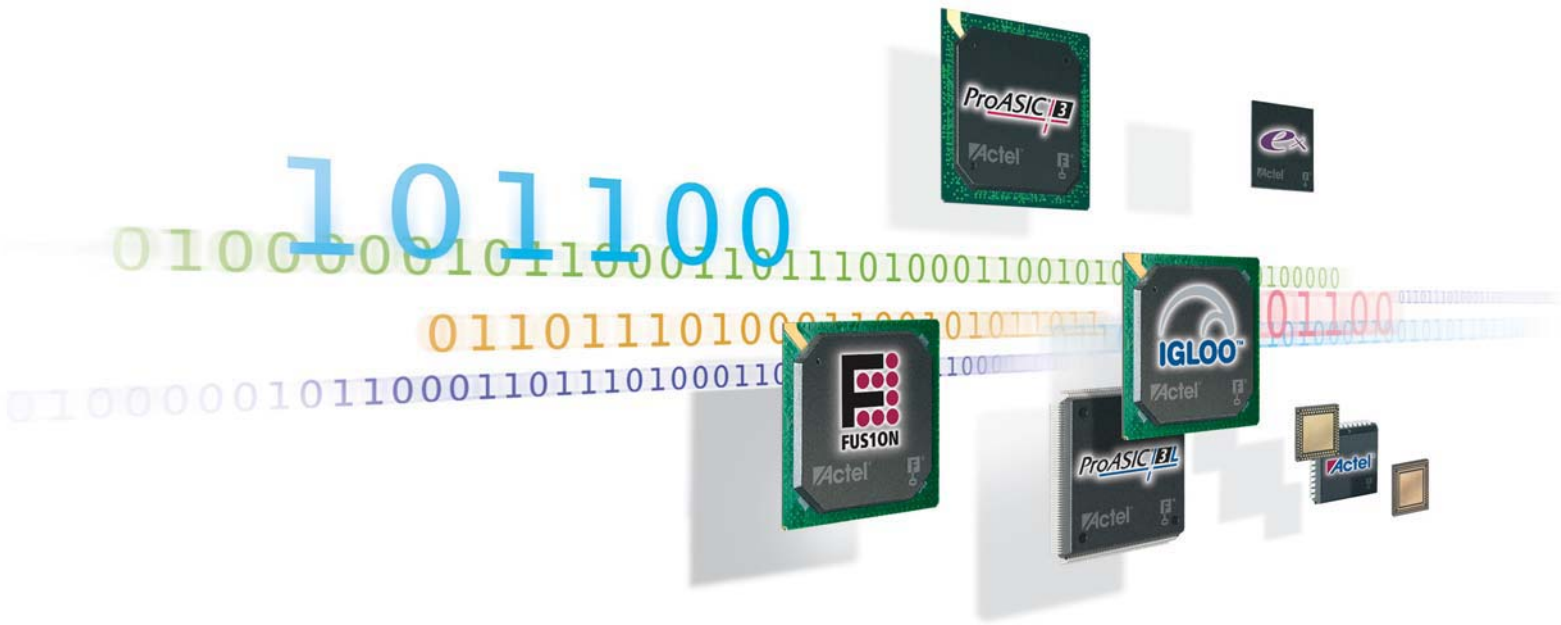
Rate This Document

IGLOO® PLUS Handbook



Actel®
POWER MATTERS

IGLOO+®
With Flash*Freeze

Table of Contents

Board-Level Requirements

Rate This Document

Low-Power Flash Device Handbooks Introduction

Device Handbooks contain all the information available to help designers understand and use Actel's devices. Handbook chapters are grouped into sections on the website to simplify navigation. Each chapter of the handbook may be viewed as an individual PDF file.

At the top of the handbook web page, you will see a PDF file for each product family. This file contains the complete device handbook. Please register for product updates to be notified when a section of the handbook changes.

Table i-1 • **Differences between Former Datasheets and Device Handbooks**

Description of Change	Comparison between Handbook and Datasheet
The silicon datasheet in the handbook does not contain the same chapters as the previous versions of the datasheets.	The former version of the silicon datasheet contained the following: • General Description • Device Architecture • DC and Switching Characteristics • Packaging The current datasheet now contains: • Product Brief (same information as the General Description) • DC and Switching Characteristics • Packaging The information previously contained in the Device Architecture chapter has been separated into individual chapters and merged with relevant application note content to provide one location for information on each architectural feature.
The General Description section no longer exists in datasheets.	The Product Brief and the General Description consisted of basically the same information but with different titles. To eliminate the duplicated information, we changed the document name to Product Brief.
Change tables were carried forward through this process; they contain information from the old datasheets and the new datasheets.	It is important that all earlier technical changes from the datasheets are listed so customers can determine if any of the changes affect their designs. Changes are listed chronologically, with the most current at the top and the earlier changes listed below them.
Version numbers were restarted.	The version numbers were restarted when the handbooks were created. For example, a datasheet may have been v2.1 and is now v1.0. The category (i.e., advance or production) of the datasheet did not change. The only change occurred to the actual numbering of the datasheet. All version numbers are located in the footer of the page.
Publication date	The publication date indicates when the document was published and posted to the Actel website.
The former datasheets were two columns and the current datasheets and handbooks are formatted with one column.	We changed the datasheet format to accommodate the large graphics and to improve readability.

Table i-1 • **Differences between Former Datasheets and Device Handbooks (continued)**

Description of Change	Comparison between Handbook and Datasheet
Each chapter within a handbook can have a different version number.	Each chapter in the handbook has a its own version number. Since the chapters are independent documents and can be updated at different times, the version numbers for each chapter increment only when a change is made to that chapter. For example, in the ProASIC3L handbook, the DC and Switching section is Advance v0.2, the Packaging Pin Assignments is v1.0, and the Global Resources section is v1.1. The DC and Switching section is advance because the data has not been fully characterized. The Packaging and Global Resources chapters are both production versions because the data is final. They have different version numbers because they were updated at different times.
Chapters can be numbered differently in each of the handbooks, but the version number for that chapter should be consistent throughout all handbooks. Chapters are also published individually without chapter numbers.	Chapters with shared content are reused across multiple handbooks. The number of chapters in each handbook varies depending on content, so the Pin Descriptions chapter could be chapter 7 in one handbook and chapter 9 in another. This is shown only in the combined handbook version of the document. The content within the chapters and version numbers is the same across each handbook for that chapter. If the chapter is updated it will be reposted for all associated handbooks.
There are three versions of the I/O Structure chapter.	There are several major differences between the following device groups: • IGLOO PLUS • IGLOO/ProASIC3/ProASIC3L • IGLOOe/ProASIC3EL/ProASIC3E As a result, we have three different I/O Structures chapters to describe the features in detail.
The part number of the datasheet has changed.	Part numbers are used internally to track documents. Each document has its own unique part number. The number after second the dash indicates the revision of the document. For example, in the part number, 51700094-006-1, 51700094-006 is the part number and 1 is the revision of the document. We start with 0 for all new documents. There is a part number for the current version of the datasheet on the back page with the addresses. In addition, all chapters in the datasheet and handbooks have their own unique part numbers. When we implemented the current handbook format, existing documents were assigned new part numbers.
The information contained in the Core Architecture section of the silicon handbook includes information previously found in application notes.	The chapters in the silicon handbook combine application notes that previously had very detailed information about an architectural feature and information from the former datasheets. In addition, because the information was very similar among several of Actel's low-power flash devices, we combined the information into one document. The Supported Families tables describe which devices are supported in the document. The application notes that contained specific architecture information and were combined into the handbook and many no longer exist as standalone application notes. Those that do exist in standalone version have been assigned an AC number (top right of first page) to help identify them. The AC number appears in the standalone version and in the handbook chapters where they occur.

Table i-1 • **Differences between Former Datasheets and Device Handbooks (continued)**

Description of Change	Comparison between Handbook and Datasheet	
The location of the following information has changed in the current handbook format:		
	Former Datasheet Location	**Current Handbook Location**
CCC/PLL Specification Table	Core Architecture	DC and Switching section > Clock Conditioning Circuits
Peak-to-Peak Jitter Waveform	Core Architecture	DC and Switching section > Clock Conditioning Circuits

Versions

Device handbook chapters may have different version numbers. Actel's goal is to provide customers with the latest information in a timely matter. As a result, the handbook chapters will be updated independently of the handbook.

Categories

In order to provide the latest information to designers, some datasheets are published before data has been fully characterized. Datasheets are designated as "Product Brief," "Advance," and "Production". The definition of these categories are as follows:

Product Brief

The product brief is a summarized version of a datasheet (advance or production) and contains general product information. This document gives an overview of specific device and family information.

Advance

This version contains initial estimated information based on simulation, other products, devices, or speed grades. This information can be used as estimates, but not for production. This label only applies to the DC and Switching Characteristics chapter of the datasheet and will only be used when the data has not been fully characterized.

Preliminary

The datasheet contains information based on simulation and/or initial characterization. The information is believed to be correct, but changes are possible.

Unmarked (production)

This version contains information that is considered to be final.

Export Administration Regulations (EAR)

The products described in this document are subject to the Export Administration Regulations (EAR). They could require an approved export license prior to export from the United States. An export includes release of product or disclosure of technology to a foreign national inside or outside the United States.

Part Number and Revision Date

Part Number 51700094-001-1
Revised October 2008

List of Changes

The following table lists critical changes that were made in the current version of the chapter.

Previous Version	Changes in Current Version (v1.1)	Page
v1.0 (January 2008)	This document was rewritten to address the differences between the former datasheets and new device handbooks.	N/A

Section I – IGLOO PLUS Datasheet

IGLOO PLUS Low-Power Flash FPGAs
with Flash*Freeze Technology

FlashLock

Features and Benefits

Low Power
- 1.2 V to 1.5 V Core Voltage Support for Low Power
- Supports Single-Voltage System Operation
- 5 µW Power Consumption in Flash*Freeze Mode
- Low-Power Active FPGA Operation
- Flash*Freeze Technology Enables Ultra-Low Power Consumption while Maintaining FPGA Content
- Configurable Hold Previous State, Tristate, HIGH, or LOW State per I/O in Flash*Freeze Mode
- Easy Entry To / Exit From Ultra-Low-Power Flash*Freeze Mode

Feature Rich
- 30 k to 125 k System Gates
- Up to 36 kbits of True Dual-Port SRAM
- Up to 212 User I/Os

Reprogrammable Flash Technology
- 130-nm, 7-Layer Metal, Flash-Based CMOS Process
- Live-at-Power-Up (LAPU) Level 0 Support
- Single-Chip Solution
- Retains Programmed Design When Powered Off

In-System Programming (ISP) and Security
- Secure ISP Using On-Chip 128-Bit Advanced Encryption Standard (AES) Decryption via JTAG (IEEE 1532–compliant)†
- FlashLock® to Secure FPGA Contents

High-Performance Routing Hierarchy
- Segmented, Hierarchical Routing and Clock Structure

Advanced I/O
- 1.2 V, 1.5 V, 1.8 V, 2.5 V, and 3.3 V Mixed-Voltage Operation
- Bank-Selectable I/O Voltages—4 Banks per Chip on All IGLOO® PLUS Devices
- Single-Ended I/O Standards: LVTTL, LVCMOS 3.3 V / 2.5 V / 1.8 V / 1.5 V / 1.2 V
- Selectable Schmitt Trigger Inputs
- I/O Registers on Input, Output, and Enable Paths
- Hot-Swappable and Cold-Sparing I/Os
- Programmable Output Slew Rate and Drive Strength
- Weak Pull-Up/-Down
- IEEE 1149.1 (JTAG) Boundary Scan Test
- Pin-Compatible Small-Footprint Packages across the IGLOO PLUS Family

Clock Conditioning Circuit (CCC) and PLL†
- Six CCC Blocks, One with an Integrated PLL
- Configurable Phase Shift, Multiply/Divide, Delay Capabilities, and External Feedback
- Wide Input Frequency Range (1.5 MHz up to 250 MHz)

Embedded Memory
- 1 kbit of FlashROM User Nonvolatile Memory
- SRAMs and FIFOs with Variable-Aspect-Ratio 4,608-Bit RAM Blocks (×1, ×2, ×4, ×9, and ×18 organizations)†
- True Dual-Port SRAM (except ×18)†

Table 1-1 • **IGLOO PLUS Product Family**

IGLOO PLUS Devices	AGLP030	AGLP060	AGLP125
System Gates	30 k	60 k	125 k
Typical Equivalent Macrocells	256	512	1,024
VersaTiles (D-flip-flops)	792	1,584	3,120
Flash*Freeze Mode (typical, µW)	5	10	16
RAM kbits (1,024 bits)	–	18	36
4,608-Bit Blocks	–	4	8
Secure (AES) ISP	–	Yes	Yes
FlashROM Bits	1 k	1 k	1 k
Integrated PLL in CCCs	–	1	1
VersaNet Globals*	6	18	18
I/O Banks	4	4	4
Maximum User I/Os	120	157	212
Package Pins CS VQ	 CS201, CS289 VQ128	 CS201, CS289 VQ176	 CS281, CS289

*Note: *Six chip (main) and twelve quadrant global networks are available for AGLP060 and AGLP125.*

† *The AGLP030 device does not support this feature.*

I/Os Per Package[1]

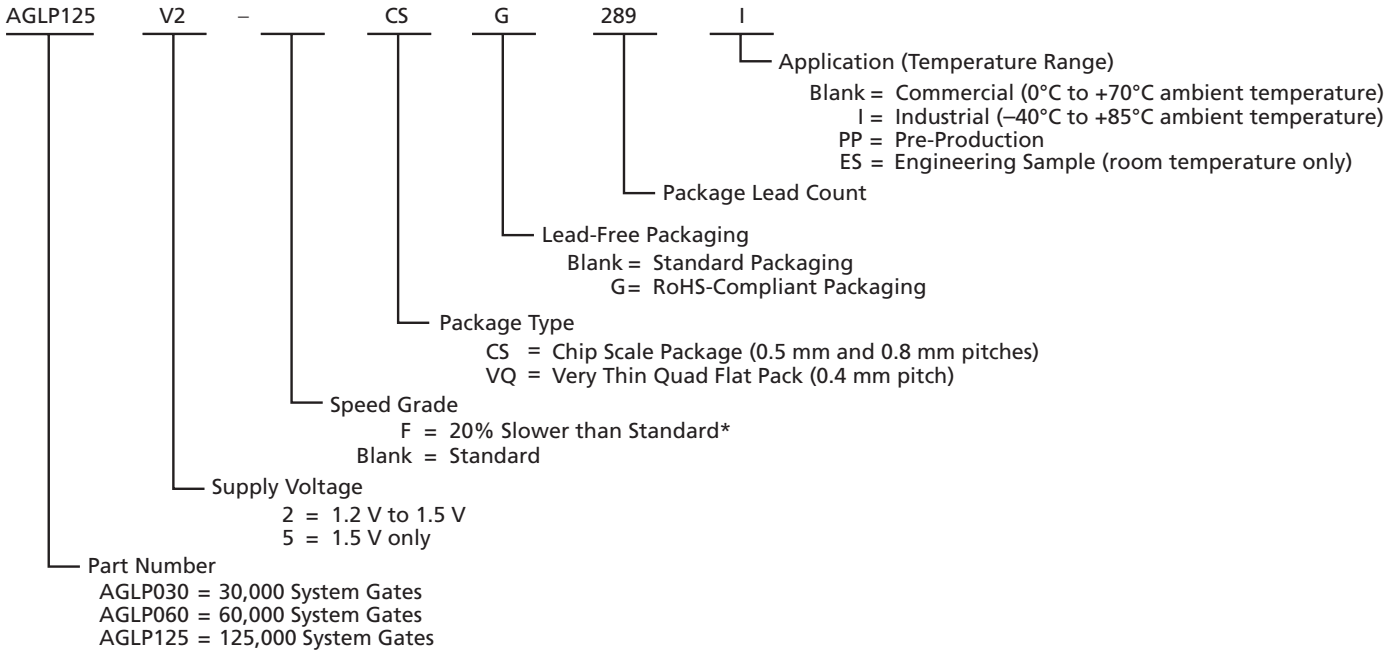
IGLOO PLUS Devices	AGLP030	AGLP060	AGLP125
Package	**Single-Ended I/Os**		
CS201	120	157	–
CS281	–	–	212
CS289	120	157	212
VQ128	101	–	–
VQ176	–	137	–

