



Welcome to [E-XFL.COM](https://www.e-xfl.com)

### Understanding [Embedded - FPGAs \(Field Programmable Gate Array\)](#)

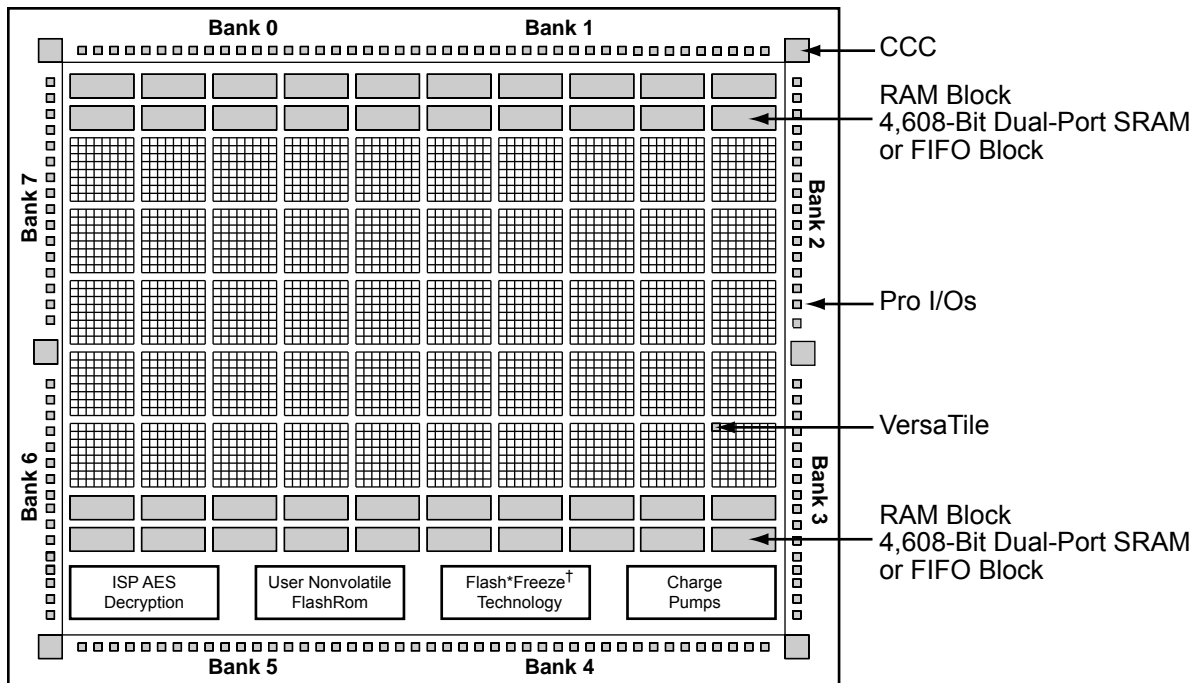
Embedded - FPGAs, or Field Programmable Gate Arrays, are advanced integrated circuits that offer unparalleled flexibility and performance for digital systems. Unlike traditional fixed-function logic devices, FPGAs can be programmed and reprogrammed to execute a wide array of logical operations, enabling 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	516096
Number of I/O	341
Number of Gates	3000000
Voltage - Supply	1.14V ~ 1.575V
Mounting Type	Surface Mount
Operating Temperature	-40°C ~ 100°C (TJ)
Package / Case	484-BGA
Supplier Device Package	484-FPBGA (23x23)
Purchase URL	<a href="https://www.e-xfl.com/product-detail/microchip-technology/a3pe3000l-1fgg484i">https://www.e-xfl.com/product-detail/microchip-technology/a3pe3000l-1fgg484i</a>



