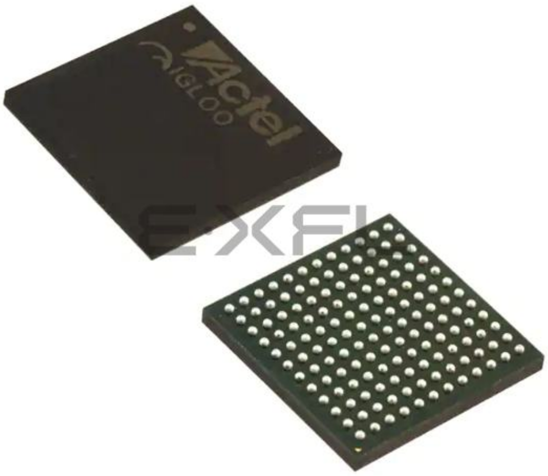


Welcome to E-XFL.COM

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	147456
Number of I/O	97
Number of Gates	1000000
Voltage - Supply	1.14V ~ 1.575V
Mounting Type	Surface Mount
Operating Temperature	0°C ~ 85°C (TJ)
Package / Case	144-LBGA
Supplier Device Package	144-FPBGA (13x13)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/a3p1000l-1fgg144

FlashROM Security Use Models 311
Generating Programming Files 313
Conclusion 324
Glossary 324
References 324
Related Documents 325
List of Changes 325

13 In-System Programming (ISP) of Microsemi's Low Power Flash Devices Using FlashPro4/3/3X . . . 327
Introduction 327
ISP Architecture 327
ISP Support in Flash-Based Devices 328
Programming Voltage (VPUMP) and VJTAG 329
Nonvolatile Memory (NVM) Programming Voltage 329
IEEE 1532 (JTAG) Interface 330
Security 330
Security in ARM-Enabled Low Power Flash Devices 331
FlashROM and Programming Files 333
Programming Solution 334
ISP Programming Header Information 335
Board-Level Considerations 337
Conclusion 338
Related Documents 338
List of Changes 339

14 Core Voltage Switching Circuit for IGLOO and ProASIC3L In-System Programming 341
Introduction 341
Microsemi's Flash Families Support Voltage Switching Circuit 342
Circuit Description 343
Circuit Verification 344
DirectC 346
Conclusion 346
List of Changes 347

15 Microprocessor Programming of Microsemi's Low Power Flash Devices 349
Introduction 349
Microprocessor Programming Support in Flash Devices 350
Programming Algorithm 351
Implementation Overview 351
Hardware Requirement 354
Security 354
Conclusion 355
List of Changes 356

16 Boundary Scan in Low Power Flash Devices. 357
Boundary Scan 357
TAP Controller State Machine 357
Microsemi's Flash Devices Support the JTAG Feature 358
Boundary Scan Support in Low Power Devices 359
Boundary Scan Opcodes 359

Array Coordinates

During many place-and-route operations in the Microsemi Designer software tool, it is possible to set constraints that require array coordinates. Table 1-2 provides array coordinates of core cells and memory blocks for IGLOO and ProASIC3 devices. Table 1-3 provides the information for IGLOO PLUS devices. Table 1-4 on page 17 provides the information for IGLOO nano and ProASIC3 nano devices. The array coordinates are measured from the lower left (0, 0). They can be used in region constraints for specific logic groups/blocks, designated by a wildcard, and can contain core cells, memories, and I/Os.

I/O and cell coordinates are used for placement constraints. Two coordinate systems are needed because there is not a one-to-one correspondence between I/O cells and core cells. In addition, the I/O coordinate system changes depending on the die/package combination. It is not listed in Table 1-2. The Designer ChipPlanner tool provides the array coordinates of all I/O locations. I/O and cell coordinates are used for placement constraints. However, I/O placement is easier by package pin assignment.

Figure 1-9 on page 17 illustrates the array coordinates of a 600 k gate device. For more information on how to use array coordinates for region/placement constraints, see the *Designer User's Guide* or online help (available in the software) for software tools.

Table 1-2 **• IGLOO and ProASIC3 Array Coordinates**

Device		**VersaTiles**				**Memory Rows**		**Entire Die**	
		Min.		**Max.**		**Bottom**	**Top**	**Min.**	**Max.**
IGLOO	**ProASIC3/ ProASIC3L**	**x**	**y**	**x**	**y**	**(x, y)**	**(x, y)**	**(x, y)**	**(x, y)**
AGL015	A3P015	3	2	34	13	None	None	(0, 0)	(37, 15)
AGL030	A3P030	3	3	66	13	None	None	(0, 0)	(69, 15)
AGL060	A3P060	3	2	66	25	None	(3, 26)	(0, 0)	(69, 29)
AGL125	A3P125	3	2	130	25	None	(3, 26)	(0, 0)	(133, 29)
AGL250	A3P250/L	3	2	130	49	None	(3, 50)	(0, 0)	(133, 53)
AGL400	A3P400	3	2	194	49	None	(3, 50)	(0, 0)	(197, 53)
AGL600	A3P600/L	3	4	194	75	(3, 2)	(3, 76)	(0, 0)	(197, 79)
AGL1000	A3P1000/L	3	4	258	99	(3, 2)	(3, 100)	(0, 0)	(261, 103)
AGLE600	A3PE600/L, RT3PE600L	3	4	194	75	(3, 2)	(3, 76)	(0, 0)	(197, 79)
	A3PE1500	3	4	322	123	(3, 2)	(3, 124)	(0, 0)	(325, 127)
AGLE3000	A3PE3000/L, RT3PE3000L	3	6	450	173	(3, 2) or (3, 4)	(3, 174) or (3, 176)	(0, 0)	(453, 179)

Table 1-3 **• IGLOO PLUS Array Coordinates**

Device	**VersaTiles**				**Memory Rows**		**Entire Die**	
	Min.		**Max.**		**Bottom**	**Top**	**Min.**	**Max.**
IGLOO PLUS	**x**	**y**	**x**	**y**	**(x, y)**	**(x, y)**	**(x, y)**	**(x, y)**
AGLP030	2	3	67	13	None	None	(0, 0)	(69, 15)
AGLP060	2	2	67	25	None	(3, 26)	(0, 0)	(69, 29)
AGLP125	2	2	131	25	None	(3, 26)	(0, 0)	(133, 29)

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

Table 2-4 summarizes the Flash*Freeze mode implementations.

