



Welcome to [E-XFL.COM](https://www.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

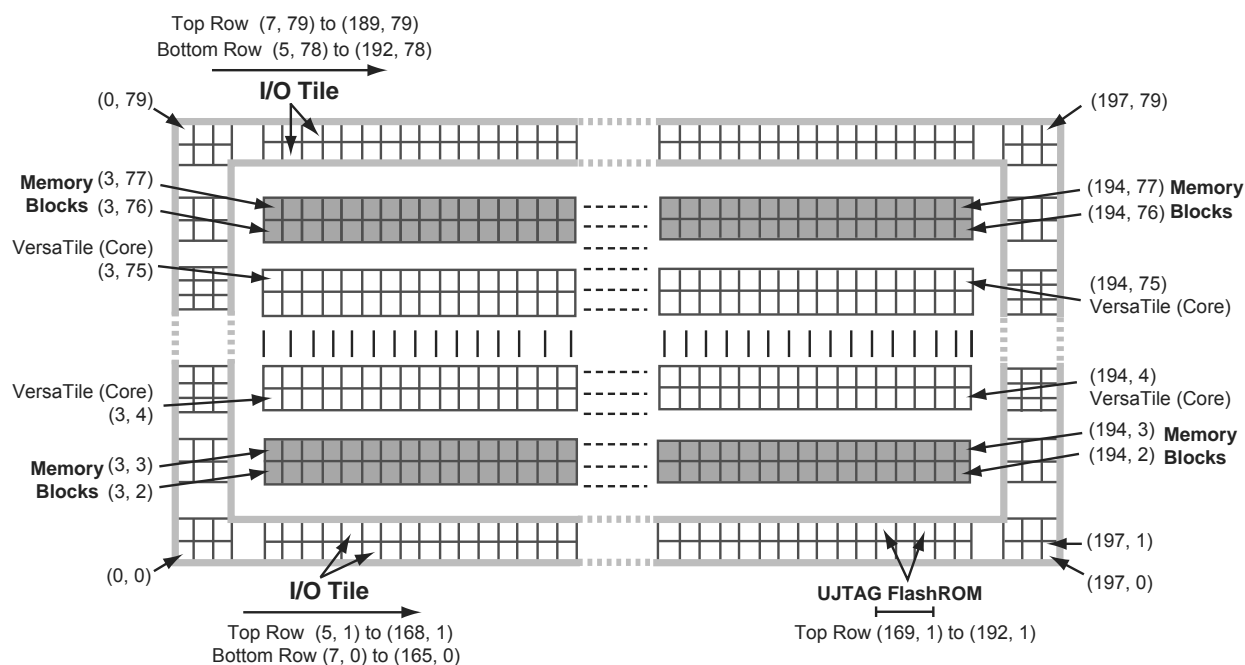
The versatility of Embedded - FPGAs makes them indispensable in numerous fields. In telecommunications.

Details

Product Status	Active
Number of LABs/CLBs	-
Number of Logic Elements/Cells	-
Total RAM Bits	110592
Number of I/O	235
Number of Gates	600000
Voltage - Supply	1.14V ~ 1.575V
Mounting Type	Surface Mount
Operating Temperature	0°C ~ 85°C (TJ)
Package / Case	484-BGA
Supplier Device Package	484-FPBGA (23x23)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/m1a3p600l-1fgg484

Table 1-4 • IGLOO nano and ProASIC3 nano Array Coordinates

Device		VersaTiles		Memory Rows		Entire Die	
		Min.	Max.	Bottom	Top	Min.	Max.
IGLOO nano	ProASIC3 nano	(x, y)	(x, y)	(x, y)	(x, y)	(x, y)	(x, y)
AGLN010	A3P010	(0, 2)	(32, 5)	None	None	(0, 0)	(34, 5)
AGLN015	A3PN015	(0, 2)	(32, 9)	None	None	(0, 0)	(34, 9)
AGLN020	A3PN020	(0, 2)	32, 13)	None	None	(0, 0)	(34, 13)
AGLN060	A3PN060	(3, 2)	(66, 25)	None	(3, 26)	(0, 0)	(69, 29)
AGLN125	A3PN125	(3, 2)	(130, 25)	None	(3, 26)	(0, 0)	(133, 29)
AGLN250	A3PN250	(3, 2)	(130, 49)	None	(3, 50)	(0, 0)	(133, 49)



Note: The vertical I/O tile coordinates are not shown. West-side coordinates are {(0, 2) to (2, 2)} to {(0, 77) to (2, 77)}; east-side coordinates are {(195, 2) to (197, 2)} to {(195, 77) to (197, 77)}.

