTgen, you may create a binary file with 128 rows of 8 bits and use as a MEM file. Actel recommends that you use different file names if you plan to generate multiple MEM files. During simulation, Libero IDE passes the MEM file, used as the generic file in the netlist, along with the design files and testbench. If the you want to use different MEM files during simulation, you need to modify the generic file reference in the netlist.

```
.....................
UFROM0: UFROM
 --generic map(MEMORYFILE => "F:\Appsnotes\FROM\test_designs\testa\actgen\FROM_a.mem")
   generic map(MEMORYFILE => "F:\Appsnotes\FROM\test_designs\testa\actgen\FROM_b.mem")
.....................
```

The VITAL and Verilog simulation models accept the generics passed by the netlist, read the MEM file, and perform simulation with the data in the file.

3. Programming File Generation for FROM Design

FlashPoint is the programming software used to generate the programming files for ProASIC3/E devices. Depending on the applications, you can use the FlashPoint software to generate a STAPL file with different FROM contents, and in each case optional AES decryption is available. In order to generate a STAPL file that contains the same FPGA core content and different FROM contents, the FlashPoint software needs an Array Map file for the core and UFC file(s) for the FROM. This final STAPL file represents the combination of the logic of the FPGA core and FROM content.

FlashPoint generates the STAPL files that you can use to program the desired FROM page and/or FPGA core of the FPGA device contents. FlashPoint supports the encryption of the FROM content and/or the FPGA Array configuration data. In the case of using the FROM for device serialization, a sequence of unique FROM contents will be generated. When generating a programming file with multiple unique FROM contents, you can specify in FlashPoint whether to include all FROM content in a single STAPL file or generate a different STAPL file for each FROM (Figure 9 on page 10). The programming software (FlashPro) handles the single STAPL file that contains the FROM content from multiple devices. It enables you to program the FROM content into a series of devices sequentially (Figure 9 on page 10). See the *FlashPro User's Guide* for information on serial programming.

Figure 10 on page 12 shows the programming file generator, which enables different STAPL file generation methods. When you select **Program FlashROM** and choose the UFC file, the FlashROM Settings window appears, as shown in Figure 11 on page 12. In this window, you can select the FlashROM

page that you want to program and the data value for the configured regions. This enables you to use a different page for different programming files.

The programming hardware and software can load the FROM with the appropriate STAPL file. Programming software handles the single STAPL file that contains multiple FROM contents for multiple devices and programs the FROMs in sequential order (e.g., for device serialization). This feature is supported in the programming software. After programming with the STAPLE file, you can run **DEVICE_INFO** to check the FROM content.

DEVICE_INFO displays the FROM content, serial number, Design Name and checksum as shown below:

```
EXPORT IDCODE[32] = 123261CF
EXPORT SILSIG[32] = 00000000
User information :
CHECKSUM: 61A0
Design Name:        TOP
Programming Method: STAPL
Algorithm Version: 1
Programmer: UNKNOWN
========================================
FlashROM Information :
EXPORT Region_7_0[128] = FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
========================================
Security Setting :
Encrypted FlashROM Programming Enabled.
Encrypted FPGA Array Programming Enabled.
========================================
```

The Libero IDE file manager recognizes the UFC and MEM files and displays them in the appropriate view. Libero IDE also recognizes the multiple programming files, if you choose the option to generate multiple files for multiple FROM content in Designer. These features enable a user-friendly flow for the FROM generation and programming in Libero IDE.

