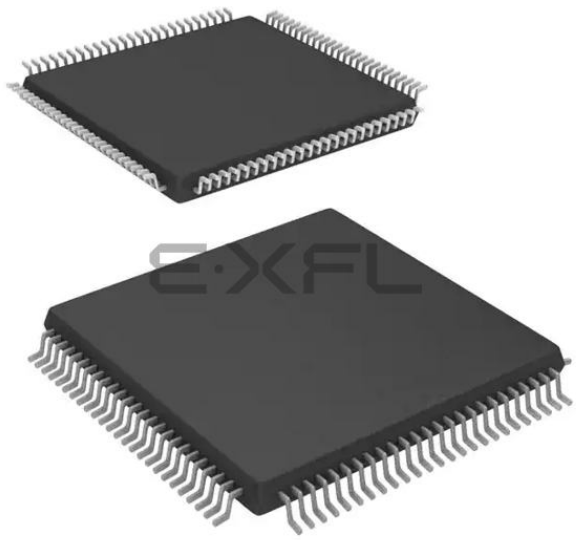


Welcome to E-XFL.COM

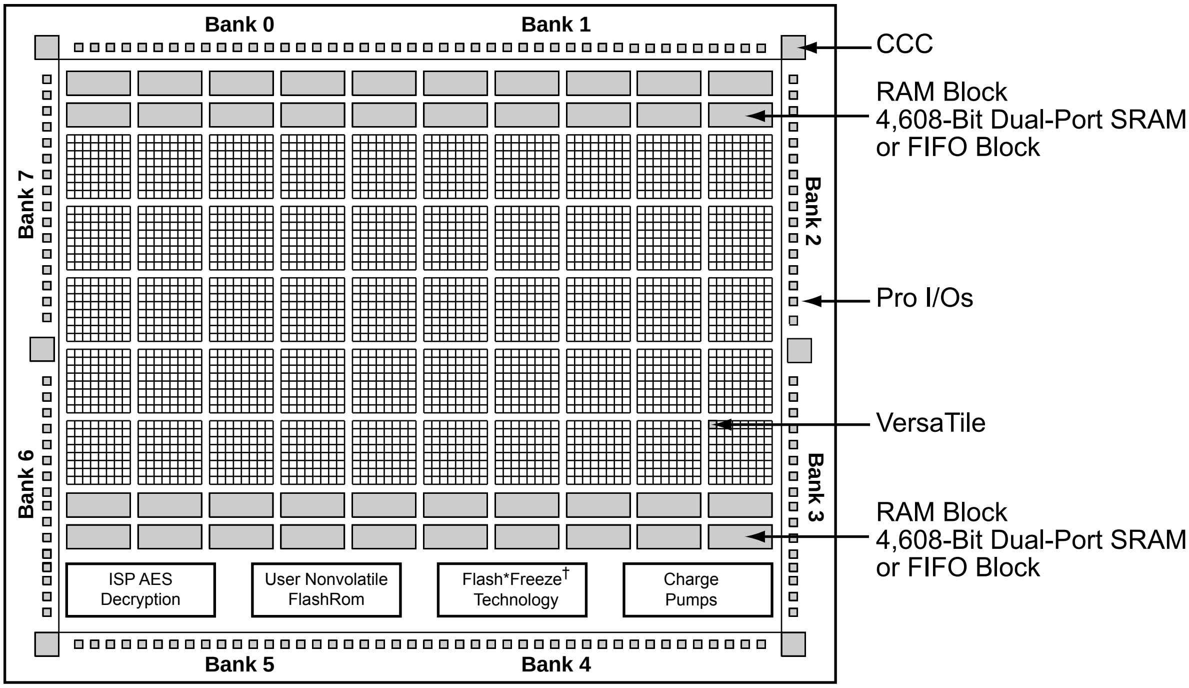
Understanding Embedded - FPGAs (Field Programmable Gate Array)

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	36864
Number of I/O	68
Number of Gates	250000
Voltage - Supply	1.425V ~ 1.575V
Mounting Type	Surface Mount
Operating Temperature	-40°C ~ 100°C (TJ)
Package / Case	100-TQFP
Supplier Device Package	100-VQFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/a3pn250-zvqg100i



Note: *Flash*Freeze technology only applies to IGLOOe devices.*

Figure 1-7 • **IGLOOe and ProASIC3E Device Architecture Overview (AGLE600 device is shown)**

I/O State of Newly Shipped Devices

Devices are shipped from the factory with a test design in the device. The power-on switch for VCC is OFF by default in this test design, so I/Os are tristated by default. Tristated means the I/O is not actively driven and floats. The exact value cannot be guaranteed when it is floating. Even in simulation software, a tristate value is marked as unknown. Due to process variations and shifts, tristated I/Os may float toward High or Low, depending on the particular device and leakage level.

If there is concern regarding the exact state of unused I/Os, weak pull-up/pull-down should be added to the floating I/Os so their state is controlled and stabilized.

Routing Architecture

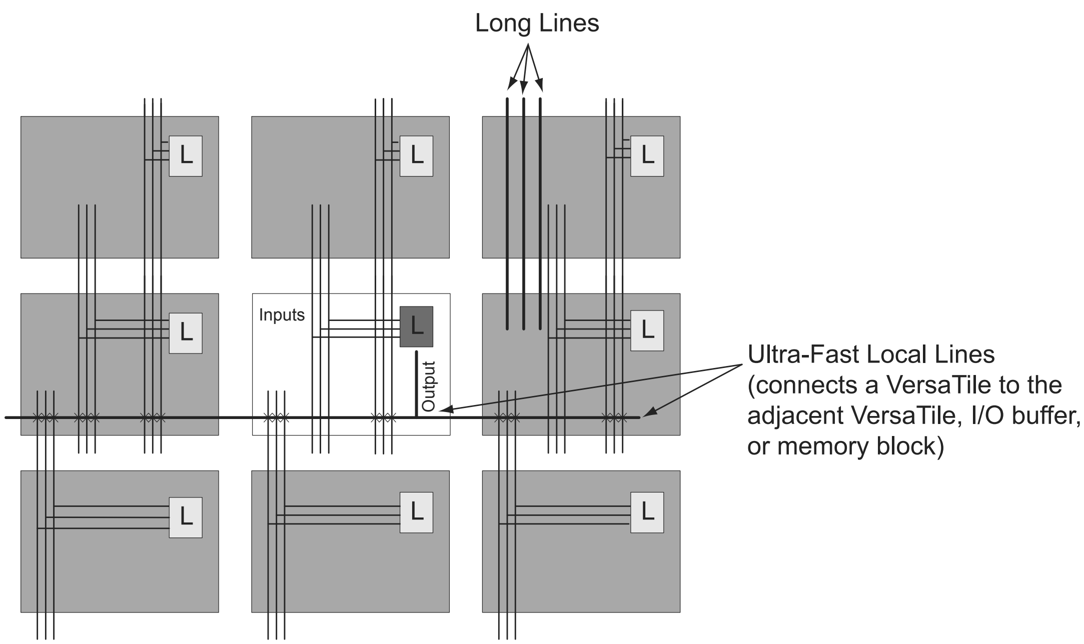
The routing structure of low power flash devices is designed to provide high performance through a flexible four-level hierarchy of routing resources: ultra-fast local resources; efficient long-line resources; high-speed, very-long-line resources; and the high-performance VersaNet networks.