Figure 1-9 • Array Coordinates for AGL600, AGL600, A3P600, and A3PE600

- 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

Date	Changes	Page
v1.1 (March 2008)	The "Global Architecture" section was updated to include the IGLOO PLUS family. The bullet was revised to include that the west CCC does not contain a PLL core in 15 k and 30 k devices. Instances of "A3P030 and AGL030 devices" were replaced with "15 k and 30 k gate devices."	47
v1.1 (continued)	Table 3-1 • Flash-Based FPGAs and the accompanying text was updated to include the IGLOO PLUS family. The "IGLOO Terminology" section and "ProASIC3 Terminology" section are new.	48
	The "VersaNet Global Network Distribution" section, "Spine Architecture" section, the note in Figure 3-1 • Overview of VersaNet Global Network and Device Architecture, and the note in Figure 3-3 • Simplified VersaNet Global Network (60 k gates and above) were updated to include mention of 15 k gate devices.	49, 50
	Table 3-4 • Globals/Spines/Rows for IGLOO and ProASIC3 Devices was updated to add the A3P015 device, and to revise the values for clock trees, globals/spines per tree, and globals/spines per device for the A3P030 and AGL030 devices.	57
	Table 3-5 • Globals/Spines/Rows for IGLOO PLUS Devices is new.	58
	CLKBUF_LVCMOS12 was added to Table 3-9 • I/O Standards within CLKBUF.	63
	The "User's Guides" section was updated to include the three different I/O Structures chapters for ProASIC3 and IGLOO device families.	74
v1.0 (January 2008)	Figure 3-3 • Simplified VersaNet Global Network (60 k gates and above) was updated.	50
	The "Naming of Global I/Os" section was updated.	51
	The "Using Global Macros in Synplicity" section was updated.	66
	The "Global Promotion and Demotion Using PDC" section was updated.	67
	The "Designer Flow for Global Assignment" section was updated.	69
	The "Simple Design Example" section was updated.	71
51900087-0/1.05 (January 2005)	Table 3-4 • Globals/Spines/Rows for IGLOO and ProASIC3 Devices was updated.	57

Device-Specific Layout

Two kinds of CCCs are offered in low power flash devices: CCCs with integrated PLLs, and CCCs without integrated PLLs (simplified CCCs). Table 4-5 lists the number of CCCs in various devices.

Table 4-5 • Number of CCCs by Device Size and Package

Device		Package	CCCs with Integrated PLLs	CCCs without Integrated PLLs (simplified CCC)
ProASIC3	IGLOO			
A3PN010	AGLN010	All	0	2
A3PN015	AGLN015	All	0	2
A3PN020	AGLN020	All	0	2
	AGLN060	CS81	0	6
A3PN060	AGLN060	All other packages	1	5
	AGLN125	CS81	0	6
A3PN125	AGLN125	All other packages	1	5
	AGLN250	CS81	0	6
A3PN250	AGLN250	All other packages	1	5
A3P015	AGL015	All	0	2
A3P030	AGL030/AGLP030	All	0	2
	AGL060/AGLP060	CS121/CS201	0	6
A3P060	AGL060/AGLP060	All other packages	1	5
A3P125	AGL125/AGLP125	All	1	5
A3P250/L	AGL250	All	1	5
A3P400	AGL400	All	1	5
A3P600/L	AGL600	All	1	5
A3P1000/L	AGL1000	All	1	5
A3PE600	AGLE600	PQ208	2	4
A3PE600/L		All other packages	6	0
A3PE1500		PQ208	2	4
A3PE1500		All other packages	6	0
A3PE3000/L		PQ208	2	4
A3PE3000/L	AGLE3000	All other packages	6	0
Fusion Devices				
AFS090		All	1	5
AFS250, M1AFS250		All	1	5
AFS600, M7AFS600, M1AFS600		All	2	4
AFS1500, M1AFS1500		All	2	4

Note: nano 10 k, 15 k, and 20 k offer 6 global MUXes instead of CCCs.

256×18 FIFO is full, even though a 128×18 FIFO was requested. For this example, the Almost-Full flag can be used instead of the Full flag to signal when the 128th data word is reached.

To accommodate different aspect ratios, the almost-full and almost-empty values are expressed in terms of data bits instead of data words. SmartGen translates the user's input, expressed in data words, into data bits internally. SmartGen allows the user to select the thresholds for the Almost-Empty and Almost-Full flags in terms of either the read data words or the write data words, and makes the appropriate conversions for each flag.

