



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

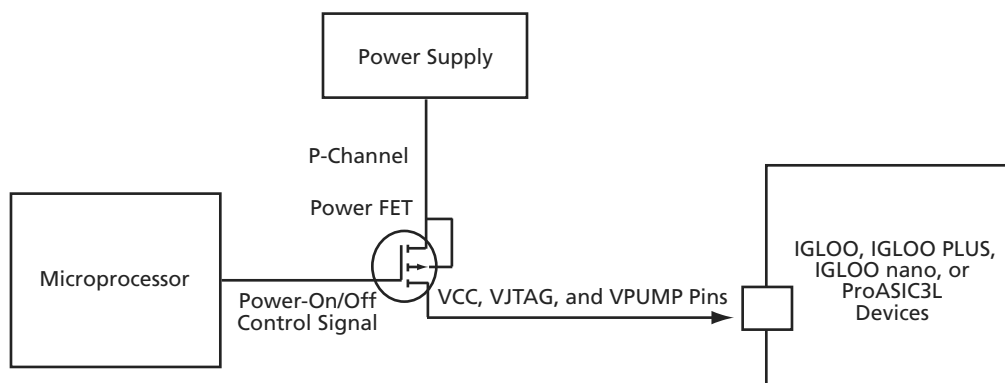
The versatility of Embedded - FPGAs makes them indispensable in numerous fields. In telecommunications.

#### Details

Product Status	Obsolete
Number of LABs/CLBs	-
Number of Logic Elements/Cells	-
Total RAM Bits	147456
Number of I/O	300
Number of Gates	1000000
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	<a href="https://www.e-xfl.com/product-detail/microsemi/a3p1000l-fgg484">https://www.e-xfl.com/product-detail/microsemi/a3p1000l-fgg484</a>

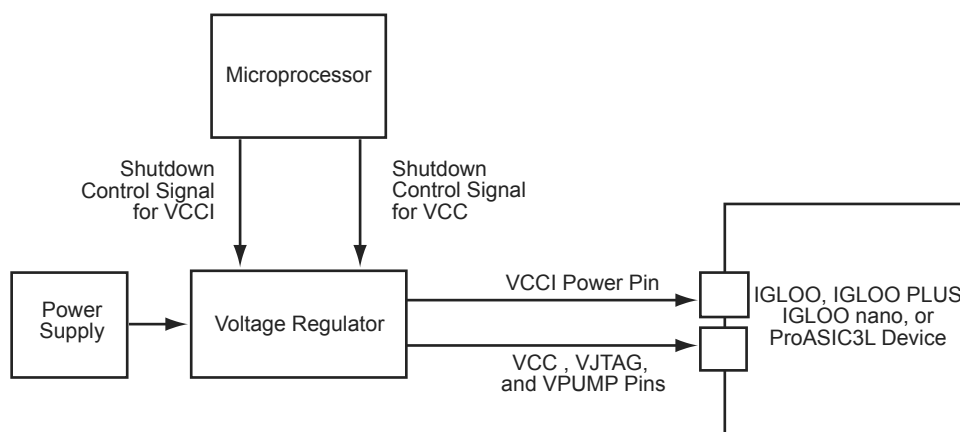
## Using Sleep and Shutdown Modes in the System

Depending on the power supply and the components used in an application, there are many ways to power on or off the power supplies connected to the device. For example, Figure 2-6 shows how a microprocessor can be used to control a power FET. Microsemi recommends that power FETs with low resistance be used to perform the switching action.



**Figure 2-6 • Controlling Power-On/Off State Using Microprocessor and Power FET**

Figure 2-7 shows how a microprocessor can be used with a voltage regulator's shutdown pin to turn on or off the power supplies connected to the device.



**Figure 2-7 • Controlling Power-On/Off State Using Microprocessor and Voltage Regulator**

## Power-Up/-Down Behavior

By design, all IGLOO, IGLOO nano, IGLOO PLUS, ProASIC3L, and RT ProASIC3 I/Os are in tristate mode before device power-up. The I/Os remain tristated until the last voltage supply ( $V_{CC}$  or  $V_{CCI}$ ) is powered to its activation level. After the last supply reaches its functional level, the outputs exit the tristate mode and drive the logic at the input of the output buffer. The behavior of user I/Os is independent of the  $V_{CC}$  and  $V_{CCI}$  sequence or the state of other voltage supplies of the FPGA ( $V_{PUMP}$  and  $V_{JTAG}$ ). During power-down, device I/Os become tristated once the first power supply ( $V_{CC}$  or  $V_{CCI}$ ) drops below its deactivation voltage level. The I/O behavior during power-down is also independent of voltage supply sequencing.

Figure 2-8 on page 34 shows a timing diagram when the  $V_{CC}$  power supply crosses the activation and deactivation trip points in a typical application when the  $V_{CC}$  power supply ramp-rate is 100  $\mu$ s (ramping from 0 V to 1.5 V in this example). This is the timing diagram for the FPGA entering and exiting Sleep mode, as this function is dependent on powering  $V_{CC}$  down or up. Depending on the ramp-rate of the

- The INBUF\_FF must be driven by a top-level input port of the design.
- The INBUF\_FF AND the ULSICC macro must be used to enable type 2 Flash\*Freeze mode.
- For type 2 Flash\*Freeze mode, the INBUF\_FF MUST drive some logic in the design.
- For type 1 Flash\*Freeze mode, the INBUF\_FF may drive some logic in the design, but it may also be left floating.
- Only one INBUF\_FF may be instantiated in a device.
- The FF pin threshold voltages are defined by VCCI and the supported single-ended I/O standard in the corresponding I/O bank.
- The FF pin Schmitt trigger option may be configured in the I/O attribute editor in Microsemi's Designer software. The Schmitt trigger option is only available for IGLOOe, IGLOO nano, IGLOO PLUS, ProASIC3EL, and RT ProASIC3 devices.
- A 2 ns glitch filter resides in the Flash\*Freeze Technology block to filter unwanted glitches on the FF pin.

## ULSICC

The User Low Static ICC (ULSICC) macro allows the FPGA core to access the Flash\*Freeze Technology block so that entering and exiting Flash\*Freeze mode can be controlled by the user's design. The ULSICC macro enables a hard block with an available LSICC input port, as shown in Figure 2-3 on page 27 and Figure 2-10 on page 37. Design rules for the ULSICC macro are as follows:

- The ULSICC macro by itself cannot enable Flash\*Freeze mode. The INBUF\_FF AND the ULSICC macro must both be used to enable type 2 Flash\*Freeze mode.
- The ULSICC controls entering the Flash\*Freeze mode by asserting the LSICC input (logic '1') of the ULSICC macro. The FF pin must also be asserted (logic '0') to enter Flash\*Freeze mode.
- When the LSICC signal is '0', the device cannot enter Flash\*Freeze mode; and if already in Flash\*Freeze mode, it will exit.
- When the ULSICC macro is not instantiated in the user's design, the LSICC port will be tied High.