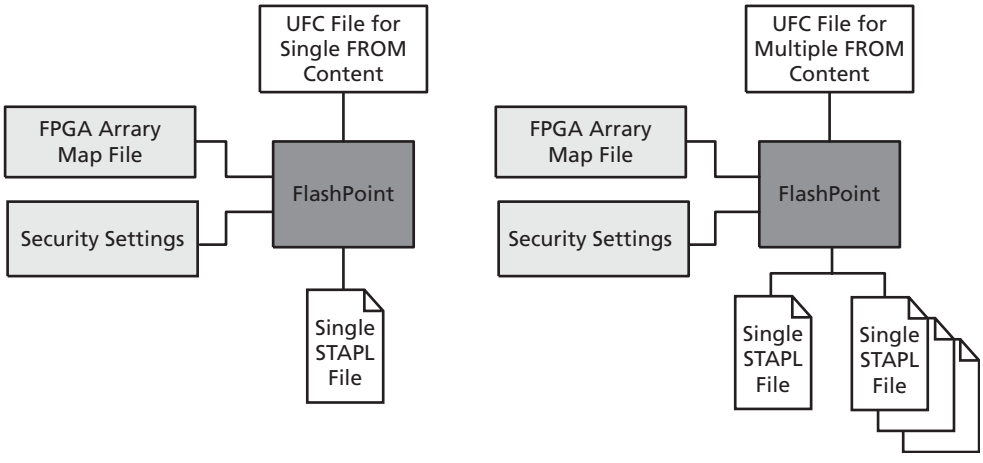

Figure 9 • **Single or Multiple Programming File Generation**

Custom Serialization Using FROM

You can use FROM for device serialization or inventory control by using the Auto Inc region or Read From File region. FlashPoint will automatically generate the serial number sequence for the Auto Inc region with the **Start Value**, **Max Value**, and **Step Value** provided. If you have a unique serial number generation scheme that you prefer, the Read From File Region allows you to import the file with your serial number scheme programmed into the region. See the *FlashPro User's Guide* for custom serialization file format information.

The following steps describe how to perform device serialization or inventory control using FROM:

1. Generate FROM using ACTgen. From the **Properties** section in the **FlashROM Settings** dialog box, select **Auto Inc** or **Read From File** region. For **Auto Inc** region, specify the desired step value. You will not be able to modify this value in the FlashPoint software.
2. Go through the regular design flow and finish place-and-route.
3. Select **Programming File** in Designer and open **Generate Programming File** (Figure 10 on page 12).
4. Select **Program FlashROM** and browse to the UFC file and select **Next**. The **FlashROM Settings** window appears, as shown in Figure 11 on page 12.
5. Select the FROM page you want to program and the data value for the configured regions. The STAPL file generated will contain only the data that targets the selected FROM page.
6. Modify properties for the serialization.
   – For Auto Inc region specify the Start and Max values.
   – For Read From File region select the file name of the custom serialization file.
7. Select the FlashROM programming file type you want to generate from the two options below:
   – Single programming file for all devices option: generates one programming file with all FROM values.
   – One programming file per device: generates a separate programming file for each FROM value.
8. Enter the number of devices you want to program and generate the required programming file.
9. Open the programming software and load the programming file. The ProASIC3/E programming software, FlashPro3 and Silicon Sculptor II, supports the device serialization feature. If for some reason the device fails to program a part during serialization, the software allows you to reuse the serial data or skip the serial data. Refer to the *FlashPro User's Guide* for details.

