Welcome to E-XFL.COM

Understanding Embedded - FPGAs (Field Programmable Gate Array)

Embedded - FPGAs, or Field Programmable Gate Arrays, are advanced integrated circuits that offer unparalleled flexibility and performance for digital systems. Unlike traditional fixed-function logic devices, FPGAs can be programmed and reprogrammed to execute a wide array of logical operations, enabling customized functionality tailored to specific applications. This reprogrammability allows developers to iterate designs quickly and implement complex functions without the need for custom hardware.

Applications of Embedded - FPGAs

The versatility of Embedded - FPGAs makes them indispensable in numerous fields. In telecommunications,

Details

Product Status	Active
Number of LABs/CLBs	-
Number of Logic Elements/Cells	-
Total RAM Bits	-
Number of I/O	49
Number of Gates	30000
Voltage - Supply	1.425V ~ 1.575V
Mounting Type	Surface Mount
Operating Temperature	-20°C ~ 85°C (TJ)
Package / Case	68-VFQFN Exposed Pad
Supplier Device Package	68-QFN (8x8)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/a3pn030-zqng68

Related Documents

User's Guides

Designer User's Guide

http://www.microsemi.com/soc/documents/designer_ug.pdf

List of Changes

The following table lists critical changes that were made in each revision of the chapter.

Date	Changes	Page
August 2012	The "I/O State of Newly Shipped Devices" section is new (SAR 39542).	14
July 2010	This chapter is no longer published separately with its own part number and version but is now part of several FPGA fabric user's guides.	N/A
v1.4 (December 2008)	IGLOO nano and ProASIC3 nano devices were added to Table 1-1 • Flash-Based FPGAs.	10
	Figure 1-2 • IGLOO and ProASIC3 nano Device Architecture Overview with Two I/O Banks (applies to 10 k and 30 k device densities, excluding IGLOO PLUS devices) through Figure 1-5 • IGLOO, IGLOO nano, ProASIC3 nano, and ProASIC3/L Device Architecture Overview with Four I/O Banks (AGL600 device is shown) are new.	11, 12
	Table 1-4 • IGLOO nano and ProASIC3 nano Array Coordinates is new.	17
v1.3 (October 2008)	The title of this document was changed from "Core Architecture of IGLOO and ProASIC3 Devices" to "FPGA Array Architecture in Low Power Flash Devices."	9
	The "FPGA Array Architecture Support" section was revised to include new families and make the information more concise.	10
	Table 1-2 • IGLOO and ProASIC3 Array Coordinates was updated to include Military ProASIC3/EL and RT ProASIC3 devices.	16
v1.2 (June 2008)	The following changes were made to the family descriptions in Table 1-1 • Flash-Based FPGAs: • ProASIC3L was updated to include 1.5 V. • The number of PLLs for ProASIC3E was changed from five to six.	10
v1.1 (March 2008)	Table 1-1 • Flash-Based FPGAs and the accompanying text was updated to include the IGLOO PLUS family. The "IGLOO Terminology" section and "Device Overview" section are new.	10
	The "Device Overview" section was updated to note that 15 k devices do not support SRAM or FIFO.	11
	Figure 1-6 • IGLOO PLUS Device Architecture Overview with Four I/O Banks is new.	13
	Table 1-2 • IGLOO and ProASIC3 Array Coordinates was updated to add A3P015 and AGL015.	16
	Table 1-3 • IGLOO PLUS Array Coordinates is new.	16