After the empty or full states are reached, the FIFO can be configured so the FIFO counters either stop or continue counting. For timing numbers, refer to the appropriate family datasheet.

Signal Descriptions for FIFO4K18

The following signals are used to configure the FIFO4K18 memory element:

WW and RW

These signals enable the FIFO to be configured in one of the five allowable aspect ratios (Table 6-6).

Table 6-6 • Aspect Ratio Settings for WW[2:0]

WW[2:0]	RW[2:0]	D×W
000	000	4k×1
001	001	2k×2
010	010	1k×4
011	011	512×9
100	100	256×18
101, 110, 111	101, 110, 111	Reserved

WBLK and RBLK

These signals are active-low and will enable the respective ports when LOW. When the RBLK signal is HIGH, that port's outputs hold the previous value.

WEN and REN

Read and write enables. WEN is active-low and REN is active-high by default. These signals can be configured as active-high or -low.

WCLK and RCLK

These are the clock signals for the synchronous read and write operations. These can be driven independently or with the same driver.

Note: For the Automotive ProASIC3 FIFO4K18, for the same clock, 180° out of phase (inverted) between clock pins should be used.

RPIPE

This signal is used to specify pipelined read on the output. A LOW on RPIPE indicates a nonpipelined read, and the data appears on the output in the same clock cycle. A HIGH indicates a pipelined read, and data appears on the output in the next clock cycle.

RESET

This active-low signal resets the control logic and forces the output hold state registers to zero when asserted. It does not reset the contents of the memory array (Table 6-7 on page 160).

While the RESET signal is active, read and write operations are disabled. As with any asynchronous RESET signal, care must be taken not to assert it too close to the edges of active read and write clocks.

WD

This is the input data bus and is 18 bits wide. Not all 18 bits are valid in all configurations. When a data width less than 18 is specified, unused higher-order signals must be grounded (Table 6-7 on page 160).

Example of RAM Initialization

This section of the document presents a sample design in which a 4×4 RAM block is being initialized through the JTAG port. A test feature has been implemented in the design to read back the contents of the RAM after initialization to verify the procedure.

The interface block of this example performs two major functions: initialization of the RAM block and running a test procedure to read back the contents. The clock output of the interface is either the write clock (for initialization) or the read clock (for reading back the contents). The Verilog code for the interface block is included in the "Sample Verilog Code" section on page 167.

For simulation purposes, users can declare the input ports of the UJTAG macro for easier assignment in the testbench. However, the UJTAG input ports should not be declared on the top level during synthesis. If the input ports of the UJTAG are declared during synthesis, the synthesis tool will instantiate input buffers on these ports. The input buffers on the ports will cause Compile to fail in Designer.

Figure 6-10 shows the simulation results for the initialization step of the example design.

The CLK_OUT signal, which is the clock output of the interface block, is the inverted DR_UPDATE output of the UJTAG macro. It is clear that it gives sufficient time (while the TAP Controller is in the Data Register Update state) for the write address and data to become stable before loading them into the RAM block.

Figure 6-11 presents the test procedure of the example. The data read back from the memory block matches the written data, thus verifying the design functionality.

Figure 6-10 • Simulation of Initialization Step

Figure 6-11 • Simulation of the Test Procedure of the Example

SmartGen enables the user to configure the desired RAM element to use either a single clock for read and write, or two independent clocks for read and write. The user can select the type of RAM as well as the width/depth and several other parameters (Figure 6-13).

Figure 6-13 • SmartGen Memory Configuration Interface

SmartGen also has a Port Mapping option that allows the user to specify the names of the ports generated in the memory block (Figure 6-14).

Figure 6-14 • Port Mapping Interface for SmartGen-Generated Memory

SmartGen also configures the FIFO according to user specifications. Users can select no flags, static flags, or dynamic flags. Static flag settings are configured using configuration flash and cannot be altered

Conclusion

Fusion, IGLOO, and ProASIC3 devices provide users with extremely flexible SRAM blocks for most design needs, with the ability to choose between an easy-to-use dual-port memory or a wide-word two-port memory. Used with the built-in FIFO controllers, these memory blocks also serve as highly efficient FIFOs that do not consume user gates when implemented. The SmartGen core generator provides a fast and easy way to configure these memory elements for use in designs.