Table 2-4 • **Flash*Freeze Mode Usage**

Flash*Freeze Mode Type	Description	Flash*Freeze Pin State	Instantiate ULSICC Macro	LSICC Signal	Operating Mode
1	Flash*Freeze mode is controlled only by the FF pin.	Deasserted	No	N/A	Normal operation
		Asserted	No	N/A	Flash*Freeze mode
2	Flash*Freeze mode is controlled by the FF pin and LSICC signal.	"Don't care"	Yes	Deasserted	Normal operation
		Deasserted	Yes	"Don't care"	Normal operation
		Asserted	Yes	Asserted	Flash*Freeze mode

Note: *Refer to Table 2-3 on page 26 for Flash*Freeze pin and LSICC signal assertion and deassertion values.*

IGLOO, ProASIC3L, and RT ProASIC3 I/O State in Flash*Freeze Mode

In IGLOO and ProASIC3L devices, when the device enters Flash*Freeze mode, I/Os become tristated. If the weak pull-up or pull-down feature is used, the I/Os will maintain the configured weak pull-up or pull-down status. This feature enables the design to set the I/O state to a certain level that is determined by the pull-up/-down configuration.

Table 2-5 shows the I/O pad state based on the configuration and buffer type.

Note that configuring weak pull-up or pull-down for the FF pin is not allowed. The FF pin can be configured as a Schmitt trigger input in IGLOOe, IGLOO nano, IGLOO PLUS, and ProASIC3EL devices.

Table 2-5 • **IGLOO, ProASIC3L, and RT ProASIC3 Flash*Freeze Mode (type 1 and type 2)—I/O Pad State**

Buffer Type		I/O Pad Weak Pull-Up/-Down	I/O Pad State in Flash*Freeze Mode
Input/Global		Enabled	Weak pull-up/pull-down*
		Disabled	Tristate*
Output		Enabled	Weak pull-up/pull-down
		Disabled	Tristate
Bidirectional / Tristate Buffer	E = 0 (input/tristate)	Enabled	Weak pull-up/pull-down*
		Disabled	Tristate*
	E = 1 (output)	Enabled	Weak pull-up/pull-down
		Disabled	Tristate

* *Internal core logic driven by this input/global buffer will be tied High as long as the device is in Flash*Freeze mode.*

- The device is reset upon exiting Flash*Freeze mode or internal state saving is not required.
- State saving is required, but data and clock management is performed external to the FPGA. In other words, incoming data is externally guaranteed and held valid prior to entering Flash*Freeze mode.

Type 2 Flash*Freeze mode is ideally suited for applications with the following design criteria:

- Entering Flash*Freeze mode is dependent on an internal or external signal in addition to the external FF pin.
- State saving is required and incoming data is not externally guaranteed valid.
- The designer wants to use his/her own Flash*Freeze management IP for clock and data management.
- The designer wants to use his/her own Flash*Freeze management logic for clock and data management.
- Internal housekeeping is required prior to entering Flash*Freeze mode. Housekeeping activities may include loading data to SRAM, system shutdown, completion of current task, or ensuring valid Flash*Freeze pin assertion.

There is no downside to type 2 mode, and Microsemi's Flash*Freeze management IP offers a very low tile count clock and data management solution. Microsemi's recommendation for most designs is to use type 2 Flash*Freeze mode with Flash*Freeze management IP.

Design Solutions

Clocks

- Microsemi recommends using a completely synchronous design in Type 2 mode with Flash*Freeze management IP cleanly gating all internal and external clocks. This will prevent narrow pulses upon entrance and exit from Flash*Freeze mode (Figure 2-5 on page 30).
- Upon entering Flash*Freeze mode, external clocks become tied off High, internal to the clock pin (unless hold state is used on IGLOO nano or IGLOO PLUS), and PLLs are turned off. Any clock that is externally Low will realize a Low to High transition internal to the device while entering Flash*Freeze. If clocks will float during Flash*Freeze mode, Microsemi recommends using the weak pull-up feature. If clocks will continue to drive the device during Flash*Freeze mode, the clock gating (filter) available in Flash*Freeze management IP can help to filter unwanted narrow clock pulses upon Flash*Freeze mode entry and exit.
- Clocks may continue to drive FPGA pins while the device is in Flash*Freeze mode, with virtually no power consumption. The weak pull-up/-down configuration will result in unnecessary power consumption if used in this scenario.
- Floating clocks can cause totem pole currents on the input I/O circuitry when the device is in active mode. If clocks are externally gated prior to entering Flash*Freeze mode, Microsemi recommends gating them to a known value (preferably '1', to avoid a possible narrow pulse upon Flash*Freeze mode exit), and not leaving them floating. However, during Flash*Freeze mode, all inputs and clocks are internally tied off to prevent totem pole currents, so they can be left floating.
- Upon exiting Flash*Freeze mode, the design must allow maximum acquisition time for the PLL to acquire the lock signal, and for a PLL clock to become active. If a PLL output clock is used as the primary clock for Flash*Freeze management IP, it is important to note that the clock gating circuit will only release other clocks after the primary PLL output clock becomes available.

Date	Changes	Page
51900147-2/5.07	In the following sentence, located in the "Flash*Freeze Mode" section, the bold text was changed from active high to active Low. The Flash*Freeze pin (**active low**) is a dedicated pin used to enter or exit Flash*Freeze mode directly, or alternatively the pin can be routed internally to the FPGA core to allow the user's logic to decide if it is safe to transition to this mode.	24
	Figure 2-2 • Flash*Freeze Mode Type 1 – Timing Diagram was updated.	25
	Information about ULSICC was added to the "Prototyping for IGLOO and ProASIC3L Devices Using ProASIC3" section.	2-21
51900147-1/3.07	In the "Flash*Freeze Mode" section, "active high" was changed to "active low."	24
	The "Prototyping for IGLOO and ProASIC3L Devices Using ProASIC3" section was updated with information concerning the Flash*Freeze pin.	2-21