*Note: Flash\*Freeze technology only applies to IGL00e devices.*

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

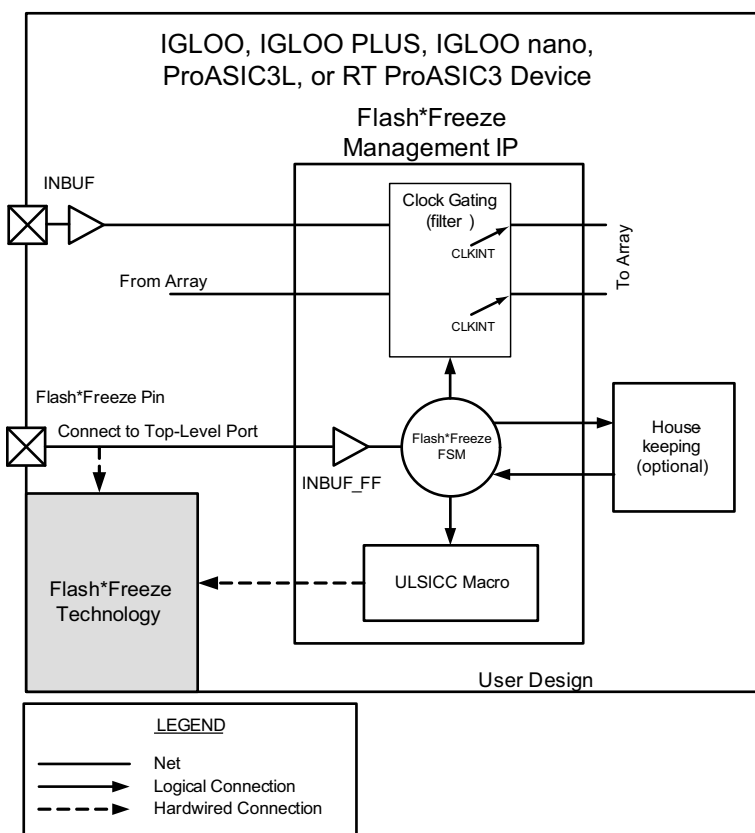
## I/O State of Newly Shipped Devices

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.

Flash\*Freeze management IP. Additional information on this IP core can be found in the Libero online help.

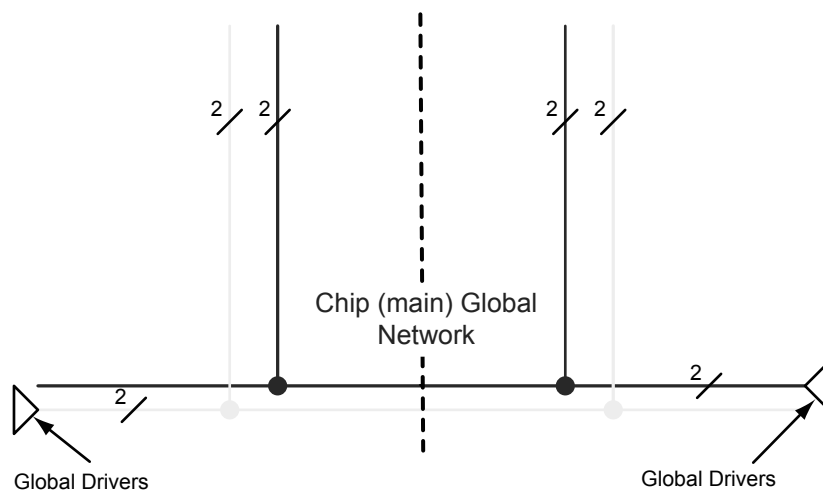
The Flash\*Freeze management IP is comprised of three blocks: the Flash\*Freeze finite state machine (FSM), the clock gating (filter) block, and the ULSICC macro, as shown in Figure 2-10.



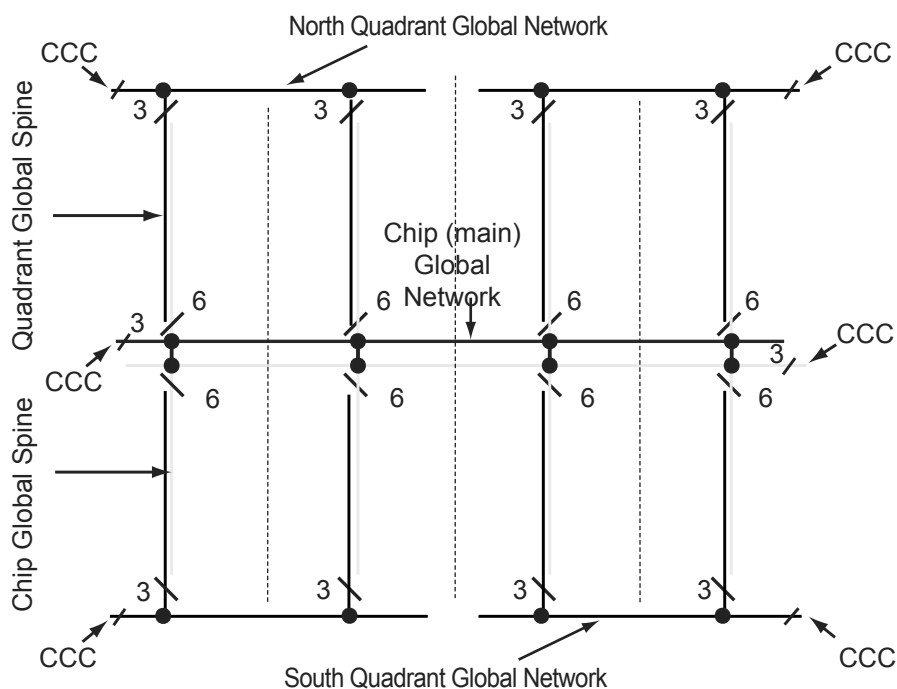
**Figure 2-10 • Flash\*Freeze Management IP Block Diagram**