The ultra-fast local resources are dedicated lines that allow the output of each VersaTile to connect directly to every input of the eight surrounding VersaTiles (Figure 1-10). The exception to this is that the SET/CLR input of a VersaTile configured as a D-flip-flop is driven only by the VersaTile global network.

The efficient long-line resources provide routing for longer distances and higher-fanout connections. These resources vary in length (spanning one, two, or four VersaTiles), run both vertically and horizontally, and cover the entire device (Figure 1-11 on page 19). Each VersaTile can drive signals onto the efficient long-line resources, which can access every input of every VersaTile. Routing software automatically inserts active buffers to limit loading effects.

The high-speed, very-long-line resources, which span the entire device with minimal delay, are used to route very long or high-fanout nets: length ±12 VersaTiles in the vertical direction and length ±16 in the horizontal direction from a given core VersaTile (Figure 1-12 on page 19). Very long lines in low power flash devices have been enhanced over those in previous ProASIC families. This provides a significant performance boost for long-reach signals.

The high-performance VersaNet global networks are low-skew, high-fanout nets that are accessible from external pins or internal logic. These nets are typically used to distribute clocks, resets, and other high-fanout nets requiring minimum skew. The VersaNet networks are implemented as clock trees, and signals can be introduced at any junction. These can be employed hierarchically, with signals accessing every input of every VersaTile. For more details on VersaNets, refer to the "Global Resources in Low Power Flash Devices" section on page 31.



Note: *Input to the core cell for the D-flip-flop set and reset is only available via the VersaNet global network connection.*

Figure 1-10 **• Ultra-Fast Local Lines Connected to the Eight Nearest Neighbors**

Table 3-2 • Chip Global Pin Name

I/O Type	Beginning of I/O Name	Notes
Single-Ended	GFAO/IOuxwByVz GFA1/IOuxwByVz GFA2/IOuxwByVz	Only one of the I/Os can be directly connected to a chip global at a time.
	GFBO/IOuxwByVz GFB1/IOuxwByVz GFB2/IOuxwByVz	Only one of the I/Os can be directly connected to a chip global at a time.
	GFC0/IOuxwByVz GFC1/IOuxwByVz GFC2/IOuxwByVz	Only one of the I/Os can be directly connected to a chip global at a time.
	GCAO/IOuxwByVz GCA1/IOuxwByVz GCA2/IOuxwByVz	Only one of the I/Os can be directly connected to a chip global at a time.
	GCBO/IOuxwByVz GCB1/IOuxwByVz GCB2/IOuxwByVz	Only one of the I/Os can be directly connected to a chip global at a time.
	GCC0/IOuxwByVz GCC1/IOuxwByVz GCC2/IOuxwByVz	Only one of the I/Os can be directly connected to a chip global at a time.
Differential I/O Pairs	GFAO/IOuxwByVz GFA1/IOuxwByVz	The output of the different pair will drive the chip global.
	GFBO/IOuxwByVz GFB1/IOuxwByVz	The output of the different pair will drive the chip global.
	GFCO/IOuxwByVz GFC1/IOuxwByVz	The output of the different pair will drive the chip global.
	GCAO/IOuxwByVz GCA1/IOuxwByVz	The output of the different pair will drive the chip global.
	GCBO/IOuxwByVz GCB1/IOuxwByVz	The output of the different pair will drive the chip global.
	GCCO/IOuxwByVz GCC1/IOuxwByVz	The output of the different pair will drive the chip global.

Note: Only one of the I/Os can be directly connected to a quadrant at a time.

Table 3-3 **• Quadrant Global Pin Name**

I/O Type	Beginning of I/O Name	Notes
Single-Ended	GAAO/IOuxwByVz GAA1/IOuxwByVz GAA2/IOuxwByVz	Only one of the I/Os can be directly connected to a quadrant global at a time
	GABO/IOuxwByVz GAB1/IOuxwByVz GAB2/IOuxwByVz	Only one of the I/Os can be directly connected to a quadrant global at a time.
	GAC0/IOuxwByVz GAC1/IOuxwByVz GAC2/IOuxwByVz	Only one of the I/Os can be directly connected to a quadrant global at a time.
	GBAO/IOuxwByVz GBA1/IOuxwByVz GBA2/IOuxwByVz	Only one of the I/Os can be directly connected to a global at a time.
	GBBO/IOuxwByVz GBB1/IOuxwByVz GBB2/IOuxwByVz	Only one of the I/Os can be directly connected to a global at a time.
	GBC0/IOuxwByVz GBC1/IOuxwByVz GBC2/IOuxwByVz	Only one of the I/Os can be directly connected to a global at a time.
	GDAO/IOuxwByVz GDA1/IOuxwByVz GDA2/IOuxwByVz	Only one of the I/Os can be directly connected to a global at a time.
	GDBO/IOuxwByVz GDB1/IOuxwByVz GDB2/IOuxwByVz	Only one of the I/Os can be directly connected to a global at a time.
	GDC0/IOuxwByVz GDC1/IOuxwByVz GDC2/IOuxwByVz	Only one of the I/Os can be directly connected to a global at a time.
	GEAO/IOuxwByVz GEA1/IOuxwByVz GEA2/IOuxwByVz	Only one of the I/Os can be directly connected to a global at a time.
	GEBO/IOuxwByVz GEB1/IOuxwByVz GEB2/IOuxwByVz	Only one of the I/Os can be directly connected to a global at a time.
	GEC0/IOuxwByVz GEC1/IOuxwByVz GEC2/IOuxwByVz	Only one of the I/Os can be directly connected to a global at a time.

Note: Only one of the I/Os can be directly connected to a quadrant at a time.

Clock Aggregation Architecture

This clock aggregation feature allows a balanced clock tree, which improves clock skew. The physical regions for clock aggregation are defined from left to right and shift by one spine. For chip global networks, there are three types of clock aggregation available, as shown in Figure 3-10:

- Long lines that can drive up to four adjacent spines (A)
- Long lines that can drive up to two adjacent spines (B)
- Long lines that can drive one spine (C)

There are three types of clock aggregation available for the quadrant spines, as shown in Figure 3-10:

- I/Os or local resources that can drive up to four adjacent spines
- I/Os or local resources that can drive up to two adjacent spines
- I/Os or local resources that can drive one spine

As an example, A3PE600 and AFS600 devices have twelve spine locations: T1, T2, T3, T4, T5, T6, B1, B2, B3, B4, B5, and B6. Table 3-7 shows the clock aggregation you can have in A3PE600 and AFS600.