External I/O or Local signal as Clock Source

External I/O refers to regular I/O pins are labeled with the I/O convention IOuxwByVz. You can allow the external I/O or internal signal to access the global. To allow the external I/O or internal signal to access the global network, you need to instantiate the CLKINT macro. Refer to Figure 3-4 on page 35 for an example illustration of the connections. Instead of using CLKINT, you can also use PDC to promote signals from external I/O or internal signal to the global network. However, it may cause layout issues because of synthesis logic replication. Refer to the "Global Promotion and Demotion Using PDC" section on page 51 for details.
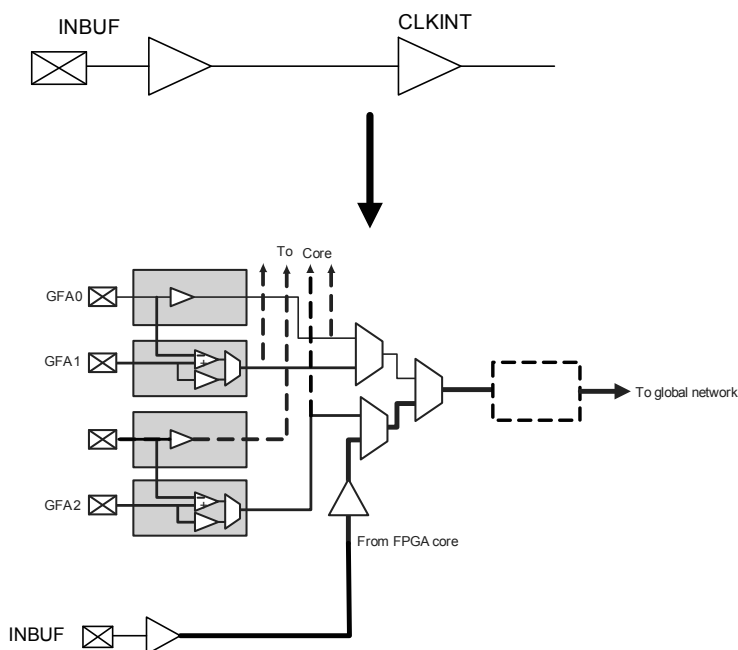

***Figure 3-14 •** CLKINT Macro*

Using Global Macros in Synplicity

The Synplify® synthesis tool automatically inserts global buffers for nets with high fanout during synthesis. By default, Synplicity® puts six global macros (CLKBUF or CLKINT) in the netlist, including any global instantiation or PLL macro. Synplify always honors your global macro instantiation. If you have a PLL (only primary output is used) in the design, Synplify adds five more global buffers in the netlist. Synplify uses the following global counting rule to add global macros in the netlist:

1. CLKBUF: 1 global buffer
2. CLKINT: 1 global buffer
3. CLKDLY: 1 global buffer
4. PLL: 1 to 3 global buffers
   – GLA, GLB, GLC, YB, and YC are counted as 1 buffer.
   – GLB or YB is used or both are counted as 1 buffer.
   – GLC or YC is used or both are counted as 1 buffer.

Table 4-18 • **Fusion Dynamic CCC Division by Half Configuration**

OADIVHALF / OBDIVHALF / OCDIVHALF	OADIV<4:0> / OBDIV<4:0> / OCDIV<4:0> (in decimal)	Divider Factor	Input Clock Frequency	Output Clock Frequency (MHz)
1	2	1.5	100 MHz RC Oscillator	66.7
	4	2.5		40.0
	6	3.5		28.6
	8	4.5		22.2
	10	5.5		18.2
	12	6.5		15.4
	14	7.5		13.3
	16	8.5		11.8
	18	9.5		10.5
	20	10.5		9.5
	22	11.5		8.7
	24	12.5		8.0
	26	13.5		7.4
	28	14.5		6.9
0	0–31	1–32	Other Clock Sources	Depends on other divider settings

Table 4-19 • **Configuration Bit <76:75> / VCOSEL<2:1> Selection for All Families**

	VCOSEL[2:1]							
	00		01		10		11	
Voltage	Min. (MHz)	Max. (MHz)	Min. (MHz)	Max. (MHz)	Min. (MHz)	Max. (MHz)	Min. (MHz)	Max. (MHz)
IGLOO and IGLOO PLUS								
1.2 V ± 5%	24	35	30	70	60	140	135	160
1.5 V ± 5%	24	43.75	30	87.5	60	175	135	250
ProASIC3L, RT ProASIC3, and Military ProASIC3/L								
1.2 V ± 5%	24	35	30	70	60	140	135	250
1.5 V ± 5%	24	43.75	30	70	60	175	135	350
ProASIC3 and Fusion								
1.5 V ± 5%	24	43.75	33.75	87.5	67.5	175	135	350

Table 4-20 • **Configuration Bit <74> / VCOSEL<0> Selection for All Families**

VCOSEL[0]	Description
0	Fast PLL lock acquisition time with high tracking jitter. Refer to the corresponding datasheet for specific value and definition.
1	Slow PLL lock acquisition time with low tracking jitter. Refer to the corresponding datasheet for specific value and definition.

Dynamic PLL Configuration

To generate a dynamically reconfigurable CCC, the user should select **Dynamic CCC** in the configuration section of the SmartGen GUI (Figure 4-26). This will generate both the CCC core and the configuration shift register / control bit MUX.

Figure 4-26 • **SmartGen GUI**

Even if dynamic configuration is selected in SmartGen, the user must still specify the static configuration data for the CCC (Figure 4-27). The specified static configuration is used whenever the MODE signal is set to LOW and the CCC is required to function in the static mode. The static configuration data can be used as the default behavior of the CCC where required.