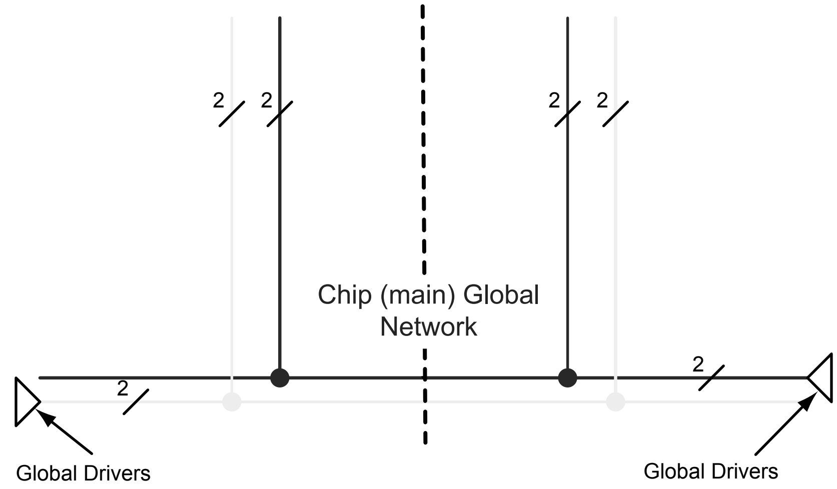


Figure 3-2 **• Simplified VersaNet Global Network (30 k gates and below)**

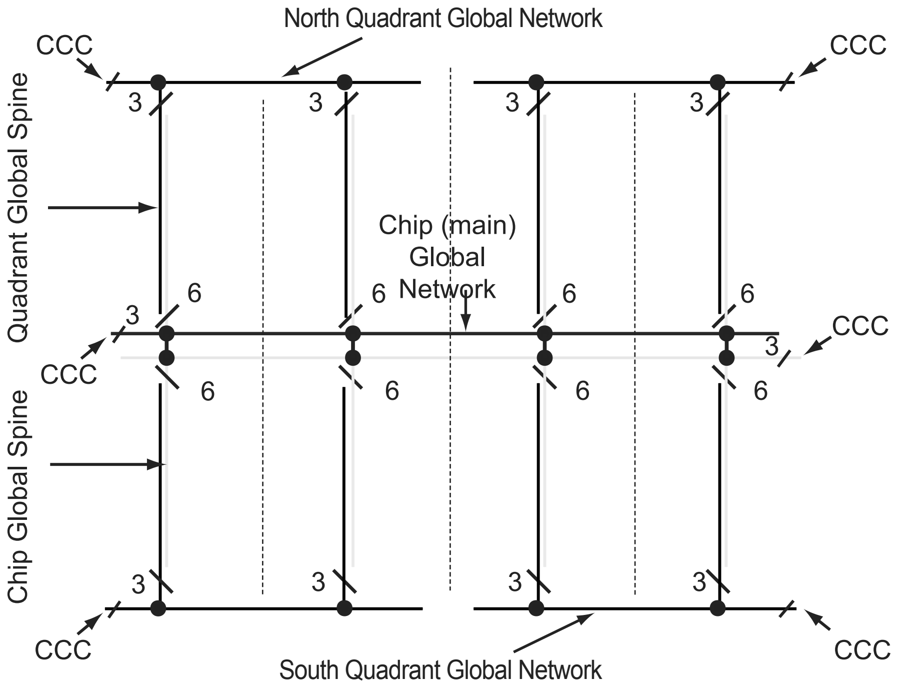

North Quadrant Global Network
CCC
CCC
Quadrant Global Spine
3
3
3
3
Chip (main) Global Network
6
6
6
6
3
CCC
CCC
3
6
6
6
6
Chip Global Spine
3
3
3
3
CCC
CCC
South Quadrant Global Network

Figure 3-3 **• Simplified VersaNet Global Network (60 k gates and above)**

Global Macro and Placement Selections

Low power flash devices provide the flexibility of choosing one of the three global input pad locations available to connect to a global / quadrant global network. For 60K gate devices and above, if the single-ended I/O standard is chosen, there is flexibility to choose one of the global input pads (the first, second, and fourth input). Once chosen, the other I/O locations are used as regular I/Os. If the differential I/O standard is chosen, the first and second inputs are considered as paired, and the third input is paired with a regular I/O. The user then has the choice of selecting one of the two sets to be used as the global input source. There is also the option to allow an internal clock signal to feed the global network. A multiplexer tree selects the appropriate global input for routing to the desired location. Note that the global I/O pads do not need to feed the global network; they can also be used as regular I/O pads.

Hardwired I/O Clock Source

Hardwired I/O refers to global input pins that are hardwired to the multiplexer tree, which directly accesses the global network. These global input pins have designated pin locations and are indicated with the I/O naming convention Gmn (m refers to any one of the positions where the global buffers is available, and n refers to any one of the three global input MUXes and the pin number of the associated global location, m). Choosing this option provides the benefit of directly connecting to the global buffers, which provides less delay. See Figure 3-11 for an example illustration of the connections, shown in red. If a CLKBUF macro is initiated, the clock input can be placed at one of nine dedicated global input pin locations: GmA0, GmA1, GmA2, GmB0, GmB1, GmB2, GmC0, GmC1, or GmC2. Note that the placement of the global will determine whether you are using chip global or quadrant global. For example, if the CLKBIF is placed in one of the GF pin locations, it will use the chip global network; if the CLKBIF is placed in one of the GA pin locations, it will use quadrant global network. This is shown in Figure 3-12 on page 65 and Figure 3-13 on page 65.

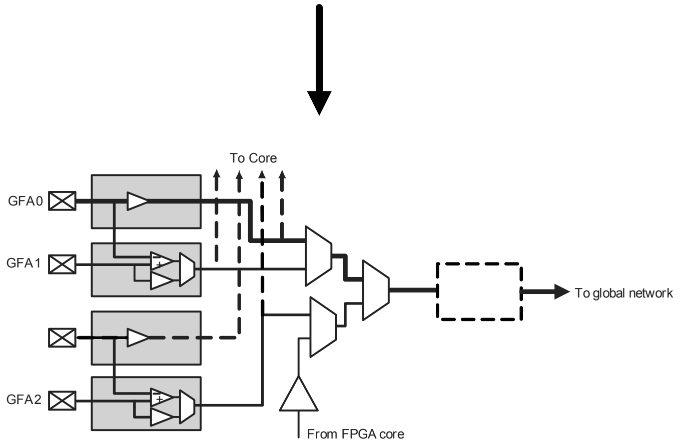


***Figure 3-11* • CLKBUF Macro**

YB and YC are identical to GLB and GLC, respectively, with the exception of a higher selectable final output delay. The SmartGen PLL Wizard will configure these outputs according to user specifications and can enable these signals with or without the enabling of Global Output Clocks.