### **Flash\*Freeze Management FSM**

The Flash\*Freeze FSM block is a simple, robust, fully encoded 3-bit state machine that ensures clean entrance to and exit from Flash\*Freeze mode by controlling activities of the clock gating, ULSICC, and optional housekeeping blocks. The state diagram for the FSM is shown in Figure 2-11 on page 38. In normal operation, the state machine waits for Flash\*Freeze pin assertion, and upon detection of a request, it waits for a short period of time to ensure the assertion persists; then it asserts WAIT\_HOUSEKEEPING (active High) synchronous to the user's designated system clock. This flag can be used by user logic to perform any needed shutdown processes prior to entering Flash\*Freeze mode, such as storing data into SRAM, notifying other system components of the request, or timing/validating the Flash\*Freeze request. The FSM also asserts Flash\_Freeze\_Enabled whenever the device enters Flash\*Freeze mode. This occurs after all housekeeping and clock gating functions have completed. The Flash\_Freeze\_Enabled signal remains asserted, even during Flash\*Freeze mode, until the Flash\*Freeze pin is deasserted. Use the Flash\_Freeze\_Enabled signal to drive any logic in the design that needs to be in a particular state during Flash\*Freeze mode. The DONE\_HOUSEKEEPING (active High) signal should be asserted to notify the FSM when all the housekeeping tasks are completed. If the user chooses not to use housekeeping, the Flash\*Freeze management IP core generator in Libero SoC will connect WAIT\_HOUSEKEEPING to DONE\_HOUSEKEEPING.



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



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



---

**Figure 3-18 • Globals Management GUI in Designer**

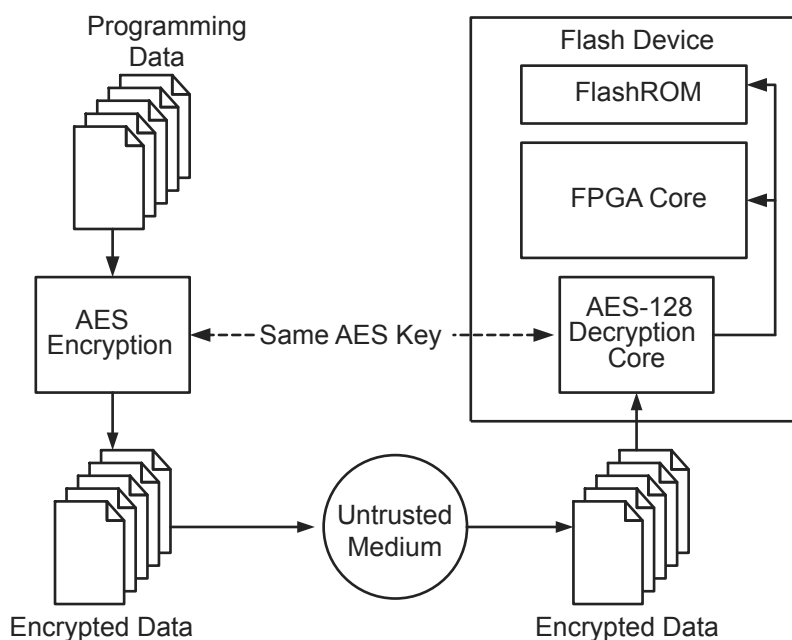
3. Occasionally, the synthesis tool assigns a global macro to clock nets, even though the fanout is significantly less than other asynchronous signals. Select **Demote global nets whose fanout is less than** and enter a reasonable value for fanouts. This frees up some global networks from the signals that have very low fanouts. This can also be done using PDC.
4. Use a local clock network for the signals that do not need to go to the whole chip but should have low skew. This local clock network assignment can only be done using PDC.
5. Assign the I/O buffer using MVN if you have fixed I/O assignment. As shown in Figure 3-10 on page 61, there are three sets of global pins that have a hardwired connection to each global network. Do not try to put multiple CLKBUF macros in these three sets of global pins. For example, do not assign two CLKBUFs to GAA0x and GAA2x pins.
6. You must click **Commit** at the end of MVN assignment. This runs the pre-layout checker and checks the validity of global assignment.
7. Always run Compile with the **Keep existing physical constraints** option on. This uses the quadrant clock network assignment in the MVN assignment and checks if you have the desired signals on the global networks.
8. Run Layout and check the timing.

## 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  $3.4 \times 10^{38}$  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**

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]**

<b>WW[2:0]</b>	<b>RW[2:0]</b>	<b>D×W</b>
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).