*Note: When the Flash*Freeze pin is used to directly enable Flash*Freeze mode and not used as a regular I/O, the number of single-ended user I/Os available is reduced by one.*

Table 1-2 • **Package Dimensions**

Package	CS201	CS281	CS289	VQ128	VQ176
Length × Width (mm/mm)	8 × 8	10 × 10	14 × 14	14 × 14	20 × 20
Nominal Area (mm^2)	64	100	196	196	100
Pitch (mm)	0.5	0.5	0.8	0.4	0.4
Height (mm)	0.89	1.05	1.20	1.0	1.0

IGLOO PLUS Ordering Information



Notes:

1. *Marking information: IGLOO PLUS V2 devices do not have a V2 marking, but IGLOO PLUS V5 devices are marked accordingly.*
2. *The DC and switching characteristics for the –F speed grade targets are based only on simulation.*
   The characteristics provided for the –F speed grade are subject to change after establishing FPGA specifications. Some restrictions might be added and will be reflected in future revisions of this document. The –F speed grade is only supported in the Commercial temperature range.
3. *"G" indicates RoHS-compliant packages.*

Temperature Grade Offerings

Package	AGLP030	AGLP060	AGLP125
CS201	C, I	C, I	–
CS281	–	–	C, I
CS289	C, I	C, I	C, I
VQ128	C, I	–	–
VQ176	–	C, I	–

Notes:

1. *C = Commercial temperature range: 0°C to 70°C ambient temperature.*
2. *I = Industrial temperature range: –40°C to 85°C ambient temperature.*

Speed Grade and Temperature Grade Matrix

Temperature Grade	–F [1]	Std.
C [2]	✓	✓
I [3]	–	✓

Notes:

1. *The characteristics provided for the –F speed grade are subject to change after establishing FPGA specifications. Some restrictions might be added and will be reflected in future revisions of this document. The –F speed grade is only supported in the Commercial temperature range.*
2. *C = Commercial temperature range: 0°C to 70°C ambient temperature.*
3. *I = Industrial temperature range: –40°C to 85°C ambient temperature.*

Contact your local Actel representative for device availability:
http://www.actel.com/company/contact/default.aspx.

1 – IGLOO PLUS Device Family Overview

General Description

The IGLOO PLUS family of flash FPGAs, based on a 130 nm flash process, offers the lowest power FPGA, a single-chip solution, small-footprint packages, reprogrammability, and an abundance of advanced features.

The Flash*Freeze technology used in IGLOO PLUS devices enables entering and exiting an ultra-low-power mode that consumes as little as 5 µW while retaining the design information, SRAM content, registers, and I/O states. Flash*Freeze technology simplifies power management through I/O and clock management with rapid recovery to operation mode.

The Low Power Active capability (static idle) allows for ultra-low-power consumption while the IGLOO PLUS device is completely functional in the system. This allows the IGLOO PLUS device to control system power management based on external inputs (e.g., scanning for keyboard stimulus) while consuming minimal power.

Nonvolatile flash technology gives IGLOO PLUS devices the advantage of being a secure, low-power, single-chip solution that is live at power-up (LAPU). IGLOO PLUS is reprogrammable and offers time-to-market benefits at an ASIC-level unit cost.

These features enable designers to create high-density systems using existing ASIC or FPGA design flows and tools.

IGLOO PLUS devices offer 1 kbit of on-chip, reprogrammable, nonvolatile FlashROM storage as well as clock conditioning circuitry based on an integrated phase-locked loop (PLL). IGLOO PLUS devices have up to 125 k system gates, supported with up to 36 kbits of true dual-port SRAM and up to 212 user I/Os. The AGLP030 devices have no PLL or RAM support.

Flash*Freeze Technology

The IGLOO PLUS device offers unique Flash*Freeze technology, allowing the device to enter and exit ultra-low-power Flash*Freeze mode. IGLOO PLUS devices do not need additional components to turn off I/Os or clocks while retaining the design information, SRAM content, registers, and I/O states. Flash*Freeze technology is combined with in-system programmability, which enables users to quickly and easily upgrade and update their designs in the final stages of manufacturing or in the field. The ability of IGLOO PLUS V2 devices to support a wide range of core and I/O voltages (1.2 V to 1.5 V) allows further reduction in power consumption, thus achieving the lowest total system power.

During Flash*Freeze mode, each I/O can be set to the following configurations: hold previous state, tristate, or set as HIGH or LOW.

The availability of low-power modes, combined with reprogrammability, a single-chip and single-voltage solution, and availability of small-footprint, high-pin-count packages, make IGLOO PLUS devices the best fit for portable electronics.

Flash Advantages

Low Power

IGLOO PLUS devices exhibit power characteristics similar to those of an ASIC, making them an ideal choice for power-sensitive applications. IGLOO PLUS devices have only a very limited power-on current surge and no high-current transition period, both of which occur on many FPGAs.

IGLOO PLUS devices also have low dynamic power consumption to further maximize power savings; power is even further reduced by the use of a 1.2 V core voltage.

Low dynamic power consumption, combined with low static power consumption and Flash*Freeze technology, gives the IGLOO PLUS device the lowest total system power offered by any FPGA.

Security

The nonvolatile, flash-based IGLOO PLUS devices do not require a boot PROM, so there is no vulnerable external bitstream that can be easily copied. IGLOO PLUS devices incorporate FlashLock, which provides a unique combination of reprogrammability and design security without external overhead, advantages that only an FPGA with nonvolatile flash programming can offer.

IGLOO PLUS devices (except AGLP030) utilize a 128-bit flash-based lock and a separate AES key to secure programmed intellectual property and configuration data. In addition, all FlashROM data in IGLOO PLUS devices can be encrypted prior to loading, using the industry-leading AES-128 (FIPS192) bit block cipher encryption standard. AES was adopted by the National Institute of Standards and Technology (NIST) in 2000 and replaces the 1977 DES standard. IGLOO PLUS devices have a built-in AES decryption engine and a flash-based AES key that make them the most comprehensive programmable logic device security solution available today. IGLOO PLUS devices with AES-based security allow for secure, remote field updates over public networks such as the Internet, and ensure that valuable IP remains out of the hands of system overbuilders, system cloners, and IP thieves. The contents of a programmed IGLOO PLUS device cannot be read back, although secure design verification is possible.

Security, built into the FPGA fabric, is an inherent component of the IGLOO PLUS family. The flash cells are located beneath seven metal layers, and many device design and layout techniques have been used to make invasive attacks extremely difficult. The IGLOO PLUS family, with FlashLock and AES security, is unique in being highly resistant to both invasive and noninvasive attacks. Your valuable IP is protected and secure, making remote ISP possible. An IGLOO PLUS device provides the most impenetrable security for programmable logic designs.

Single Chip

Flash-based FPGAs store their configuration information in on-chip flash cells. Once programmed, the configuration data is an inherent part of the FPGA structure, and no external configuration data needs to be loaded at system power-up (unlike SRAM-based FPGAs). Therefore, flash-based IGLOO PLUS FPGAs do not require system configuration components such as EEPROMs or microcontrollers to load device configuration data. This reduces bill-of-materials costs and PCB area, and increases security and system reliability.

The IGLOO PLUS devices can be operated with a 1.2 V or 1.5 V single-voltage supply for core and I/Os, eliminating the need for additional supplies while minimizing total power consumption.

Live at Power-Up

The Actel flash-based IGLOO PLUS devices support Level 0 of the LAPU classification standard. This feature helps in system component initialization, execution of critical tasks before the processor wakes up, setup and configuration of memory blocks, clock generation, and bus activity management. The LAPU feature of flash-based IGLOO PLUS devices greatly simplifies total system design and reduces total system cost, often eliminating the need for CPLDs and clock generation PLLs. In addition, glitches and brownouts in system power will not corrupt the IGLOO PLUS device's flash configuration, and unlike SRAM-based FPGAs, the device will not have to be reloaded when system power is restored. This enables the reduction or complete removal of the configuration PROM, expensive voltage monitor, brownout detection, and clock generator devices from the PCB design. Flash-based IGLOO PLUS devices simplify total system design and reduce cost and design risk while increasing system reliability and improving system initialization time.

IGLOO PLUS flash FPGAs allow the user to quickly enter and exit Flash*Freeze mode. This is done almost instantly (within 1 µs), and the device retains configuration and data in registers and RAM. Unlike SRAM-based FPGAs, the device does not need to reload configuration and design state from external memory components; instead, it retains all necessary information to resume operation immediately.

Reduced Cost of Ownership

Advantages to the designer extend beyond low unit cost, performance, and ease of use. Unlike SRAM-based FPGAs, flash-based IGLOO PLUS devices allow all functionality to be live at power-up; no external boot PROM is required. On-board security mechanisms prevent access to all the programming information and enable secure remote updates of the FPGA logic. Designers can perform secure remote in-system reprogramming to support future design iterations and field

upgrades with confidence that valuable intellectual property cannot be compromised or copied. Secure ISP can be performed using the industry-standard AES algorithm. The IGLOO PLUS family device architecture mitigates the need for ASIC migration at higher user volumes. This makes the IGLOO PLUS family a cost-effective ASIC replacement solution, especially for applications in the consumer, networking/communications, computing, and avionics markets.

Firm-Error Immunity

Firm errors occur most commonly when high-energy neutrons, generated in the upper atmosphere, strike a configuration cell of an SRAM FPGA. The energy of the collision can change the state of the configuration cell and thus change the logic, routing, or I/O behavior in an unpredictable way. These errors are impossible to prevent in SRAM FPGAs. The consequence of this type of error can be a complete system failure. Firm errors do not exist in the configuration memory of IGLOO PLUS flash-based FPGAs. Once it is programmed, the flash cell configuration element of IGLOO PLUS FPGAs cannot be altered by high-energy neutrons and is therefore immune to them. Recoverable (or soft) errors occur in the user data SRAM of all FPGA devices. These can easily be mitigated by using error detection and correction (EDAC) circuitry built into the FPGA fabric.

Advanced Flash Technology

The IGLOO PLUS family offers many benefits, including nonvolatility and reprogrammability, through an advanced flash-based, 130 nm LVCMOS process with seven layers of metal. Standard CMOS design techniques are used to implement logic and control functions. The combination of fine granularity, enhanced flexible routing resources, and abundant flash switches allows for very high logic utilization without compromising device routability or performance. Logic functions within the device are interconnected through a four-level routing hierarchy.

IGLOO PLUS family FPGAs utilize design and process techniques to minimize power consumption in all modes of operation.

Advanced Architecture

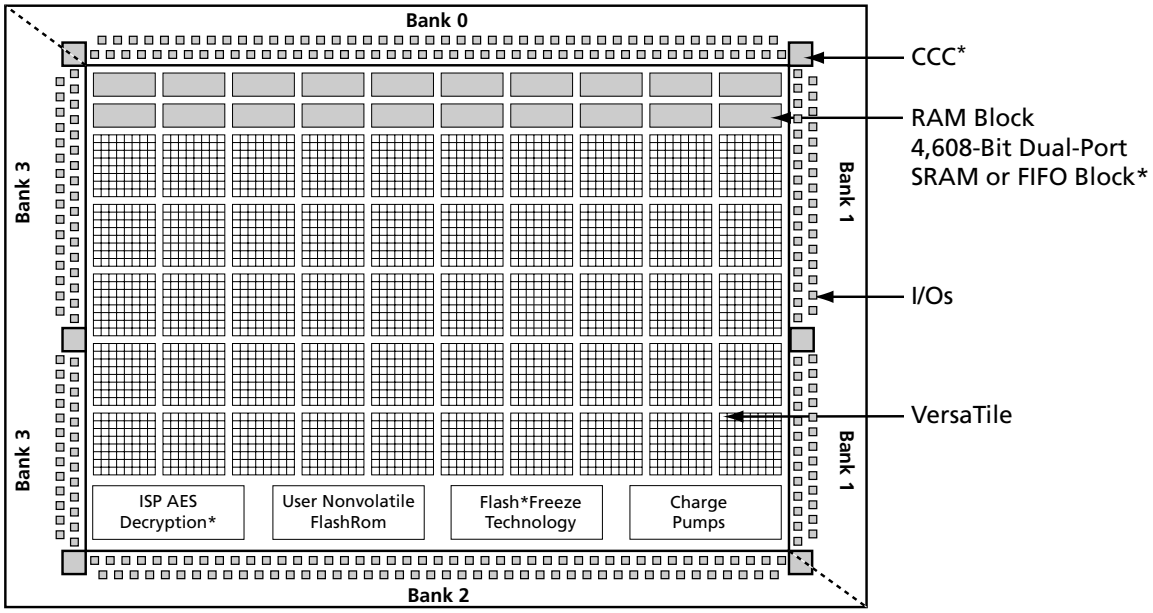
The proprietary IGLOO PLUS architecture provides granularity comparable to standard-cell ASICs. The IGLOO PLUS device consists of five distinct and programmable architectural features (Figure 1-1 on page 1-4):

- Flash*Freeze technology
- FPGA VersaTiles
- Dedicated FlashROM
- Dedicated SRAM/FIFO memory†
- Extensive CCCs and PLLs†
- Advanced I/O structure

The FPGA core consists of a sea of VersaTiles. Each VersaTile can be configured as a three-input logic function, a D-flip-flop (with or without enable), or a latch by programming the appropriate flash switch interconnections. The versatility of the IGLOO PLUS core tile as either a three-input lookup table (LUT) equivalent or a D-flip-flop/latch with enable allows for efficient use of the FPGA fabric. The VersaTile capability is unique to the Actel ProASIC family of third-generation-architecture flash FPGAs. VersaTiles are connected with any of the four levels of routing hierarchy. Flash switches are distributed throughout the device to provide nonvolatile, reconfigurable interconnect programming. Maximum core utilization is possible for virtually any design.

In addition, extensive on-chip programming circuitry allows for rapid, single-voltage (3.3 V) programming of IGLOO PLUS devices via an IEEE 1532 JTAG interface.