The above signals can be enabled in the following output groupings in both internal and external feedback configurations of the static PLL:

- One output – GLA only
- Two outputs – GLA + (GLB and/or YB)
- Three outputs – GLA + (GLB and/or YB) + (GLC and/or YC)

PLL Macro Block Diagram

As illustrated, the PLL supports three distinct output frequencies from a given input clock. Two of these (GLB and GLC) can be routed to the B and C global network access, respectively, and/or routed to the device core (YB and YC).

There are five delay elements to support phase control on all five outputs (GLA, GLB, GLC, YB, and YC).

There are delay elements in the feedback loop that can be used to advance the clock relative to the reference clock.

The PLL macro reference clock can be driven in the following ways:

1. By an INBUF* macro to create a composite macro, where the I/O macro drives the global buffer (with programmable delay) using a hardwired connection. In this case, the I/O must be placed in one of the dedicated global I/O locations.
2. Directly from the FPGA core.
3. From an I/O that is routed through the FPGA regular routing fabric. In this case, users must instantiate a special macro, PLLINT, to differentiate from the hardwired I/O connection described earlier.

During power-up, the PLL outputs will toggle around the maximum frequency of the voltage-controlled oscillator (VCO) gear selected. Toggle frequencies can range from 40 MHz to 250 MHz. This will continue as long as the clock input (CLKA) is constant (HIGH or LOW). This can be prevented by LOW assertion of the POWERDOWN signal.

The visual PLL configuration in SmartGen, a component of the Libero SoC and Designer tools, will derive the necessary internal divider ratios based on the input frequency and desired output frequencies selected by the user.

PLL Core Specifications

PLL core specifications can be found in the DC and Switching Characteristics chapter of the appropriate family datasheet.

Loop Bandwidth

Common design practice for systems with a low-noise input clock is to have PLLs with small loop bandwidths to reduce the effects of noise sources at the output. Table 4-6 shows the PLL loop bandwidth, providing a measure of the PLL's ability to track the input clock and jitter.

Table 4-6 **• –3 dB Frequency of the PLL**

	Minimum **(T_a = +125°C, VCCA = 1.4 V)**	**Typical** **(T_a = +25°C, VCCA = 1.5 V)**	**Maximum** **(T_a = –55°C, VCCA = 1.6 V)**
–3 dB Frequency	15 kHz	25 kHz	45 kHz

PLL Core Operating Principles

This section briefly describes the basic principles of PLL operation. The PLL core is composed of a phase detector (PD), a low-pass filter (LPF), and a four-phase voltage-controlled oscillator (VCO). Figure 4-19 illustrates a basic single-phase PLL core with a divider and delay in the feedback path.

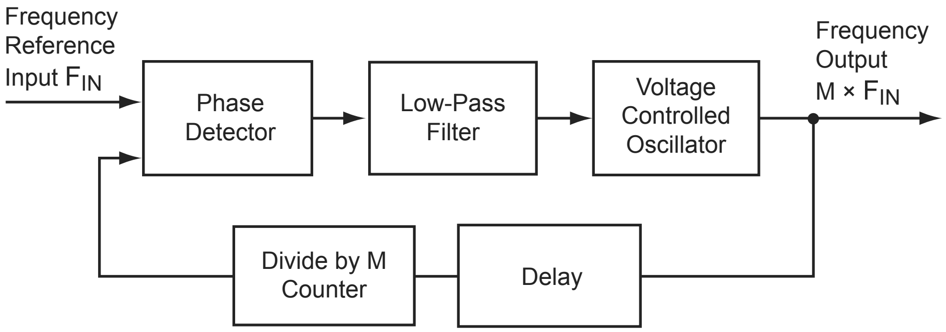

Frequency Reference Input F_{IN}
Phase Detector
Low-Pass Filter
Voltage Controlled Oscillator
Frequency Output M × F_{IN}
Divide by M Counter
Delay

Figure 4-19 **• Simplified PLL Core with Feedback Divider and Delay**

The PLL is an electronic servo loop that phase-aligns the PD feedback signal with the reference input. To achieve this, the PLL dynamically adjusts the VCO output signal according to the average phase difference between the input and feedback signals.

The first element is the PD, which produces a voltage proportional to the phase difference between its inputs. A simple example of a digital phase detector is an Exclusive-OR gate. The second element, the LPF, extracts the average voltage from the phase detector and applies it to the VCO. This applied voltage alters the resonant frequency of the VCO, thus adjusting its output frequency.

Consider Figure 4-19 with the feedback path bypassing the divider and delay elements. If the LPF steadily applies a voltage to the VCO such that the output frequency is identical to the input frequency, this steady-state condition is known as lock. Note that the input and output phases are also identical. The PLL core sets a LOCK output signal HIGH to indicate this condition.

Should the input frequency increase slightly, the PD detects the frequency/phase difference between its reference and feedback input signals. Since the PD output is proportional to the phase difference, the change causes the output from the LPF to increase. This voltage change increases the resonant frequency of the VCO and increases the feedback frequency as a result. The PLL dynamically adjusts in this manner until the PD senses two phase-identical signals and steady-state lock is achieved. The opposite (decreasing PD output signal) occurs when the input frequency decreases.

Now suppose the feedback divider is inserted in the feedback path. As the division factor M (shown in Figure 4-20 on page 101) is increased, the average phase difference increases. The average phase