List of Changes

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

Date	Changes	Page
August 2012	The note connected with Figure 6-3 • Supported Basic RAM Macros, regarding RAM4K9, was revised to explain that it applies only to part numbers of certain revisions and earlier (SAR 29574).	152
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 (December 2008)	IGLOO nano and ProASIC3 nano devices were added to Table 6-1 • Flash-Based FPGAs.	150
	IGLOO nano and ProASIC3 nano devices were added to Figure 6-8 • Interfacing TAP Ports and SRAM Blocks.	164
v1.4 (October 2008)	The "SRAM/FIFO Support in Flash-Based Devices" section was revised to include new families and make the information more concise.	150
	The "SRAM and FIFO Architecture" section was modified to remove "IGLOO and ProASIC3E" from the description of what the memory block includes, as this statement applies to all memory blocks.	151
	Wording in the "Clocking" section was revised to change "IGLOO and ProASIC3 devices support inversion" to "Low power flash devices support inversion." The reference to IGLOO and ProASIC3 development tools in the last paragraph of the section was changed to refer to development tools in general.	157
	The "ESTOP and FSTOP Usage" section was updated to refer to FIFO counters in devices in general rather than only IGLOO and ProASIC3E devices.	160
v1.3 (August 2008)	The note was removed from Figure 6-7 • RAM Block with Embedded FIFO Controller and placed in the WCLK and RCLK description.	158
	The "WCLK and RCLK" description was revised.	159
v1.2 (June 2008)	The following changes were made to the family descriptions in Table 6-1 • Flash-Based FPGAs: <ul style="list-style-type: none"> ProASIC3L was updated to include 1.5 V. The number of PLLs for ProASIC3E was changed from five to six. 	150
v1.1 (March 2008)	The "Introduction" section was updated to include the IGLOO PLUS family.	147
	The "Device Architecture" section was updated to state that 15 k gate devices do not support SRAM and FIFO.	147
	The first note in Figure 6-1 • IGLOO and ProASIC3 Device Architecture Overview was updated to include mention of 15 k gate devices, and IGLOO PLUS was added to the second note.	149

I/O Register Combining

Every I/O has several embedded registers in the I/O tile that are close to the I/O pads. Rather than using the internal register from the core, the user has the option of using these registers for faster clock-to-out timing, and external hold and setup. When combining these registers at the I/O buffer, some architectural rules must be met. Provided these rules are met, the user can enable register combining globally during Compile (as shown in the "Compiling the Design" section on page 261).

This feature is supported by all I/O standards.

Rules for Registered I/O Function

1. The fanout between an I/O pin (D, Y, or E) and a register must be equal to one for combining to be considered on that pin.
2. All registers (Input, Output, and Output Enable) connected to an I/O must share the same clear or preset function:
 - If one of the registers has a CLR pin, all the other registers that are candidates for combining in the I/O must have a CLR pin.
 - 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. 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 buffer 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

IGLOO and ProASIC3 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 VCCI of its corresponding I/O bank. When it is pulled down, it is connected to GND. Refer to the datasheet for more information.