† *The AGLP030 device does not support PLL or SRAM.*



* *Not supported by AGLP030 devices*

Figure 1-1 • **IGLOO PLUS Device Architecture Overview with Four I/O Banks (AGLP030, AGLP060, and AGLP125)**

*Flash*Freeze Technology*

The IGLOO PLUS device has an ultra-low-power static mode, called Flash*Freeze mode, which retains all SRAM and register information and can still quickly return to normal operation. Flash*Freeze technology enables the user to quickly (within 1 µs) enter and exit Flash*Freeze mode by activating the Flash*Freeze pin while all power supplies are kept at their original values. In addition, I/Os and global I/Os can still be driven and can be toggling without impact on power consumption, clocks can still be driven or can be toggling without impact on power consumption, and the device retains all core registers, SRAM information, and I/O states. I/Os can be individually configured to either hold their previous state or be tristated during Flash*Freeze mode. Alternatively, they can be set to a certain state using weak pull-up or pull-down I/O attribute configuration. No power is consumed by the I/O banks, clocks, JTAG pins, or PLL, and the device consumes as little as 5 µW in this mode.

Flash*Freeze technology allows the user to switch to Active mode on demand, thus simplifying the power management of the device.

The Flash*Freeze pin (active low) can be routed internally to the core to allow the user's logic to decide when it is safe to transition to this mode. Refer to Figure 1-2 for an illustration of entering/exiting Flash*Freeze mode. It is also possible to use the Flash*Freeze pin as a regular I/O if Flash*Freeze mode usage is not planned.

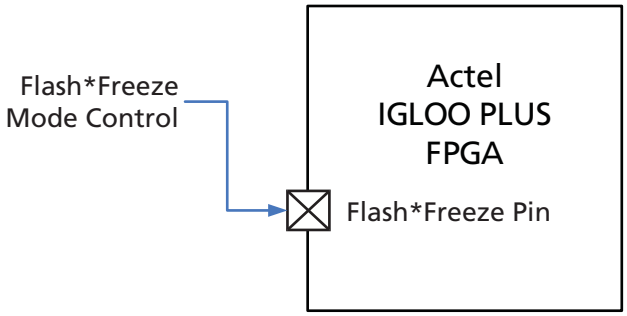


Figure 1-2 • **IGLOO PLUS Flash*Freeze Mode**

VersaTiles

The IGLOO PLUS core consists of VersaTiles, which have been enhanced beyond the ProASIC[PLUS]® core tiles. The IGLOO PLUS VersaTile supports the following:

- All 3-input logic functions—LUT-3 equivalent
- Latch with clear or set
- D-flip-flop with clear or set
- Enable D-flip-flop with clear or set

Refer to Figure 1-3 for VersaTile configurations.

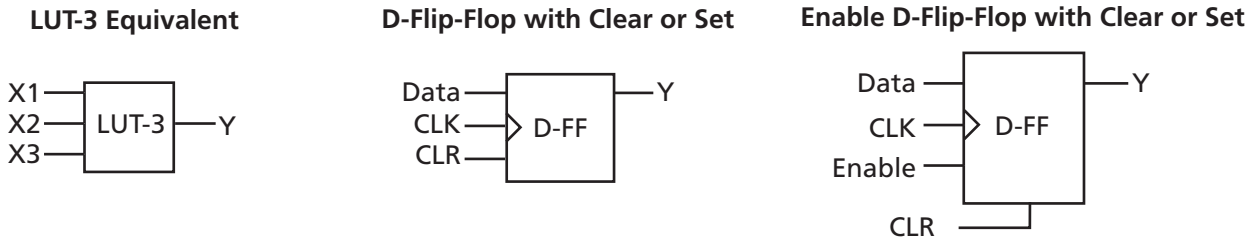


Figure 1-3 • **VersaTile Configurations**

User Nonvolatile FlashROM

Actel IGLOO PLUS devices have 1 kbit of on-chip, user-accessible, nonvolatile FlashROM. The FlashROM can be used in diverse system applications:

- Internet protocol addressing (wireless or fixed)
- System calibration settings
- Device serialization and/or inventory control
- Subscription-based business models (for example, set-top boxes)
- Secure key storage for secure communications algorithms
- Asset management/tracking
- Date stamping
- Version management

The FlashROM is written using the standard IGLOO PLUS IEEE 1532 JTAG programming interface. The core can be individually programmed (erased and written), and on-chip AES decryption can be used selectively to securely load data over public networks (except in AGLP030 devices), as in security keys stored in the FlashROM for a user design.

The FlashROM can be programmed via the JTAG programming interface, and its contents can be read back either through the JTAG programming interface or via direct FPGA core addressing. Note that the FlashROM can only be programmed from the JTAG interface and cannot be programmed from the internal logic array.

The FlashROM is programmed as 8 banks of 128 bits; however, reading is performed on a byte-by-byte basis using a synchronous interface. A 7-bit address from the FPGA core defines which of the 8 banks and which of the 16 bytes within that bank are being read. The three most significant bits (MSBs) of the FlashROM address determine the bank, and the four least significant bits (LSBs) of the FlashROM address define the byte.

The Actel IGLOO PLUS development software solutions, Libero® Integrated Design Environment (IDE) and Designer, have extensive support for the FlashROM. One such feature is auto-generation of sequential programming files for applications requiring a unique serial number in each part. Another feature allows the inclusion of static data for system version control. Data for the FlashROM can be generated quickly and easily using Actel Libero IDE and Designer software tools. Comprehensive programming file support is also included to allow for easy programming of large numbers of parts with differing FlashROM contents.

SRAM and FIFO

IGLOO PLUS devices (except AGLP030 devices) have embedded SRAM blocks along their north side. Each variable-aspect-ratio SRAM block is 4,608 bits in size. Available memory configurations are 256×18, 512×9, 1k×4, 2k×2, and 4k×1 bits. The individual blocks have independent read and write ports that can be configured with different bit widths on each port. For example, data can be sent through a 4-bit port and read as a single bitstream. The embedded SRAM blocks can be initialized via the device JTAG port (ROM emulation mode) using the UJTAG macro (except in AGLP030 devices).

In addition, every SRAM block has an embedded FIFO control unit. The control unit allows the SRAM block to be configured as a synchronous FIFO without using additional core VersaTiles. The FIFO width and depth are programmable. The FIFO also features programmable Almost Empty (AEMPTY) and Almost Full (AFULL) flags in addition to the normal Empty and Full flags. The embedded FIFO control unit contains the counters necessary for generation of the read and write address pointers. The embedded SRAM/FIFO blocks can be cascaded to create larger configurations.

PLL and CCC

IGLOO PLUS devices provide designers with very flexible clock conditioning circuit (CCC) capabilities. Each member of the IGLOO PLUS family contains six CCCs. One CCC (center west side) has a PLL. The AGLP030 device does not have a PLL or CCCs; it contains only inputs to six globals.

The six CCC blocks are located at the four corners and the centers of the east and west sides. One CCC (center west side) has a PLL.

The four corner CCCs and the east CCC allow simple clock delay operations as well as clock spine access.

The inputs of the six CCC blocks are accessible from the FPGA core or from one of several inputs located near the CCC that have dedicated connections to the CCC block.

The CCC block has these key features:

- Wide input frequency range (f_{IN_CCC}) = 1.5 MHz up to 250 MHz
- Output frequency range (f_{OUT_CCC}) = 0.75 MHz up to 250 MHz
- 2 programmable delay types for clock skew minimization
- Clock frequency synthesis (for PLL only)

Additional CCC specifications:

- Internal phase shift = 0°, 90°, 180°, and 270°. Output phase shift depends on the output divider configuration (for PLL only).
- Output duty cycle = 50% ± 1.5% or better (for PLL only)
- Low output jitter: worst case < 2.5% × clock period peak-to-peak period jitter when single global network used (for PLL only)
- Maximum acquisition time is 300 µs (for PLL only)
- Exceptional tolerance to input period jitter—allowable input jitter is up to 1.5 ns (for PLL only)
- Four precise phases; maximum misalignment between adjacent phases of 40 ps × 250 MHz / f_{OUT_CCC} (for PLL only)

Global Clocking

IGLOO PLUS devices have extensive support for multiple clocking domains. In addition to the CCC and PLL support described above, there is a comprehensive global clock distribution network.

Each VersaTile input and output port has access to nine VersaNets: six chip (main) and three quadrant global networks. The VersaNets can be driven by the CCC or directly accessed from the core via multiplexers (MUXes). The VersaNets can be used to distribute low-skew clock signals or for rapid distribution of high-fanout nets.

I/Os with Advanced I/O Standards

The IGLOO PLUS family of FPGAs features a flexible I/O structure, supporting a range of voltages (1.2 V, 1.5 V, 1.8 V, 2.5 V, and 3.3 V). IGLOO PLUS FPGAs support many different I/O standards.

The I/Os are organized into four banks. All devices in IGLOO PLUS have four banks. The configuration of these banks determines the I/O standards supported.

Each I/O module contains several input, output, and output enable registers.

Part Number and Revision Date

Part Number 51700102-001-2

Revised August 2008

List of Changes

The following table lists critical changes that were made in the current version of the document.

Previous Version	Changes in Current Version (v1.2)	Page
v1.1 (July 2008)	The VQ128 and VQ176 packages were added to Table 1-1 · IGLOO PLUS Product Family, the "I/Os Per Package1" table, Table 1-2 · Package Dimensions, "IGLOO PLUS Ordering Information", and the "Temperature Grade Offerings" table.	I to IV
v1.0 (March 2008)	As a result of the Libero IDE v8.4 release, Actel now offers a wide range of core voltage support. The document was updated to change 1.2 V / 1.5 V to 1.2 V to 1.5 V.	N/A

Datasheet Categories

Categories

In order to provide the latest information to designers, some datasheets are published before data has been fully characterized. Datasheets are designated as "Product Brief," "Advance," "Preliminary," and "Production." The definition of these categories are as follows:

Product Brief

The product brief is a summarized version of a datasheet (advance or production) and contains general product information. This document gives an overview of specific device and family information.

Advance

This version contains initial estimated information based on simulation, other products, devices, or speed grades. This information can be used as estimates, but not for production. This label only applies to the DC and Switching Characteristics chapter of the datasheet and will only be used when the data has not been fully characterized.

Preliminary

The datasheet contains information based on simulation and/or initial characterization. The information is believed to be correct, but changes are possible.

Unmarked (production)

This version contains information that is considered to be final.

Export Administration Regulations (EAR)

The products described in this document are subject to the Export Administration Regulations (EAR). They could require an approved export license prior to export from the United States. An export includes release of product or disclosure of technology to a foreign national inside or outside the United States.

Actel Safety Critical, Life Support, and High-Reliability Applications Policy

The Actel products described in this advance status document may not have completed Actel's qualification process. Actel may amend or enhance products during the product introduction and qualification process, resulting in changes in device functionality or performance. It is the responsibility of each customer to ensure the fitness of any Actel product (but especially a new product) for a particular purpose, including appropriateness for safety-critical, life-support, and other high-reliability applications. Consult Actel's Terms and Conditions for specific liability exclusions relating to life-support applications. A reliability report covering all of Actel's products is available on the Actel website at http://www.actel.com/documents/ORT_Report.pdf. Actel also offers a variety of enhanced qualification and lot acceptance screening procedures. Contact your local Actel sales office for additional reliability information.

2 – IGLOO PLUS DC and Switching Characteristics

General Specifications

DC and switching characteristics for –F speed grade targets are based only on simulation.

The characteristics provided for the –F speed grade are subject to change after establishing FPGA specifications. Some restrictions might be added and will be reflected in future revisions of this document. The –F speed grade is only supported in the commercial temperature range.

Operating Conditions

Stresses beyond those listed in Table 2-1 may cause permanent damage to the device.

Exposure to absolute maximum rating conditions for extended periods may affect device reliability. Absolute Maximum Ratings are stress ratings only; functional operation of the device at these or any other conditions beyond those listed under the Recommended Operating Conditions specified in Table 2-2 on page 2-2 is not implied.

Table 2-1 • **Absolute Maximum Ratings**

Symbol	Parameter	Limits	Units
V_{CC}	DC core supply voltage	–0.3 to 1.65	V
V_{JTAG}	JTAG DC voltage	–0.3 to 3.75	V
V_{PUMP}	Programming voltage	–0.3 to 3.75	V
V_{CCPLL}	Analog power supply (PLL)	–0.3 to 1.65	V
V_{CCI}	DC I/O buffer supply voltage	–0.3 to 3.75	V
VI	I/O input voltage	–0.3 V to 3.6 V (when I/O hot insertion mode is enabled) –0.3 V to (V_{CCI} + 1 V) or 3.6 V, whichever voltage is lower (when I/O hot-insertion mode is disabled)	V
T_{STG} [2]	Storage temperature	–65 to +150	°C
T_J [2]	Junction temperature	+125	°C

Notes:

1. *The device should be operated within the limits specified by the datasheet. During transitions, the input signal may undershoot or overshoot according to the limits shown in Table 2-4 on page 2-3.*
2. *For flash programming and retention maximum limits, refer to Table 2-3 on page 2-2, and for recommended operating limits, refer to Table 2-2 on page 2-2.*

Table 2-2 • Recommended Operating Conditions [4]

Symbol	Parameter		Commercial	Industrial	Units
T_A	Ambient temperature		0 to +70 [6]	–40 to +85 [7]	°C
T_J	Junction temperature [8]		0 to + 85	–40 to +100	°C
V_{CC} [3]	1.5 V DC core supply voltage[1]		1.425 to 1.575	1.425 to 1.575	V
	1.2 V–1.5 V wide range core voltage[2]		1.14 to 1.575	1.14 to 1.575	V
V_{JTAG}	JTAG DC voltage		1.4 to 3.6	1.4 to 3.6	V
V_{PUMP} [5]	Programming voltage	Programming mode	3.15 to 3.45	3.15 to 3.45	V
		Operation	0 to 3.45	0 to 3.45	V
V_{CCPLL} [9]	Analog power supply (PLL)	1.5 V DC core supply voltage[1]	1.4 to 1.6	1.4 to 1.6	V
		1.2 V–1.5 V wide range core voltage[2]	1.14 to 1.575	1.14 to 1.575	V
V_{CCI}	1.2 V DC supply voltage[2]		1.14 to 1.26	1.14 to 1.26	V
	1.5 V DC supply voltage		1.425 to 1.575	1.425 to 1.575	V
	1.8 V DC supply voltage		1.7 to 1.9	1.7 to 1.9	V
	2.5 V DC supply voltage		2.3 to 2.7	2.3 to 2.7	V
	3.3 V DC supply voltage		3.0 to 3.6	3.0 to 3.6	V

Notes:

1. *For IGLOO® PLUS V5 devices*
2. *For IGLOO PLUS V2 devices only, operating at $V_{CCI} \geq V_{CC}$*
3. *The ranges given here are for power supplies only. The recommended input voltage ranges specific to each I/O standard are given in Table 2-20 on page 2-19. V_{CCI} should be at the same voltage within a given I/O bank.*
4. *All parameters representing voltages are measured with respect to GND unless otherwise specified.*
5. *V_{PUMP} can be left floating during operation (not programming mode).*
6. *Maximum T_J = 85°C.*
7. *Maximum T_J = 100°C.*
8. *To ensure targeted reliability standards are met across ambient and junction operating temperatures, Actel recommends that the user follow best design practices using Actel's timing and power simulation tools.*
9. *V_{CCPLL} pins should be tied to V_{CC} pins. See Pin Descriptions for further information.*

Table 2-3 • Flash Programming Limits – Retention, Storage, and Operating Temperature[1]

Product Grade	Programming Cycles	Program Retention (biased/unbiased)	Maximum Storage Temperature T_{STG} (°C) [2]	Maximum Operating Junction Temperature T_J (°C) [2]
Commercial	500	20 years	110	100
Industrial	500	20 years	110	100

Notes:

1. *This is a stress rating only; functional operation at any condition other than those indicated is not implied.*
2. *These limits apply for program/data retention only. Refer to Table 2-1 on page 2-1 and Table 2-2 for device operating conditions and absolute limits.*

Table 2-4 • **Overshoot and Undershoot Limits [1]**

V_{CCI}	Average V_{CCI}–GND Overshoot or Undershoot Duration as a Percentage of Clock Cycle[2]	Maximum Overshoot/ Undershoot[2]
2.7 V or less	10%	1.4 V
	5%	1.49 V
3 V	10%	1.1 V
	5%	1.19 V
3.3 V	10%	0.79 V
	5%	0.88 V
3.6 V	10%	0.45 V
	5%	0.54 V

Notes:

1. *Based on reliability requirements at 85°C.*
2. *The duration is allowed at one out of six clock cycles. If the overshoot/undershoot occurs at one out of two cycles, the maximum overshoot/undershoot has to be reduced by 0.15 V.*

I/O Power-Up and Supply Voltage Thresholds for Power-On Reset (Commercial and Industrial)

Sophisticated power-up management circuitry is designed into every IGLOO PLUS device. These circuits ensure easy transition from the powered-off state to the powered-up state of the device. The many different supplies can power up in any sequence with minimized current spikes or surges. In addition, the I/O will be in a known state through the power-up sequence. The basic principle is shown in Figure 2-1 on page 2-4.