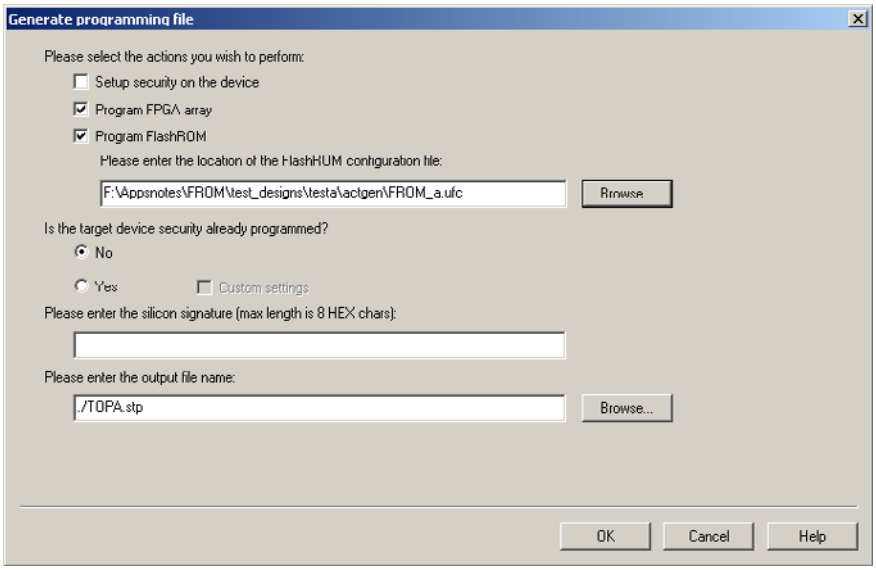

Figure 10 • **Programming File Generator**

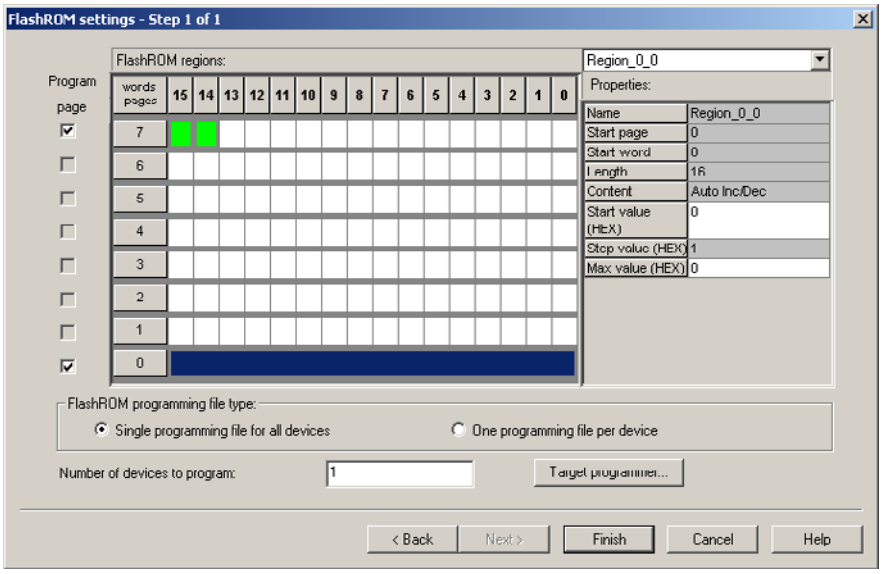

Figure 11 • **Setting FROM during Programming File Generation**

Conclusion

The ProASIC3/E families are the only FPGAs that offer on-chip FROM support. This application note presented information on the FROM architecture, possible applications, programming, access through the JTAG and UJTAG interface, and integration into your design. In addition, the Libero IDE tool set enables easy creation and modification of the FROM content.

The nonvolatile FROM block in the FPGA can be customized, enabling multiple applications.

Additionally, the security offered by the ProASIC3/E devices keeps both the contents of FROM and the FPGA design safe from system over-builders, system cloners, and IP thieves.

Related Documents

Application Notes

ProASIC3/E Security
http://www.actel.com/documents/PA3_E_Security_AN.pdf
UJTAG Applications in ProASIC3/E Devices
http://www.actel.com/documents/PA3_E_UJTAG_AN.pdf

User's Guides

FlashPro User's Guide
http://www.actel.com/documents/flashproUG.pdf

List of Changes

Previous Version	Changes in the current version 51900053-2/6.05	Page
51900053-1/1.05	Figure 5 and Figure 6 are new.	5
	Figure 7 was updated.	6
	The "1. FROM Generation and Instantiation in the Design" section was updated.	7
	The "3. Programming File Generation for FROM Design" section was updated.	9

Note: *The part number is located on the last page of the document.*

www.actel.com

Actel Corporation

2061 Stierlin Court
Mountain View, CA
94043-4655 USA
Phone 650.318.4200
Fax 650.318.4600

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
www.jp.actel.com

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
www.actel.com.cn

Suite 2114, Two Pacific Place
88 Queensway, Admiralty
Hong Kong
Phone +852 2185 6460
Fax +852 2185 6488

51900053-2/6.05