Date	Changes	Page
v1.1 (continued)	Table 6-1 • Flash-Based FPGAs and associated text were updated to include the IGLOO PLUS family. The "IGLOO Terminology" section and "ProASIC3 Terminology" section are new.	150
	The text introducing Table 6-8 • Memory Availability per IGLOO and ProASIC3 Device was updated to replace "A3P030 and AGL030" with "15 k and 30 k gate devices." Table 6-8 • Memory Availability per IGLOO and ProASIC3 Device was updated to remove AGL400 and AGL1500 and include IGLOO PLUS and ProASIC3L devices.	162

- In Active and Static modes:
  - Input buffers with pull-up, driven Low
  - Input buffers with pull-down, driven High
  - Bidirectional buffers with pull-up, driven Low
  - Bidirectional buffers with pull-down, driven High
  - Output buffers with pull-up, driven Low
  - Output buffers with pull-down, driven High
  - Tristate buffers with pull-up, driven Low
  - Tristate buffers with pull-down, driven High
- In Flash\*Freeze mode:
  - Input buffers with pull-up, driven Low
  - Input buffers with pull-down, driven High
  - Bidirectional buffers with pull-up, driven Low
  - Bidirectional buffers with pull-down, driven High

## Electrostatic Discharge Protection

Low power flash devices are tested per JEDEC Standard JESD22-A114-B.

These devices contain clamp diodes at every I/O, global, and power pad. Clamp diodes protect all device pads against damage from ESD as well as from excessive voltage transients.

All IGLOO and ProASIC3 devices are tested to the Human Body Model (HBM) and the Charged Device Model (CDM).

Each I/O has two clamp diodes. One diode has its positive (P) side connected to the pad and its negative (N) side connected to VCCI. The second diode has its P side connected to GND and its N side connected to the pad. During operation, these diodes are normally biased in the off state, except when transient voltage is significantly above VCCI or below GND levels.

In 30K gate devices, the first diode is always off. In other devices, the clamp diode is always on and cannot be switched off.

By selecting the appropriate I/O configuration, the diode is turned on or off. Refer to Table 7-12 on page 193 for more information about the I/O standards and the clamp diode.

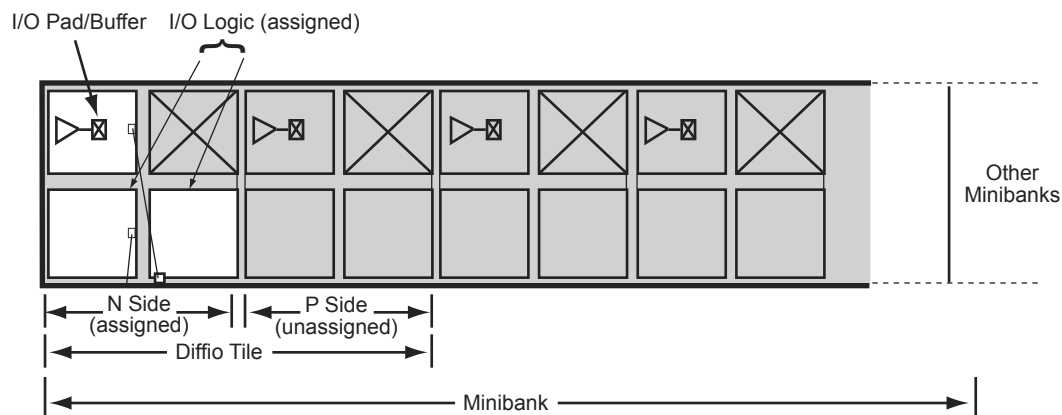
The second diode is always connected to the pad, regardless of the I/O configuration selected.

## I/O Bank Structure

Low power flash device I/Os are divided into multiple technology banks. The number of banks is device-dependent. The IGLOOe, ProASIC3EL, and ProASIC3E devices have eight banks (two per side); and IGLOO, ProASIC3L, and ProASIC3 devices have two to four banks. Each bank has its own  $V_{CCI}$  power supply pin. Multiple I/O standards can co-exist within a single I/O bank.

In IGLOOe, ProASIC3EL, and ProASIC3E devices, each I/O bank is subdivided into  $V_{REF}$  minibanks. These are used by voltage-referenced I/Os. VREF minibanks contain 8 to 18 I/Os. All I/Os in a given minibank share a common VREF line (only one VREF pin is needed per VREF minibank). Therefore, if an I/O in a VREF minibank is configured as a VREF pin, the remaining I/Os in that minibank will be able to use the voltage assigned to that pin. If the location of the VREF pin is selected manually in the software, the user must satisfy VREF rules (refer to the "I/O Software Control in Low Power Flash Devices" section on page 251). If the user does not pick the VREF pin manually, the software automatically assigns it.

Figure 8-4 is a snapshot of a section of the I/O ring, showing the basic elements of an I/O tile, as viewed from the Designer place-and-route tool's MultiView Navigator (MVN).



**Figure 8-4 • Snapshot of an I/O Tile**

Low power flash device I/Os are implemented using two tile types: I/O and differential I/O (diffio).

The diffio tile is built up using two I/O tiles, which form an I/O pair (P side and N side). These I/O pairs are used according to differential I/O standards. Both the P and N sides of the diffio tile include an I/O buffer and two I/O logic blocks (auxiliary and main logic).