## Flash\*Freeze Management IP

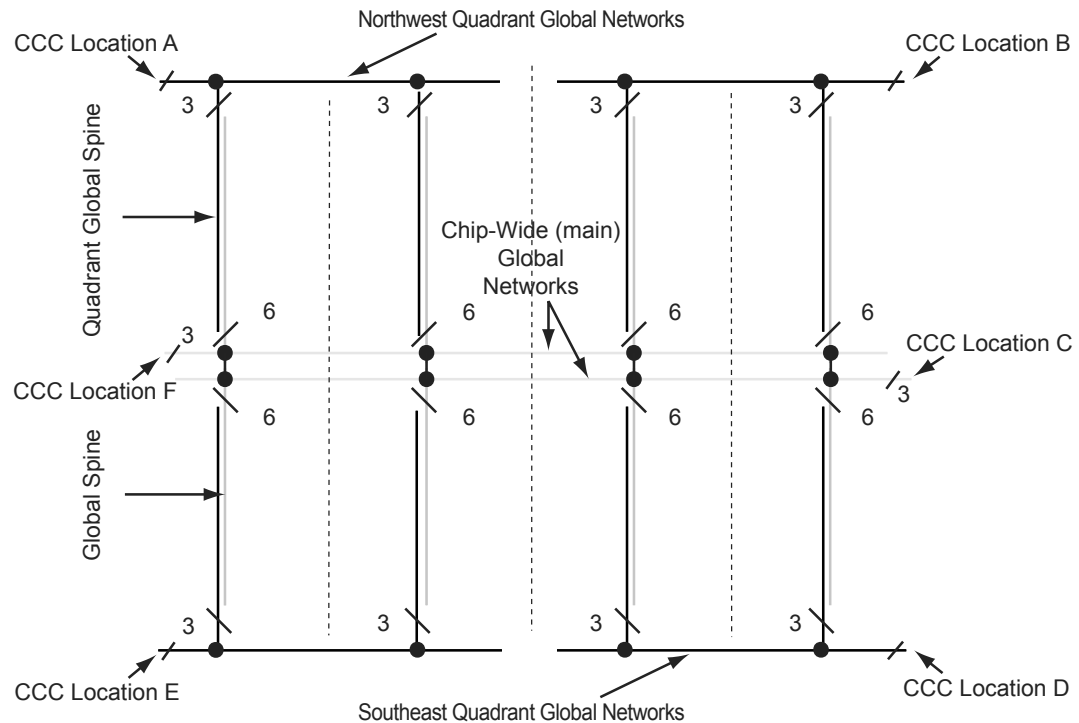
The Flash\*Freeze management IP can be configured with the Libero (or SmartGen) core generator in a simple, intuitive interface. With the core configuration tool, users can select the number of clocks to be gated, and select whether or not to implement housekeeping. All port names on the Flash\*Freeze management IP block can be renamed by the user.

- The clock gating (filter) blocks include CLKINT buffers for each gated clock output (version 8.3).
- When housekeeping is NOT used, the WAIT\_HOUSEKEEPING signal will be automatically fed back into DONE\_HOUSEKEEPING inside the core, and the ports will not be available at the IP core interface.
- The INBUF\_FF macro is automatically instantiated within the IP core.
- The INBUF\_FF port (default name is "Flash\_Freeze\_N") must be connected to a top-level input port of the design.
- The ULSICC macro is automatically instantiated within the IP core, and the LSICC signal is driven by the FSM.
- Timing analysis can be performed on the clock domain of the source clock (i.e., input to the clock gating filters). For example, if CLKIn becomes CLKIn\_gated, the timing can be performed on the CLKIn domain in SmartTime.
- The gated clocks can be added to the clock list if the user wishes to analyze these clocks specifically. The user can locate the gated clocks by looking for instance names such as those below:

```
Top/ff1/ff_1_wrapper_inst/user_ff_1_wrapper/Primary_Filter_Instance/  
Latch_For_Clock_Gating:Q  
Top/ff1/ff_1_wrapper_inst/user_ff_1_wrapper/genblk1.genblk2.secondary_filter[0].  
secondary_filter_instance/Latch_For_Clock_Gating:Q  
Top/ff1/ff_1_wrapper_inst/user_ff_1_wrapper/genblk1.genblk2.secondary_filter[1].  
secondary_filter_instance/Latch_For_Clock_Gating:Q
```

## CCC Locations

CCCs located in the middle of the east and west sides of the device access the three VersaNet global networks on each side (six total networks), while the four CCCs located in the four corners access three quadrant global networks (twelve total networks). See Figure 4-13.



**Figure 4-13 • Global Network Architecture for 60 k Gate Devices and Above**

The following explains the locations of the CCCs in IGLOO and ProASIC3 devices:

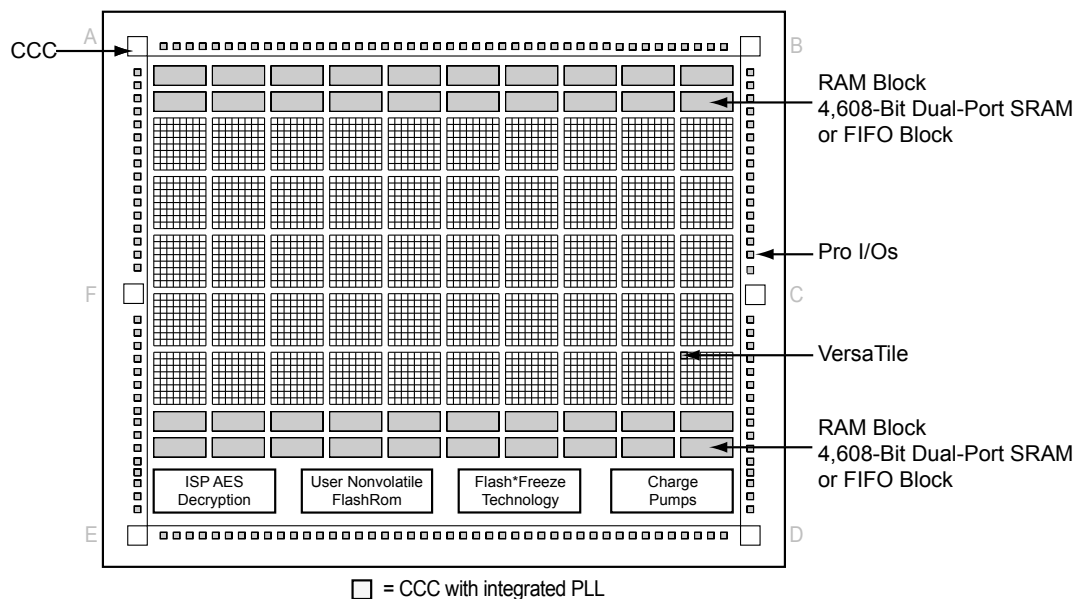
In Figure 4-15 on page 98 through Figure 4-16 on page 98, CCCs with integrated PLLs are indicated in red, and simplified CCCs are indicated in yellow. There is a letter associated with each location of the CCC, in clockwise order. The upper left corner CCC is named "A," the upper right is named "B," and so on. These names finish up at the middle left with letter "F."