There are five regions to consider during power-up.

IGLOO PLUS I/Os are activated only if ALL of the following three conditions are met:

1. V_{CC} and V_{CCI} are above the minimum specified trip points (Figure 2-1 and Figure 2-2 on page 2-5).
2. $V_{CCI} > V_{CC} – 0.75$ V (typical)
3. Chip is in the operating mode.

V_{CCI} Trip Point:
Ramping up (V5 devices): 0.6 V < trip_point_up < 1.2 V
Ramping down (V5 devices): 0.5 V < trip_point_down < 1.1 V
Ramping up (V2 devices): 0.75 V < trip_point_up < 1.05 V
Ramping down (V2 devices): 0.65 V < trip_point_down < 0.95 V

V_{CC} Trip Point:
Ramping up (V5 devices): 0.6 V < trip_point_up < 1.1 V
Ramping down (V5 devices): 0.5 V < trip_point_down < 1.0 V
Ramping up (V2 devices): 0.65 V < trip_point_up < 1.05 V
Ramping down (V2 devices): 0.55 V < trip_point_down < 0.95 V

V_{CC} and V_{CCI} ramp-up trip points are about 100 mV higher than ramp-down trip points. This specifically built-in hysteresis prevents undesirable power-up oscillations and current surges. Note the following:

- During programming, I/Os become tristated and weakly pulled up to V_{CCI}.
- JTAG supply, PLL power supplies, and charge pump V_{PUMP} supply have no influence on I/O behavior.

PLL Behavior at Brownout Condition

Actel recommends using monotonic power supplies or voltage regulators to ensure proper power-up behavior. Power ramp-up should be monotonic at least until V_{CC} and V_{CCPLX} exceed brownout activation levels (see Figure 2-1 and Figure 2-2 on page 2-5 for more details).

When PLL power supply voltage and/or V_{CC} levels drop below the V_{CC} brownout levels (0.75 V ± 0.25 V for V5 devices, and 0.75 V ± 0.2 V for V2 devices), the PLL output lock signal goes LOW and/or the output clock is lost. Refer to the "Brownout Voltage" section in the *Power-Up/-Down Behavior of Low-Power Flash Devices* chapter of the *ProASIC3* and *ProASIC3E* handbooks for information on clock and lock recovery.

Internal Power-Up Activation Sequence

1. Core
2. Input buffers
3. Output buffers, after 200 ns delay from input buffer activation

To make sure the transition from input buffers to output buffers is clean, ensure that there is no path longer than 100 ns from input buffer to output buffer in your design.

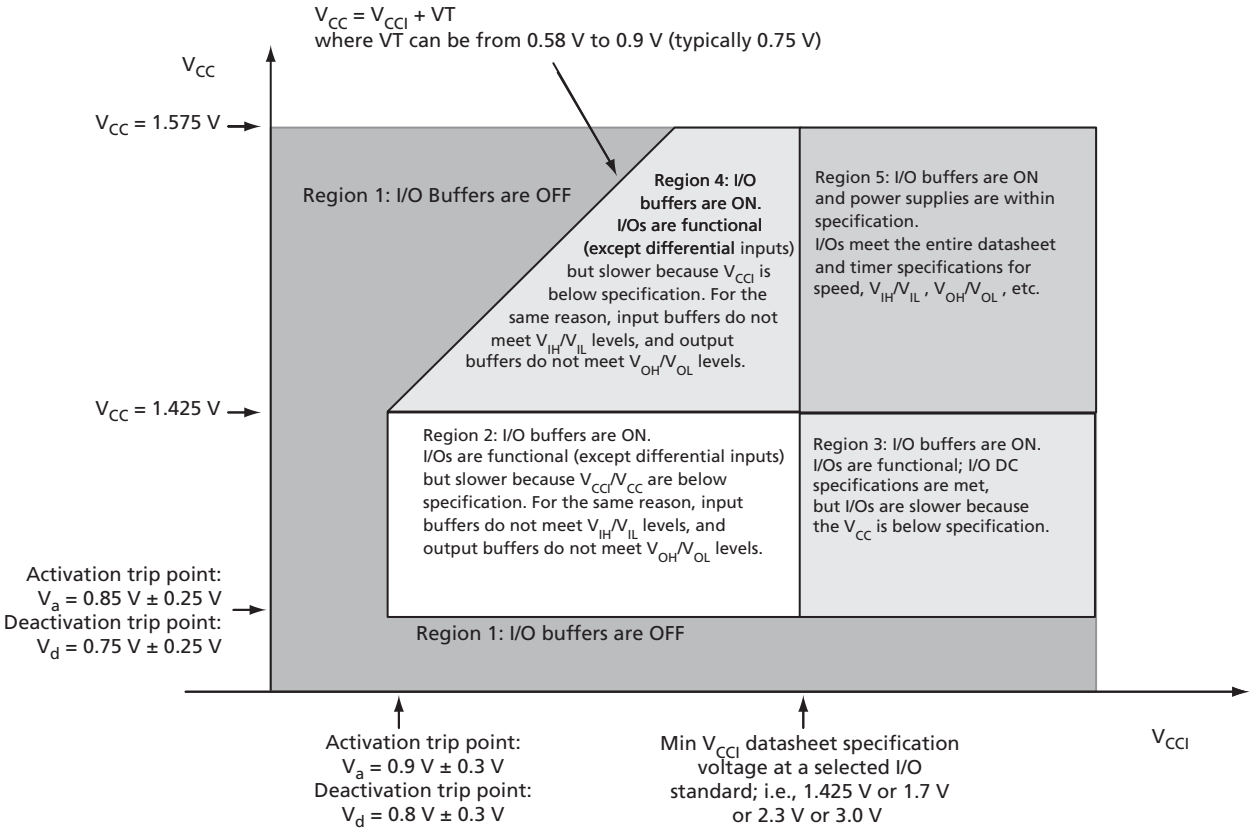


Figure 2-1 • **V5 Devices – I/O State as a Function of V_{CCI} and V_{CC} Voltage Levels**

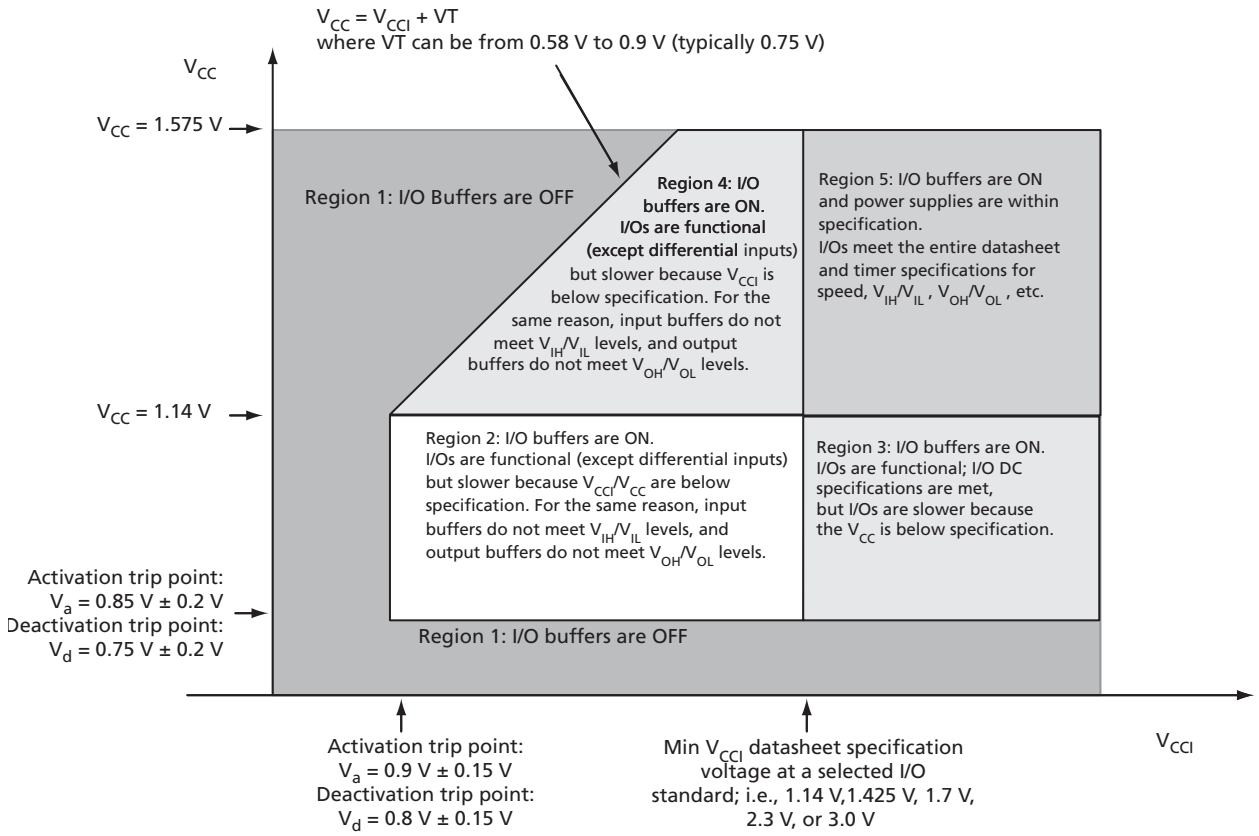


Figure 2-2 • **V2 Devices – I/O State as a Function of V_{CCI} and V_{CC} Voltage Levels**

Thermal Characteristics

Introduction

The temperature variable in the Actel Designer software refers to the junction temperature, not the ambient temperature. This is an important distinction because dynamic and static power consumption cause the chip junction temperature to be higher than the ambient temperature.

EQ 2-1 can be used to calculate junction temperature.

$$T_J = \text{Junction Temperature} = \Delta T + T_A$$

EQ 2-1

where:

T_A = Ambient temperature

ΔT = Temperature gradient between junction (silicon) and ambient $\Delta T = \theta_{ja} * P$

θ_{ja} = Junction-to-ambient of the package. θ_{ja} numbers are located in Figure 2-5.

P = Power dissipation

Package Thermal Characteristics

The device junction-to-case thermal resistivity is θ_{jc} and the junction-to-ambient air thermal resistivity is θ_{ja}. The thermal characteristics for θ_{ja} are shown for two air flow rates. The maximum operating junction temperature is 100°C. EQ 2-2 shows a sample calculation of the maximum operating power dissipation allowed for a 484-pin FBGA package at commercial temperature and in still air.

$$\text{Maximum Power Allowed} = \frac{\text{Max. junction temp. (°C)} - \text{Max. ambient temp. (°C)}}{\theta_{ja}(\text{°C/W})} = \frac{100\text{°C} - 70\text{°C}}{20.5\text{°C/W}} = 1.46\ \text{W}$$

EQ 2-2

Table 2-5 • **Package Thermal Resistivities**

Package Type	Pin Count	θ_{jc}	θ_{ja}			Units
			Still Air	200 ft./ min.	500 ft./ min.	
Chip Scale Package (CSP)	201	TBD	TBD	TBD	TBD	C/W
	281	TBD	TBD	TBD	TBD	C/W
	289	TBD	TBD	TBD	TBD	C/W

Temperature and Voltage Derating Factors

Table 2-6 • **Temperature and Voltage Derating Factors for Timing Delays (normalized to T_J = 70°C, V_{CC} = 1.425 V)**
For IGLOO PLUS V2 or V5 devices, 1.5 V DC Core Supply Voltage

Array Voltage V_{CC} (V)	Junction Temperature (°C)					
	–40°C	0°C	25°C	70°C	85°C	110°C
1.425	0.95	0.97	0.98	1.00	1.01	1.02
1.5	0.87	0.89	0.90	0.92	0.93	0.94
1.575	0.81	0.83	0.84	0.86	0.87	0.87

Table 2-7 • Temperature and Voltage Derating Factors for Timing Delays (normalized to T_J = 70°C, V_{CC} = 1.14 V)
For IGLOO PLUS V2, 1.2 V DC Core Supply Voltage

Array Voltage V_{CC} (V)	Junction Temperature (°C)					
	–40°C	0°C	25°C	70°C	85°C	110°C
1.14	0.97	0.98	0.99	1.00	1.01	1.01
1.2	0.86	0.87	0.88	0.89	0.89	0.90
1.26	0.79	0.80	0.81	0.81	0.82	0.82

Calculating Power Dissipation

Quiescent Supply Current

Quiescent supply current (I_{DD}) calculation depends on multiple factors, including operating voltages (V_{CC}, V_{CCI}, and V_{JTAG}), operating temperature, system clock frequency, and power mode usage. Actel recommends using the Power Calculator and SmartPower software estimation tools to evaluate the projected static and active power based on the user design, power mode usage, operating voltage, and temperature.