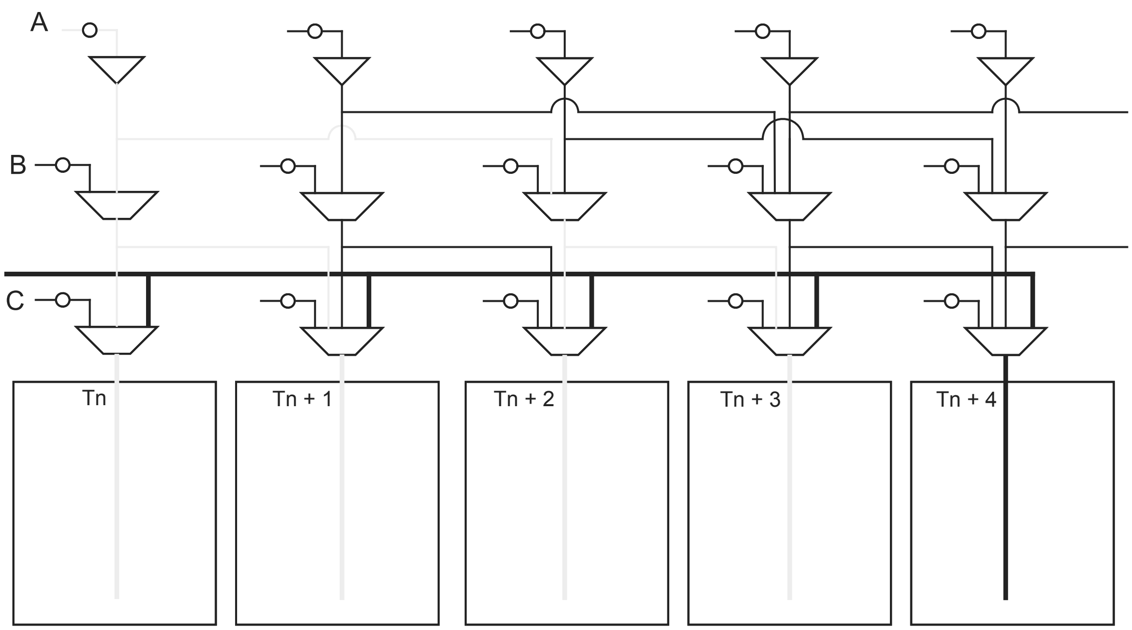


Figure 3-10 **• Four Spines Aggregation**

Table 3-7 **• Spine Aggregation in A3PE600 or AFS600**

Clock Aggregation	Spine
1 spine	T1, T2, T3, T4, T5, T6, B1, B2, B3, B4, B5, B6
2 spines	T1:T2, T2:T3, T3:T4, T4:T5, T5:T6, B1:B2, B2:B3, B3:B4, B4:B5, B5:B6
4 spines	B1:B4, B2:B5, B3:B6, T1:T4, T2:T5, T3:T6

The clock aggregation for the quadrant spines can cross over from the left to right quadrant, but not from top to bottom. The quadrant spine assignment T1:T4 is legal, but the quadrant spine assignment T1:B1 is not legal. Note that this clock aggregation is hardwired. You can always assign signals to spine T1 and B2 by instantiating a buffer, but this may add skew in the signal.

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
	Notes were added where appropriate to point out that IGLOO nano and ProASIC3 nano devices do not support differential inputs (SAR 21449).	N/A
	The "Global Architecture" section and "VersaNet Global Network Distribution" section were revised for clarity (SARs 20646, 24779).	31, 33
	The "I/O Banks and Global I/Os" section was moved earlier in the document, renamed to "Chip and Quadrant Global I/Os", and revised for clarity. Figure 3-4 • Global Connections Details, Figure 3-6 • Global Inputs, Table 3-2 • Chip Global Pin Name, and Table 3-3 • Quadrant Global Pin Name are new (SARs 20646, 24779).	35
	The "Clock Aggregation Architecture" section was revised (SARs 20646, 24779).	41
	Figure 3-7 • Chip Global Aggregation was revised (SARs 20646, 24779).	43
	The "Global Macro and Placement Selections" section is new (SARs 20646, 24779).	48
v1.4 (December 2008)	The "Global Architecture" section was updated to include 10 k devices, and to include information about VersaNet global support for IGLOO nano devices.	31
	The Table 3-1 • Flash-Based FPGAs was updated to include IGLOO nano and ProASIC3 nano devices.	32
	The "VersaNet Global Network Distribution" section was updated to include 10 k devices and to note an exception in global lines for nano devices.	33
	Figure 3-2 • Simplified VersaNet Global Network (30 k gates and below) is new.	34
	The "Spine Architecture" section was updated to clarify support for 10 k and nano devices.	41
	Table 3-4 • Globals/Spines/Rows for IGLOO and ProASIC3 Devices was updated to include IGLOO nano and ProASIC3 nano devices.	41
	The figure in the CLKBUF_LVDS/LVPECL row of Table 3-8 • Clock Macros was updated to change CLKBIBUF to CLKBUF.	46
v1.3 (October 2008)	A third bullet was added to the beginning of the "Global Architecture" section: In Fusion devices, the west CCC also contains a PLL core. In the two larger devices (AFS600 and AFS1500), the west and east CCCs each contain a PLL.	31
	The "Global Resource Support in Flash-Based Devices" section was revised to include new families and make the information more concise.	32
	Table 3-4 • Globals/Spines/Rows for IGLOO and ProASIC3 Devices was updated to include A3PE600/L in the device column.	41
	Table note 1 was revised in Table 3-9 • I/O Standards within CLKBUF to include AFS600 and AFS1500.	47
v1.2 (June 2008)	The following changes were made to the family descriptions in Table 3-1 • Flash-Based FPGAs: • ProASIC3L was updated to include 1.5 V. • The number of PLLs for ProASIC3E was changed from five to six.	32

This section outlines the following device information: CCC features, PLL core specifications, functional descriptions, software configuration information, detailed usage information, recommended board-level considerations, and other considerations concerning global networks in low power flash devices.

Clock Conditioning Circuits with Integrated PLLs

Each of the CCCs with integrated PLLs includes the following:

- 1 PLL core, which consists of a phase detector, a low-pass filter, and a four-phase voltage-controlled oscillator
- 3 global multiplexer blocks that steer signals from the global pads and the PLL core onto the global networks
- 6 programmable delays and 1 fixed delay for time advance/delay adjustments
- 5 programmable frequency divider blocks to provide frequency synthesis (automatically configured by the SmartGen macro builder tool)

Clock Conditioning Circuits without Integrated PLLs

There are two types of simplified CCCs without integrated PLLs in low power flash devices.

1. The simplified CCC with programmable delays, which is composed of the following:
   – 3 global multiplexer blocks that steer signals from the global pads and the programmable delay elements onto the global networks
   – 3 programmable delay elements to provide time delay adjustments
2. The simplified CCC (referred to as CCC-GL) without programmable delay elements, which is composed of the following:
   – A global multiplexer block that steer signals from the global pads onto the global networks

Phase Adjustment

The four phases available (0, 90, 180, 270) are phases with respect to VCO (PLL output). The VCO is divided to achieve the user's CCC required output frequency (GLA, YB/GLB, YC/GLC). The division happens after the selection of the VCO phase. The effective phase shift is actually the VCO phase shift divided by the output divider. This is why the visual CCC shows both the actual achievable phase and more importantly the actual delay that is equivalent to the phase shift that can be achieved.