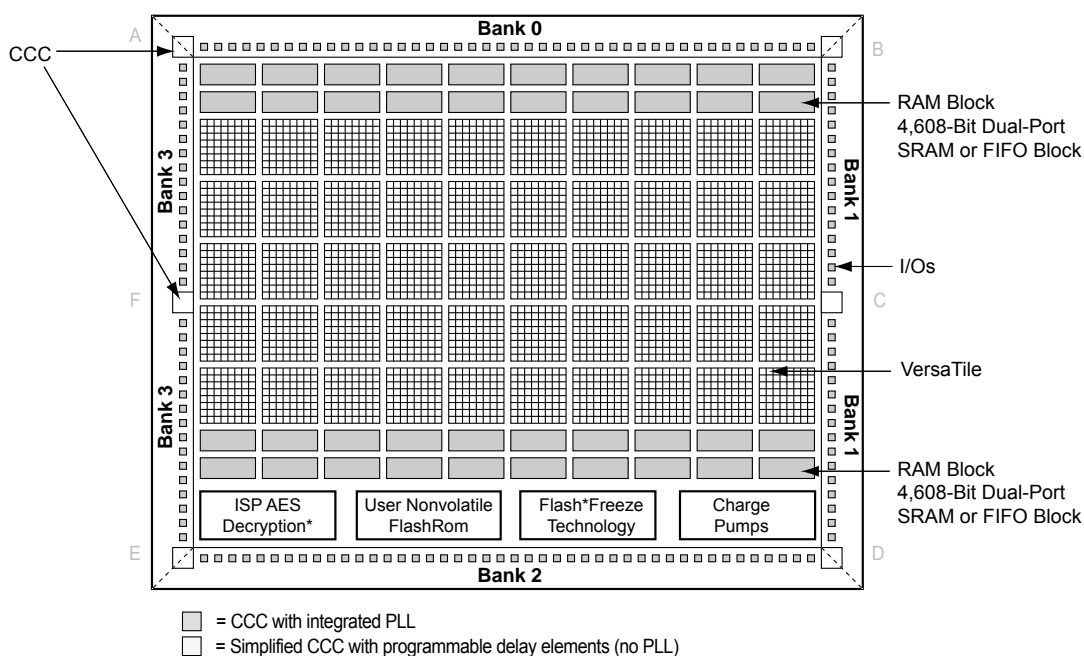
## IGLOOe and ProASIC3E CCC Locations

IGLOOe and ProASIC3E devices have six CCCs—one in each of the four corners and one each in the middle of the east and west sides of the device (Figure 4-15).

All six CCCs are integrated with PLLs, except in PQFP-208 package devices. PQFP-208 package devices also have six CCCs, of which two include PLLs and four are simplified CCCs. The CCCs with PLLs are implemented in the middle of the east and west sides of the device (middle right and middle left). The simplified CCCs without PLLs are located in the four corners of the device (Figure 4-16).



**Figure 4-15 • CCC Locations in IGLOOe and ProASIC3E Family Devices (except PQFP-208 package)**



**Figure 4-16 • CCC Locations in ProASIC3E Family Devices (PQFP-208 package)**

**Table 4-13 • 2-Bit Feedback MUX**

FBSEL<1:0> State	MUX Input Selected
0	Ground. Used for power-down mode in power-down logic block.
1	PLL VCO 0° phase shift
2	PLL delayed VCO 0° phase shift
3	N/A

**Table 4-14 • Programmable Delay Selection for Feedback Delay and Secondary Core Output Delays**

FBDLY<4:0>; DLYYB<4:0>; DLYYC<4:0> State	Delay Value
0	Typical delay = 600 ps
1	Typical delay = 760 ps
2	Typical delay = 920 ps
⋮	⋮
31	Typical delay = 5.56 ns

**Table 4-15 • Programmable Delay Selection for Global Clock Output Delays**

DLYGLA<4:0>; DLYGLB<4:0>; DLYGLC<4:0> State	Delay Value
0	Typical delay = 225 ps
1	Typical delay = 760 ps
2	Typical delay = 920 ps
⋮	⋮
31	Typical delay = 5.56 ns

**Table 4-16 • Fusion Dynamic CCC Clock Source Selection**

RXASEL	DYNASEL	Source of CLKA
1	0	RC Oscillator
1	1	Crystal Oscillator
RXBSEL	DYNBSEL	Source of CLKB
1	0	RC Oscillator
1	1	Crystal Oscillator
RXCSEL	DYNCSEL	Source of CLKC
1	0	RC Oscillator
1	1	Crystal Oscillator

**Table 4-17 • Fusion Dynamic CCC NGMUX Configuration**

GLMUXCFG<1:0>	NGMUX Select Signal	Supported Input Clocks to NGMUX
00	0	GLA
	1	GLC
01	0	GLA
	1	GLINT
10	0	GLC
	1	GLINT

## Conclusion

The advanced CCCs of the IGLOO and ProASIC3 devices are ideal for applications requiring precise clock management. They integrate easily with the internal low-skew clock networks and provide flexible frequency synthesis, clock deskewing, and/or time-shifting operations.