Table 2-8 • Quiescent Supply Current (I_{DD}) Characteristics, IGLOO PLUS Flash*Freeze Mode*

	Core Voltage	AGLP030	AGLP060	AGLP125	Units
Typical (25°C)	1.2 V	4	8	13	µA
	1.5 V	6	10	18	µA

* *I_{DD} includes V_{CC}, V_{PUMP}, V_{CCI}, V_{JTAG}, and V_{CCPLL} currents.*

Table 2-9 • Quiescent Supply Current (I_{DD}) Characteristics, IGLOO PLUS Sleep Mode (V_{CC} = 0 V)*

	Core Voltage	AGLP030	AGLP060	AGLP125	Units
V_{CCI}/V_{JTAG} = 1.2 V (per bank) Typical (25°C)	1.2 V	1.7	1.7	1.7	µA
V_{CCI}/V_{JTAG} = 1.5 V (per bank) Typical (25°C)	1.2 V / 1.5 V	1.8	1.8	1.8	µA
V_{CCI}/V_{JTAG} = 1.8 V (per bank) Typical (25°C)	1.2 V / 1.5 V	1.9	1.9	1.9	µA
V_{CCI}/V_{JTAG} = 2.5 V (per bank) Typical (25°C)	1.2 V / 1.5 V	2.2	2.2	2.2	µA
V_{CCI}/V_{JTAG} = 3.3 V (per bank) Typical (25°C)	1.2 V / 1.5 V	2.5	2.5	2.5	µA

* *I_{DD} includes V_{CC}, V_{PUMP}, and V_{CCPLL} currents.*

Table 2-10 • Quiescent Supply Current (I_{DD}) Characteristics, IGLOO PLUS Shutdown Mode (V_{CC}, V_{CCI} = 0 V)*

	Core Voltage	AGLP030	AGLP060	AGLP125	Units
Typical (25°C)	1.2 V / 1.5 V	0	0	0	µA

* *I_{DD} includes V_{CC}, V_{PUMP}, V_{CCI}, V_{JTAG}, and V_{CCPLL} currents.*

Table 2-11 • Quiescent Supply Current (I_{DD}), No IGLOO PLUS Flash*Freeze Mode[1]

	Core Voltage	AGLP030	AGLP060	AGLP125	Units
I_{CCA} Current[2]					
Typical (25°C)	1.2 V	6	10	13	µA
	1.5 V	16	20	28	µA
I_{CCI} or I_{JTAG} Current[3]					
V_{CCI} / V_{JTAG} = 1.2 V (per bank) Typical (25°C)	1.2 V	1.7	1.7	1.7	µA
V_{CCI} / V_{JTAG} = 1.5 V (per bank) Typical (25°C)	1.2 V / 1.5 V	1.8	1.8	1.8	µA
V_{CCI} / V_{JTAG} = 1.8 V (per bank) Typical (25°C)	1.2 V / 1.5 V	1.9	1.9	1.9	µA
V_{CCI} / V_{JTAG} = 2.5 V (per bank) Typical (25°C)	1.2 V / 1.5 V	2.2	2.2	2.2	µA
V_{CCI} / V_{JTAG} = 3.3 V (per bank) Typical (25°C)	1.2 V / 1.5 V	2.5	2.5	2.5	µA

Notes:

1. *To calculate total device I_{DD}, multiply the number of banks used by I_{CCI} and add I_{CCA} contribution.*
2. *Includes V_{CC}, V_{CCPLL}, and V_{PUMP} currents.*
3. *Per V_{CCI} or V_{JTAG} bank*

Power per I/O Pin

Table 2-12 • Summary of I/O Input Buffer Power (per pin) – Default I/O Software Settings

	V_{CCI} (V)	Dynamic Power P_{AC9} (µW/MHz) [1]
Single-Ended		
3.3 V LVTTL / 3.3 V LVCMOS	3.3	16.88
3.3 V LVTTL / 3.3 V LVCMOS – Schmitt Trigger	3.3	19.54
2.5 V LVCMOS	2.5	5.20
2.5 V LVCMOS – Schmitt Trigger	2.5	6.60
1.8 V LVCMOS	1.8	2.22
1.8 V LVCMOS – Schmitt Trigger	1.8	2.29
1.5 V LVCMOS (JESD8-11)	1.5	1.57
1.5 V LVCMOS (JESD8-11) – Schmitt Trigger	1.5	1.49
1.2 V LVCMOS [2]	1.2	0.55
1.2 V LVCMOS [2] – Schmitt Trigger	1.2	0.47

Notes:

1. *P_{AC9} is the total dynamic power measured on V_{CCI}.*
2. *Applicable to IGLOO PLUS V2 devices only.*

Table 2-13 • Summary of I/O Output Buffer Power (per pin) – Default I/O Software Settings[1]

	C_{LOAD} (pF)	V_{CCI} (V)	Dynamic Power P_{AC10} (µW/MHz)[2]
Single-Ended			
3.3 V LVTTL / 3.3 V LVCMOS	5	3.3	128.60
2.5 V LVCMOS	5	2.5	72.14
1.8 V LVCMOS	5	1.8	36.94
1.5 V LVCMOS (JESD8-11)	5	1.5	25.65
1.2 V LVCMOS[3]	5	1.2	15.22

Notes:

1. *Dynamic power consumption is given for standard load and software default drive strength and output slew.*
2. *P_{AC10} is the total dynamic power measured on V_{CCI}.*
3. *Applicable for IGLOO PLUS V2 devices only.*

Power Consumption of Various Internal Resources

Table 2-14 • Different Components Contributing to Dynamic Power Consumption in IGLOO PLUS Devices For IGLOO PLUS V2 or V5 Devices, 1.5 V Core Supply Voltage

Parameter	Definition	Device Specific Dynamic Power (µW/MHz)		
		AGLP125	AGLP060	AGLP030
P_{AC1}	Clock contribution of a Global Rib	11.03	9.3	9.3
P_{AC2}	Clock contribution of a Global Spine	0.81	0.81	0.41
P_{AC3}	Clock contribution of a VersaTile row	0.81		
P_{AC4}	Clock contribution of a VersaTile used as a sequential module	0.11		
P_{AC5}	First contribution of a VersaTile used as a sequential module	0.057		
P_{AC6}	Second contribution of a VersaTile used as a sequential module	0.207		
P_{AC7}	Contribution of a VersaTile used as a combinatorial module	0.17		
P_{AC8}	Average contribution of a routing net	0.7		
P_{AC9}	Contribution of an I/O input pin (standard-dependent)	See Table 2-12 on page 2-8.		
P_{AC10}	Contribution of an I/O output pin (standard-dependent)	See Table 2-13.		
P_{AC11}	Average contribution of a RAM block during a read operation	25.00		
P_{AC12}	Average contribution of a RAM block during a write operation	30.00		
P_{AC13}	Dynamic contribution for PLL	2.70		

Table 2-15 • Different Components Contributing to the Static Power Consumption in IGLOO PLUS Devices For IGLOO PLUS V2 or V5 Devices, 1.5 V Core Supply Voltage

Parameter	Definition	Device -Specific Static Power (mW)		
		AGLP125	AGLP060	AGLP030
P_{DC1}	Array static power in Active mode	See Table 2-11 on page 2-8		
P_{DC2}	Array static power in Static (Idle) mode	See Table 2-11 on page 2-8		
P_{DC3}	Array static power in Flash*Freeze mode	See Table 2-8 on page 2-7		
P_{DC4} [2]	Static PLL contribution	1.84		
P_{DC5}	Bank quiescent power (V_{CCI}-dependent)	See Table 2-11 on page 2-8		

Notes:

1. *For a different output load, drive strength, or slew rate, Actel recommends using the Actel power spreadsheet calculator or the SmartPower tool in Actel Libero® Integrated Design Environment (IDE).*
2. *Minimum contribution of the PLL when running at lowest frequency.*

Table 2-16 • Different Components Contributing to Dynamic Power Consumption in IGLOO PLUS Devices For IGLOO PLUS V2 Devices, 1.2 V Core Supply Voltage

Parameter	Definition	Device-Specific Dynamic Power (µW/MHz)		
		AGLP125	AGLP060	AGLP030
P_{AC1}	Clock contribution of a Global Rib	7.07	5.96	5.96
P_{AC2}	Clock contribution of a Global Spine	0.52	0.52	0.26
P_{AC3}	Clock contribution of a VersaTile row	0.52		
P_{AC4}	Clock contribution of a VersaTile used as a sequential module	0.07		
P_{AC5}	First contribution of a VersaTile used as a sequential module	0.045		
P_{AC6}	Second contribution of a VersaTile used as a sequential module	0.186		
P_{AC7}	Contribution of a VersaTile used as a combinatorial module	0.11		
P_{AC8}	Average contribution of a routing net	0.45		
P_{AC9}	Contribution of an I/O input pin (standard-dependent)	See Table 2-12 on page 2-8		
P_{AC10}	Contribution of an I/O output pin (standard-dependent)	See Table 2-13 on page 2-9		
P_{AC11}	Average contribution of a RAM block during a read operation	25.00		
P_{AC12}	Average contribution of a RAM block during a write operation	30.00		
P_{AC13}	Dynamic contribution for PLL	2.10		

Table 2-17 • Different Components Contributing to the Static Power Consumption in IGLOO PLUS Devices For IGLOO PLUS V2 Devices, 1.2 V Core Supply Voltage

Parameter	Definition	Device-Specific Static Power (mW)		
		AGLP125	AGLP060	AGLP030
P_{DC1}	Array static power in Active mode	See Table 2-11 on page 2-8		
P_{DC2}	Array static power in Static (Idle) mode	See Table 2-11 on page 2-8		
P_{DC3}	Array static power in Flash*Freeze mode	See Table 2-8 on page 2-7		
P_{DC4} [2]	Static PLL contribution	0.90		
P_{DC5}	Bank quiescent power (V_{CCI}-dependent)	See Table 2-11 on page 2-8		

Notes:

1. *For a different output load, drive strength, or slew rate, Actel recommends using the Actel power spreadsheet calculator or the SmartPower tool in Actel Libero IDE.*
2. *Minimum contribution of the PLL when running at lowest frequency.*

Power Calculation Methodology

This section describes a simplified method to estimate power consumption of an application. For more accurate and detailed power estimations, use the SmartPower tool in Actel Libero IDE software.

The power calculation methodology described below uses the following variables:

- The number of PLLs as well as the number and the frequency of each output clock generated
- The number of combinatorial and sequential cells used in the design
- The internal clock frequencies
- The number and the standard of I/O pins used in the design
- The number of RAM blocks used in the design
- Toggle rates of I/O pins as well as VersaTiles—guidelines are provided in Table 2-18 on page 2-14.
- Enable rates of output buffers—guidelines are provided for typical applications in Table 2-19 on page 2-14.
- Read rate and write rate to the memory—guidelines are provided for typical applications in Table 2-19 on page 2-14. The calculation should be repeated for each clock domain defined in the design.

Methodology

Total Power Consumption—P_{TOTAL}

$P_{TOTAL} = P_{STAT} + P_{DYN}$

P_{STAT} is the total static power consumption.

P_{DYN} is the total dynamic power consumption.

Total Static Power Consumption—P_{STAT}

$P_{STAT} = (P_{DC1}$ or P_{DC2} or $P_{DC3}) + N_{BANKS} * P_{DC5}$

N_{BANKS} is the number of I/O banks powered in the design.

Total Dynamic Power Consumption—P_{DYN}

$P_{DYN} = P_{CLOCK} + P_{S\text{-}CELL} + P_{C\text{-}CELL} + P_{NET} + P_{INPUTS} + P_{OUTPUTS} + P_{MEMORY} + P_{PLL}$

Global Clock Contribution—P_{CLOCK}

$P_{CLOCK} = (P_{AC1} + N_{SPINE}*P_{AC2} + N_{ROW}*P_{AC3} + N_{S\text{-}CELL}* P_{AC4}) * F_{CLK}$

N_{SPINE} is the number of global spines used in the user design—guidelines are provided in Table 2-18 on page 2-14.

N_{ROW} is the number of VersaTile rows used in the design—guidelines are provided in Table 2-18 on page 2-14.

F_{CLK} is the global clock signal frequency.

$N_{S\text{-}CELL}$ is the number of VersaTiles used as sequential modules in the design.

P_{AC1}, P_{AC2}, P_{AC3}, and P_{AC4} are device-dependent.

Sequential Cells Contribution—$P_{S\text{-}CELL}$

$P_{S\text{-}CELL} = N_{S\text{-}CELL} * (P_{AC5} + \alpha_1 / 2 * P_{AC6}) * F_{CLK}$

$N_{S\text{-}CELL}$ is the number of VersaTiles used as sequential modules in the design. When a multi-tile sequential cell is used, it should be accounted for as 1.

α_1 is the toggle rate of VersaTile outputs—guidelines are provided in Table 2-18 on page 2-14.

F_{CLK} is the global clock signal frequency.

Combinatorial Cells Contribution—$P_{C\text{-}CELL}$

$P_{C\text{-}CELL} = N_{C\text{-}CELL} * \alpha_1 / 2 * P_{AC7} * F_{CLK}$

$N_{C\text{-}CELL}$ is the number of VersaTiles used as combinatorial modules in the design.

α_1 is the toggle rate of VersaTile outputs—guidelines are provided in Table 2-18 on page 2-14.

F_{CLK} is the global clock signal frequency.

Routing Net Contribution—P_{NET}

$P_{NET} = (N_{S\text{-}CELL} + N_{C\text{-}CELL}) * \alpha_1 / 2 * P_{AC8} * F_{CLK}$

$N_{S\text{-}CELL}$ is the number of VersaTiles used as sequential modules in the design.

$N_{C\text{-}CELL}$ is the number of VersaTiles used as combinatorial modules in the design.

α_1 is the toggle rate of VersaTile outputs—guidelines are provided in Table 2-18 on page 2-14.

F_{CLK} is the global clock signal frequency.

I/O Input Buffer Contribution—P_{INPUTS}

$P_{INPUTS} = N_{INPUTS} * \alpha_2 / 2 * P_{AC9} * F_{CLK}$

N_{INPUTS} is the number of I/O input buffers used in the design.

α_2 is the I/O buffer toggle rate—guidelines are provided in Table 2-18 on page 2-14.

F_{CLK} is the global clock signal frequency.

I/O Output Buffer Contribution—$P_{OUTPUTS}$

$P_{OUTPUTS} = N_{OUTPUTS} * \alpha_2 / 2 * \beta_1 * P_{AC10} * F_{CLK}$

$N_{OUTPUTS}$ is the number of I/O output buffers used in the design.

α_2 is the I/O buffer toggle rate—guidelines are provided in Table 2-18 on page 2-14.

β_1 is the I/O buffer enable rate—guidelines are provided in Table 2-19 on page 2-14.

F_{CLK} is the global clock signal frequency.

RAM Contribution—P_{MEMORY}

$P_{MEMORY} = P_{AC11} * N_{BLOCKS} * F_{READ\text{-}CLOCK} * \beta_2 + P_{AC12} * N_{BLOCK} * F_{WRITE\text{-}CLOCK} * \beta_3$

N_{BLOCKS} is the number of RAM blocks used in the design.

$F_{READ\text{-}CLOCK}$ is the memory read clock frequency.

β_2 is the RAM enable rate for read operations.

$F_{WRITE\text{-}CLOCK}$ is the memory write clock frequency.

β_3 is the RAM enable rate for write operations—guidelines are provided in Table 2-19 on page 2-14.