Dynamic PLL Configuration

The CCCs can be configured both statically and dynamically.

In addition to the ports available in the Static CCC, the Dynamic CCC has the dynamic shift register signals that enable dynamic reconfiguration of the CCC. With the Dynamic CCC, the ports CLKB and CLKC are also exposed. All three clocks (CLKA, CLKB, and CLKC) can be configured independently.

The CCC block is fully configurable. The following two sources can act as the CCC configuration bits.

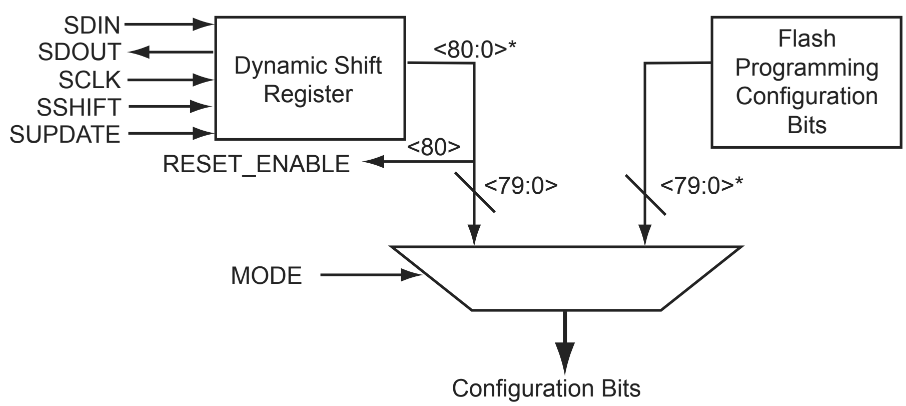
Flash Configuration Bits

The flash configuration bits are the configuration bits associated with programmed flash switches. These bits are used when the CCC is in static configuration mode. Once the device is programmed, these bits cannot be modified. They provide the default operating state of the CCC.

Dynamic Shift Register Outputs

This source does not require core reprogramming and allows core-driven dynamic CCC reconfiguration. When the dynamic register drives the configuration bits, the user-defined core circuit takes full control over SDIN, SDOUT, SCLK, SSHIFT, and SUPDATE. The configuration bits can consequently be dynamically changed through shift and update operations in the serial register interface. Access to the logic core is accomplished via the dynamic bits in the specific tiles assigned to the PLLs.

Figure 4-21 illustrates a simplified block diagram of the MUX architecture in the CCCs.



*Note: *For Fusion, bit <88:81> is also needed.*

Figure 4-21 **• The CCC Configuration MUX Architecture**

The selection between the flash configuration bits and the bits from the configuration register is made using the MODE signal shown in Figure 4-21. If the MODE signal is logic HIGH, the dynamic shift register configuration bits are selected. There are 81 control bits to configure the different functions of the CCC.

5 – FlashROM in Microsemi's Low Power Flash Devices

Introduction

The Fusion, IGLOO, and ProASIC3 families of low power flash-based devices have a dedicated nonvolatile FlashROM memory of 1,024 bits, which provides a unique feature in the FPGA market. The FlashROM can be read, modified, and written using the JTAG (or UJTAG) interface. It can be read but not modified from the FPGA core. Only low power flash devices contain on-chip user nonvolatile memory (NVM).

Architecture of User Nonvolatile FlashROM

Low power flash devices have 1 kbit of user-accessible nonvolatile flash memory on-chip that can be read from the FPGA core fabric. The FlashROM is arranged in eight banks of 128 bits (16 bytes) during programming. The 128 bits in each bank are addressable as 16 bytes during the read-back of the FlashROM from the FPGA core. Figure 5-1 shows the FlashROM logical structure.

The FlashROM can only be programmed via the IEEE 1532 JTAG port. It cannot be programmed directly from the FPGA core. When programming, each of the eight 128-bit banks can be selectively reprogrammed. The FlashROM can only be reprogrammed on a bank boundary. Programming involves an automatic, on-chip bank erase prior to reprogramming the bank. The FlashROM supports synchronous read. The address is latched on the rising edge of the clock, and the new output data is stable after the falling edge of the same clock cycle. For more information, refer to the timing diagrams in the DC and Switching Characteristics chapter of the appropriate datasheet. The FlashROM can be read on byte boundaries. The upper three bits of the FlashROM address from the FPGA core define the bank being accessed. The lower four bits of the FlashROM address from the FPGA core define which of the 16 bytes in the bank is being accessed.

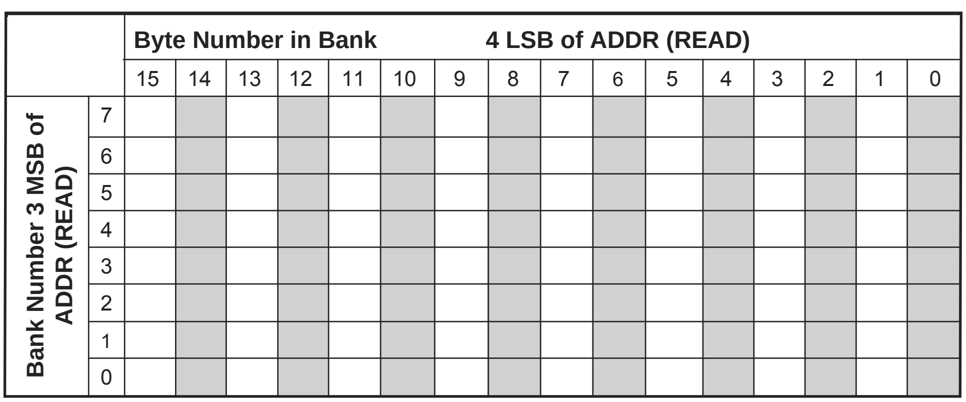
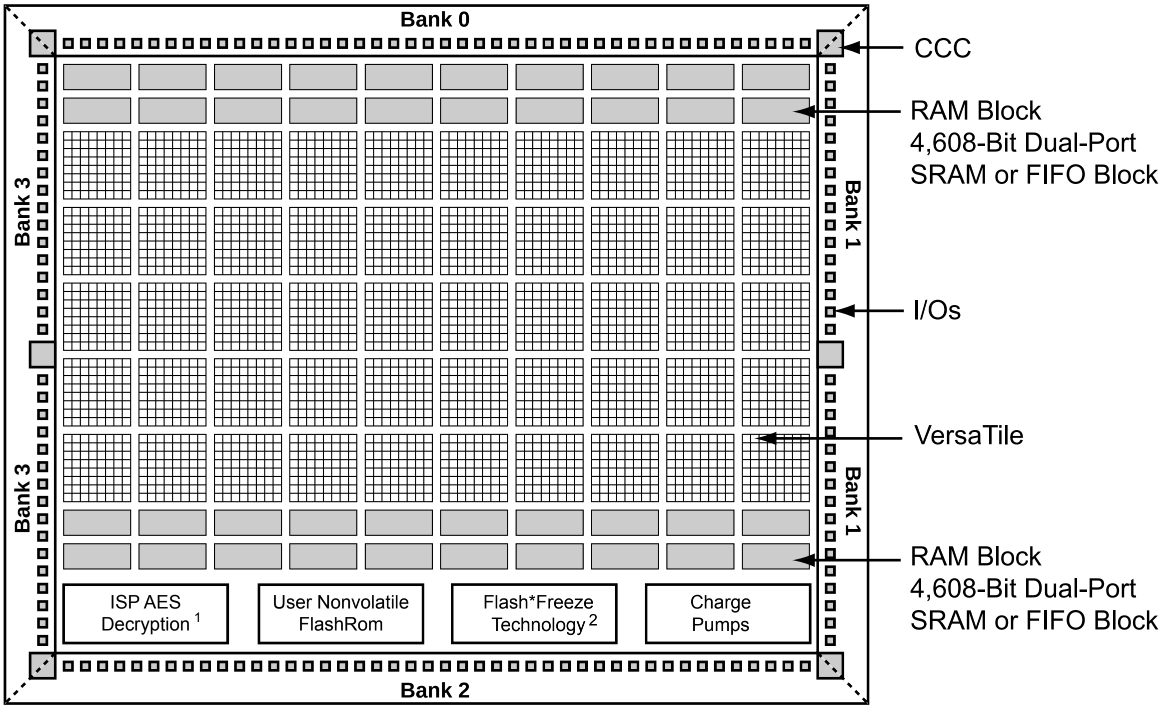