## Related Documents

### Application Notes

*Board-Level Considerations*

[http://www.microsemi.com/soc/documents/ALL\\_AC276\\_AN.pdf](http://www.microsemi.com/soc/documents/ALL_AC276_AN.pdf)

### Datasheets

*Fusion Family of Mixed Signal FPGAs*

[http://www.microsemi.com/soc/documents/Fusion\\_DS.pdf](http://www.microsemi.com/soc/documents/Fusion_DS.pdf)

### User's Guides

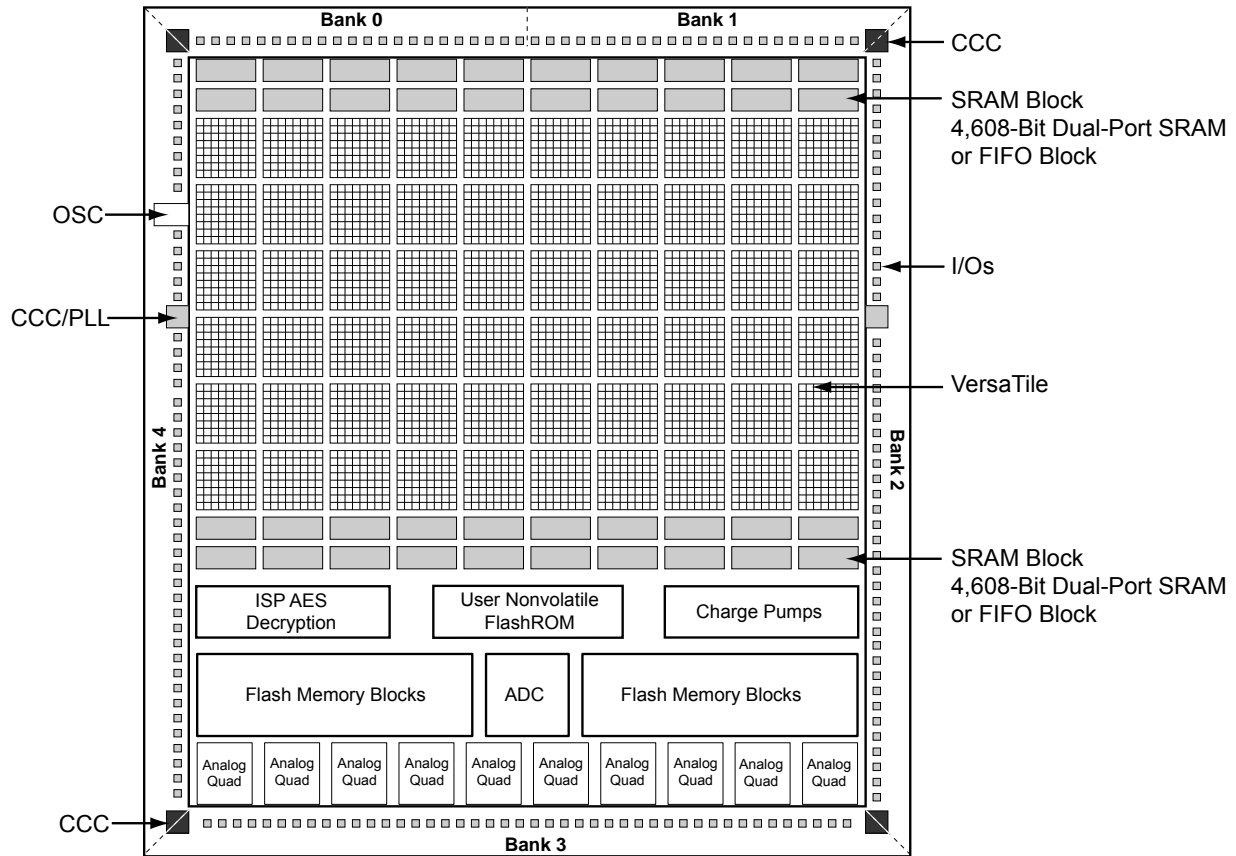
*IGLOO, ProASIC3, SmartFusion, and Fusion Macro Library Guide*

[http://www.microsemi.com/soc/documents/pa3\\_libguide\\_ug.pdf](http://www.microsemi.com/soc/documents/pa3_libguide_ug.pdf)

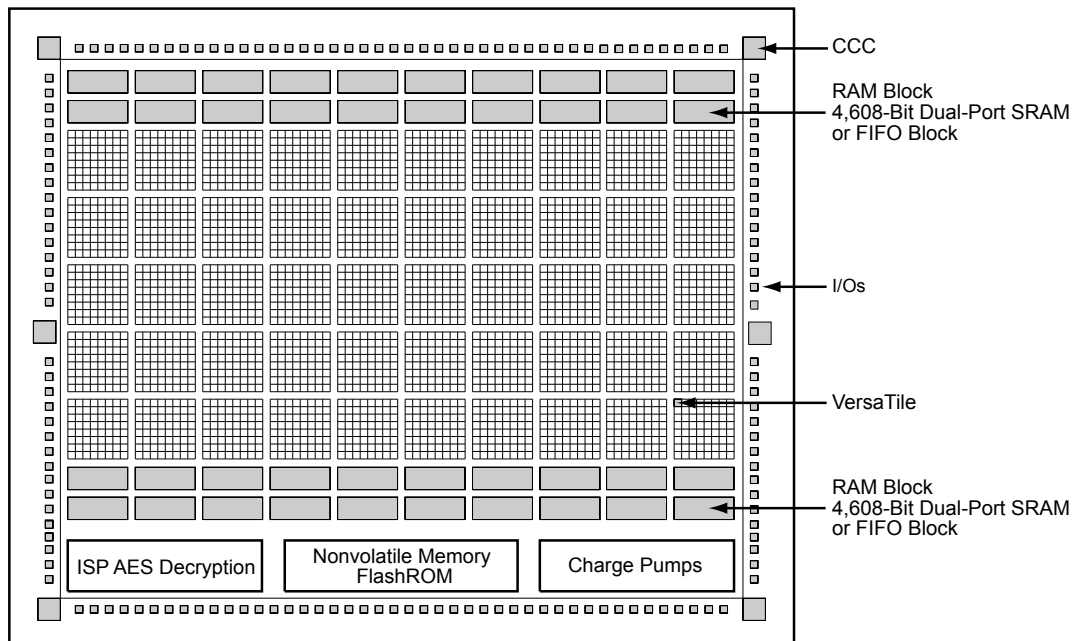
## List of Changes

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

Date	Changes	Page
August 2012	The "Implementing EXTFB in ProASIC3/E Devices" section is new (SAR 36647).	86
	Table 4-7 • Delay Values in Libero SoC Software per Device Family was added to the "Clock Delay Adjustment" section (SAR 22709).	102
	The "Phase Adjustment" section was rewritten to explain better why the visual CCC shows both the actual phase and the actual delay that is equivalent to this phase shift (SAR 29647).	103
	The hyperlink for the <i>Board-Level Considerations</i> application note was corrected (SAR 36663)	128, 129
December 2011	Figure 4-20 • PLL Block Diagram, Figure 4-22 • CCC Block Control Bits – Graphical Representation of Assignments, and Table 4-12 • MUXA, MUXB, MUXC were revised to change the phase shift assignments for PLLs 4 through 7 (SAR 33791).	101, 105, 109
June 2011	The description for RESETEN in Table 4-8 • Configuration Bit Descriptions for the CCC Blocks was revised. The phrase "and should not be modified via dynamic configuration" was deleted because RESETEN is read only (SAR 25949).	106
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



**Figure 5-2 • Fusion Device Architecture Overview (AFS600)**



**Figure 5-3 • ProASIC3 and IGLOO Device Architecture**

## RD

This is the output data bus and is 18 bits wide. Not all 18 bits are valid in all configurations. Like the WD bus, high-order bits become unusable if the data width is less than 18. The output data on unused pins is undefined (Table 6-7).