Every minibank (E devices only) is built up from multiple diffio tiles. The number of the minibank depends on the different-size dies. Refer to the "Pro I/Os—IGLOOe, ProASIC3EL, and ProASIC3E" section on page 215 for an illustration of the minibank structure.

Figure 8-5 on page 222 shows a simplified diagram of the I/O buffer circuitry. The Output Enable signal (OE) enables the output buffer to pass the signal from the core logic to the pin. The output buffer contains ESD protection circuitry, an n-channel transistor that shunts all ESD surges (up to the limit of the device ESD specification) to GND. This transistor also serves as an output pull-down resistor.

Each output buffer also contains programmable slew rate, drive strength, programmable power-up state (pull-up/-down resistor), hot-swap, 5 V tolerance, and clamp diode control circuitry. Multiple flash switches (not shown in Figure 8-5 on page 222) are programmed by user selections in the software to activate different I/O features.

## I/O Features

Low power flash devices support multiple I/O features that make board design easier. For example, an I/O feature like Schmitt Trigger in the ProASIC3E input buffer saves the board space that would be used by an external Schmitt trigger for a slow or noisy input signal. These features are also programmable for each I/O, which in turn gives flexibility in interfacing with other components. The following is a detailed description of all available features in low power flash devices.

### I/O Programmable Features

Low power flash devices offer many flexible I/O features to support a wide variety of board designs. Some of the features are programmable, with a range for selection. Table 8-8 lists programmable I/O features and their ranges.

**Table 8-8 • Programmable I/O Features (user control via I/O Attribute Editor)**

Feature <sup>1</sup>	Description	Range
Slew Control	Output slew rate	HIGH, LOW
Output Drive (mA)	Output drive strength	2, 4, 6, 8, 12, 16, 24
Skew Control	Output tristate enable delay option	ON, OFF
Resistor Pull	Resistor pull circuit	Up, Down, None
Input Delay <sup>2</sup>	Input delay	OFF, 0–7
Schmitt Trigger	Schmitt trigger for input only	ON, OFF

Notes:

1. Limitations of these features with respect to different devices are discussed in later sections.
2. Programmable input delay is applicable only to ProASIC3E, IGLOOe, ProASIC3EL, and RT ProASIC3 devices.

### Hot-Swap Support

A pull-up clamp diode must not be present in the I/O circuitry if the hot-swap feature is used. The 3.3 V PCI standard requires a pull-up clamp diode on the I/O, so it cannot be selected if hot-swap capability is required. The A3P030 device does not support 3.3 V PCI, so it is the only device in the ProASIC3 family that supports the hot-swap feature. All devices in the ProASIC3E family are hot-swappable. All standards except LVCMOS 2.5/5.0 V and 3.3 V PCI/PCI-X support the hot-swap feature.

The hot-swap feature appears as a read-only check box in the I/O Attribute Editor that shows whether an I/O is hot-swappable or not. Refer to the *"Power-Up/Down Behavior of Low Power Flash Devices"* section on page 373 for details on hot-swapping.

Hot-swapping (also called hot-plugging) is the operation of hot insertion or hot removal of a card in a powered-up system. The levels of hot-swap support and examples of related applications are described in Table 8-9 on page 228 to Table 8-12 on page 229. The I/Os also need to be configured in hot-insertion mode if hot-plugging compliance is required. The AGL030 and A3P030 devices have an I/O structure that allows the support of Level 3 and Level 4 hot-swap with only two levels of staging.

**Table 8-11 • Hot-Swap Level 3**

<b>Description</b>	Hot-swap while bus idle
<b>Power Applied to Device</b>	Yes
<b>Bus State</b>	Held idle (no ongoing I/O processes during insertion/removal)
<b>Card Ground Connection</b>	Reset must be maintained for 1 ms before, during, and after insertion/removal.
<b>Device Circuitry Connected to Bus Pins</b>	Must remain glitch-free during power-up or power-down
<b>Example Application</b>	Board bus shared with card bus is "frozen," and there is no toggling activity on the bus. It is critical that the logic states set on the bus signal not be disturbed during card insertion/removal.
<b>Compliance of IGLOO and ProASIC3 Devices</b>	30 k gate devices, all IGLOOe/ProASIC3E devices: Compliant with two levels of staging (first: GND; second: all other pins) Other IGLOO/ProASIC3 devices: Compliant: Option A – Two levels of staging (first: GND; second: all other pins) together with bus switch on the I/Os Option B – Three levels of staging (first: GND; second: supplies; third: all other pins)