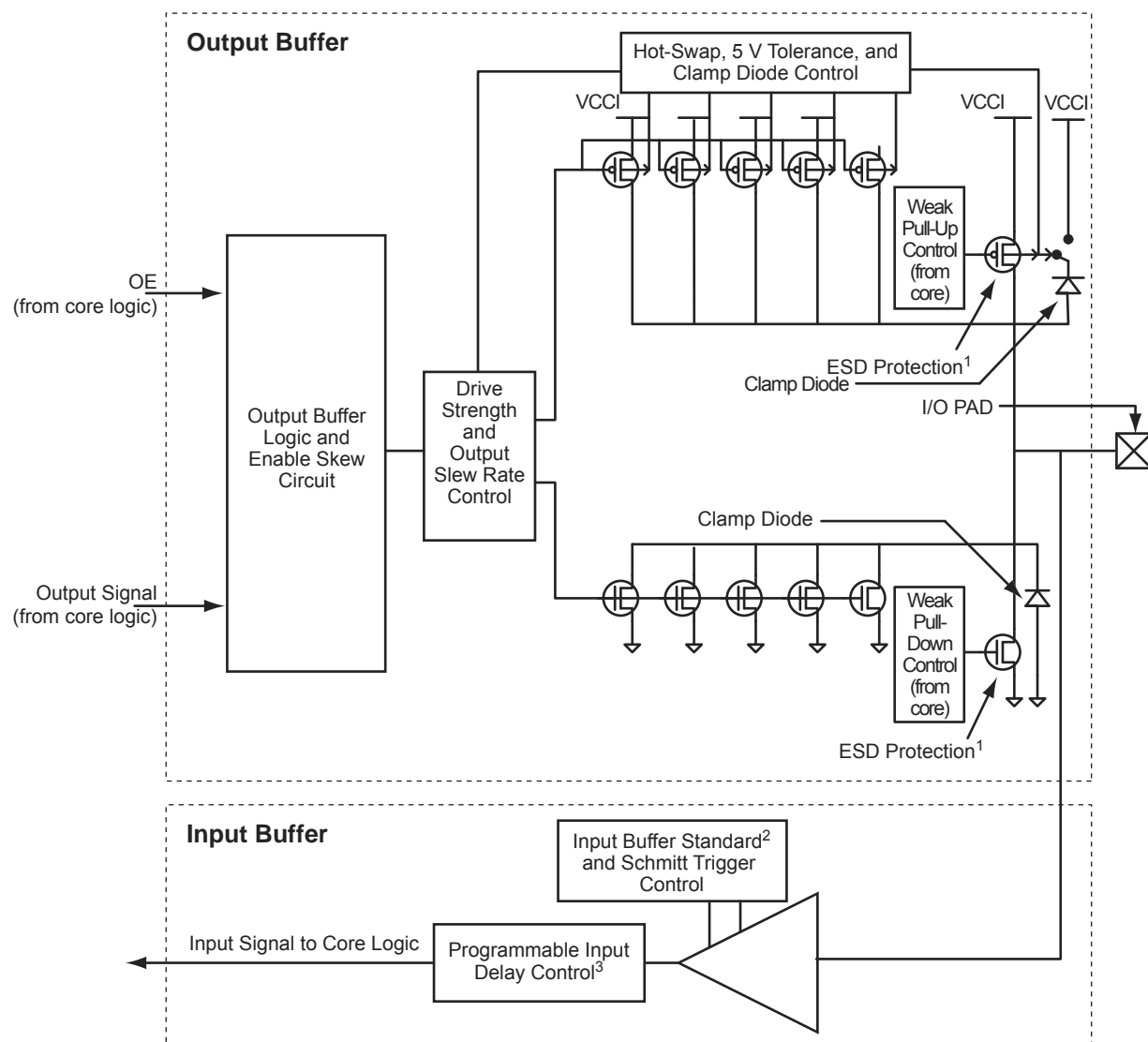
For low power applications, configuration of the pull-up or pull-down of the I/O can be used to set the I/O to a known state while the device is in Flash*Freeze mode. Refer to the "Flash*Freeze Technology and Low Power Modes in IGLOO and ProASIC3L Devices" chapter of the *IGLOO FPGA Fabric User's Guide* or *ProASIC3L 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. Slew rate control is available for LVTTTL, LVCMOS, and PCI-X I/O standards. The other I/O standards have a preset slew value.

The slew rate can be implemented by using a PDC command (Table 7-5 on page 179), 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."



Notes:

1. All NMOS transistors connected to the I/O pad serve as ESD protection.
2. See Table 8-2 on page 215 for available I/O standards.
3. Programmable input delay is applicable only to ProASIC3E, IGLOOe, ProASIC3EL, and RT ProASIC3 devices.

Figure 8-5 • Simplified I/O Buffer Circuitry

I/O Registers

Each I/O module contains several input, output, and enable registers. Refer to Figure 8-5 for a simplified representation of the I/O block. The number of input registers is selected by a set of switches (not shown in Figure 8-3 on page 220) between registers to implement single-ended or differential data transmission to and from the FPGA core. The Designer software sets these switches for the user. A common CLR/PRE signal is employed by all I/O registers when I/O register combining is used. Input Register 2 does not have a CLR/PRE pin, as this register is used for DDR implementation. The I/O register combining must satisfy certain rules.

If the assignment is not successful, an error message appears in the Output window.

To undo the I/O bank assignments, choose **Undo** from the **Edit** menu. Undo removes the I/O technologies assigned by the IOBA. It does not remove the I/O technologies previously assigned.

To redo the changes undone by the Undo command, choose **Redo** from the **Edit** menu.

To clear I/O bank assignments made before using the Undo command, manually unassign or reassign I/O technologies to banks. To do so, choose **I/O Bank Settings** from the **Edit** menu to display the I/O Bank Settings dialog box.

Conclusion

Fusion, IGLOO, and ProASIC3 support for multiple I/O standards minimizes board-level components and makes possible a wide variety of applications. The Microsemi Designer software, integrated with Libero SoC, presents a clear visual display of I/O assignments, allowing users to verify I/O and board-level design requirements before programming the device. The device I/O features and functionalities ensure board designers can produce low-cost and low power FPGA applications fulfilling the complexities of contemporary design needs.