**Table 6-7 • Input Data Signal Usage for Different Aspect Ratios**

D×W	WD/RD Unused
4k×1	WD[17:1], RD[17:1]
2k×2	WD[17:2], RD[17:2]
1k×4	WD[17:4], RD[17:4]
512×9	WD[17:9], RD[17:9]
256×18	—

## ESTOP, FSTOP

ESTOP is used to stop the FIFO read counter from further counting once the FIFO is empty (i.e., the EMPTY flag goes HIGH). A HIGH on this signal inhibits the counting.

FSTOP is used to stop the FIFO write counter from further counting once the FIFO is full (i.e., the FULL flag goes HIGH). A HIGH on this signal inhibits the counting.

For more information on these signals, refer to the "ESTOP and FSTOP Usage" section.

## FULL, EMPTY

When the FIFO is full and no more data can be written, the FULL flag asserts HIGH. The FULL flag is synchronous to WCLK to inhibit writing immediately upon detection of a full condition and to prevent overflows. Since the write address is compared to a resynchronized (and thus time-delayed) version of the read address, the FULL flag will remain asserted until two WCLK active edges after a read operation eliminates the full condition.

When the FIFO is empty and no more data can be read, the EMPTY flag asserts HIGH. The EMPTY flag is synchronous to RCLK to inhibit reading immediately upon detection of an empty condition and to prevent underflows. Since the read address is compared to a resynchronized (and thus time-delayed) version of the write address, the EMPTY flag will remain asserted until two RCLK active edges after a write operation removes the empty condition.

For more information on these signals, refer to the "FIFO Flag Usage Considerations" section on page 161.

## AFULL, AEMPTY

These are programmable flags and will be asserted on the threshold specified by AFVAL and AEVAL, respectively.

When the number of words stored in the FIFO reaches the amount specified by AEVAL while reading, the AEMPTY output will go HIGH. Likewise, when the number of words stored in the FIFO reaches the amount specified by AFVAL while writing, the AFULL output will go HIGH.

## AFVAL, AEVAL

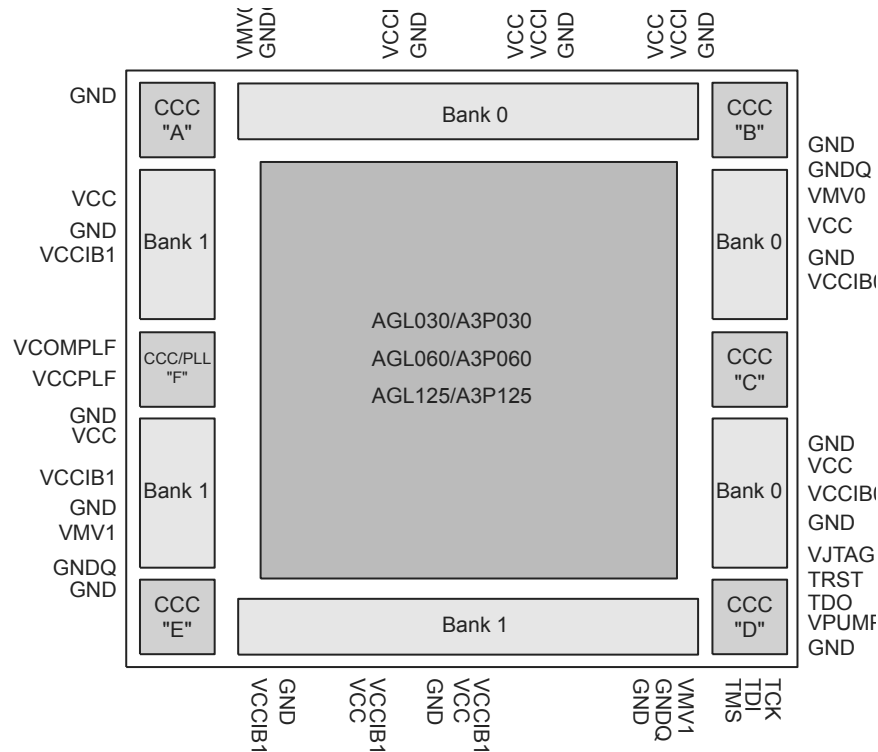
The AEVAL and AFVAL pins are used to specify the almost-empty and almost-full threshold values. They are 12-bit signals. For more information on these signals, refer to the "FIFO Flag Usage Considerations" section on page 161.

## FIFO Usage

### ESTOP and FSTOP Usage

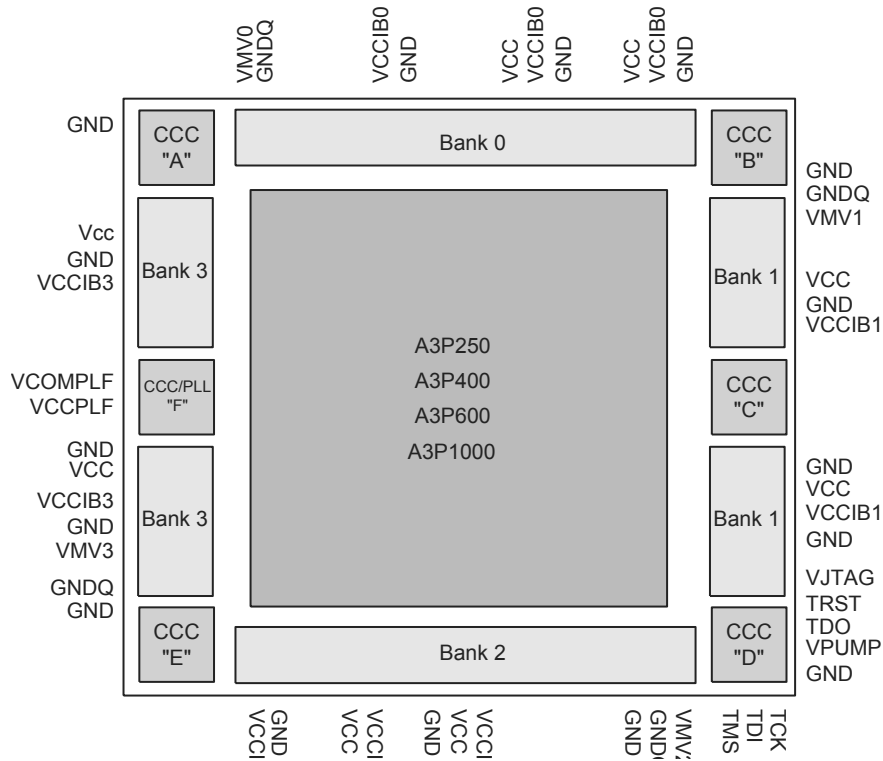
The ESTOP pin is used to stop the read counter from counting any further once the FIFO is empty (i.e., the EMPTY flag goes HIGH). Likewise, the FSTOP pin is used to stop the write counter from counting any further once the FIFO is full (i.e., the FULL flag goes HIGH).