**Table 8-12 • Hot-Swap Level 4**

<b>Description</b>	Hot-swap on an active bus
<b>Power Applied to Device</b>	Yes
<b>Bus State</b>	Bus may have active I/O processes ongoing, but device being inserted or removed must be idle.
<b>Card Ground Connection</b>	Reset must be maintained for 1 ms before, during, and after insertion/removal.
<b>Device Circuitry Connected to Bus Pins</b>	Must remain glitch-free during power-up or power-down
<b>Example Application</b>	There is activity on the system bus, and it is critical that the logic states set on the bus signal not be disturbed during card insertion/removal.
<b>Compliance of IGLOO and ProASIC3 Devices</b>	30 k gate devices, all IGLOOe/ProASIC3E devices: Compliant with two levels of staging (first: GND; second: all other pins) Other IGLOO/ProASIC3 devices: Compliant: Option A – Two levels of staging (first: GND; second: all other pins) together with bus switch on the I/Os Option B – Three levels of staging (first: GND; second: supplies; third: all other pins)



### 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 9-4).
  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 9-5 on page 257).
- 

---

#### Figure 9-4 • 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:

---

**Figure 12-15 • Programming Fusion Security Settings Only**

2. Choose the desired security level setting and enter the key(s).
    - The **High** security level employs FlashLock Pass Key with AES Key protection.
    - The **Medium** security level employs FlashLock Pass Key protection only.
- 

---

**Figure 12-16 • High Security Level to Implement FlashLock Pass Key and AES Key Protection**

## Related Documents

### User's Guides

*FlashPro User's Guide*

[http://www.microsemi.com/soc/documents/flashpro\\_ug.pdf](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: <ul style="list-style-type: none"> <li>ProASIC3L was updated to include 1.5 V.</li> <li>The number of PLLs for ProASIC3E was changed from five to six.</li> </ul>	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

## 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<sup>PLUS</sup> 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 13-3 • 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 <sup>PLUS</sup> based boards to ProASIC3 based boards.

TCK	1	2	GND
TDO	3	4	NC (FlashPro3/3X); Prog_Mode* (FlashPro4)
TMS	5	6	VJTAG
VPUMP	7	8	TRST
TDI	9	10	GND

*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 13-5 • Programming Header (top view)**

# Programming Algorithm

## JTAG Interface

The low power flash families are fully compliant with the IEEE 1149.1 (JTAG) standard. They support all the mandatory boundary scan instructions (EXTEST, SAMPLE/PRELOAD, and BYPASS) as well as six optional public instructions (USERCODE, IDCODE, HIGHZ, and CLAMP).

## IEEE 1532

The low power flash families are also fully compliant with the IEEE 1532 programming standard. The IEEE 1532 standard adds programming instructions and associated data registers to devices that comply with the IEEE 1149.1 standard (JTAG). These instructions and registers extend the capabilities of the IEEE 1149.1 standard such that the Test Access Port (TAP) can be used for configuration activities. The IEEE 1532 standard greatly simplifies the programming algorithm, reducing the amount of time needed to implement microprocessor ISP.

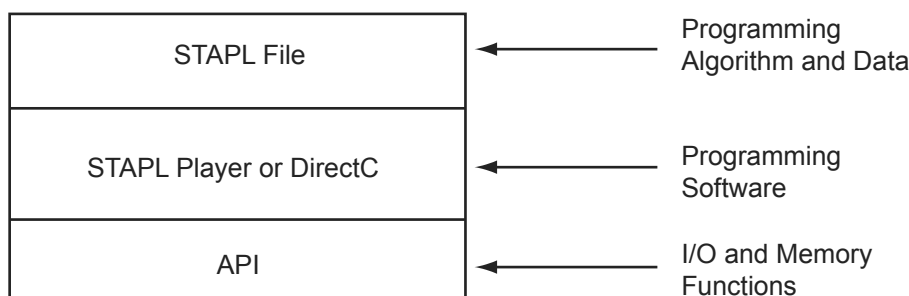
# Implementation Overview

To implement device programming with a microprocessor, the user should first download the C-based STAPL player or DirectC code from the Microsemi SoC Products Group website. Refer to the website for future updates regarding the STAPL player and DirectC code.

[http://www.microsemi.com/soc/download/program\\_debug/stapl/default.aspx](http://www.microsemi.com/soc/download/program_debug/stapl/default.aspx)

[http://www.microsemi.com/soc/download/program\\_debug/directc/default.aspx](http://www.microsemi.com/soc/download/program_debug/directc/default.aspx)

Using the easy-to-follow user's guide, create the low-level application programming interface (API) to provide the necessary basic functions. These API functions act as the interface between the programming software and the actual hardware (Figure 15-2).