Figure 4-27 • **Dynamic CCC Configuration in SmartGen**

When SmartGen is used to define the configuration that will be shifted in via the serial interface, SmartGen prints out the values of the 81 configuration bits. For ease of use, several configuration bits are automatically inferred by SmartGen when the dynamic PLL core is generated; however, <71:73> (STATASEL, STATBSEL, STATCSEL) and <77:79> (DYNASEL, DYNBSEL, DYNCSEL) depend on the input clock source of the corresponding CCC. Users must first run Layout in Designer to determine the exact setting for these ports. After Layout is complete, generate the "CCC_Configuration" report by choosing **Tools** > **Reports** > **CCC_Configuration** in the Designer software. Refer to "PLL Configuration Bits Description" on page 90 for descriptions of the PLL configuration bits. For simulation purposes, bits <71:73> and <78:80> are "don't care." Therefore, it is strongly suggested that SmartGen be used to generate the correct configuration bit settings for the dynamic PLL core.

After setting all the required parameters, users can generate one or more PLL configurations with HDL or EDIF descriptions by clicking the **Generate** button. SmartGen gives the option of saving session results and messages in a log file:

```
****************
Macro Parameters
****************

Name                         : dyn_pll_hardio
Family                       : ProASIC3E
Output Format                : VERILOG
Type                         : Dynamic CCC
Input Freq(MHz)              : 30.000
CLKA Source                  : Hardwired I/O
Feedback Delay Value Index   : 1
Feedback Mux Select          : 1
XDLY Mux Select              : No
Primary Freq(MHz)            : 33.000
Primary PhaseShift           : 0
Primary Delay Value Index    : 1
Primary Mux Select           : 4
Secondary1 Freq(MHz)         : 40.000
Use GLB                      : YES
Use YB                       : NO
GLB Delay Value Index        : 1
YB Delay Value Index         : 1
Secondary1 PhaseShift        : 0
Secondary1 Mux Select        : 0
Secondary1 Input Freq(MHz)   : 40.000
CLKB Source                  : Hardwired I/O
Secondary2 Freq(MHz)         : 50.000
Use GLC                      : YES
Use YC                       : NO
GLC Delay Value Index        : 1
YC Delay Value Index         : 1
Secondary2 PhaseShift        : 0
Secondary2 Mux Select        : 0
Secondary2 Input Freq(MHz)   : 50.000
CLKC Source                  : Hardwired I/O

Configuration Bits:
FINDIV[6:0]     0000101
FBDIV[6:0]      0100000
OADIV[4:0]      00100
OBDIV[4:0]      00000
OCDIV[4:0]      00000
OAMUX[2:0]      100
OBMUX[2:0]      000
OCMUX[2:0]      000
FBSEL[1:0]      01
FBDLY[4:0]      00000
XDLYSEL         0
DLYGLA[4:0]     00000
DLYGLB[4:0]     00000
```

Figure 5-12 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 5-13. In this window, you can select the FlashROM page you want to program and the data value for the configured regions. This enables you to use a different page for different programming files.