The FIFO counters in the device start the count at zero, reach the maximum depth for the configuration (e.g., 511 for a 512×9 configuration), and then restart at zero. An example application for ESTOP, where the read counter keeps counting, would be writing to the FIFO once and reading the same content over and over without doing another write.



Note: The 30 k gate devices do not support a PLL ( $V_{COMPLF}$  and  $V_{CCPLF}$  pins).

**Figure 7-19 • Naming Conventions of IGLOO and ProASIC3 Devices with Two I/O Banks – Top View**



**Figure 7-20 • Naming Conventions of IGLOO and ProASIC3 Devices with Four I/O Banks – Top View**

## 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

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 in the *IGLOOe FPGA Fabric User's Guide* or *ProASIC3E 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 8-6 on page 218), 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."

## Volume Programming Services

### Device Type Supported: Flash and Antifuse

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

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

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

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

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

- **Microsemi In-House Programming**

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

- **Distributor Programming Centers**

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



- Programming Centers

Microsemi programming hardware policy also applies to programming centers. Microsemi expects all programming centers to use certified programmers to program Microsemi devices. If a programming center uses noncertified programmers to program Microsemi devices, the "Noncertified Programmers" policy applies.

## Important Programming Guidelines

### Preprogramming Setup

Before programming, several steps are required to ensure an optimal programming yield.

#### ***Use Proper Handling and Electrostatic Discharge (ESD) Precautions***

Microsemi FPGAs are sensitive electronic devices that are susceptible to damage from ESD and other types of mishandling. For more information about ESD, refer to the *Quality and Reliability Guide*, beginning with page 41.

#### ***Use the Latest Version of the Designer Software to Generate Your Programming File (recommended)***

The files used to program Microsemi flash devices (\*.bit, \*.stp, \*.pdb) contain important information about the switches that will be programmed in the FPGA. Find the latest version and corresponding release notes at <http://www.microsemi.com/soc/download/software/designer/>. Also, programming files must always be zipped during file transfer to avoid the possibility of file corruption.

#### ***Use the Latest Version of the Programming Software***

The programming software is frequently updated to accommodate yield enhancements in FPGA manufacturing. These updates ensure maximum programming yield and minimum programming times. Before programming, always check the version of software being used to ensure it is the most recent. Depending on the programming software, refer to one of the following:

- FlashPro: [http://www.microsemi.com/soc/download/program\\_debug/flashpro/](http://www.microsemi.com/soc/download/program_debug/flashpro/)
- Silicon Sculptor: [http://www.microsemi.com/soc/download/program\\_debug/ss/](http://www.microsemi.com/soc/download/program_debug/ss/)

#### ***Use the Most Recent Adapter Module with Silicon Sculptor***

Occasionally, Microsemi makes modifications to the adapter modules to improve programming yields and programming times. To identify the latest version of each module before programming, visit [http://www.microsemi.com/soc/products/hardware/program\\_debug/ss/modules.aspx](http://www.microsemi.com/soc/products/hardware/program_debug/ss/modules.aspx).

### ***Perform Routine Hardware Self-Diagnostic Test***

- Adapter modules must be regularly cleaned. Adapter modules need to be inserted carefully into the programmer to make sure the DIN connectors (pins at the back side) are not damaged.
- FlashPro

The self-test is only applicable when programming with FlashPro and FlashPro3 programmers. It is not supported with FlashPro4 or FlashPro Lite. To run the self-diagnostic test, follow the instructions given in the "Performing a Self-Test" section of [http://www.microsemi.com/soc/documents/FlashPro\\_UG.pdf](http://www.microsemi.com/soc/documents/FlashPro_UG.pdf).

- Silicon Sculptor

The self-diagnostic test verifies correct operation of the pin drivers, power supply, CPU, memory, and adapter module. This test should be performed with an adapter module installed and before every programming session. At minimum, the test must be executed every week. To perform self-diagnostic testing using the Silicon Sculptor software, perform the following steps, depending on the operating system:

- DOS: From anywhere in the software, type **ALT + D**.
- Windows: Click **Device** > choose **Actel Diagnostic** > select the **Test** tab > click **OK**.

Silicon Sculptor programmers must be verified annually for calibration. Refer to the *Silicon Sculptor Verification of Calibration Work Instruction* document on the website.

## Cortex-M1 Device Security

Cortex-M1-enabled devices are shipped with the following security features:

- FPGA array enabled for AES-encrypted programming and verification
- FlashROM enabled for AES-encrypted Write and Verify
- Fusion Embedded Flash Memory enabled for AES-encrypted Write

## AES Encryption of Programming Files

Low power flash devices employ AES as part of the security mechanism that prevents invasive and noninvasive attacks. The mechanism entails encrypting the programming file with AES encryption and then passing the programming file through the AES decryption core, which is embedded in the device. The file is decrypted there, and the device is successfully programmed. The AES master key is stored in on-chip nonvolatile memory (flash). The AES master key can be preloaded into parts in a secure programming environment (such as the Microsemi In-House Programming center), and then "blank" parts can be shipped to an untrusted programming or manufacturing center for final personalization with an AES-encrypted bitstream. Late-stage product changes or personalization can be implemented easily and securely by simply sending a STAPL file with AES-encrypted data. Secure remote field updates over public networks (such as the Internet) are possible by sending and programming a STAPL file with AES-encrypted data.

The AES key protects the programming data for file transfer into the device with 128-bit AES encryption. If AES encryption is used, the AES key is stored or preprogrammed into the device. To program, you must use an AES-encrypted file, and the encryption used on the file must match the encryption key already in the device.

The AES key is protected by a FlashLock security Pass Key that is also implemented in each device. The AES key is always protected by the FlashLock Key, and the AES-encrypted file does NOT contain the FlashLock Key. This FlashLock Pass Key technology is exclusive to the Microsemi flash-based device families. FlashLock Pass Key technology can also be implemented without the AES encryption option, providing a choice of different security levels.

In essence, security features can be categorized into the following three options:

- AES encryption with FlashLock Pass Key protection
- FlashLock protection only (no AES encryption)
- No protection

Each of the above options is explained in more detail in the following sections with application examples and software implementation options.