Figure 5-1 • **FlashROM Architecture**



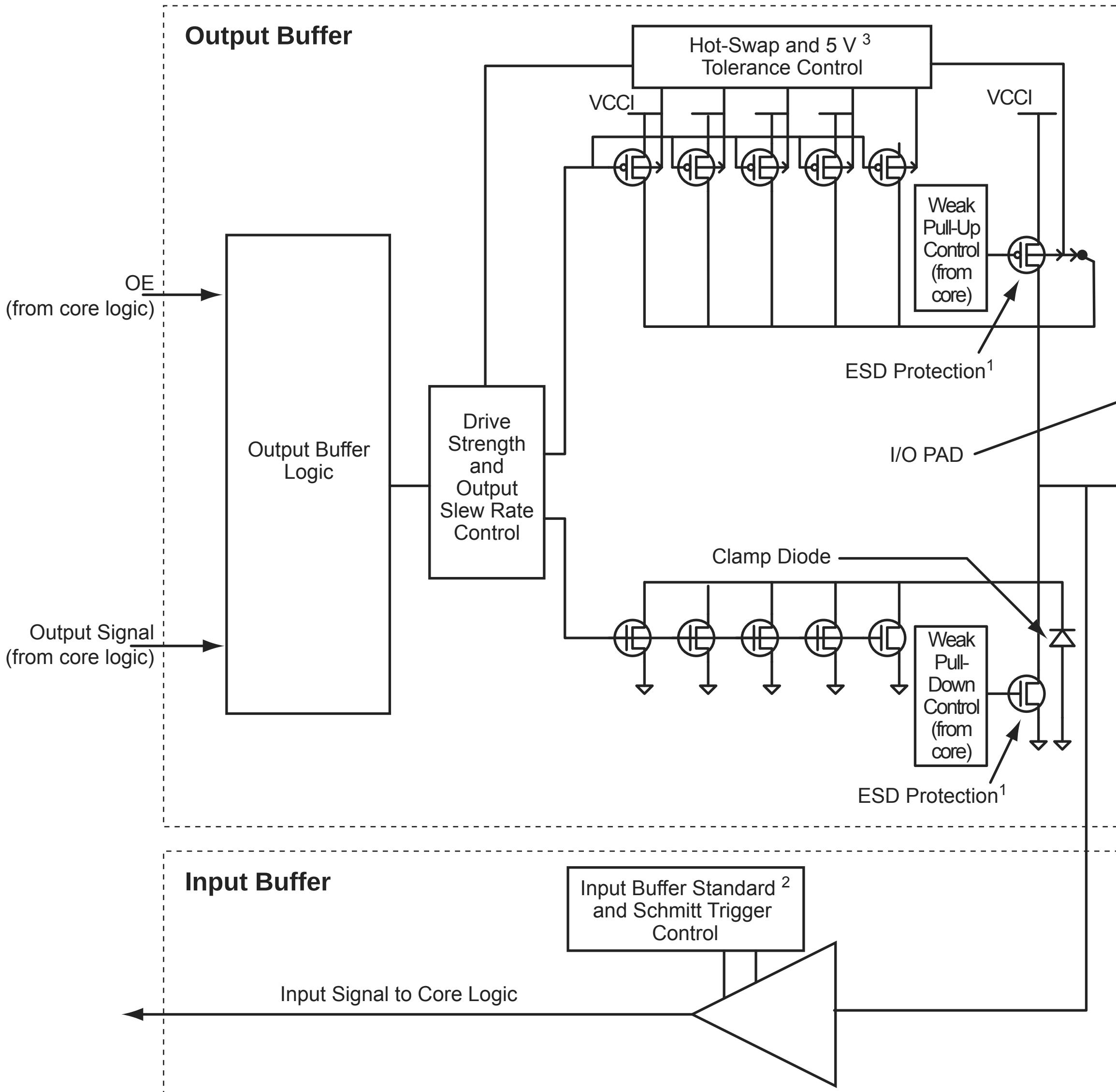
Notes:

1. *AES decryption not supported in 30 k gate devices and smaller.*
2. *Flash*Freeze is supported in all IGLOO devices and the ProASIC3L devices.*

Figure 6-1 **• IGLOO and ProASIC3 Device Architecture Overview**

I/O Bank Structure

Low power flash device I/Os are divided into multiple technology banks. The number of banks is device-dependent, supporting two, three, or four banks. Each bank has its own V_{CCI} power supply pin. Refer to Figure 7-2 on page 160 for more information.



Notes:

1. *All NMOS transistors connected to the I/O pad serve as ESD protection.*
2. *See Table 7-2 on page 162 for available I/O standards.*
3. *5 V tolerance requires external resistor.*

Figure 7-3 **• Simplified I/O Buffer Circuitry**

Table 7-8 • **Hot-Swap Level 1**

Description	Cold-swap
Power Applied to Device	No
Bus State	–
Card Ground Connection	–
Device Circuitry Connected to Bus Pins	–
Example Application	System and card with Microsemi FPGA chip are powered down, and the card is plugged into the system. Then the power supplies are turned on for the system but not for the FPGA on the card.
Compliance of nano Devices	Compliant

Table 7-9 • **Hot-Swap Level 2**

Description	Hot-swap while reset
Power Applied to Device	Yes
Bus State	Held in reset state
Card Ground Connection	Reset must be maintained for 1 ms before, during, and after insertion/removal.
Device Circuitry Connected to Bus Pins	–
Example Application	In the PCI hot-plug specification, reset control circuitry isolates the card busses until the card supplies are at their nominal operating levels and stable.
Compliance of nano Devices	Compliant

- If one of the registers has a PRE pin, all the other registers that are candidates for combining in the I/O must have a PRE pin.
- If one of the registers has neither a CLR nor a PRE pin, all the other registers that are candidates for combining must have neither a CLR nor a PRE pin.
- If the clear or preset pins are present, they must have the same polarity.
- If the clear or preset pins are present, they must be driven by the same signal (net).