Related Documents

User's Guides

Libero SoC User's Guide

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

IGLOO, ProASIC3, SmartFusion, and Fusion Macro Library Guide

http://www.microsemi.com/soc/documents/pa3_libguide_ug.pdf

SmartGen Core Reference Guide

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

DDR Support in Flash-Based Devices

The flash FPGAs listed in Table 10-1 support the DDR feature and the functions described in this document.

Table 10-1 • 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
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 10-1. 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 10-1. 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*.

The AES key is securely stored on-chip in dedicated low power flash device flash memory and cannot be read out. In the first step, the AES key is generated and programmed into the device (for example, at a secure or trusted programming site). The Microsemi Designer software tool provides AES key generation capability. After the key has been programmed into the device, the device will only correctly decrypt programming files that have been encrypted with the same key. If the individual programming file content is incorrect, a Message Authentication Control (MAC) mechanism inside the device will fail in authenticating the programming file. In other words, when an encrypted programming file is being loaded into a device that has a different programmed AES key, the MAC will prevent this incorrect data from being loaded, preventing possible device damage. See Figure 12-3 on page 304 and Figure 12-4 on page 306 for graphical representations of this process.

It is important to note that the user decides what level of protection will be implemented for the device. When AES protection is desired, the FlashLock Pass Key must be set. The AES key is a content protection mechanism, whereas the FlashLock Pass Key is a device protection mechanism. When the AES key is programmed into the device, the device still needs the Pass Key to protect the FPGA and FlashROM contents and the security settings, including the AES key. Using the FlashLock Pass Key prevents modification of the design contents by means of simply programming the device with a different AES key.

AES Decryption and MAC Authentication

Low power flash devices have a built-in 128-bit AES decryption core, which decrypts the encrypted programming file and performs a MAC check that authenticates the file prior to programming.

MAC authenticates the entire programming data stream. After AES decryption, the MAC checks the data to make sure it is valid programming data for the device. This can be done while the device is still operating. If the MAC validates the file, the device will be erased and programmed. If the MAC fails to validate, then the device will continue to operate uninterrupted.

This will ensure the following:

- Correct decryption of the encrypted programming file
- Prevention of erroneous or corrupted data being programmed during the programming file transfer
- Correct bitstream passed to the device for decryption

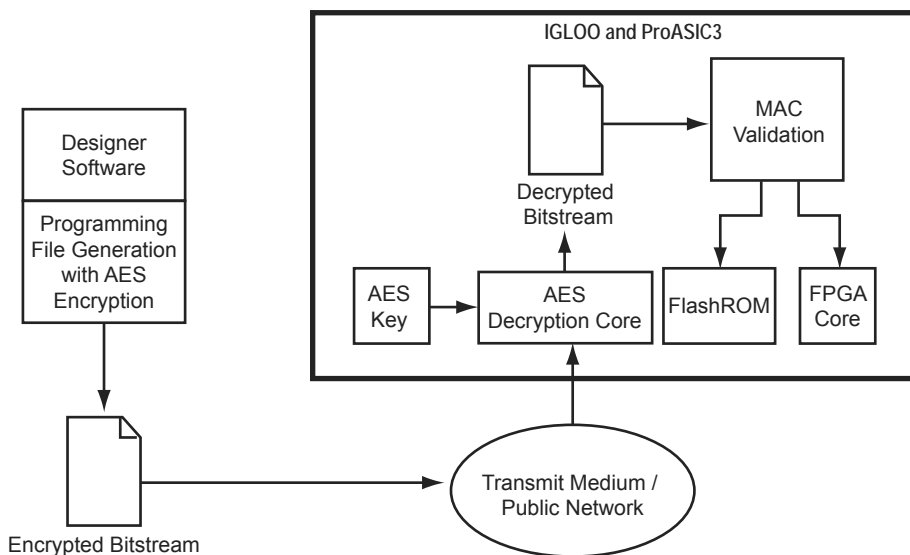


Figure 12-4 • Example Application Scenario Using AES in IGLOO and ProASIC3 Devices

1. 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.

Circuit Description

All IGLOO devices as well as the ProASIC3L product family are available in two versions: V5 devices, which are powered by a 1.5 V supply and V2 devices, which are powered by a supply anywhere in the range of 1.2 V to 1.5 V in 50 mV increments. Applications that use IGLOO or ProASIC3L devices powered by a 1.2 V core supply must have a mechanism that switches the core voltage from 1.2 V (or other voltage below 1.5 V) to 1.5 V during in-system programming (ISP). There are several possible techniques to meet this requirement. Microsemi recommends utilizing a linear voltage regulator, a resistor voltage divider, and an N-Channel Digital FET to set the appropriate VCC voltage, as shown in Figure 14-1.