PLL Contribution—P_{PLL}

$P_{PLL} = P_{DC4} + P_{AC13} * F_{CLKOUT}$

F_{CLKOUT} is the output clock frequency.[1]

1. *If a PLL is used to generate more than one output clock, include each output clock in the formula by adding its corresponding contribution ($P_{AC13} * F_{CLKOUT}$ product) to the total PLL contribution.*

Guidelines

Toggle Rate Definition

A toggle rate defines the frequency of a net or logic element relative to a clock. It is a percentage. If the toggle rate of a net is 100%, this means that this net switches at half the clock frequency. Below are some examples:

- The average toggle rate of a shift register is 100% because all flip-flop outputs toggle at half of the clock frequency.
- The average toggle rate of an 8-bit counter is 25%:
  - Bit 0 (LSB) = 100%
  - Bit 1 = 50%
  - Bit 2 = 25%
  - …
  - Bit 7 (MSB) = 0.78125%
  - Average toggle rate = (100% + 50% + 25% + 12.5% + . . . + 0.78125%) / 8

Enable Rate Definition

Output enable rate is the average percentage of time during which tristate outputs are enabled. When nontristate output buffers are used, the enable rate should be 100%.

Table 2-18 • **Toggle Rate Guidelines Recommended for Power Calculation**

Component	Definition	Guideline
α_1	Toggle rate of VersaTile outputs	10%
α_2	I/O buffer toggle rate	10%

Table 2-19 • **Enable Rate Guidelines Recommended for Power Calculation**

Component	Definition	Guideline
β_1	I/O output buffer enable rate	100%
β_2	RAM enable rate for read operations	12.5%
β_3	RAM enable rate for write operations	12.5%

User I/O Characteristics

Timing Model

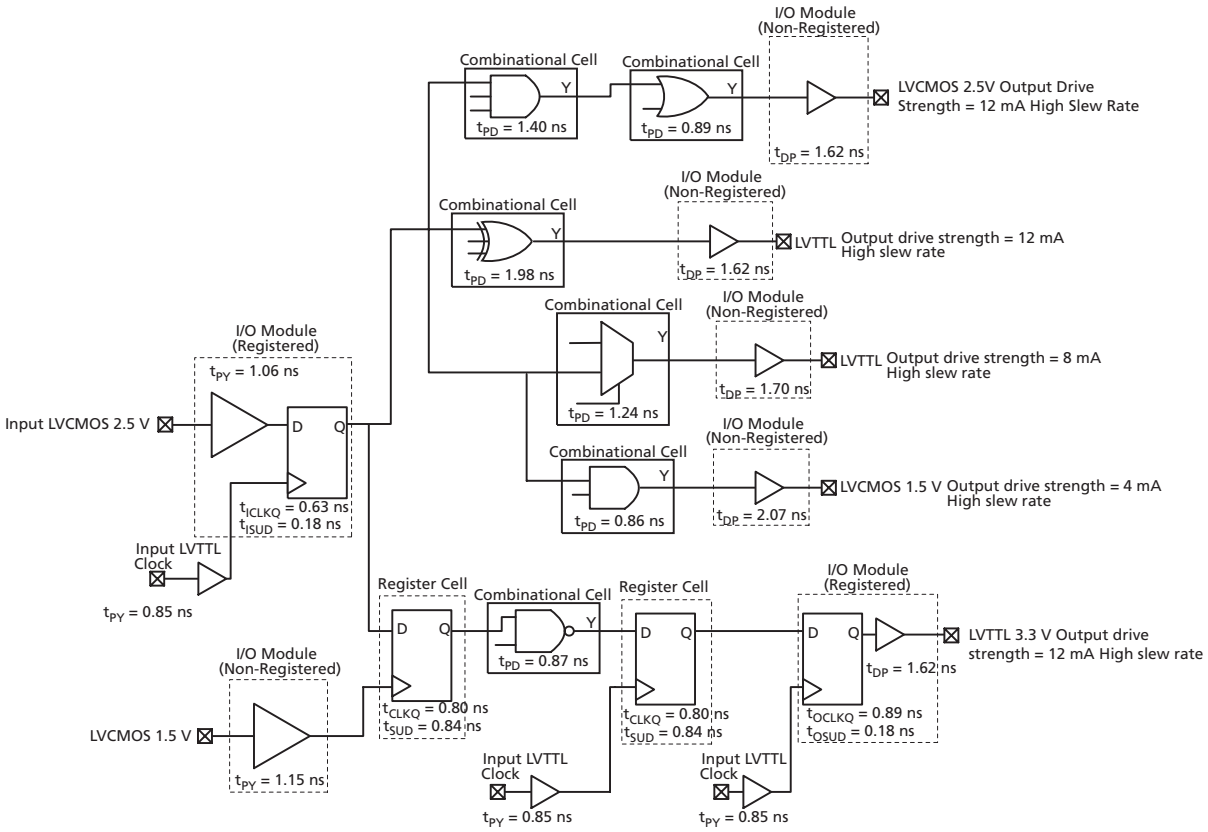


Figure 2-3 • **Timing Model**
Operating Conditions: STD Speed, Commercial Temperature Range (T_J = 70°C), Worst-Case V_{CC} = 1.425 V, for DC 1.5 V Core Voltage, Applicable to V2 and V5 Devices

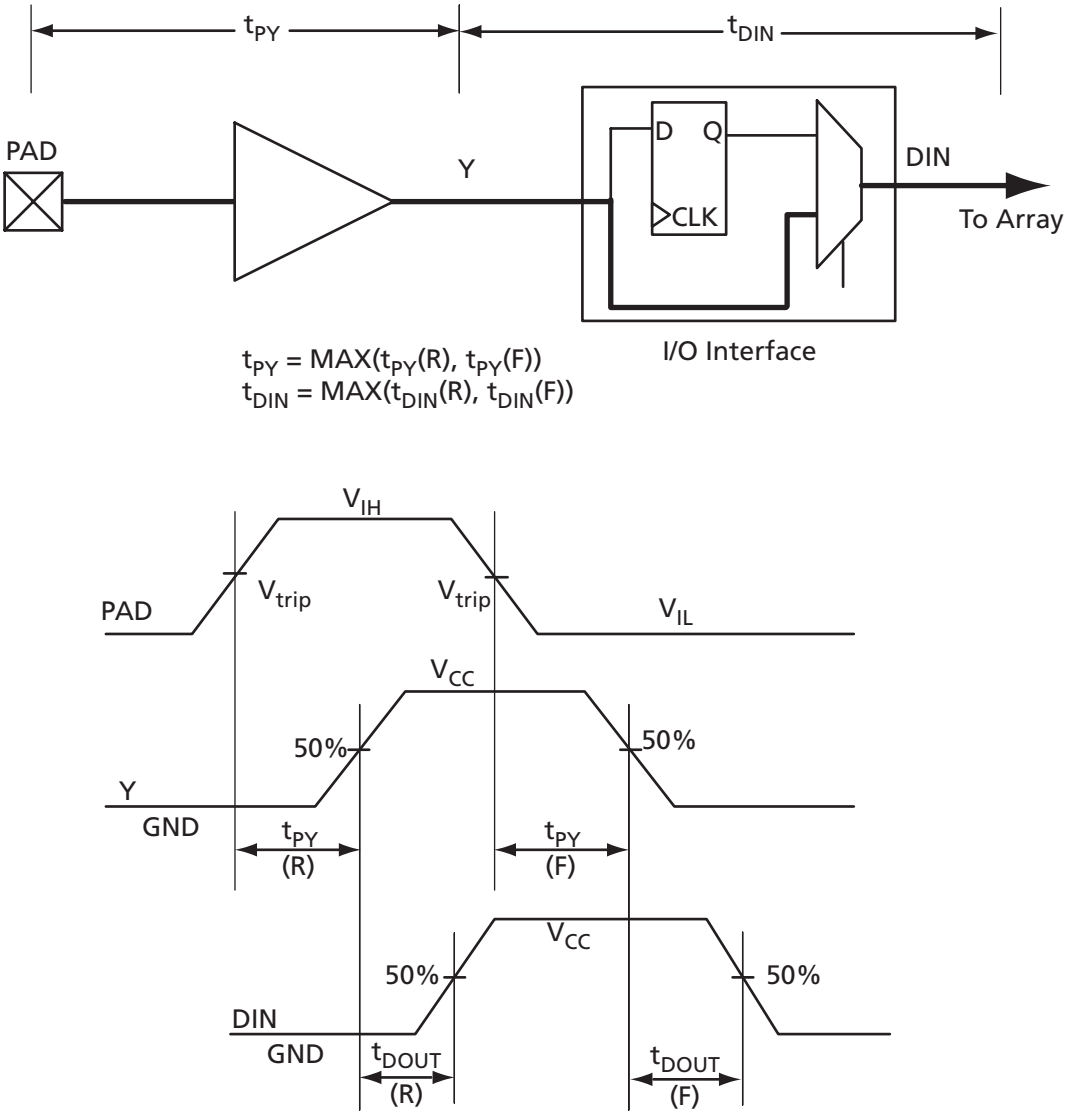


Figure 2-4 • **Input Buffer Timing Model and Delays (example)**

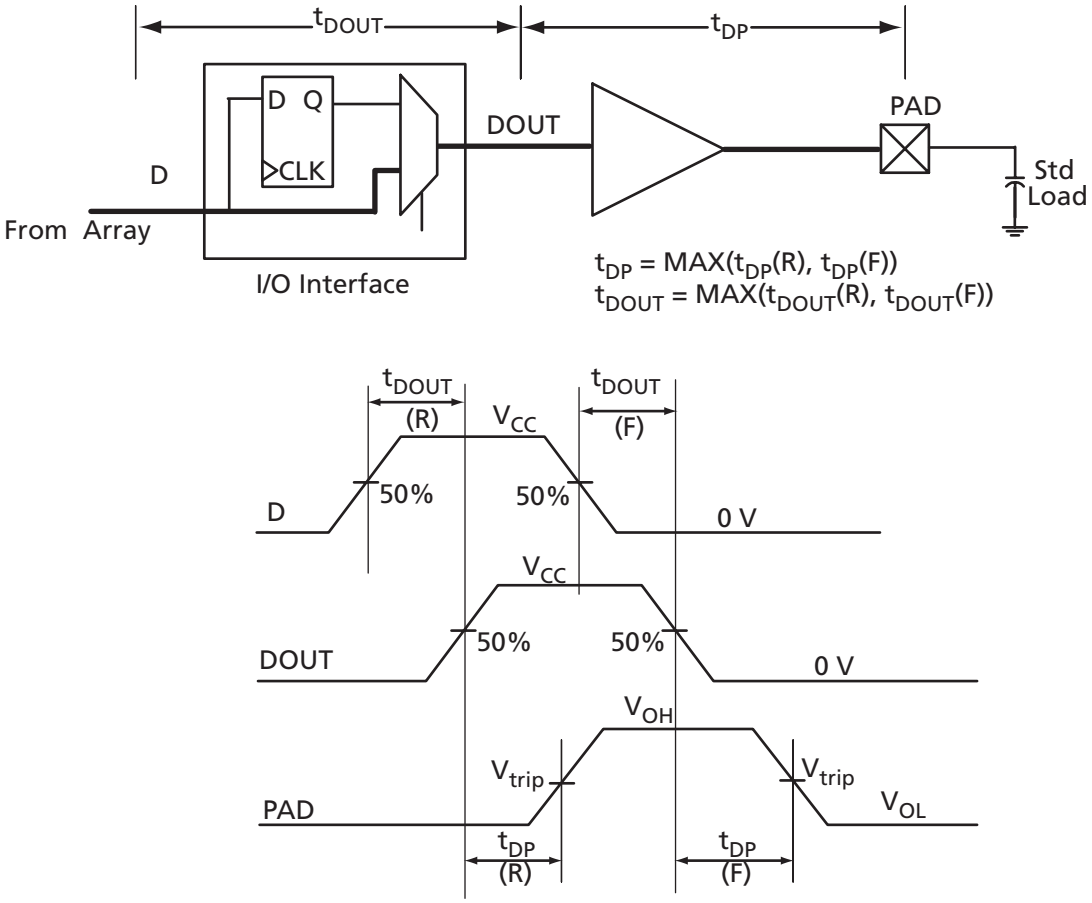


Figure 2-5 • **Output Buffer Model and Delays (example)**

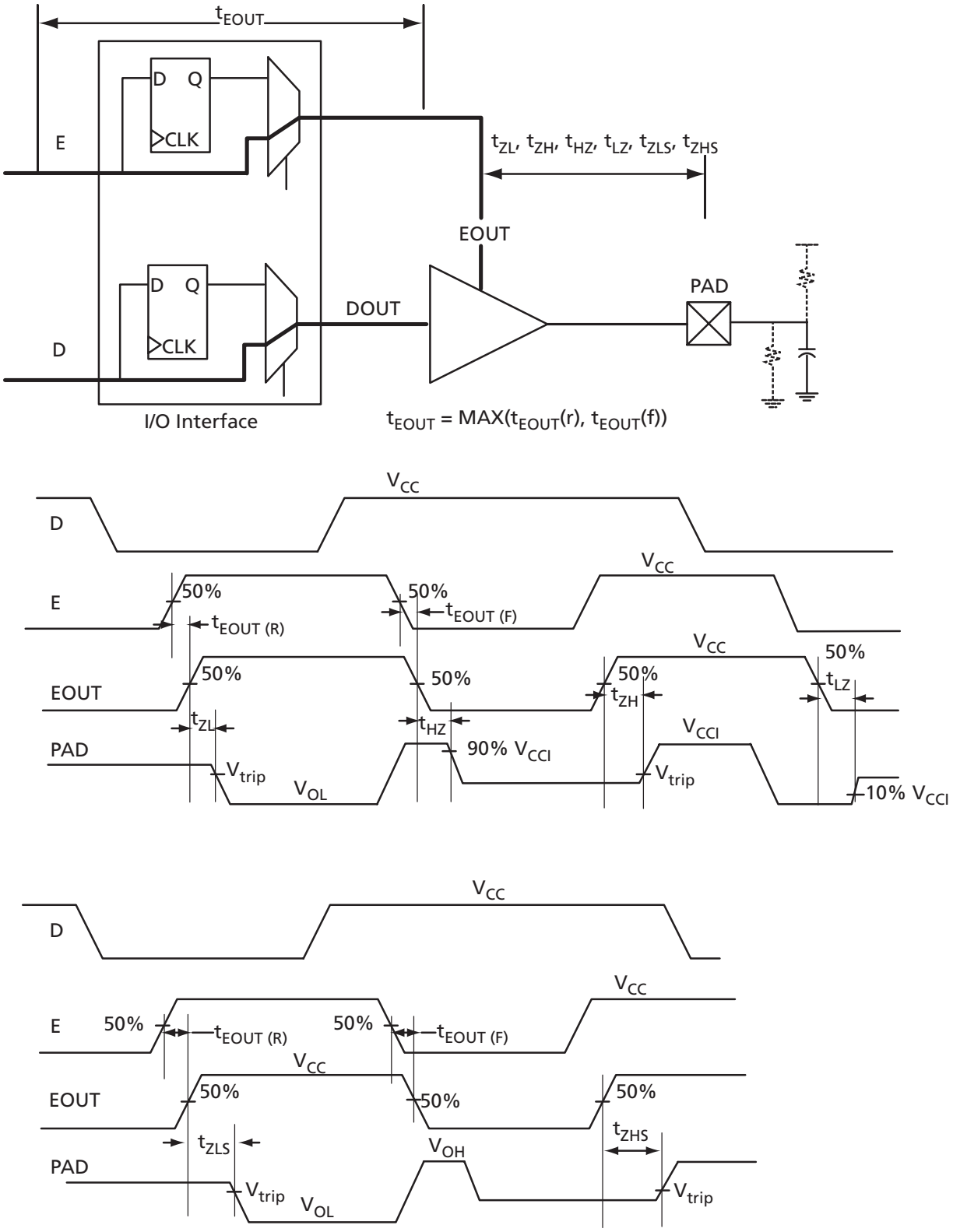


Figure 2-6 • **Tristate Output Buffer Timing Model and Delays (example)**

Overview of I/O Performance

Summary of I/O DC Input and Output Levels – Default I/O Software Settings

Table 2-20 • Summary of Maximum and Minimum DC Input and Output Levels Applicable to Commercial and Industrial Conditions—Software Default Settings