3. For single-tile devices (10 k, 15 k, and 20 k): Registers connected to an I/O on the Output and Output Enable pins must have the same clock function (both CLR and CLK are shared among all registers):
   - Both the Output and Output Enable registers must not have an E pin (clock enable).
4. For dual-tile devices (60 k, 125 k, and 250 k): Registers connected to an I/O on the Output and Output Enable pins must have the same clock and enable function:
   - Both the Output and Output Enable registers must have an E pin (clock enable), or none at all.
   - If the E pins are present, they must have the same polarity. The CLK pins must also have the same polarity.

   In some cases, the user may want registers to be combined with the input of a bibuf while maintaining the output as-is. This can be achieved by using PDC commands as follows:

```
set_io <signal name> -REGISTER yes ------register will combine
set_preserve <signal name> ----register will not combine
```

Weak Pull-Up and Weak Pull-Down Resistors

nano devices support optional weak pull-up and pull-down resistors on each I/O pin. When the I/O is pulled up, it is connected to the V_{CCI} of its corresponding I/O bank. When it is pulled down, it is connected to GND. Refer to the datasheet for more information.

For low power applications and when using IGLOO nano devices, configuration of the pull-up or pull-down of the I/O can be used to set the I/O to a known state while the device is in Flash*Freeze mode. Refer to "Flash*Freeze Technology and Low Power Modes" in an applicable FPGA fabric user's guide for more information.

The Flash*Freeze (FF) pin cannot be configured with a weak pull-down or pull-up I/O attribute, as the signal needs to be driven at all times.

Output Slew Rate Control

The slew rate is the amount of time an input signal takes to get from logic LOW to logic HIGH or vice versa.

It is commonly defined as the propagation delay between 10% and 90% of the signal's voltage swing. Slew rate control is available for the output buffers of low power flash devices. The output buffer has a programmable slew rate for both HIGH-to-LOW and LOW-to-HIGH transitions.

The slew rate can be implemented by using a PDC command (Table 7-5 on page 163), setting it "High" or "Low" in the I/O Attribute Editor in Designer, or instantiating a special I/O macro. The default slew rate value is "High."

Microsemi recommends the high slew rate option to minimize the propagation delay. This high-speed option may introduce noise into the system if appropriate signal integrity measures are not adopted. Selecting a low slew rate reduces this kind of noise but adds some delays in the system. Low slew rate is recommended when bus transients are expected.

Output Drive

The output buffers of nano devices can provide multiple drive strengths to meet signal integrity requirements. The LVTTL and LVCMOS (except 1.2 V LVCMOS) standards have selectable drive strengths.

Drive strength should also be selected according to the design requirements and noise immunity of the system.

Output Buffers

There are two variations: Regular and Special.

If the **Regular** variation is selected, only the Width (1 to 128) needs to be entered. The default value for Width is 1.

The **Special** variation has Width, Technology, Output Drive, and Slew Rate options.

Bidirectional Buffers

There are two variations: Regular and Special.

The **Regular** variation has Enable Polarity (Active High, Active Low) in addition to the Width option.

The **Special** variation has Width, Technology, Output Drive, Slew Rate, and Resistor Pull-Up/-Down options.

Tristate Buffers

Same as Bidirectional Buffers.

DDR

There are eight variations: DDR with Regular Input Buffers, Special Input Buffers, Regular Output Buffers, Special Output Buffers, Regular Tristate Buffers, Special Tristate Buffers, Regular Bidirectional Buffers, and Special Bidirectional Buffers.

These variations resemble the options of the previous I/O macro. For example, the Special Input Buffers variation has Width, Technology, Voltage Level, and Resistor Pull-Up/-Down options. DDR is not available on IGLOO PLUS devices.

4. Once the desired configuration is selected, click the **Generate** button. The Generate Core window opens (Figure 8-4).
5. Enter a name for the macro. Click **OK**. The core will be generated and saved to the appropriate location within the project files (Figure 8-5 on page 191).

Figure 8-4 **• Generate Core Window**

6. Instantiate the I/O macro in the top-level code.

   The user must instantiate the DDR_REG or DDR_OUT macro in the design. Use SmartGen to generate both these macros and then instantiate them in your top level. To combine the DDR macros with the I/O, the following rules must be met:

Application 3: Nontrusted Environment—Field Updates/Upgrades

Programming or reprogramming of devices may occur at remote locations. Reconfiguration of devices in consumer products/equipment through public networks is one example. Typically, the remote system is already programmed with particular design contents. When design update (FPGA array contents update) and/or data upgrade (FlashROM and/or FB contents upgrade) is necessary, an updated programming file with AES encryption can be generated, sent across public networks, and transmitted to the remote system. Reprogramming can then be done using this AES-encrypted programming file, providing easy and secure field upgrades. Low power flash devices support this secure ISP using AES. The detailed flow for this application is shown in Figure 11-8. Refer to the "Microprocessor Programming of Microsemi's Low Power Flash Devices" chapter of an appropriate FPGA fabric user's guide for more information.

To prepare devices for this scenario, the user can initially generate a programming file with the available security setting options. This programming file is programmed into the devices before shipment. During the programming file generation step, the user has the option of making the security settings permanent or not. In situations where no changes to the security settings are necessary, the user can select this feature in the software to generate the programming file with permanent security settings. Microsemi recommends that the programming file use encryption with an AES key, especially when ISP is done via public domain.

For example, if the designer wants to use an AES key for the FPGA array and the FlashROM, **Permanent** needs to be chosen for this setting. At first, the user chooses the options to use an AES key for the FPGA array and the FlashROM, and then chooses **Permanently lock the security settings**. A unique AES key is chosen. Once this programming file is generated and programmed to the devices, the AES key is permanently stored in the on-chip memory, where it is secured safely. The devices are sent to distant locations for the intended application. When an update is needed, a new programming file must be generated. The programming file must use the same AES key for encryption; otherwise, the authentication will fail and the file will not be programmed in the device.