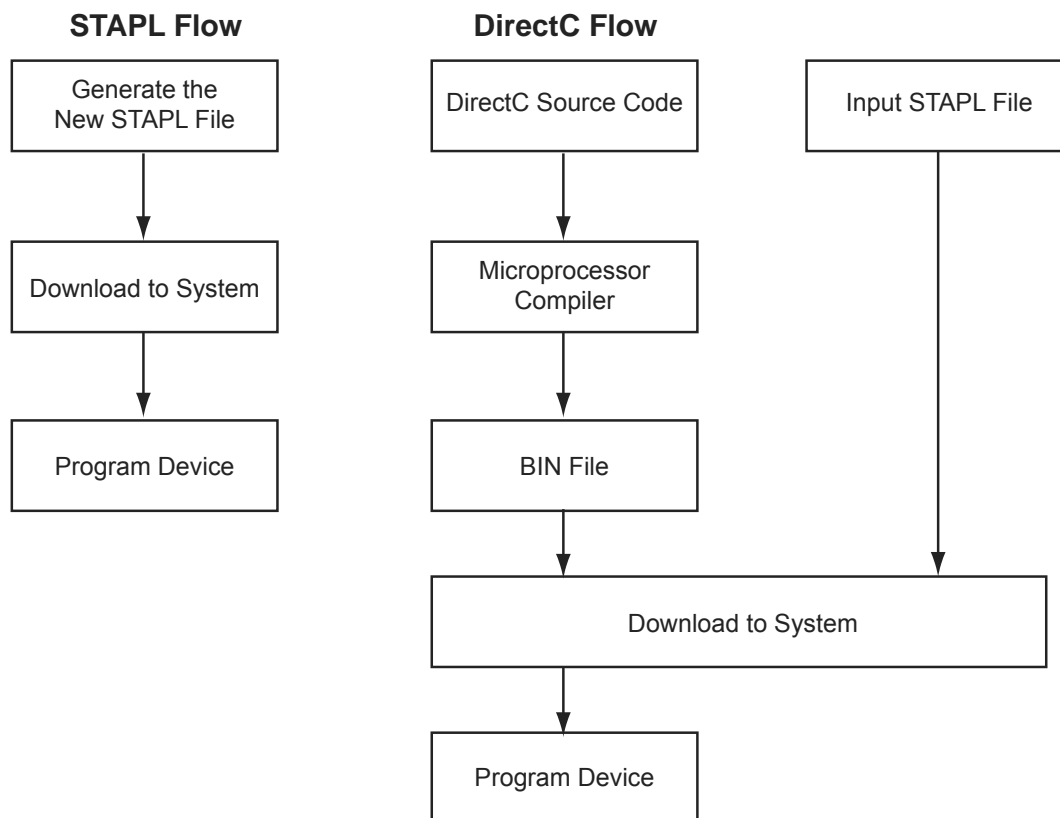
**Figure 15-2 • Device Programming Code Relationship**

The API is then linked with the STAPL player or DirectC and compiled using the microprocessor's compiler. Once the entire code is compiled, the user must download the resulting binary into the MCU system's program memory (such as ROM, EEPROM, or flash). The system is now ready for programming.

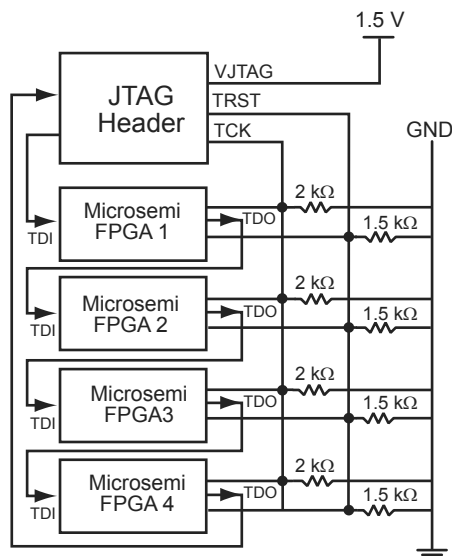
To program a design into the FPGA, the user creates a bitstream or STAPL file using the Microsemi Designer software, downloads it into the MCU system's volatile memory, and activates the stored programming binary file (Figure 15-3 on page 352). Once the programming is completed, the bitstream or STAPL file can be removed from the system, as the configuration profile is stored in the flash FPGA fabric and does not need to be reloaded at every system power-on.

## STAPL vs. DirectC

Programming the low power flash devices is performed using DirectC or the STAPL player. Both tools use the STAPL file as an input. DirectC is a compiled language, whereas STAPL is an interpreted language. Microprocessors will be able to load the FPGA using DirectC much more quickly than STAPL. This speed advantage becomes more apparent when lower clock speeds of 8- or 16-bit microprocessors are used. DirectC also requires less memory than STAPL, since the programming algorithm is directly implemented. STAPL does have one advantage over DirectC—the ability to upgrade. When a new programming algorithm is required, the STAPL user simply needs to regenerate a STAPL file using the latest version of the Designer software and download it to the system. The DirectC user must download the latest version of DirectC from Microsemi, compile everything, and download the result into the system (Figure 15-4).