Figure 5-12 • **Programming File Generator**

Figure 5-13 • **Setting FlashROM during Programming File Generation**

The programming hardware and software can load the FlashROM with the appropriate STAPL file. Programming software handles the single STAPL file that contains multiple FlashROM contents for multiple devices, and programs the FlashROM in sequential order (e.g., for device serialization). This feature is supported in the programming software. After programming with the STAPL file, you can run DEVICE_INFO to check the FlashROM content.
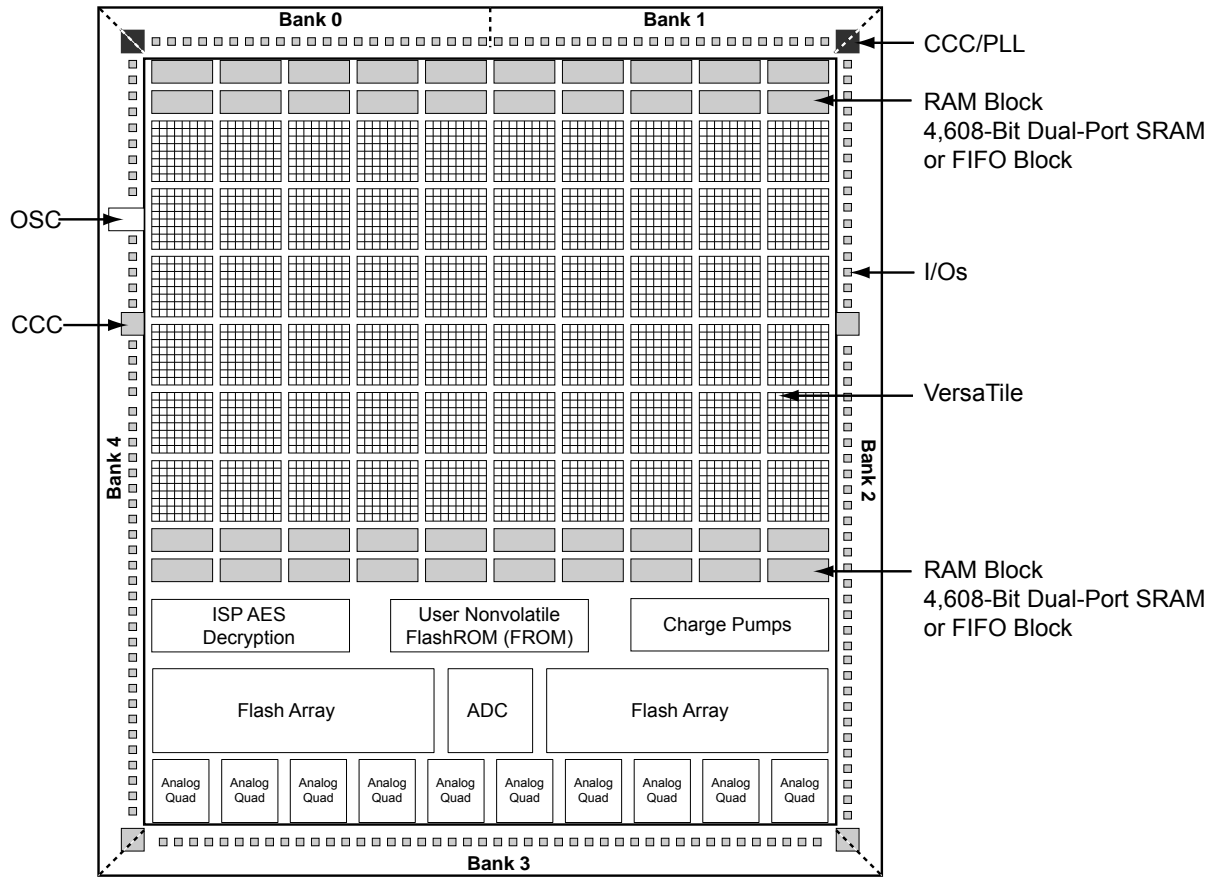
Figure 6-2 • **Fusion Device Architecture Overview (AFS600)**

Flash FPGAs I/O Support

The flash FPGAs listed in Table 8-1 support I/Os and the functions described in this document.

Table 8-1 • **Flash-Based FPGAs**

Series	Family*	Description
IGLOO	IGLOO	Ultra-low power 1.2 V to 1.5 V FPGAs with Flash*Freeze technology
	IGLOOe	Higher density IGLOO FPGAs with six PLLs and additional I/O standards
	IGLOO nano	The industry's lowest-power, smallest-size solution
	IGLOO PLUS	IGLOO FPGAs with enhanced I/O capabilities
ProASIC3	ProASIC3	Low power, high-performance 1.5 V FPGAs
	ProASIC3E	Higher density ProASIC3 FPGAs with six PLLs and additional I/O standards
	ProASIC3 nano	Lowest-cost solution with enhanced I/O capabilities
	ProASIC3L	ProASIC3 FPGAs supporting 1.2 V to 1.5 V with Flash*Freeze technology
	RT ProASIC3	Radiation-tolerant RT3PE600L and RT3PE3000L
	Military ProASIC3/EL	Military temperature A3PE600L, A3P1000, and A3PE3000L
	Automotive ProASIC3	ProASIC3 FPGAs qualified for automotive applications
Fusion	Fusion	Mixed signal FPGA integrating ProASIC3 FPGA fabric, programmable analog block, support for ARM® Cortex™-M1 soft processors, and flash memory into a monolithic device

Note: *The device names link to the appropriate datasheet, including product brief, DC and switching characteristics, and packaging information.*

IGLOO Terminology

In documentation, the terms IGLOO series and IGLOO devices refer to all of the IGLOO devices as listed in Table 8-1. Where the information applies to only one product line or limited devices, these exclusions will be explicitly stated.