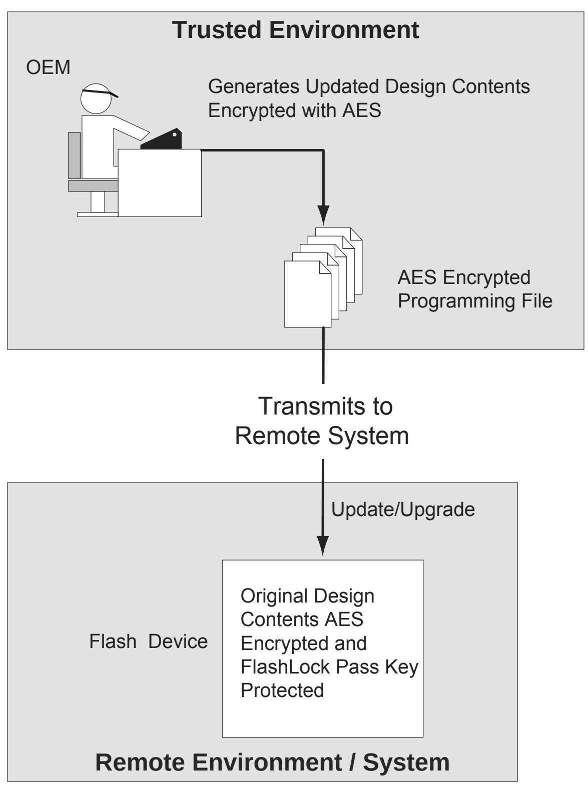


Figure 11-8 • **Application 3: Nontrusted Environment—Field Updates/Upgrades**

Programming Voltage (VPUMP) and VJTAG

Low-power flash devices support on-chip charge pumps, and therefore require only a single 3.3 V programming voltage for the VPUMP pin during programming. When the device is not being programmed, the VPUMP pin can be left floating or can be tied (pulled up) to any voltage between 0 V and 3.6 V[2]. During programming, the target board or the FlashPro4/3/3X programmer can provide VPUMP. FlashPro4/3/3X is capable of supplying VPUMP to a single device. If more than one device is to be programmed using FlashPro4/3/3X on a given board, FlashPro4/3/3X should not be relied on to supply the VPUMP voltage. A FlashPro4/3/3X programmer is not capable of providing reliable VJTAG voltage. The board must supply VJTAG voltage to the device and the VJTAG pin of the programmer header must be connected to the device VJTAG pin. Microsemi recommends that VPUMP[3] and VJTAG power supplies be kept separate with independent filtering capacitors rather than supplying them from a common rail. Refer to the "Board-Level Considerations" section on page 271 for capacitor requirements.

Low power flash device I/Os support a bank-based, voltage-supply architecture that simultaneously supports multiple I/O voltage standards (Table 12-2). By isolating the JTAG power supply in a separate bank from the user I/Os, low power flash devices provide greater flexibility with supply selection and simplify power supply and printed circuit board (PCB) design. The JTAG pins can be run at any voltage from 1.5 V to 3.3 V (nominal). Microsemi recommends that TCK be tied to GND through a 200 ohm to 1 Kohm resistor. This prevents a possible totempole current on the input buffer stage. For TDI, TMS, and TRST pins, the devices provide an internal nominal 10 Kohm pull-up resistor. During programming, all I/O pins, except for JTAG interface pins, are tristated and weakly pulled up to VCCI. This isolates the part and prevents the signals from floating. The JTAG interface pins are driven by the FlashPro4/3/3X during programming, including the TRST pin, which is driven HIGH.

Table 12-2 **• Power Supplies**

Power Supply	Programming Mode	Current during Programming
VCC	1.2 V / 1.5 V	< 70 mA
VCCI	1.2 V / 1.5 V / 1.8 V / 2.5 V / 3.3 V (bank-selectable)	I/Os are weakly pulled up.
VJTAG	1.2 V / 1.5 V / 1.8 V / 2.5 V / 3.3 V	< 20 mA
VPUMP	3.15 V to 3.45 V	< 80 mA

Note: *All supply voltages should be at 1.5 V or higher, regardless of the setting during normal operation, except for IGLOO nano, where 1.2 V VCC and VJTAG programming is allowed.*

Nonvolatile Memory (NVM) Programming Voltage

SmartFusion and Fusion devices need stable VCCNVM/VCCENVM[3] (1.5 V power supply to the embedded nonvolatile memory blocks) and VCCOSC/VCCROSC[4] (3.3 V power supply to the integrated RC oscillator). The tolerance of VCCNVM/VCCENVM is ± 5% and VCCOSC/VCCROSC is ± 5%.

Unstable supply voltage on these pins can cause an NVM programming failure due to NVM page corruption. The NVM page can also be corrupted if the NVM reset pin has noise. This signal must be tied off properly.

Microsemi recommends installing the following capacitors[5] on the VCCNVM/VCCENVM and VCCOSC/VCCROSC pins:

- Add one bypass capacitor of 10 µF for each power supply plane followed by an array of decoupling capacitors of 0.1 µF.
- Add one 0.1 µF capacitor near each pin.

2. *During sleep mode in IGLOO devices connect VPUMP to GND.*
3. *VPUMP has to be quiet for successful programming. Therefore VPUMP must be separate and required capacitors must be installed close to the FPGA VPUMP pin.*
4. *VCCROSC is for SmartFusion.*
5. *The capacitors cannot guarantee reliable operation of the device if the board layout is not done properly.*

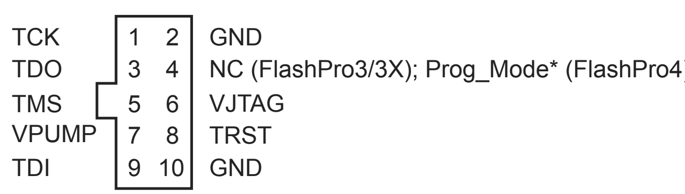
ISP Programming Header Information

The FlashPro4/3/3X programming cable connector can be connected with a 10-pin, 0.1"-pitch programming header. The recommended programming headers are manufactured by AMP (103310-1) and 3M (2510-6002UB). If you have limited board space, you can use a compact programming header manufactured by Samtec (FTSH-105-01-L-D-K). Using this compact programming header, you are required to order an additional header adapter manufactured by Microsemi SoC Products Group (FP3-10PIN-ADAPTER-KIT).

Existing ProASIC^{PLUS} family customers who are using the Samtec Small Programming Header (FTSH-113-01-L-D-K) and are planning to migrate to IGLOO or ProASIC3 devices can also use FP3-10PIN-ADAPTER-KIT.

***Table 12-3* • Programming Header Ordering Codes**

Manufacturer	Part Number	Description
AMP	103310-1	10-pin, 0.1"-pitch cable header (right-angle PCB mount angle)
3M	2510-6002UB	10-pin, 0.1"-pitch cable header (straight PCB mount angle)
Samtec	FTSH-113-01-L-D-K	Small programming header supported by FlashPro and Silicon Sculptor
Samtec	FTSH-105-01-L-D-K	Compact programming header
Samtec	FFSD-05-D-06.00-01-N	10-pin cable with 50 mil pitch sockets; included in FP3-10PIN-ADAPTER-KIT.
Microsemi	FP3-10PIN-ADAPTER-KIT	Transition adapter kit to allow FP3 to be connected to a micro 10-pin header (50 mil pitch). Includes a 6 inch Samtec FFSD-05-D-06.00-01-N cable in the kit. The transition adapter board was previously offered as FP3-26PIN-ADAPTER and includes a 26-pin adapter for design transitions from ProASIC^{PLUS} based boards to ProASIC3 based boards.