Date	Changes	Page
v1.4 (December 2008)	The"CCC Support in Microsemi's Flash Devices" section was updated to include IGLOO nano and ProASIC3 nano devices.	79
	Figure 4-2 • CCC Options: Global Buffers with No Programmable Delay was revised to add the CLKBIBUF macro.	80
	The description of the reference clock was revised in Table 4-2 • Input and Output Description of the CLKDLY Macro.	81
	Figure 4-7 • Clock Input Sources (30 k gates devices and below) is new. Figure 4-8 • Clock Input Sources Including CLKBUF, CLKBUF_LVDS/LVPECL, and CLKINT (60 k gates devices and above) applies to 60 k gate devices and above.	88
	The "IGLOO and ProASIC3" section was updated to include information for IGLOO nano devices.	89
	A note regarding Fusion CCCs was added to Figure 4-9 • Illustration of Hardwired I/O (global input pins) Usage for IGLOO and ProASIC3 devices 60 k Gates and Larger and the name of the figure was changed from Figure 4-8 • Illustration of Hardwired I/O (global input pins) Usage. Figure 4-10 • Illustration of Hardwired I/O (global input pins) Usage for IGLOO and ProASIC3 devices 30 k Gates and Smaller is new.	90
	Table 4-5 • Number of CCCs by Device Size and Package was updated to include IGLOO nano and ProASIC3 nano devices. Entries were added to note differences for the CS81, CS121, and CS201 packages.	94
	The "Clock Conditioning Circuits without Integrated PLLs" section was rewritten.	95
	The "IGLOO and ProASIC3 CCC Locations" section was updated for nano devices.	97
	Figure 4-13 • CCC Locations in the 15 k and 30 k Gate Devices was deleted.	4-20
v1.3 (October 2008)	This document was updated to include Fusion and RT ProASIC3 device information. Please review the document very carefully.	N/A
	The "CCC Support in Microsemi's Flash Devices" section was updated.	79
	In the "Global Buffer with Programmable Delay" section, the following sentence was changed from: "In this case, the I/O must be placed in one of the dedicated global I/O locations." To "In this case, the software will automatically place the dedicated global I/O in the appropriate locations."	80
	Figure 4-4 • CCC Options: Global Buffers with PLL was updated to include OADIVRST and OADIVHALF.	83
	In Figure 4-6 • CCC with PLL Block "fixed delay" was changed to "programmable delay".	83
	Table 4-3 • Input and Output Signals of the PLL Block was updated to include OADIVRST and OADIVHALF descriptions.	84
	Table 4-8 • Configuration Bit Descriptions for the CCC Blocks was updated to include configuration bits 88 to 81. Note 2 is new. In addition, the description for bit <76:74> was updated.	106
	Table 4-16 • Fusion Dynamic CCC Clock Source Selection and Table 4-17 • Fusion Dynamic CCC NGMUX Configuration are new.	110
	Table 4-18 • Fusion Dynamic CCC Division by Half Configuration and Table 4-19 • Configuration Bit <76:75> / VCOSEL<2:1> Selection for All Families are new.	111

FlashROM Security

Low power flash devices have an on-chip Advanced Encryption Standard (AES) decryption core, combined with an enhanced version of the Microsemi 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 FlashROM content. These 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 low power flash devices works via a FlashLock security Pass Key mechanism, where the user locks or unlocks the device with a user-defined key. Refer to the "Security in Low Power Flash Devices" section on page 301.

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. To gain access to the FPGA, the device first must be unlocked using the correct Pass Key. During programming of the FlashROM 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 FlashROM and FPGA core content. The FlashLock Pass Key is not stored in the FlashROM.

Low power flash 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 5-5 shows this flow diagram.

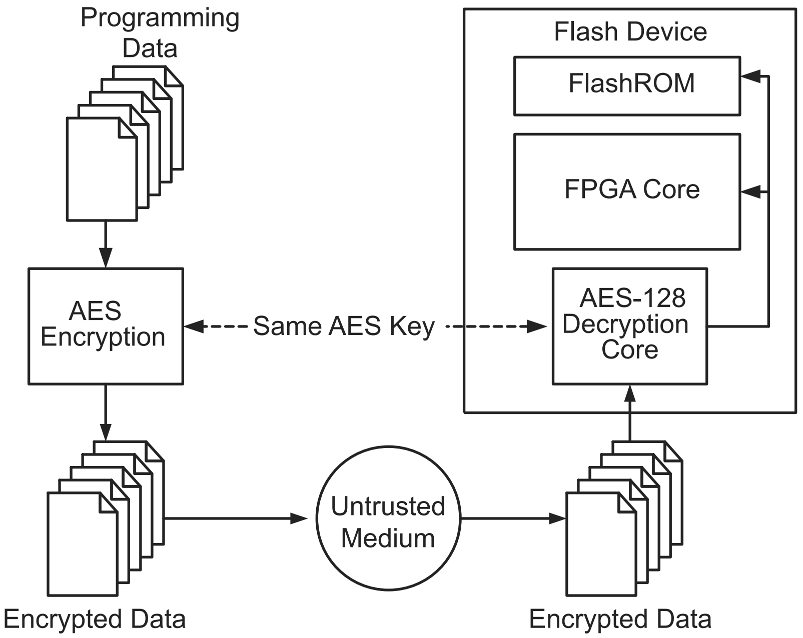
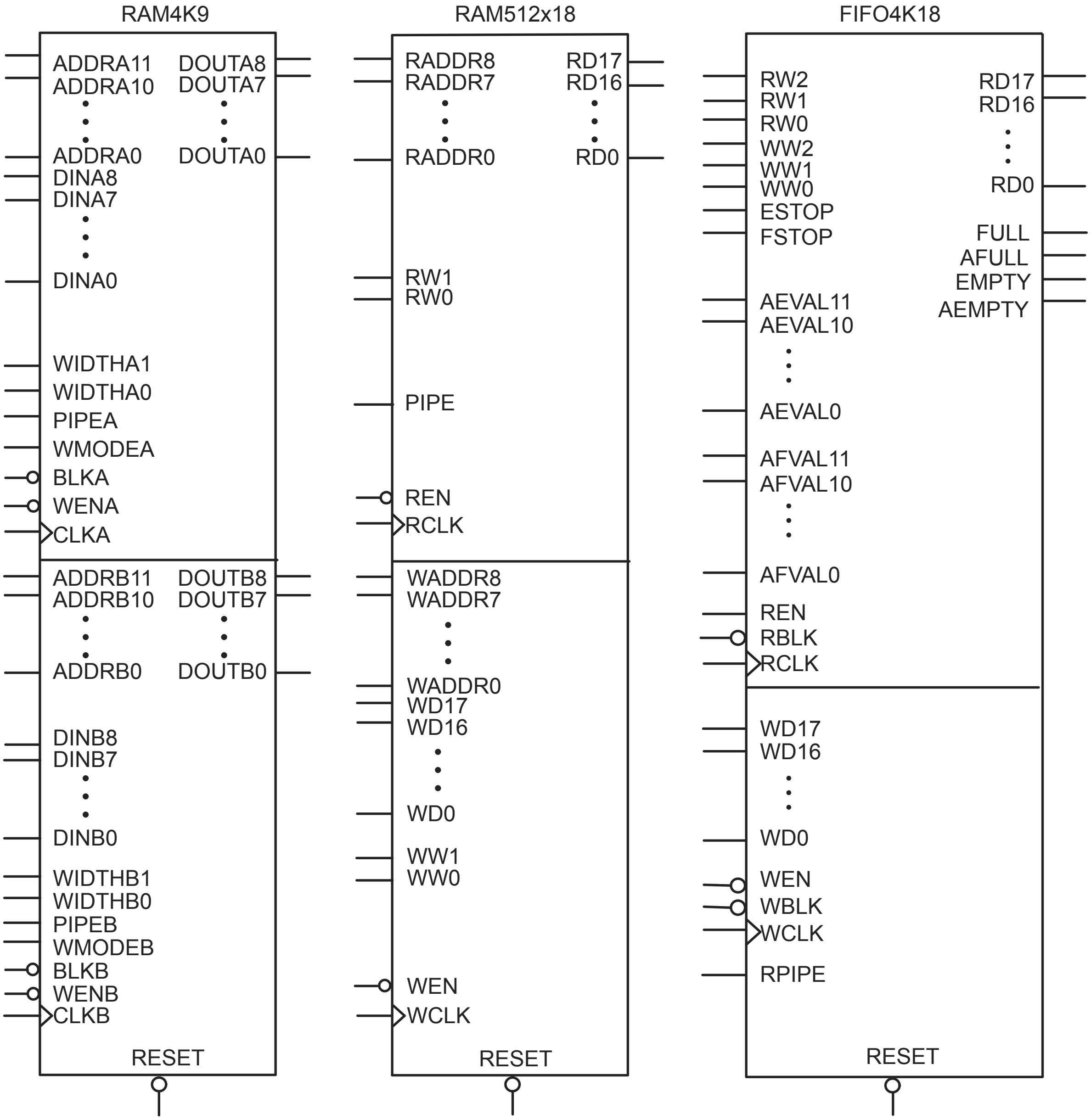