ProASIC3 Terminology

In documentation, the terms ProASIC3 series and ProASIC3 devices refer to all of the ProASIC3 devices as listed in Table 8-1. Where the information applies to only one product line or limited devices, these exclusions will be explicitly stated.

To further understand the differences between the IGLOO and ProASIC3 devices, refer to the *Industry's Lowest Power FPGAs Portfolio.*

Software-Controlled I/O Attributes

Users may modify these programmable I/O attributes using the I/O Attribute Editor. Modifying an I/O attribute may result in a change of state in Designer. Table 8-2 details which steps have to be re-run as a function of modified I/O attribute.

Table 8-2 • **Designer State (resulting from I/O attribute modification)**

I/O Attribute	Designer States[1]				
	Compile	**Layout**	**Fuse**	**Timing**	**Power**
Slew Control[2]	No	No	Yes	Yes	Yes
Output Drive (mA)	No	No	Yes	Yes	Yes
Skew Control	No	No	Yes	Yes	Yes
Resistor Pull	No	No	Yes	Yes	Yes
Input Delay	No	No	Yes	Yes	Yes
Schmitt Trigger	No	No	Yes	Yes	Yes
OUT_LOAD	No	No	No	Yes	Yes
COMBINE_REGISTER	Yes	Yes	N/A	N/A	N/A

Notes:

1. *No = Remains the same, Yes = Re-run the step, N/A = Not applicable*

2. *Skew control does not apply to IGLOO nano, IGLOO PLUS, and ProASIC3 nano devices.*

3. *Programmable input delay is applicable only for ProASIC3E, ProASIC3EL, RT ProASIC3, and IGLOOe devices.*

Input Support for DDR

The basic structure to support a DDR input is shown in Figure 9-2. Three input registers are used to capture incoming data, which is presented to the core on each rising edge of the I/O register clock. Each I/O tile supports DDR inputs.
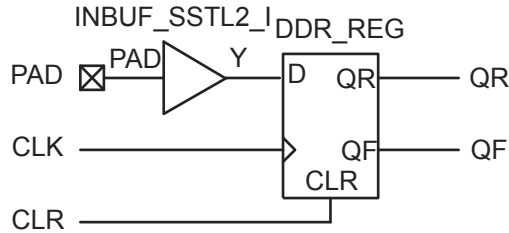

Figure 9-2 • **DDR Input Register Support in Low Power Flash Devices**

Output Support for DDR

The basic DDR output structure is shown in Figure 9-1 on page 205. New data is presented to the output every half clock cycle.

Note: DDR macros and I/O registers do not require additional routing. The combiner automatically recognizes the DDR macro and pushes its registers to the I/O register area at the edge of the chip. The routing delay from the I/O registers to the I/O buffers is already taken into account in the DDR macro.
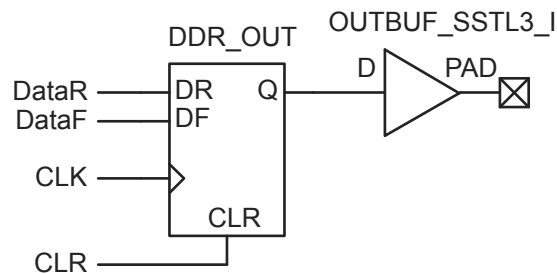

Figure 9-3 • **DDR Output Register (SSTL3 Class I)**

11 – Security in Low Power Flash Devices

Security in Programmable Logic

The need for security on FPGA programmable logic devices (PLDs) has never been greater than today. If the contents of the FPGA can be read by an external source, the intellectual property (IP) of the system is vulnerable to unauthorized copying. Fusion, IGLOO, and ProASIC3 devices contain state-of-the-art circuitry to make the flash-based devices secure during and after programming. Low power flash devices have a built-in 128-bit Advanced Encryption Standard (AES) decryption core (except for 30 k gate devices and smaller). The decryption core facilitates secure in-system programming (ISP) of the FPGA core array fabric, the FlashROM, and the Flash Memory Blocks (FBs) in Fusion devices. The FlashROM, Flash Blocks, and FPGA core fabric can be programmed independently of each other, allowing the FlashROM or Flash Blocks to be updated without the need for change to the FPGA core fabric.

Microsemi has incorporated the AES decryption core into the low power flash devices and has also included the Microsemi flash-based lock technology, FlashLock.$^{®}$ Together, they provide leading-edge security in a programmable logic device. Configuration data loaded into a device can be decrypted prior to being written to the FPGA core using the AES 128-bit block cipher standard. The AES encryption key is stored in on-chip, nonvolatile flash memory.