Where 1.2 V is mentioned in the following text, the meaning applies to any voltage below the 1.5 V range. Resistor values in the figures have been calculated for 1.2 V, so refer to power regulator datasheets if a different core voltage is required.

The main component of Microsemi's recommended circuit is the LTC3025 linear voltage regulator from LinearTech. The output voltage of the LTC3025 on the OUT pin is set by the ratio of two external resistors, R37 and R38, in a voltage divider. The linear voltage regulator adjusts the voltage on the OUT pin to maintain the ADJ pin voltage at 0.4 V (referenced to ground). By using an R38 value of 40.2 k Ω and an R37 value of 80.6 k Ω , the output voltage on the OUT pin is 1.2 V. To achieve 1.5 V on the OUT pin, R44 can be used in parallel with R38. The OUT pin can now be used as a switchable source for the VCC supply. Refer to the *LTC3025 Linear Voltage Regulator datasheet* for more information.

In Figure 14-1, the N-Channel Digital FET is used to enable and disable R44. This FET is controlled by the JTAG TRST signal driven by the FlashPro3 programmer. During programming of the device, the TRST signal is driven HIGH by the FlashPro3, and turns the N-Channel Digital FET ON. When the FET is ON, R44 becomes enabled as a parallel resistance to R38, which forces the regulator to set OUT to 1.5 V.

When the FlashPro3 is connected and not in programming mode or when it is not connected, the pull-down resistor, R10, will pull the TRST signal LOW. When this signal is LOW, the N-Channel Digital FET is "open" and R44 is not part of the resistance seen by the LTC3025. The new resistance momentarily changes the voltage value on the ADJ pin, which in turn causes the output of the LTC3025 to compensate by setting OUT to 1.2 V. Now the device will run in regular active mode at the regular 1.2 V core voltage.

Figure 14-1 • Circuit Diagram

Boundary Scan Support in Low Power Devices

The information in this document applies to all Fusion, IGLOO, and ProASIC3 devices. For IGLOO, IGLOO PLUS, and ProASIC3L devices, the Flash*Freeze pin must be deasserted for successful boundary scan operations. Devices cannot enter JTAG mode directly from Flash*Freeze mode.

Boundary Scan Opcodes

Low power flash devices support all mandatory IEEE 1149.1 instructions (EXTEST, SAMPLE/PRELOAD, and BYPASS) and the optional IDCODE instruction (Table 16-2).

Table 16-2 • Boundary Scan Opcodes

	Hex Opcode
EXTEST	00
HIGHZ	07
USERCODE	0E
SAMPLE/PRELOAD	01
IDCODE	0F
CLAMP	05
BYPASS	FF

Boundary Scan Chain

The serial pins are used to serially connect all the boundary scan register cells in a device into a boundary scan register chain (Figure 16-2 on page 360), which starts at the TDI pin and ends at the TDO pin. The parallel ports are connected to the internal core logic I/O tile and the input, output, and control ports of an I/O buffer to capture and load data into the register to control or observe the logic state of each I/O.

Each test section is accessed through the TAP, which has five associated pins: TCK (test clock input), TDI, TDO (test data input and output), TMS (test mode selector), and TRST (test reset input). TMS, TDI, and TRST are equipped with pull-up resistors to ensure proper operation when no input data is supplied to them. These pins are dedicated for boundary scan test usage. Refer to the "JTAG Pins" section in the "Pin Descriptions and Packaging" chapter of the appropriate device datasheet for pull-up/-down recommendations for TCK and TRST pins. Pull-down recommendations are also given in Table 16-3 on page 360

UJTAG Macro

The UJTAG tiles can be instantiated in a design using the UJTAG macro from the Fusion, IGLOO, or ProASIC3 macro library. Note that "UJTAG" is a reserved name and cannot be used for any other user-defined blocks. A block symbol of the UJTAG tile macro is presented in Figure 17-2. In this figure, the ports on the left side of the block are connected to the JTAG TAP Controller, and the right-side ports are accessible by the FPGA core VersaTiles. The TDI, TMS, TDO, TCK, and TRST ports of UJTAG are only provided for design simulation purposes and should be treated as external signals in the design netlist. However, these ports must NOT be connected to any I/O buffer in the netlist. Figure 17-3 on page 366 illustrates the correct connection of the UJTAG macro to the user design netlist. Microsemi Designer software will automatically connect these ports to the TAP during place-and-route. Table 17-2 gives the port descriptions for the rest of the UJTAG ports:

Table 17-2 • UJTAG Port Descriptions

Port	Description
UIREG [7:0]	This 8-bit bus carries the contents of the JTAG Instruction Register of each device. Instruction Register values 16 to 127 are not reserved and can be employed as user-defined instructions.
URSTB	URSTB is an active-low signal and will be asserted when the TAP Controller is in Test-Logic-Reset mode. URSTB is asserted at power-up, and a power-on reset signal resets the TAP Controller. URSTB will stay asserted until an external TAP access changes the TAP Controller state.
UTDI	This port is directly connected to the TAP's TDI signal.
UTDO	This port is the user TDO output. Inputs to the UTDO port are sent to the TAP TDO output MUX when the IR address is in user range.
UDRSH	Active-high signal enabled in the ShiftDR TAP state
UDRCAP	Active-high signal enabled in the CaptureDR TAP state
UDRCK	This port is directly connected to the TAP's TCK signal.
UDRUPD	Active-high signal enabled in the UpdateDR TAP state

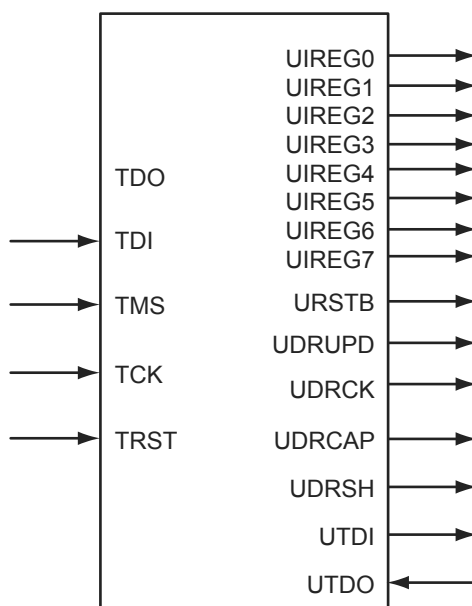


Figure 17-2 • UJTAG Tile Block Symbol

Typical UJTAG Applications

Bidirectional access to the JTAG port from VersaTiles—without putting the device into test mode—creates flexibility to implement many different applications. This section describes a few of these. All are based on importing/exporting data through the UJTAG tiles.

Clock Conditioning Circuitry—Dynamic Reconfiguration

In low power flash devices, CCCs, which include PLLs, can be configured dynamically through either an 81-bit embedded shift register or static flash programming switches. These 81 bits control all the characteristics of the CCC: routing MUX architectures, delay values, divider values, etc. Table 17-3 lists the 81 configuration bits in the CCC.

Table 17-3 • Configuration Bits of Fusion, IGLOO, and ProASIC3 CCC Blocks

Bit Number(s)	Control Function
80	RESET ENABLE
79	DYNCSSEL
78	DYNBSSEL
77	DYNASEL
<76:74>	VCOSSEL [2:0]
73	STATCSSEL
72	STATBSSEL
71	STATASEL
<70:66>	DLYC [4:0]
<65:61>	DLYB [4:0]
<60:56>	DLYGLC [4:0]
<55:51>	DLYGLB [4:0]
<50:46>	DLYGLA [4:0]
45	XDLYSEL
<44:40>	FBDLY [4:0]
<39:38>	FBSEL
<37:35>	OCMUX [2:0]
<34:32>	OBMUX [2:0]
<31:29>	OAMUX [2:0]
<28:24>	OCDIV [4:0]
<23:19>	OBDIV [4:0]
<18:14>	OADIV [4:0]
<13:7>	FBDIV [6:0]
<6:0>	FINDIV [6:0]

The embedded 81-bit shift register (for the dynamic configuration of the CCC) is accessible to the VersaTiles, which, in turn, have access to the UJTAG tiles. Therefore, the CCC configuration shift register can receive and load the new configuration data stream from JTAG.

Dynamic reconfiguration eliminates the need to reprogram the device when reconfiguration of the CCC functional blocks is needed. The CCC configuration can be modified while the device continues to operate. Employing the UJTAG core requires the user to design a module to provide the configuration data and control the CCC configuration shift register. In essence, this is a user-designed TAP Controller requiring chip resources.

Similar reconfiguration capability exists in the ProASIC^{PLUS}® family. The only difference is the number of shift register bits controlling the CCC (27 in ProASIC^{PLUS} and 81 in IGLOO, ProASIC3, and Fusion).

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 \pm 0.075 V for IGLOO, IGLOO nano, IGLOO PLUS, and ProASIC3 devices operating at 1.5 V.
 - VCC is 1.2 V \pm 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 18-1 on page 376 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>.