Figure 5-5 **• Programming FlashROM Using AES**



Notes:

1. *Automotive ProASIC3 devices restrict RAM4K9 to a single port or to dual ports with the same clock 180° out of phase (inverted) between clock pins. In single-port mode, inputs to port B should be tied to ground to prevent errors during compile. This warning applies only to automotive ProASIC3 parts of certain revisions and earlier. Contact Technical Support at soc_tech@microsemi.com for information on the revision number for a particular lot and date code.*
2. *For FIFO4K18, the same clock 180° out of phase (inverted) between clock pins should be used.*

***Figure 6-3* • Supported Basic RAM Macros**

GTL+ (Gunning Transceiver Logic Plus)

This is an enhanced version of GTL that has defined slew rates and higher voltage levels. It requires a differential amplifier input buffer and an open-drain output buffer. Even though the output is open-drain, VCCI must be connected to either 2.5 V or 3.3 V. The reference voltage (VREF) is 1 V.

Differential Standards

These standards require two I/Os per signal (called a "signal pair"). Logic values are determined by the potential difference between the lines, not with respect to ground. This is why differential drivers and receivers have much better noise immunity than single-ended standards. The differential interface standards offer higher performance and lower power consumption than their single-ended counterparts. Two I/O pins are used for each data transfer channel. Both differential standards require resistor termination.

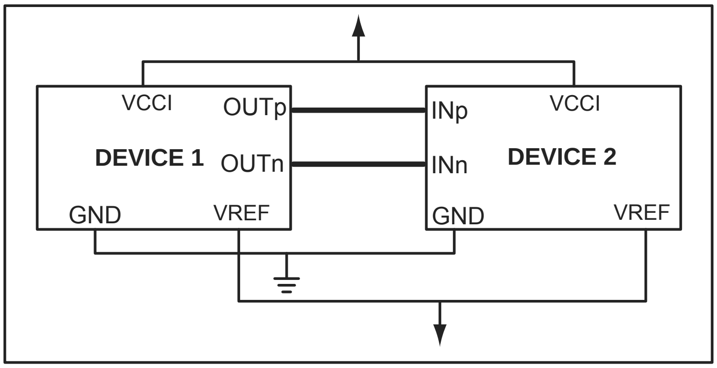


Figure 7-7 • **Differential Topology**

LVPECL (Low-Voltage Positive Emitter Coupled Logic)

LVPECL requires that one data bit be carried through two signal lines; therefore, two pins are needed per input or output. It also requires external resistor termination. The voltage swing between the two signal lines is approximately 850 mV. When the power supply is +3.3 V, it is commonly referred to as Low-Voltage PECL (LVPECL). Refer to the device datasheet for the full implementation of the LVPECL transmitter and receiver.

LVDS (Low-Voltage Differential Signal)

LVDS is a moderate-speed differential signaling system, in which the transmitter generates two different voltages that are compared at the receiver. LVDS uses a differential driver connected to a terminated receiver through a constant-impedance transmission line. It requires that one data bit be carried through two signal lines; therefore, the user will need two pins per input or output. It also requires external resistor termination. The voltage swing between the two signal lines is approximately 350 mV. VCCI is 2.5 V. Low power flash devices contain dedicated circuitry supporting a high-speed LVDS standard that has its own user specification. Refer to the device datasheet for the full implementation of the LVDS transmitter and receiver.

B-LVDS/M-LVDS

Bus LVDS (B-LVDS) refers to bus interface circuits based on LVDS technology. Multipoint LVDS (M-LVDS) specifications extend the LVDS standard to high-performance multipoint bus applications. Multidrop and multipoint bus configurations may contain any combination of drivers, receivers, and transceivers. Microsemi LVDS drivers provide the higher drive current required by B-LVDS and M-LVDS to accommodate the loading. The driver requires series terminations for better signal quality and to control voltage swing. Termination is also required at both ends of the bus, since the driver can be located anywhere on the bus. These configurations can be implemented using TRIBUF_LVDS and BIBUF_LVDS macros along with appropriate terminations. Multipoint designs using Microsemi LVDS macros can achieve up to 200 MHz with a maximum of 20 loads. A sample application is given in Figure 7-8. The input and output buffer delays are available in the LVDS sections in the datasheet.

Volume Programming Services

Device Type Supported: Flash and Antifuse

Once the design is stable for applications with large production volumes, preprogrammed devices can be purchased. Table 11-2 describes the volume programming services.