This document outlines the security features offered in low power flash devices, some applications and uses, as well as the different software settings for each application.

Figure 11-1 • **Overview on Security**

3. Choose the desired settings for the FlashROM configurations to be programmed (Figure 11-13). Click **Finish** to generate the STAPL programming file for the design.

Figure 11-13 • **FlashROM Configuration Settings for Low Power Flash Devices**

Generation of Security Header Programming File Only— Application 2

As mentioned in the "Application 2: Nontrusted Environment—Unsecured Location" section on page 243, the designer may employ FlashLock Pass Key protection or FlashLock Pass Key with AES encryption on the device before sending it to a nontrusted or unsecured location for device programming. To achieve this, the user needs to generate a programming file containing only the security settings desired (Security Header programming file).

Note: If AES encryption is configured, FlashLock Pass Key protection must also be configured.

The available security options are indicated in Table 11-4 and Table 11-5 on page 251.

Table 11-4 • **FlashLock Security Options for IGLOO and ProASIC3**

Security Option	FlashROM Only	FPGA Core Only	Both FlashROM and FPGA
No AES / no FlashLock	–	–	–
FlashLock only	✓	✓	✓
AES and FlashLock	✓	✓	✓

STAPL File with AES Encryption

- Does not contain AES key / FlashLock Key information
- Intended for transmission through web or service to unsecured locations for programming

```
==========================================
NOTE "CREATOR" "Designer Version: 6.1.1.108";
NOTE "DEVICE" "A3PE600";
NOTE "PACKAGE" "208 PQFP";
NOTE "DATE" "2005/04/08";
NOTE "STAPL_VERSION" "JESD71";
NOTE "IDCODE" "$123261CF";
NOTE "DESIGN" "counter32";
NOTE "CHECKSUM" "$EF57";
NOTE "SAVE_DATA" "FRomStream";
NOTE "SECURITY" "ENCRYPT FROM CORE ";
NOTE "ALG_VERSION" "1";
NOTE "MAX_FREQ" "20000000";
NOTE "SILSIG" "$00000000";
```

Conclusion

The new and enhanced security features offered in Fusion, IGLOO, and ProASIC3 devices provide state-of-the-art security to designs programmed into these flash-based devices. Microsemi low power flash devices employ the encryption standard used by NIST and the U.S. government—AES using the 128-bit Rijndael algorithm.

The combination of an on-chip AES decryption engine and FlashLock technology provides the highest level of security against invasive attacks and design theft, implementing the most robust and secure ISP solution. These security features protect IP within the FPGA and protect the system from cloning, wholesale "black box" copying of a design, invasive attacks, and explicit IP or data theft.

Glossary

Term	Explanation
Security Header programming file	Programming file used to program the FlashLock Pass Key and/or AES key into the device to secure the FPGA, FlashROM, and/or FBs.
AES (encryption) key	128-bit key defined by the user when the AES encryption option is set in the Microsemi Designer software when generating the programming file.
FlashLock Pass Key	128-bit key defined by the user when the FlashLock option is set in the Microsemi Designer software when generating the programming file.
	The FlashLock Key protects the security settings programmed to the device. Once a device is programmed with FlashLock, whatever settings were chosen at that time are secure.
FlashLock	The combined security features that protect the device content from attacks. These features are the following:
	• Flash technology that does not require an external bitstream to program the device
	• FlashLock Pass Key that secures device content by locking the security settings and preventing access to the device as defined by the user
	• AES key that allows secure, encrypted device reprogrammability

References

National Institute of Standards and Technology. "ADVANCED ENCRYPTION STANDARD (AES) Questions and Answers." 28 January 2002 (10 January 2005).
See http://csrc.nist.gov/archive/aes/index1.html for more information.

FlashROM and Programming Files

Each low power flash device has 1 kbit of on-chip, nonvolatile flash memory that can be accessed from the FPGA core. This nonvolatile FlashROM is arranged in eight pages of 128 bits (Figure 12-3). Each page can be programmed independently, with or without the 128-bit AES encryption. The FlashROM can only be programmed via the IEEE 1532 JTAG port and cannot be programmed from the FPGA core. In addition, during programming of the FlashROM, the FPGA core is powered down automatically by the on-chip programming control logic.