## Advanced Encryption Standard

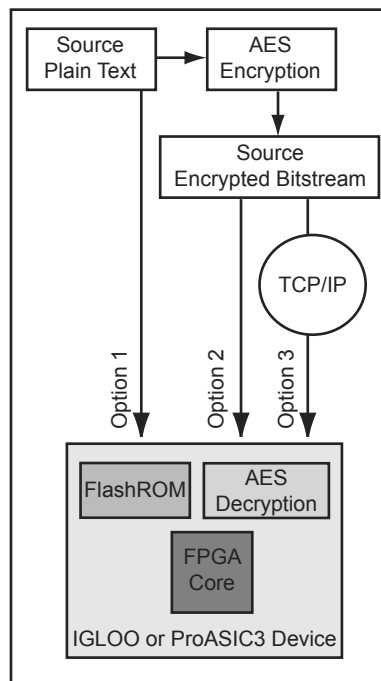
The 128-bit AES standard (FIPS-192) block cipher is the NIST (National Institute of Standards and Technology) replacement for DES (Data Encryption Standard FIPS46-2). AES has been designed to protect sensitive government information well into the 21st century. It replaces the aging DES, which NIST adopted in 1977 as a Federal Information Processing Standard used by federal agencies to protect sensitive, unclassified information. The 128-bit AES standard has  $3.4 \times 10^{38}$  possible 128-bit key variants, and it has been estimated that it would take 1,000 trillion years to crack 128-bit AES cipher text using exhaustive techniques. Keys are stored (securely) in low power flash devices in nonvolatile flash memory. All programming files sent to the device can be authenticated by the part prior to programming to ensure that bad programming data is not loaded into the part that may possibly damage it. All programming verification is performed on-chip, ensuring that the contents of low power flash devices remain secure.

Microsemi has implemented the 128-bit AES (Rijndael) algorithm in low power flash devices. With this key size, there are approximately  $3.4 \times 10^{38}$  possible 128-bit keys. DES has a 56-bit key size, which provides approximately  $7.2 \times 10^{16}$  possible keys. In their AES fact sheet, the National Institute of Standards and Technology uses the following hypothetical example to illustrate the theoretical security provided by AES. If one were to assume that a computing system existed that could recover a DES key in a second, it would take that same machine approximately 149 trillion years to crack a 128-bit AES key. NIST continues to make their point by stating the universe is believed to be less than 20 billion years old.<sup>1</sup>

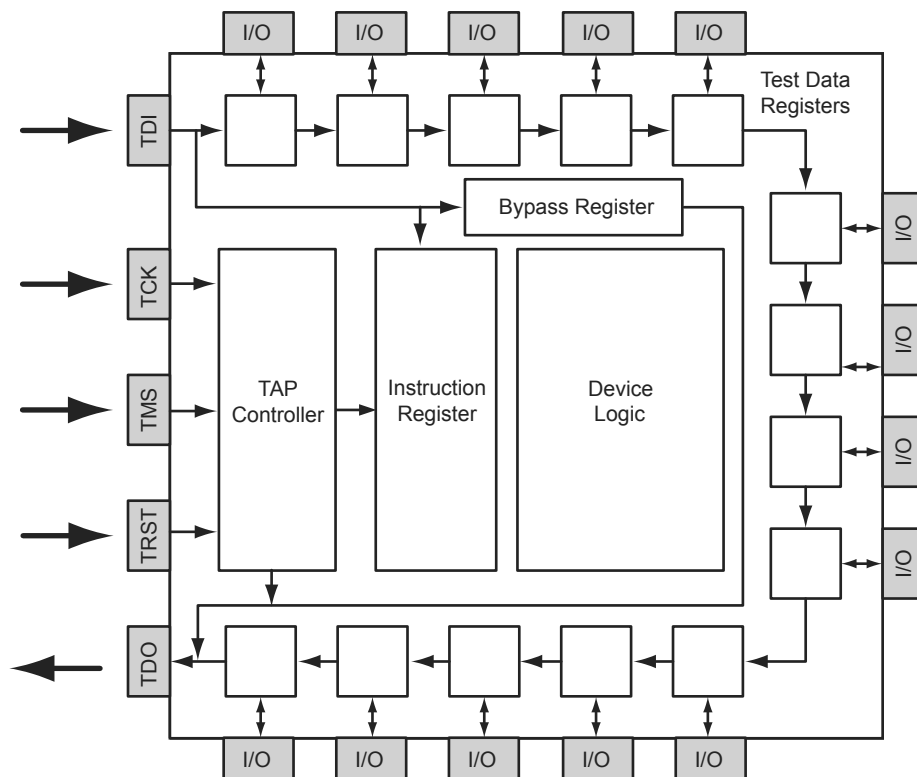
Figure 13-2 shows different applications for ISP programming.

1. In a trusted programming environment, you can program the device using the unencrypted (plaintext) programming file.
2. You can program the AES Key in a trusted programming environment and finish the final programming in an untrusted environment using the AES-encrypted (cipher text) programming file.
3. For the remote ISP updating/reprogramming, the AES Key stored in the device enables the encrypted programming bitstream to be transmitted through the untrusted network connection.

Microsemi low power flash devices also provide the unique Microsemi FlashLock feature, which protects the Pass Key and AES Key. Unless the original FlashLock Pass Key is used to unlock the device, security settings cannot be modified. Microsemi does not support read-back of FPGA core-programmed data; however, the FlashROM contents can selectively be read back (or disabled) via the JTAG port based on the security settings established by the Microsemi Designer software. Refer to the "Security in Low Power Flash Devices" section on page 301 for more information.



**Figure 13-2 • Different ISP Use Models**



**Figure 16-2 • Boundary Scan Chain**

## Board-Level Recommendations

Table 16-3 gives pull-down recommendations for the TRST and TCK pins.

**Table 16-3 • TRST and TCK Pull-Down Recommendations**

VJTAG	Tie-Off Resistance*
VJTAG at 3.3 V	200 $\Omega$ to 1 k $\Omega$
VJTAG at 2.5 V	200 $\Omega$ to 1 k $\Omega$
VJTAG at 1.8 V	500 $\Omega$ to 1 k $\Omega$
VJTAG at 1.5 V	500 $\Omega$ to 1 k $\Omega$
VJTAG at 1.2 V	TBD

*Note:* Equivalent parallel resistance if more than one device is on JTAG chain (Figure 16-3)

## Power-Up/-Down Sequence and Transient Current

Microsemi's low power flash devices use the following main voltage pins during normal operation:<sup>2</sup>

- 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>.