Table 11-2 **• Volume Programming Services**

Programmer	Vendor	Availability
In-House Programming	Microsemi	Contact Microsemi Sales
Distributor Programming Centers	Memec Unique	Contact Distribution
Independent Programming Centers	Various	Contact Vendor

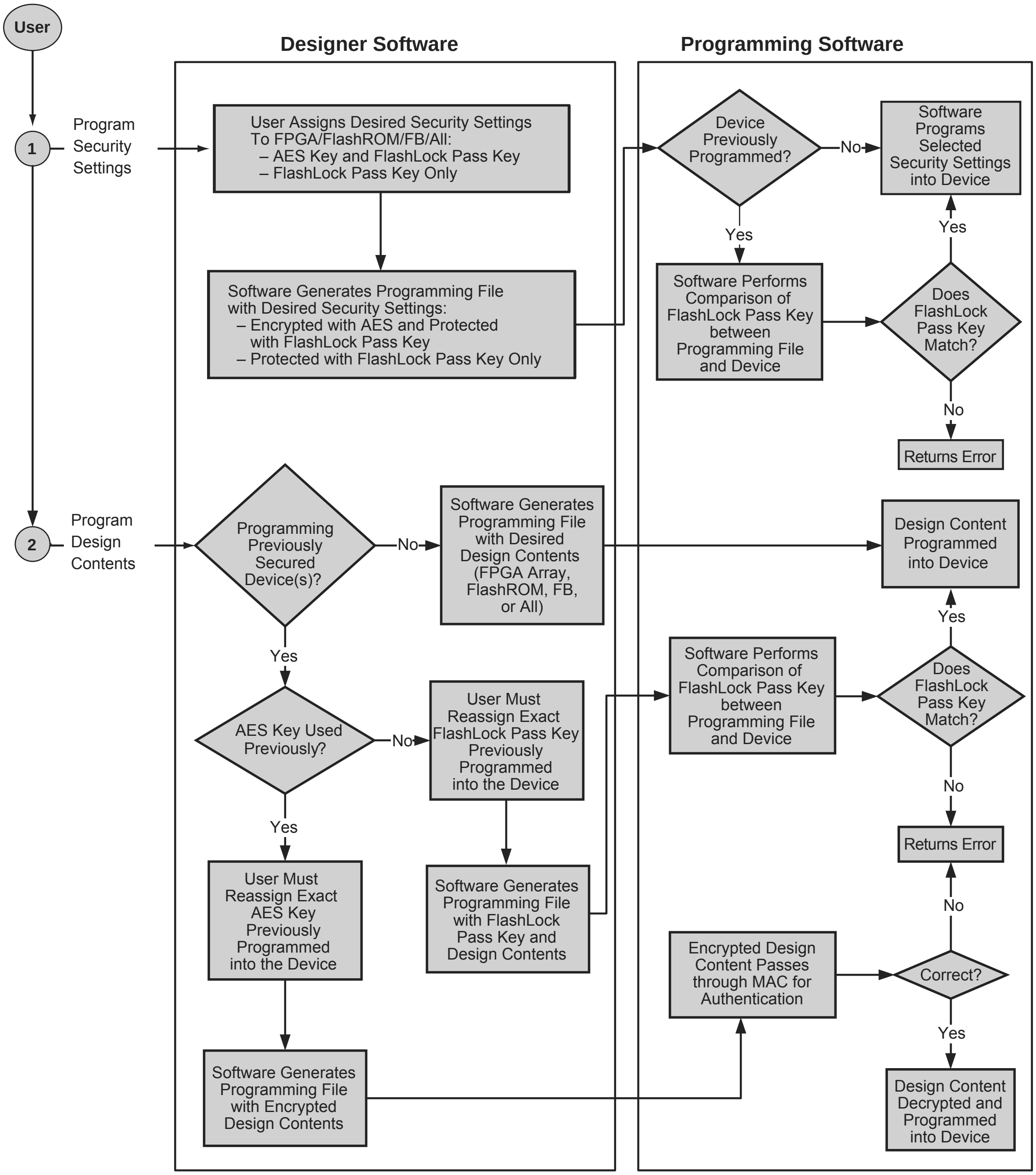
Advantages: As programming is outsourced, this solution is easier to implement than creating a substantial in-house programming capability. As programming houses specialize in large-volume programming, this is often the most cost-effective solution.

Limitations: There are some logistical issues with the use of a programming service provider, such as the transfer of programming files and the approval of First Articles. By definition, the programming file must be released to a third-party programming house. Nondisclosure agreements (NDAs) can be signed to help ensure data protection; however, for extremely security-conscious designs, this may not be an option.

- Microsemi In-House Programming

  When purchasing Microsemi devices in volume, IHP can be requested as part of the purchase. If this option is chosen, there is a small cost adder for each device programmed. Each device is marked with a special mark to distinguish it from blank parts. Programming files for the design will be sent to Microsemi. Sample parts with the design programmed, First Articles, will be returned for customer approval. Once approval of First Articles has been received, Microsemi will proceed with programming the remainder of the order. To request Microsemi IHP, contact your local Microsemi representative.
- Distributor Programming Centers

  If purchases are made through a distributor, many distributors will provide programming for their customers. Consult with your preferred distributor about this option.


User
Designer Software
Programming Software
1
Program Security Settings
User Assigns Desired Security Settings To FPGA/FlashROM/FB/All:
– AES Key and FlashLock Pass Key
– FlashLock Pass Key Only
Software Generates Programming File with Desired Security Settings:
– Encrypted with AES and Protected with FlashLock Pass Key
– Protected with FlashLock Pass Key Only
Device Previously Programmed?
No
Software Programs Selected Security Settings into Device
Yes
Software Performs Comparison of FlashLock Pass Key between Programming File and Device
Yes
Does FlashLock Pass Key Match?
No
Returns Error
2
Program Design Contents
Programming Previously Secured Device(s)?
No
Software Generates Programming File with Desired Design Contents (FPGA Array, FlashROM, FB, or All)
Design Content Programmed into Device
Yes
Yes
Software Performs Comparison of FlashLock Pass Key between Programming File and Device
Does FlashLock Pass Key Match?
AES Key Used Previously?
No
User Must Reassign Exact FlashLock Pass Key Previously Programmed into the Device
No
Returns Error
Yes
User Must Reassign Exact AES Key Previously Programmed into the Device
Software Generates Programming File with FlashLock Pass Key and Design Contents
No
Encrypted Design Content Passes through MAC for Authentication
Correct?
Yes
Software Generates Programming File with Encrypted Design Contents
Design Content Decrypted and Programmed into Device

Note: If programming the Security Header only, just perform sub-flow 1.
If programming design content only, just perform sub-flow 2.