		Byte Number in Page															
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Page Number	7																
	6																
	5																
	4																
	3																
	2																
	1																
	0																

Figure 12-3 • **FlashROM Architecture**

When using FlashROM combined with AES, many subscription-based applications or device serialization applications are possible. The FROM configurator found in the Libero SoC Catalog supports easy management of the FlashROM contents, even over large numbers of devices. The FROM configurator can support FlashROM contents that contain the following:

- Static values
- Random numbers
- Values read from a file
- Independent updates of each page

In addition, auto-incrementing of fields is possible. In applications where the FlashROM content is different for each device, you have the option to generate a single STAPL file for all the devices or individual serialization files for each device. For more information on how to generate the FlashROM content for device serialization, refer to the "FlashROM in Microsemi's Low Power Flash Devices" section on page 117.

Libero SoC includes a unique tool to support the generation and management of FlashROM and FPGA programming files. This tool is called FlashPoint.

Depending on the applications, designers can use the FlashPoint software to generate a STAPL file with different contents. In each case, optional AES encryption and/or different security settings can be set.

In Designer, when you click the Programming File icon, FlashPoint launches, and you can generate STAPL file(s) with four different cases (Figure 12-4 on page 268). When the serialization feature is used during the configuration of FlashROM, you can generate a single STAPL file that will program all the devices or an individual STAPL file for each device.

The following cases present the FPGA core and FlashROM programming file combinations that can be used for different applications. In each case, you can set the optional security settings (FlashLock Pass Key and/or AES Key) depending on the application.

1. A single STAPL file or multiple STAPL files with multiple FlashROM contents and the FPGA core content. A single STAPL file will be generated if the device serialization feature is not used. You can program the whole FlashROM or selectively program individual pages.
2. A single STAPL file for the FPGA core content

Date	Changes	Page
July 2010 (continued)	The "Chain Integrity Test Error Analyze Chain Failure" section was renamed to the "Scan Chain Failure" section, and the Analyze Chain command was changed to Scan Chain. It was noted that occasionally a faulty programmer can cause scan chain failures.	272
v1.5 (August 2009)	The "CoreMP7 Device Security" section was removed from "Security in ARM-Enabled Low Power Flash Devices", since M7-enabled devices are no longer supported.	265
v1.4 (December 2008)	The "ISP Architecture" section was revised to include information about core voltage for IGLOO V2 and ProASIC3L devices, as well as 50 mV increments allowable in Designer software.	261
	IGLOO nano and ProASIC3 nano devices were added to Table 12-1 • Flash-Based FPGAs Supporting ISP.	262
	A second capacitor was added to Figure 12-6 • Board Layout and Programming Header Top View.	271
v1.3 (October 2008)	The "ISP Support in Flash-Based Devices" section was revised to include new families and make the information more concise.	262
v1.2 (June 2008)	The following changes were made to the family descriptions in Table 12-1 • Flash-Based FPGAs Supporting ISP: • ProASIC3L was updated to include 1.5 V. • The number of PLLs for ProASIC3E was changed from five to six.	262
v1.1 (March 2008)	The "ISP Architecture" section was updated to included the IGLOO PLUS family in the discussion of family-specific support. The text, "When 1.2 V is used, the device can be reprogrammed in-system at 1.5 V only," was revised to state, "Although the device can operate at 1.2 V core voltage, the device can only be reprogrammed when all supplies (VCC, VCCI, and VJTAG) are at 1.5 V."	261
	The "ISP Support in Flash-Based Devices" section and Table 12-1 • Flash-Based FPGAs Supporting ISP were updated to include the IGLOO PLUS family. The "IGLOO Terminology" section and "ProASIC3 Terminology" section are new.	262
	The "Security" section was updated to mention that 15 k gate devices do not have a built-in 128-bit decryption core.	264
	Table 12-2 • Power Supplies was revised to remove the Normal Operation column and add a table note stating, "All supply voltages should be at 1.5 V or higher, regardless of the setting during normal operation."	263
	The "ISP Programming Header Information" section was revised to change FP3-26PIN-ADAPTER to FP3-10PIN-ADAPTER-KIT. Table 12-3 • Programming Header Ordering Codes was updated with the same change, as well as adding the part number FFSD-05-D-06.00-01-N, a 10-pin cable with 50-mil-pitch sockets.	269
	The "Board-Level Considerations" section was updated to describe connecting two capacitors in parallel across VPUMP and GND for proper programming.	271
v1.0 (January 2008)	Information was added to the "Programming Voltage (VPUMP) and VJTAG" section about the JTAG interface pin.	263
51900055-2/7.06	ACTgen was changed to SmartGen.	N/A
	In Figure 12-6 • Board Layout and Programming Header Top View, the order of the text was changed to: VJTAG from the target board VCCI from the target board VCC from the target board	271

Microsemi

15 – Boundary Scan in Low Power Flash Devices

Boundary Scan

Low power flash devices are compatible with IEEE Standard 1149.1, which defines a hardware architecture and the set of mechanisms for boundary scan testing. JTAG operations are used during boundary scan testing.