*Note: *Prog_Mode on FlashPro4 is an output signal that goes High during device programming and returns to Low when programming is complete. This signal can be used to drive a system to provide a 1.5 V programming signal to IGLOO nano, ProASIC3L, and RT ProASIC3 devices that can run with 1.2 V core voltage but require 1.5 V for programming. IGLOO nano V2 devices can be programmed at 1.2 V core voltage (when using FlashPro4 only), but IGLOO nano V5 devices are programmed with a VCC core voltage of 1.5 V.*

***Figure 12-5* • Programming Header (top view)**

***Table 12-4* • Programming Header Pin Numbers and Description**

Pin	Signal	Source	Description
1	TCK	Programmer	JTAG Clock
2	GND[1]	–	Signal Reference
3	TDO	Target Board	Test Data Output
4	NC	–	No Connect (FlashPro3/3X); Prog_Mode (FlashPro4). See note associated with Figure 12-5 on page 269 regarding Prog_Mode on FlashPro4.
5	TMS	Programmer	Test Mode Select
6	VJTAG	Target Board	JTAG Supply Voltage
7	VPUMP[2]	Programmer/Target Board	Programming Supply Voltage
8	nTRST	Programmer	JTAG Test Reset (Hi-Z with 10 kΩ pull-down, HIGH, LOW, or toggling)
9	TDI	Programmer	Test Data Input
10	GND[1]	–	Signal Reference

Notes:

1. *Both GND pins must be connected.*
2. *FlashPro4/3/3X can provide VPUMP if there is only one device on the target board.*

Remote Upgrade via TCP/IP

Transmission Control Protocol (TCP) provides a reliable bitstream transfer service between two endpoints on a network. TCP depends on Internet Protocol (IP) to move packets around the network on its behalf. TCP protects against data loss, data corruption, packet reordering, and data duplication by adding checksums and sequence numbers to transmitted data and, on the receiving side, sending back packets and acknowledging the receipt of data.

The system containing the low power flash device can be assigned an IP address when deployed in the field. When the device requires an update (core or FlashROM), the programming instructions along with the new programming data (AES-encrypted cipher text) can be sent over the Internet to the target system via the TCP/IP protocol. Once the MCU receives the instruction and data, it can proceed with the FPGA update. Low power flash devices support Message Authentication Code (MAC), which can be used to validate data for the target device. More details are given in the "Message Authentication Code (MAC) Validation/Authentication" section.

Hardware Requirement

To facilitate the programming of the low power flash families, the system must have a microprocessor (with access to the device JTAG pins) to process the programming algorithm, memory to store the programming algorithm, programming data, and the necessary programming voltage. Refer to the relevant datasheet for programming voltages.

Security

Encrypted Programming

As an additional security measure, the devices are equipped with AES decryption. AES works in two steps. The first step is to program a key into the devices in a secure or trusted programming center (such as Microsemi SoC Products Group In-House Programming (IHP) center). The second step is to encrypt any programming files with the same encryption key. The encrypted programming file will only work with the devices that have the same key. The AES used in the low power flash families is the 128-bit AES decryption engine (Rijndael algorithm).

Message Authentication Code (MAC) Validation/Authentication

As part of the AES decryption flow, the devices are equipped with a MAC validation/authentication system. MAC is an authentication tag, also called a checksum, derived by applying an on-chip authentication scheme to a STAPL file as it is loaded into the FPGA. MACs are computed and verified with the same key so they can only be verified by the intended recipient. When the MCU system receives the AES-encrypted programming data (cipher text), it can validate the data by loading it into the FPGA and performing a MAC verification prior to loading the data, via a second programming pass, into the FPGA core cells. This prevents erroneous or corrupt data from getting into the FPGA.

Low power flash devices with AES and MAC are superior to devices with only DES or 3DES encryption. Because the MAC verifies the correctness of the data, the FPGA is protected from erroneous loading of invalid programming data that could damage a device (Figure 14-5 on page 289).

The AES with MAC enables field updates over public networks without fear of having the design stolen. An encrypted programming file can only work on devices with the correct key, rendering any stolen files

Power-Up/-Down Sequence and Transient Current

Microsemi's low power flash devices use the following main voltage pins during normal operation:[2]

- VCCPLX
- VJTAG
- VCC: Voltage supply to the FPGA core
  - VCC is 1.5 V ± 0.075 V for IGLOO, IGLOO nano, IGLOO PLUS, and ProASIC3 devices operating at 1.5 V.
  - VCC is 1.2 V ± 0.06 V for IGLOO, IGLOO nano, IGLOO PLUS, and ProASIC3L devices operating at 1.2 V.
  - V5 devices will require a 1.5 V VCC supply, whereas V2 devices can utilize either a 1.2 V or 1.5 V VCC.
- VCCIBx: Supply voltage to the bank's I/O output buffers and I/O logic. Bx is the I/O bank number.
- VMVx: Quiet supply voltage to the input buffers of each I/O bank. x is the bank number. (Note: IGLOO nano, IGLOO PLUS, and ProASIC3 nano devices do not have VMVx supply pins.)

The I/O bank VMV pin must be tied to the VCCI pin within the same bank. Therefore, the supplies that need to be powered up/down during normal operation are VCC and VCCI. These power supplies can be powered up/down in any sequence during normal operation of IGLOO, IGLOO nano, IGLOO PLUS, ProASIC3L, ProASIC3, and ProASIC3 nano FPGAs. During power-up, I/Os in each bank will remain tristated until the last supply (either VCCIBx or VCC) reaches its functional activation voltage. Similarly, during power-down, I/Os of each bank are tristated once the first supply reaches its brownout deactivation voltage.

Although Microsemi's low power flash devices have no power-up or power-down sequencing requirements, Microsemi identifies the following power conditions that will result in higher than normal transient current. Use this information to help maximize power savings:

Microsemi recommends tying VCCPLX to VCC and using proper filtering circuits to decouple VCC noise from the PLL.

a. If VCCPLX is powered up before VCC, a static current of up to 5 mA (typical) per PLL may be measured on VCCPLX.

   The current vanishes as soon as VCC reaches the VCCPLX voltage level.

   The same current is observed at power-down (VCC before VCCPLX).

b. If VCCPLX is powered up simultaneously or after VCC:

   i. Microsemi's low power flash devices exhibit very low transient current on VCC. For ProASIC3 devices, the maximum transient current on V_{CC} does not exceed the maximum standby current specified in the device datasheet.

The source of transient current, also known as inrush current, varies depending on the FPGA technology. Due to their volatile technology, the internal registers in SRAM FPGAs must be initialized before configuration can start. This initialization is the source of significant inrush current in SRAM FPGAs during power-up. Due to the nonvolatile nature of flash technology, low power flash devices do not require any initialization at power-up, and there is very little or no crossbar current through PMOS and NMOS devices. Therefore, the transient current at power-up is significantly less than for SRAM FPGAs. Figure 17-1 on page 310 illustrates the types of power consumption by SRAM FPGAs compared to Microsemi's antifuse and flash FPGAs.

2. For more information on Microsemi FPGA voltage supplies, refer to the appropriate datasheet located at http://www.microsemi.com/soc/techdocs/ds.