I/O Standard	Drive Strength	Slew Rate	V_{IL}		V_{IH}		V_{OL}	V_{OH}	I_{OL}	I_{OH}
			Min, V	Max, V	Min, V	Max, V	Max, V	Min, V	mA	mA
3.3 V LVTTL / 3.3 V LVCMOS	12 mA	High	–0.3	0.8	2	3.6	0.4	2.4	12	12
2.5 V LVCMOS	12 mA	High	–0.3	0.7	1.7	2.7	0.7	1.7	12	12
1.8 V LVCMOS	8 mA	High	–0.3	0.35 * V_{CCI}	0.65 * V_{CCI}	1.9	0.45	V_{CCI} – 0.45	8	8
1.5 V LVCMOS	4 mA	High	–0.3	0.35 * V_{CCI}	0.65 * V_{CCI}	1.575	0.25 * V_{CCI}	0.75 * V_{CCI}	4	4
1.2 V LVCMOS[2]	2 mA	High	–0.3	0.35 * V_{CCI}	0.65 * V_{CCI}	1.26	0.25 * V_{CCI}	0.75 * V_{CCI}	2	2

Notes:

1. *Currents are measured at 85°C junction temperature.*
2. *Applicable to IGLOO PLUS V2 devices operating at $V_{CCI} \geq V_{CC}$*

Table 2-21 • Summary of Maximum and Minimum DC Input Levels Applicable to Commercial and Industrial Conditions

DC I/O Standards	Commercial[1]		Industrial[2]	
	I_{IL}	I_{IH}	I_{IL}	I_{IH}
	µA	µA	µA	µA
3.3 V LVTTL / 3.3 V LVCMOS	10	10	15	15
2.5 V LVCMOS	10	10	15	15
1.8 V LVCMOS	10	10	15	15
1.5 V LVCMOS	10	10	15	15
1.2 V LVCMOS[3]	10	10	15	15

Notes:

1. *Commercial range (0°C < T_A < 70°C)*
2. *Industrial range (–40°C < T_A < 85°C)*
3. *Applicable to IGLOO PLUS V2 devices operating at $V_{CCI} \geq V_{CC}$*

Summary of I/O Timing Characteristics – Default I/O Software Settings

Table 2-22 • Summary of AC Measuring Points

Standard	Measuring Trip Point (V_{trip})
3.3 V LVTTL / 3.3 V LVCMOS	1.4 V
2.5 V LVCMOS	1.2 V
1.8 V LVCMOS	0.90 V
1.5 V LVCMOS	0.75 V
1.2 V LVCMOS	0.60 V

Table 2-23 • I/O AC Parameter Definitions

Parameter	Parameter Definition
t_{DP}	Data to Pad delay through the Output Buffer
t_{PY}	Pad to Data delay through the Input Buffer
t_{DOUT}	Data to Output Buffer delay through the I/O interface
t_{EOUT}	Enable to Output Buffer Tristate Control delay through the I/O interface
t_{DIN}	Input Buffer to Data delay through the I/O interface
t_{HZ}	Enable to Pad delay through the Output Buffer—HIGH to Z
t_{ZH}	Enable to Pad delay through the Output Buffer—Z to HIGH
t_{LZ}	Enable to Pad delay through the Output Buffer—LOW to Z
t_{ZL}	Enable to Pad delay through the Output Buffer—Z to LOW
t_{ZHS}	Enable to Pad delay through the Output Buffer with delayed enable—Z to HIGH
t_{ZLS}	Enable to Pad delay through the Output Buffer with delayed enable—Z to LOW

Table 2-24 • Summary of I/O Timing Characteristics—Software Default Settings, STD Speed Grade, Commercial-Case Conditions: T_J = 70°C, Worst-Case V_{CC} = 1.425 V, Worst-Case V_{CCI} = 3.0 V

I/O Standard	Drive Strength (mA)	Slew Rate	Capacitive Load (pF)	External Resistor (Ω)	t_{DOUT}	t_{DP}	t_{DIN}	t_{PY}	t_{PYS}	t_{EOUT}	t_{ZL}	t_{ZH}	t_{LZ}	t_{HZ}	Units
3.3 V LVTTL / 3.3 V LVCMOS	12 mA	High	5 pF	–	0.97	1.62	0.18	0.85	1.14	0.66	1.65	1.27	2.20	2.64	ns
2.5 V LVCMOS	12 mA	High	5 pF	–	0.97	1.62	0.18	1.06	1.22	0.66	1.65	1.34	2.22	2.56	ns
1.8 V LVCMOS	8 mA	High	5 pF	–	0.97	1.82	0.18	0.99	1.43	0.66	1.85	1.53	2.29	2.54	ns
1.5 V LVCMOS	4 mA	High	5 pF	–	0.97	2.07	0.18	1.15	1.62	0.66	2.10	1.71	2.37	2.57	ns

Notes:

1. *For specific junction temperature and voltage supply levels, refer to Table 2-6 on page 2-6 for derating values.*

Table 2-25 • Summary of I/O Timing Characteristics—Software Default Settings, STD Speed Grade, Commercial-Case Conditions: T_J = 70°C, Worst-Case V_{CC} = 1.14 V, Worst-Case V_{CCI} = 3.0 V

I/O Standard	Drive Strength (mA)	Slew Rate	Capacitive Load (pF)	External Resistor (Ω)	t_{DOUT}	t_{DP}	t_{DIN}	t_{PY})	t_{PYS}	t_{EOUT}	t_{ZL}	t_{ZH}	t_{LZ}	t_{HZ}	Units
3.3 V LVTTL / 3.3 V LVCMOS	12 mA	High	5 pF	–	0.98	2.16	0.19	0.99	1.37	0.67	2.20	1.74	2.64	3.37	ns
2.5 V LVCMOS	12 mA	High	5 pF	–	0.98	2.13	0.19	1.20	1.40	0.67	2.17	1.77	2.65	3.25	ns
1.8 V LVCMOS	8 mA	High	5 pF	–	0.98	2.25	0.19	1.12	1.60	0.67	2.30	1.92	2.70	3.15	ns
1.5 V LVCMOS	4 mA	High	5 pF	–	0.98	2.48	0.19	1.26	1.79	0.67	2.52	2.10	2.77	3.14	ns
1.2 V LVCMOS[2]	2 mA	High	5 pF	–	0.98	2.68	0.19	1.56	2.34	0.67	2.73	2.24	2.53	2.67	ns

Notes:

1. *For specific junction temperature and voltage supply levels, refer to Table 2-6 on page 2-6 for derating values.*
2. *Applicable to IGLOO PLUS V2 devices operating at $V_{CCI} \geq V_{CC}$.*

Detailed I/O DC Characteristics

Table 2-26 • **Input Capacitance**

Symbol	Definition	Conditions	Min.	Max.	Units
C_{IN}	Input capacitance	V_{IN} = 0, f = 1.0 MHz		8	pF
C_{INCLK}	Input capacitance on the clock pin	V_{IN} = 0, f = 1.0 MHz		8	pF

Table 2-27 • **I/O Output Buffer Maximum Resistances[1]**

Standard	Drive Strength	$R_{PULL\text{-}DOWN}$ (Ω)[2]	$R_{PULL\text{-}UP}$ (Ω)[3]
3.3 V LVTTL / 3.3V LVCMOS	2 mA	100	300
	4 mA	100	300
	6 mA	50	150
	8m A	50	150
	12 mA	25	75
	16 mA	25	75
2.5 V LVCMOS	2 mA	100	200
	4 mA	100	200
	6 mA	50	100
	8 mA	50	100
	12 mA	25	50
1.8 V LVCMOS	2 mA	200	225
	4 mA	100	112
	6 mA	50	56
	8 mA	50	56
1.5 V LVCMOS	2 mA	200	224
	4 mA	100	112
1.2 V LVCMOS	2 mA	TBD	TBD

Notes:

1. *These maximum values are provided for informational reasons only. Minimum output buffer resistance values depend on V_{CCI}, drive strength selection, temperature, and process. For board design considerations and detailed output buffer resistances, use the corresponding IBIS models located on the Actel website at http://www.actel.com/download/ibis/default.aspx.*
2. $R_{(PULL\text{-}DOWN\text{-}MAX)} = (V_{OLspec}) / I_{OLspec}$
3. $R_{(PULL\text{-}UP\text{-}MAX)} = (V_{CCImax} - V_{OHspec}) / I_{OHspec}$

Table 2-28 • I/O Weak Pull-Up/Pull-Down Resistances
Minimum and Maximum Weak Pull-Up/Pull-Down Resistance Values

	$R_{(WEAK\ PULL\text{-}UP)}$[1] (Ω)		$R_{(WEAK\ PULL\text{-}DOWN)}$[2] (Ω)	
V_{CCI}	Min.	Max.	Min.	Max.
3.3 V	10 k	45 k	10 k	45 k
2.5 V	11 k	55 k	12 k	74 k
1.8 V	18 k	70 k	17 k	110 k
1.5 V	19 k	90 k	19 k	140 k
1.2 V	TBD	TBD	TBD	TBD

Notes:

1. $R_{(WEAK\ PULL\text{-}UP\text{-}MAX)} = (V_{OLspec}) / I_{(WEAK\ PULL\text{-}UP\text{-}MIN)}$
2. $R_{(WEAK\ PULL\text{-}UP\text{-}MAX)} = (V_{CCImax} – V_{OHspec}) / I_{(WEAK\ PULL\text{-}UP\text{-}MIN)}$

Table 2-29 • I/O Short Currents I_{OSH}/I_{OSL}

	Drive Strength	I_{OSL} (mA)*	I_{OSH} (mA)*
3.3 V LVTTL / 3.3 V LVCMOS	2 mA	25	27
	4 mA	25	27
	6 mA	51	54
	8 mA	51	54
	12 mA	103	109
	16 mA	103	109
2.5 V LVCMOS	2 mA	16	18
	4 mA	16	18
	6 mA	32	37
	8 mA	32	37
	12 mA	65	74
1.8 V LVCMOS	2 mA	9	11
	4 mA	17	22
	6 mA	35	44
	8 mA	35	44
1.5 V LVCMOS	2 mA	13	16
	4 mA	25	33
1.2 V LVCMOS	2 mA	TBD	TBD

* T_J = 100°C

The length of time an I/O can withstand I_{OSH}/I_{OSL} events depends on the junction temperature. The reliability data below is based on a 3.3 V, 12 mA I/O setting, which is the worst case for this type of analysis.

For example, at 110°C, the short current condition would have to be sustained for more than three months to cause a reliability concern. The I/O design does not contain any short circuit protection, but such protection would only be needed in extremely prolonged stress conditions.

Table 2-30 • **Duration of Short Circuit Event before Failure**

Temperature	Time before Failure
–40°C	> 20 years
0°C	> 20 years
25°C	> 20 years
70°C	5 years
85°C	2 years
100°C	6 months
110°C	3 months

Table 2-31 • **Schmitt Trigger Input Hysteresis**
Hysteresis Voltage Value (Typ.) for Schmitt Mode Input Buffers

Input Buffer Configuration	Hysteresis Value (typ.)
3.3 V LVTTL/LVCMOS (Schmitt trigger mode)	240 mV
2.5 V LVCMOS (Schmitt trigger mode)	140 mV
1.8 V LVCMOS (Schmitt trigger mode)	80 mV
1.5 V LVCMOS (Schmitt trigger mode)	60 mV
1.2 V LVCMOS (Schmitt trigger mode)	40 mV

Table 2-32 • **I/O Input Rise Time, Fall Time, and Related I/O Reliability**

Input Buffer	Input Rise/Fall Time (min.)	Input Rise/Fall Time (max.)	Reliability
LVTTL/LVCMOS (Schmitt trigger disabled)	No requirement	10 ns *	20 years (100°C)
LVTTL/LVCMOS (Schmitt trigger enabled)	No requirement	No requirement, but input noise voltage cannot exceed Schmitt hysteresis.	20 years (100°C)

* *The maximum input rise/fall time is related to the noise induced into the input buffer trace. If the noise is low, then the rise time and fall time of input buffers can be increased beyond the maximum value. The longer the rise/fall times, the more susceptible the input signal is to the board noise. Actel recommends signal integrity evaluation/characterization of the system to ensure that there is no excessive noise coupling into input signals.*

Single-Ended I/O Characteristics

3.3 V LVTTL / 3.3 V LVCMOS

Low-Voltage Transistor–Transistor Logic (LVTTL) is a general-purpose standard (EIA/JESD) for 3.3 V applications. It uses an LVTTL input buffer and push-pull output buffer.