## Related Documents

### Datasheets

*ProASIC3 Flash Family FPGAs*

[http://www.microsemi.com/soc/documents/PA3\\_DS.pdf](http://www.microsemi.com/soc/documents/PA3_DS.pdf)

*ProASIC3E Flash Family FPGAs*

[http://www.microsemi.com/soc/documents/PA3E\\_DS.pdf](http://www.microsemi.com/soc/documents/PA3E_DS.pdf)

## List of Changes

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

Date	Changes	Page
v1.2 (December 2008)	IGLOO nano and ProASIC3 nano devices were added to the document as supported device types.	
v1.1 (October 2008)	The "Introduction" section was updated to add Military ProASIC3EL and RT ProASIC3 devices to the list of devices that can have inputs driven in while the device is not powered.	373
	The "Flash Devices Support Power-Up Behavior" section was revised to include new families and make the information more concise.	374
	The "Cold-Sparing" section was revised to add Military ProASIC3/EL and RT ProASIC3 devices to the lists of devices with and without cold-sparing support.	382
	The "Hot-Swapping" section was revised to add Military ProASIC3/EL and RT ProASIC3 devices to the lists of devices with and without hot-swap support. AGL400 was added to the list of devices that do not support hot-swapping.	383
v1.0 (August 2008)	This document was revised, renamed, and assigned a new part number. It now includes data for the IGLOO and ProASIC3L families.	N/A
v1.3 (March 2008)	The "List of Changes" section was updated to include the three different I/O Structure handbook chapters.	384
v1.2 (February 2008)	The first sentence of the "PLL Behavior at Brownout Condition" section was updated to read, "When PLL power supply voltage and/or $V_{CC}$ levels drop below the VCC brownout levels ( $0.75\text{ V} \pm 0.25\text{ V}$ ), the PLL output lock signal goes low and/or the output clock is lost."	381
v1.1 (January 2008)	The "PLL Behavior at Brownout Condition" section was added.	381

---

# Index

---

## A

AES encryption 305

architecture 147

four I/O banks 13

global 47

IGLOO 12

IGLOO nano 11

IGLOO PLUS 13

IGLOOe 14

ProASIC3 nano 11

ProASIC3E 14

routing 18

spine 57

SRAM and FIFO 151

architecture overview 11

array coordinates 16

## B

boundary scan 357

board-level recommendations 360

chain 359

opcodes 359

brownout voltage 381

## C

CCC 98

board-level considerations 128

cascading 125

Fusion locations 99

global resources 78

hardwired I/O clock input 124

IGLOO locations 97

IGLOOe locations 98

locations 96

overview 77

ProASIC3 locations 97

ProASIC3E locations 98

programming 78

software configuration 112

with integrated PLLs 95

without integrated PLLs 95

chip global aggregation 59

CLKDLY macro 81

clock aggregation 60

clock macros 62

clock sources

core logic 92

PLL and CLKDLY macros 89

clocks

delay adjustment 102

detailed usage information 120

multipliers and dividers 101

phase adjustment 103

physical constraints for quadrant clocks 124

SmartGen settings 121

static timing analysis 123

cold-sparing 382

compiling 261

report 261

contacting Microsemi SoC Products Group

customer service 387

email 387

web-based technical support 387

context save and restore 34

customer service 387

## D

DDR

architecture 271

design example 282

I/O options 273

input/output support 275

instantiating registers 276

design example 71

design recommendations 62

device architecture 147

DirectC 346

DirectC code 351

## E

efficient long-line resources 19

encryption 355

## F

FIFO

features 157

initializing 164

memory block consumption 163

software support 170

usage 160

flash switch for programming 9

Flash\*Freeze

design flow 39

design guide 34

device behavior 30

I/O state 28

management IP 36

pin locations 31

type 1 24

type 2 26

ULSICC 40

Flash\*Freeze mode 24

- sleep 32
- static 23
- summary 23
- product support
  - customer service 387
  - email 387
  - My Cases 388
  - outside the U.S. 388
  - technical support 387
  - website 387
- programmers 291
  - device support 294
- programming
  - AES encryption 319
  - basics 289
  - features 289
  - file header definition 323
  - flash and antifuse 291
  - flash devices 289
  - glossary 324
  - guidelines for flash programming 295
  - header pin numbers 336
  - microprocessor 349
  - power supplies 329
  - security 313
  - solution 334
  - solutions 293
  - voltage 329
  - volume services 292
- programming support 287

## R

- RAM
  - memory block consumption 163
- remote upgrade via TCP/IP 354
- routing structure 18

## S

- security 330
  - architecture 303
  - encrypted programming 354
  - examples 308
  - features 304
  - FlashLock 307
  - FlashROM 137
  - FlashROM use models 311
  - in programmable logic 301
  - overview 301
- shutdown mode 32
  - context save and restore 34
- signal integrity problem 337
- silicon testing 370
- sleep mode 32

- context save and restore 34
- SmartGen 170
- spine architecture 57
- spine assignment 68
- SRAM
  - features 153
  - initializing 164
  - software support 170
  - usage 157
- STAPL player 351
- STAPL vs. DirectC 353
- static mode 23
- switching circuit 344
  - verification 344
- synthesizing 258

## T

- TAP controller state machine 357, 366
- tech support
  - ITAR 388
  - My Cases 388
  - outside the U.S. 388
- technical support 387
- transient current
  - VCC 376
  - VCCI 376
- transient current, power-up/-down 375

## U

- UJTAG
  - CCC dynamic reconfiguration 368
  - fine tuning 369
  - macro 365
  - operation 366
  - port usage 367
  - use to read FlashROM contents 363
- ULSICC 40
- ultra-fast local lines 18

## V

- variable aspect ratio and cascading 161
- VersaNet global networks 49
- VersaTile 15
- very-long-line resources 19
- ViewDraw 257
- VREF pins
  - manually assigning 265

## W

- web-based technical support 387





**Microsemi Corporate Headquarters**  
One Enterprise, Aliso Viejo CA 92656 USA  
Within the USA: +1 (949) 380-6100  
Sales: +1 (949) 380-6136  
Fax: +1 (949) 215-4996

Microsemi Corporation (NASDAQ: MSCC) offers a comprehensive portfolio of semiconductor solutions for: aerospace, defense and security; enterprise and communications; and industrial and alternative energy markets. Products include high-performance, high-reliability analog and RF devices, mixed signal and RF integrated circuits, customizable SoCs, FPGAs, and complete subsystems. Microsemi is headquartered in Aliso Viejo, Calif. Learn more at [www.microsemi.com](http://www.microsemi.com).

© 2012 Microsemi Corporation. All rights reserved. Microsemi and the Microsemi logo are trademarks of Microsemi Corporation. All other trademarks and service marks are the property of their respective owners.