The basic boundary scan logic circuit is composed of the TAP controller, test data registers, and instruction register (Figure 15-2 on page 294).

Low power flash devices support three types of test data registers: bypass, device identification, and boundary scan. The bypass register is selected when no other register needs to be accessed in a device. This speeds up test data transfer to other devices in a test data path. The 32-bit device identification register is a shift register with four fields (LSB, ID number, part number, and version). The boundary scan register observes and controls the state of each I/O pin. Each I/O cell has three boundary scan register cells, each with serial-in, serial-out, parallel-in, and parallel-out pins.

TAP Controller State Machine

The TAP controller is a 4-bit state machine (16 states) that operates as shown in Figure 15-1.

The 1s and 0s represent the values that must be present on TMS at a rising edge of TCK for the given state transition to occur. IR and DR indicate that the instruction register or the data register is operating in that state.

The TAP controller receives two control inputs (TMS and TCK) and generates control and clock signals for the rest of the test logic architecture. On power-up, the TAP controller enters the Test-Logic-Reset state. To guarantee a reset of the controller from any of the possible states, TMS must remain HIGH for five TCK cycles. The TRST pin can also be used to asynchronously place the TAP controller in the Test-Logic-Reset state.
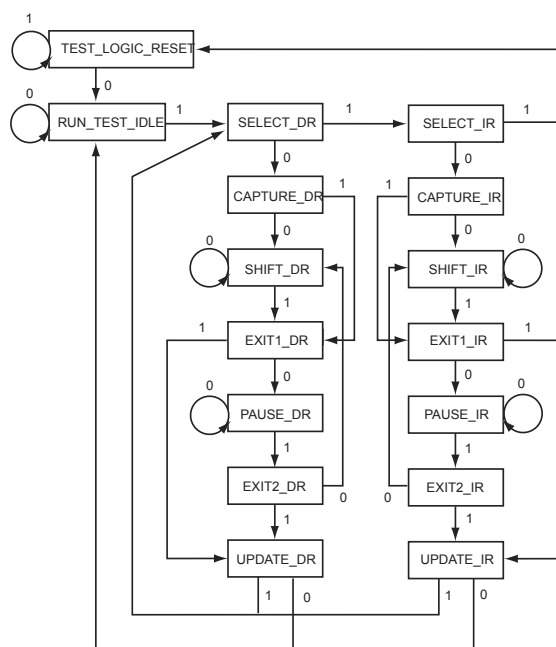
Figure 15-1 • **TAP Controller State Machine**

I/O Behavior at Power-Up/-Down

This section discusses the behavior of device I/Os, used and unused, during power-up/-down of V_{CC} and V_{CCI}. As mentioned earlier, VMVx and V_{CCI}Bx are tied together, and therefore, inputs and outputs are powered up/down at the same time.

I/O State during Power-Up/-Down

This section discusses the characteristics of I/O behavior during device power-up and power-down. Before the start of power-up, all I/Os are in tristate mode. The I/Os will remain tristated during power-up until the last voltage supply (VCC or VCCI) is powered to its functional level (power supply functional levels are discussed in the "Power-Up to Functional Time" section on page 312). After the last supply reaches the functional level, the outputs will exit the tristate mode and drive the logic at the input of the output buffer. Similarly, the input buffers will pass the external logic into the FPGA fabric once the last supply reaches the functional level. The behavior of user I/Os is independent of the VCC and VCCI sequence or the state of other voltage supplies of the FPGA (VPUMP and VJTAG). Figure 17-2 shows the output buffer driving HIGH and its behavior during power-up with 10 kΩ external pull-down. In Figure 17-2, VCC is powered first, and VCCI is powered 5 ms after VCC. Figure 17-3 on page 312 shows the state of the I/O when VCCI is powered about 5 ms before VCC. In the circuitry shown in Figure 17-3 on page 312, the output is externally pulled down.

During power-down, device I/Os become tristated once the first power supply (VCC or VCCI) drops below its brownout voltage level. The I/O behavior during power-down is also independent of voltage supply sequencing.

Figure 17-2 • **I/O State when VCC Is Powered before VCCI**