***Figure 12-9* • Security Programming Flows**

Generating Programming Files

Generation of the Programming File in a Trusted Environment—Application 1

As discussed in the "Application 1: Trusted Environment" section on page 309, in a trusted environment, the user can choose to program the device with plaintext bitstream content. It is possible to use plaintext for programming even when the FlashLock Pass Key option has been selected. In this application, it is not necessary to employ AES encryption protection. For AES encryption settings, refer to the next sections.

The generated programming file will include the security setting (if selected) and the plaintext programming file content for the FPGA array, FlashROM, and/or FBs. These options are indicated in Table 12-2 and Table 12-3.

Table 12-2 **• IGLOO and ProASIC3 Plaintext Security Options, No AES**

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

Table 12-3 **• Fusion Plaintext Security Options**

Security Protection	FlashROM Only	FPGA Core Only	FB Core Only	All
No AES / no FlashLock	✓	✓	✓	✓
FlashLock	✓	✓	✓	✓
AES and FlashLock	–	–	–	–

Note: For all instructions, the programming of Flash Blocks refers to Fusion only.

For this scenario, generate the programming file as follows:

1. Select the **Silicon features to be programmed** (Security Settings, FPGA Array, FlashROM, Flash Memory Blocks), as shown in Figure 12-10 on page 314 and Figure 12-11 on page 314. Click **Next**.

   If **Security Settings** is selected (i.e., the FlashLock security Pass Key feature), an additional dialog will be displayed to prompt you to select the security level setting. If no security setting is selected, you will be directed to Step 3.

Related Documents

User's Guides

FlashPro User's Guide

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

List of Changes

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

Date	Changes	Page
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
v1.5 (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.	304
v1.4 (December 2008)	IGLOO nano and ProASIC3 nano devices were added to Table 12-1 • Flash-Based FPGAs.	302
v1.3 (October 2008)	The "Security Support in Flash-Based Devices" section was revised to include new families and make the information more concise.	302
v1.2 (June 2008)	The following changes were made to the family descriptions in Table 12-1 • Flash-Based FPGAs: • ProASIC3L was updated to include 1.5 V. • The number of PLLs for ProASIC3E was changed from five to six.	302
v1.1 (March 2008)	The chapter was updated to include the IGLOO PLUS family and information regarding 15 k gate devices.	N/A
	The "IGLOO Terminology" section and "ProASIC3 Terminology" section are new.	302

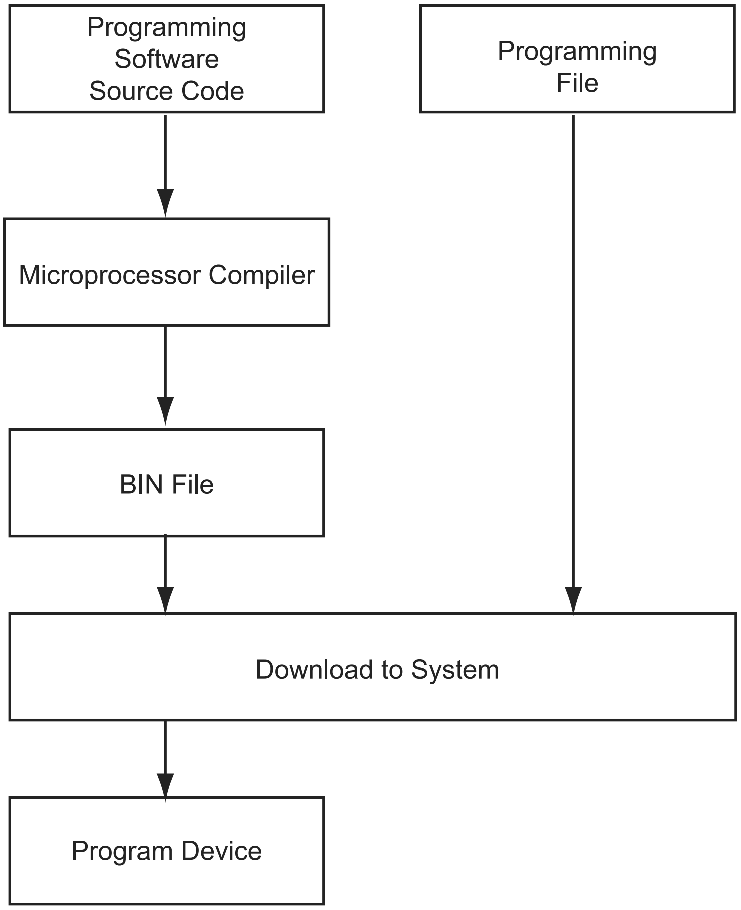


Figure 15-3 • **MCU FPGA Programming Model**

FlashROM

Microsemi low power flash devices have 1 kbit of user-accessible, nonvolatile, FlashROM on-chip. This nonvolatile FlashROM can be programmed along with the core or on its own using the standard IEEE 1532 JTAG programming interface.

The FlashROM is architected as eight pages of 128 bits. Each page can be individually programmed (erased and written). Additionally, on-chip AES security decryption can be used selectively to load data securely into the FlashROM (e.g., over public or private networks, such as the Internet). Refer to the "FlashROM in Microsemi's Low Power Flash Devices" section on page 133.

List of Changes

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

Date	Changes	Page
August 2012	In the "Boundary Scan Chain" section, the reference made to the datasheet for pull-up/-down recommendations was changed to mention TCK and TRST pins rather than TDO and TCK pins. TDO is an output, so no pull resistor is needed (SAR 35937).	359
	The "Advanced Boundary Scan Register Settings" section is new (SAR 38432).	361
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
	Table 16-3 • TRST and TCK Pull-Down Recommendations was revised to add VJTAG at 1.2 V.	360
v1.4 (December 2008)	IGLOO nano and ProASIC3 nano devices were added to Table 16-1 • Flash-Based FPGAs.	358
v1.3 (October 2008)	The "Boundary Scan Support in Low Power Devices" section was revised to include new families and make the information more concise.	359
v1.2 (June 2008)	The following changes were made to the family descriptions in Table 16-1 • Flash-Based FPGAs: • ProASIC3L was updated to include 1.5 V. • The number of PLLs for ProASIC3E was changed from five to six.	358
v1.1 (March 2008)	The chapter was updated to include the IGLOO PLUS family and information regarding 15 k gate devices.	N/A
	The "IGLOO Terminology" section and "ProASIC3 Terminology" section are new.	358