**Figure 15-4 • STAPL vs. DirectC**



*Note: TCK is correctly wired with an equivalent tie-off resistance of 500  $\Omega$ , which satisfies the table for VJTAG of 1.5 V. The resistor values for TRST are not appropriate in this case, as the tie-off resistance of 375  $\Omega$  is below the recommended minimum for VJTAG = 1.5 V, but would be appropriate for a VJTAG setting of 2.5 V or 3.3 V.*

**Figure 16-3 • Parallel Resistance on JTAG Chain of Devices**

## Advanced Boundary Scan Register Settings

You will not be able to control the order in which I/Os are released from boundary scan control. Testing has produced cases where, depending on I/O placement and FPGA routing, a 5 ns glitch has been seen on exiting programming mode. The following setting is recommended to prevent such I/O glitches:

1. In the FlashPro software, configure the advanced BSR settings for **Specify I/O Settings During Programming**.
2. Set the input BSR cell to **Low** for the input I/O.

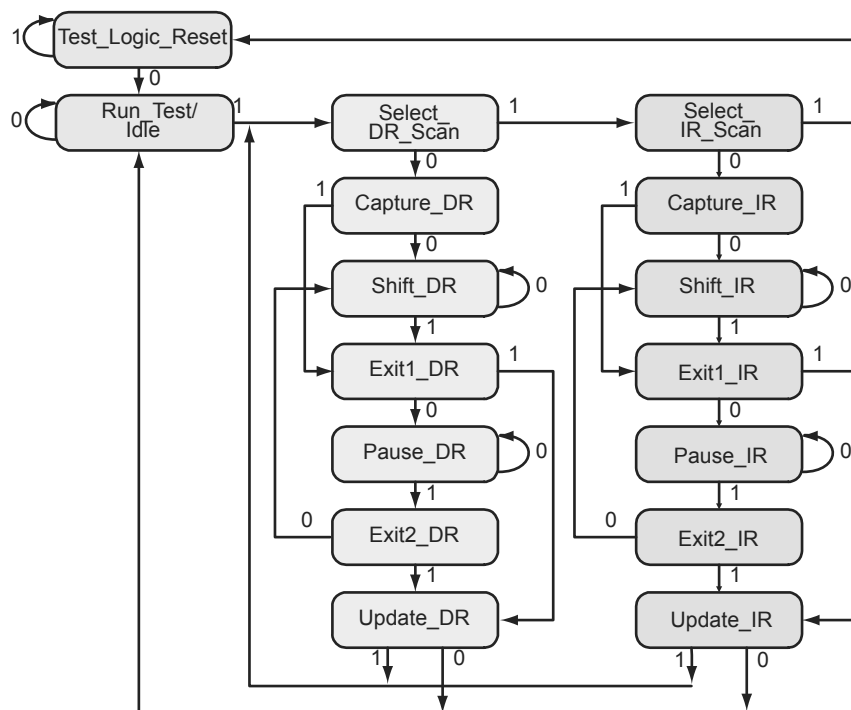


Figure 17-4 • TAP Controller State Diagram

## UJTAG Port Usage

UIREG[7:0] hold the contents of the JTAG instruction register. The UIREG vector value is updated when the TAP Controller state machine enters the Update\_IR state. Instructions 16 to 127 are user-defined and can be employed to encode multiple applications and commands within an application. Loading new instructions into the UIREG vector requires users to send appropriate logic to TMS to put the TAP Controller in a full IR cycle starting from the Select IR\_Scan state and ending with the Update\_IR state.

UTDI, UTDO, and UDRCK are directly connected to the JTAG TDI, TDO, and TCK ports, respectively. The TDI input can be used to provide either data (TAP Controller in the Shift\_DR state) or the new contents of the instruction register (TAP Controller in the Shift\_IR state).

UDRSH, UDRUPD, and UDRCAP are HIGH when the TAP Controller state machine is in the Shift\_DR, Update\_DR, and Capture\_DR states, respectively. Therefore, they act as flags to indicate the stages of the data shift process. These flags are useful for applications in which blocks of data are shifted into the design from JTAG pins. For example, an active UDRSH can indicate that UTDI contains the data bitstream, and UDRUPD is a candidate for the end-of-data-stream flag.

As mentioned earlier, users should not connect the TDI, TDO, TCK, TMS, and TRST ports of the UJTAG macro to any port or net of the design netlist. The Designer software will automatically handle the port connection.



Revision (month/year)	Chapter Affected	List of Changes (page number)
Revision 0 (continued)	"DDR for Microsemi's Low Power Flash Devices" was revised.	285
	"Programming Flash Devices" was revised.	298
	"In-System Programming (ISP) of Microsemi's Low Power Flash Devices Using FlashPro4/3/3X" was revised.	339
	"Core Voltage Switching Circuit for IGLOO and ProASIC3L In-System Programming" was revised.	347
	"Boundary Scan in Low Power Flash Devices" was revised.	362