Table 2-33 • Minimum and Maximum DC Input and Output Levels

3.3 V LVTTL / 3.3 V LVCMOS	V_{IL}		V_{IH}		V_{OL}	V_{OH}	I_{OL}	I_{OH}	I_{OSL}	I_{OSH}	I_{IL}	I_{IH}
Drive Strength	Min., V	Max., V	Min., V	Max., V	Max., V	Min., V	mA	mA	Max., mA[1]	Max., mA[1]	µA[2]	µA[2]
2 mA	–0.3	0.8	2	3.6	0.4	2.4	2	2	25	27	10	10
4 mA	–0.3	0.8	2	3.6	0.4	2.4	4	4	25	27	10	10
6 mA	–0.3	0.8	2	3.6	0.4	2.4	6	6	51	54	10	10
8 mA	–0.3	0.8	2	3.6	0.4	2.4	8	8	51	54	10	10
12 mA	–0.3	0.8	2	3.6	0.4	2.4	12	12	103	109	10	10
16 mA	–0.3	0.8	2	3.6	0.4	2.4	16	16	103	109	10	10

Notes:

1. *Currents are measured at high temperature (100°C junction temperature) and maximum voltage.*
2. *Currents are measured at 85°C junction temperature.*
3. *Software default selection highlighted in gray.*

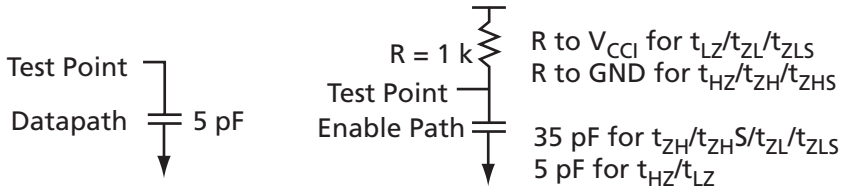


Figure 2-7 • AC Loading

Table 2-34 • AC Waveforms, Measuring Points, and Capacitive Loads

Input LOW (V)	Input HIGH (V)	Measuring Point* (V)	C_{LOAD} (pF)
0	3.3	1.4	5

* *Measuring point = V_{trip}. See Table 2-22 on page 2-20 for a complete table of trip points.*

Timing Characteristics

Applies to 1.5 V DC Core Voltage

Table 2-35 • 3.3 V LVTTL / 3.3 V LVCMOS Low Slew – Applies to 1.5 V DC Core Voltage
Commercial-Case Conditions: T_J = 70°C, Worst-Case V_{CC} = 1.425 V, Worst-Case V_{CCI} = 3.0 V

Drive Strength	Speed Grade	t_{DOUT}	t_{DP}	t_{DIN}	t_{PY}	t_{PYS}	t_{EOUT}	t_{ZL}	t_{ZH}	t_{LZ}	t_{HZ}	Units
4 mA	STD	0.97	3.33	0.18	0.85	1.14	0.66	3.39	2.95	1.82	1.87	ns
6 mA	STD	0.97	2.83	0.18	0.85	1.14	0.66	2.88	2.65	2.04	2.27	ns
8 mA	STD	0.97	2.83	0.18	0.85	1.14	0.66	2.88	2.65	2.04	2.27	ns
12 mA	STD	0.97	2.48	0.18	0.85	1.14	0.66	2.52	2.38	2.20	2.53	ns
16 mA	STD	0.97	2.48	0.18	0.85	1.14	0.66	2.52	2.38	2.20	2.53	ns

Note: For specific junction temperature and voltage supply levels, refer to Table 2-6 on page 2-6 for derating values.

Table 2-36 • 3.3 V LVTTL / 3.3 V LVCMOS High Slew – Applies to 1.5 V DC Core Voltage
Commercial-Case Conditions: T_J = 70°C, Worst-Case V_{CC} = 1.425 V, Worst-Case V_{CCI} = 3.0 V

Drive Strength	Speed Grade	t_{DOUT}	t_{DP}	t_{DIN}	t_{PY}	t_{PYS}	t_{EOUT}	t_{ZL}	t_{ZH}	t_{LZ}	t_{HZ}	Units
4 mA	STD	0.97	1.84	0.18	0.85	1.14	0.66	1.88	1.43	1.81	1.98	ns
6 mA	STD	0.97	1.70	0.18	0.85	1.14	0.66	1.73	1.32	2.04	2.38	ns
8 mA	STD	0.97	1.70	0.18	0.85	1.14	0.66	1.73	1.32	2.04	2.38	ns
12 mA	STD	0.97	1.62	0.18	0.85	1.14	0.66	1.65	1.27	2.20	2.64	ns
16 mA	STD	0.97	1.62	0.18	0.85	1.14	0.66	1.65	1.27	2.20	2.64	ns

Note: For specific junction temperature and voltage supply levels, refer to Table 2-6 on page 2-6 for derating values.

Applies to 1.2 V DC Core Voltage

Table 2-37 • 3.3 V LVTTL / 3.3 V LVCMOS Low Slew – Applies to 1.2 V DC Core Voltage
Commercial-Case Conditions: T_J = 70°C, Worst-Case V_{CC} = 1.14 V, Worst-Case V_{CCI} = 3.0 V

Drive Strength	Speed Grade	t_{DOUT}	t_{DP}	t_{DIN}	t_{PY}	t_{PYS}	t_{EOUT}	t_{ZL}	t_{ZH}	t_{LZ}	t_{HZ}	Units
4 mA	STD	0.98	3.92	0.19	0.99	1.37	0.67	3.99	3.47	2.25	2.56	ns
6 mA	STD	0.98	3.40	0.19	0.99	1.37	0.67	3.47	3.15	2.48	2.97	ns
8 mA	STD	0.98	3.40	0.19	0.99	1.37	0.67	3.47	3.15	2.48	2.97	ns
12 mA	STD	0.98	3.04	0.19	0.99	1.37	0.67	3.10	2.88	2.64	3.24	ns
16 mA	STD	0.98	3.04	0.19	0.99	1.37	0.67	3.10	2.88	2.64	3.24	ns

Note: For specific junction temperature and voltage supply levels, refer to Table 2-6 on page 2-6 for derating values.

Table 2-38 • 3.3 V LVTTL / 3.3 V LVCMOS High Slew – Applies to 1.2 V DC Core Voltage
Commercial-Case Conditions: T_J = 70°C, Worst-Case V_{CC} = 1.14 V, Worst-Case V_{CCI} = 3.0 V

Drive Strength	Speed Grade	t_{DOUT}	t_{DP}	t_{DIN}	t_{PY}	t_{PYS}	t_{EOUT}	t_{ZL}	t_{ZH}	t_{LZ}	t_{HZ}	Units
4 mA	STD	0.98	2.39	0.19	0.99	1.37	0.67	2.43	1.91	2.24	2.68	ns
6 mA	STD	0.98	2.24	0.19	0.99	1.37	0.67	2.28	1.80	2.48	3.10	ns
8 mA	STD	0.98	2.24	0.19	0.99	1.37	0.67	2.28	1.80	2.48	3.10	ns
12 mA	STD	0.98	2.16	0.19	0.99	1.37	0.67	2.20	1.74	2.64	3.37	ns
16 mA	STD	0.98	2.16	0.19	0.99	1.37	0.67	2.20	1.74	2.64	3.37	ns

Note: For specific junction temperature and voltage supply levels, refer to Table 2-6 on page 2-6 for derating values.

2.5 V LVCMOS

Low-Voltage CMOS for 2.5 V is an extension of the LVCMOS standard (JESD8-5) used for general-purpose 2.5 V applications. It uses a 5 V–tolerant input buffer and push-pull output buffer.

Table 2-39 • Minimum and Maximum DC Input and Output Levels

2.5 V LVCMOS	V_{IL}		V_{IH}		V_{OL}	V_{OH}	I_{OL}	I_{OH}	I_{OSL}	I_{OSH}	I_{IL}	I_{IH}
Drive Strength	Min., V	Max., V	Min., V	Max., V	Max., V	Min., V	mA	mA	Max., mA[1]	Max., mA[1]	µA[2]	µA[2]
2 mA	–0.3	0.7	1.7	2.7	0.7	1.7	2	2	16	18	10	10
4 mA	–0.3	0.7	1.7	2.7	0.7	1.7	4	4	16	18	10	10
6 mA	–0.3	0.7	1.7	2.7	0.7	1.7	6	6	32	37	10	10
8 mA	–0.3	0.7	1.7	2.7	0.7	1.7	8	8	32	37	10	10
12 mA	–0.3	0.7	1.7	2.7	0.7	1.7	12	12	65	74	10	10

Notes:

1. *Currents are measured at high temperature (100°C junction temperature) and maximum voltage.*
2. *Currents are measured at 85°C junction temperature.*
3. *Software default selection highlighted in gray.*

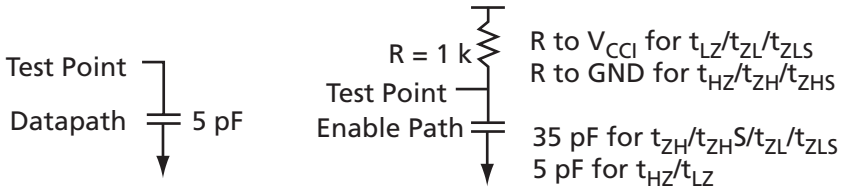


Figure 2-8 • AC Loading

Table 2-40 • AC Waveforms, Measuring Points, and Capacitive Loads

Input LOW (V)	Input HIGH (V)	Measuring Point* (V)	C_{LOAD} (pF)
0	2.5	1.2	5

* *Measuring point = V_{trip}. See Table 2-22 on page 2-20 for a complete table of trip points.*

Timing Characteristics

Applies to 1.5 V DC Core Voltage

Table 2-41 • 2.5 V LVCMOS Low Slew – Applies to 1.5 V DC Core Voltage
Commercial-Case Conditions: T_J = 70°C, Worst-Case V_{CC} = 1.425 V, Worst-Case V_{CCI} = 2.3 V

Drive Strength	Speed Grade	t_{DOUT}	t_{DP}	t_{DIN}	t_{PY}	t_{PYS}	t_{EOUT}	t_{ZL}	t_{ZH}	t_{LZ}	t_{HZ}	Units
4 mA	STD	0.97	3.80	0.18	1.06	1.22	0.66	3.87	3.47	1.80	1.70	ns
6 mA	STD	0.97	3.21	0.18	1.06	1.22	0.66	3.27	3.11	2.05	2.17	ns
8 mA	STD	0.97	3.21	0.18	1.06	1.22	0.66	3.27	3.11	2.05	2.17	ns
12 mA	STD	0.97	2.80	0.18	1.06	1.22	0.66	2.85	2.79	2.22	2.48	ns

Note: For specific junction temperature and voltage supply levels, refer to Table 2-6 on page 2-6 for derating values.

Table 2-42 • 2.5 V LVCMOS High Slew – Applies to 1.5 V DC Core Voltage
Commercial-Case Conditions: T_J = 70°C, Worst-Case V_{CC} = 1.425 V, Worst-Case V_{CCI} = 2.3 V

Drive Strength	Speed Grade	t_{DOUT}	t_{DP}	t_{DIN}	t_{PY}	t_{PYS}	t_{EOUT}	t_{ZL}	t_{ZH}	t_{LZ}	t_{HZ}	Units
4 mA	STD	0.97	1.90	0.18	1.06	1.22	0.66	1.93	1.57	1.79	1.77	ns
6 mA	STD	0.97	1.73	0.18	1.06	1.22	0.66	1.76	1.42	2.04	2.25	ns
8 mA	STD	0.97	1.73	0.18	1.06	1.22	0.66	1.76	1.42	2.04	2.25	ns
12 mA	STD	0.97	1.62	0.18	1.06	1.22	0.66	1.65	1.34	2.22	2.56	ns

Note: For specific junction temperature and voltage supply levels, refer to Table 2-6 on page 2-6 for derating values.

Applies to 1.2 V DC Core Voltage

Table 2-43 • 2.5 LVCMOS Low Slew – Applies to 1.2 V DC Core Voltage
Commercial-Case Conditions: T_J = 70°C, Worst-Case V_{CC} = 1.14 V, Worst-Case V_{CCI} = 2.3 V

Drive Strength	Speed Grade	t_{DOUT}	t_{DP}	t_{DIN}	t_{PY}	t_{PYS}	t_{EOUT}	t_{ZL}	t_{ZH}	t_{LZ}	t_{HZ}	Units
4 mA	STD	0.98	4.37	0.19	1.20	1.40	0.67	4.45	3.95	2.21	2.35	ns
6 mA	STD	0.98	3.76	0.19	1.20	1.40	0.67	3.83	3.59	2.47	2.83	ns
8 mA	STD	0.98	3.76	0.19	1.20	1.40	0.67	3.83	3.59	2.47	2.83	ns
12 mA	STD	0.98	3.34	0.19	1.20	1.40	0.67	3.41	3.26	2.65	3.15	ns

Note: For specific junction temperature and voltage supply levels, refer to Table 2-6 on page 2-6 for derating values.

Table 2-44 • 2.5 V LVCMOS High Slew – Applies to 1.2 V DC Core Voltage
Commercial-Case Conditions: T_J = 70°C, Worst-Case V_{CC} = 1.14 V, Worst-Case V_{CCI} = 2.3 V

Drive Strength	Speed Grade	t_{DOUT}	t_{DP}	t_{DIN}	t_{PY}	t_{PYS}	t_{EOUT}	t_{ZL}	t_{ZH}	t_{LZ}	t_{HZ}	Units
4 mA	STD	0.98	2.42	0.19	1.20	1.40	0.67	2.46	2.01	2.21	2.44	ns
6 mA	STD	0.98	2.24	0.19	1.20	1.40	0.67	2.28	1.85	2.46	2.93	ns
8 mA	STD	0.98	2.24	0.19	1.20	1.40	0.67	2.28	1.85	2.46	2.93	ns
12 mA	STD	0.98	2.13	0.19	1.20	1.40	0.67	2.17	1.77	2.65	3.25	ns

Note: For specific junction temperature and voltage supply levels, refer to Table 2-6 on page 2-6 for derating values.

1.8 V LVCMOS

Low-voltage CMOS for 1.8 V is an extension of the LVCMOS standard (JESD8-5) used for general-purpose 1.8 V applications. It uses a 1.8 V input buffer and a push-pull output buffer.

Table 2-45 • Minimum and Maximum DC Input and Output Levels

1.8 V LVCMOS	V_{IL}		V_{IH}		V_{OL}	V_{OH}	I_{OL}	I_{OH}	I_{OSL}	I_{OSH}	I_{IL}	I_{IH}
Drive Strength	Min., V	Max., V	Min., V	Max., V	Max., V	Min., V	mA	mA	Max., mA[1]	Max., mA[1]	µA[2]	µA[2]
2 mA	–0.3	0.35 * V_{CCI}	0.65 * V_{CCI}	1.9	0.45	V_{CCI} – 0.45	2	2	9	11	10	10
4 mA	–0.3	0.35 * V_{CCI}	0.65 * V_{CCI}	1.9	0.45	V_{CCI} – 0.45	4	4	17	22	10	10
6 mA	–0.3	0.35 * V_{CCI}	0.65 * V_{CCI}	1.9	0.45	V_{CCI} – 0.45	6	6	35	44	10	10
8 mA	–0.3	0.35 * V_{CCI}	0.65 * V_{CCI}	1.9	0.45	V_{CCI} – 0.45	8	8	35	44	10	10

Notes:

1. *Currents are measured at high temperature (100°C junction temperature) and maximum voltage.*
2. *Currents are measured at 85°C junction temperature.*
3. *Software default selection highlighted in gray.*

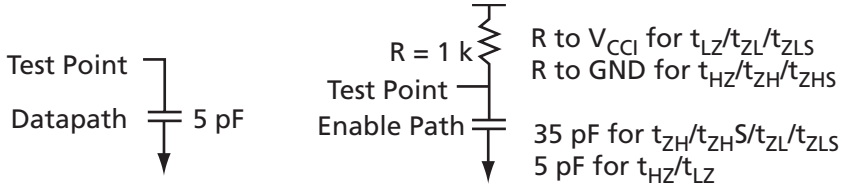


Figure 2-9 • AC Loading

Table 2-46 • AC Waveforms, Measuring Points, and Capacitive Loads

Input LOW (V)	Input HIGH (V)	Measuring Point* (V)	C_{LOAD} (pF)
0	1.8	0.9	5

* *Measuring point = V_{trip}. See Table 2-22 on page 2-20 for a complete table of trip points.*

Timing Characteristics

Applies to 1.5 V DC Core Voltage

Table 2-47 • 1.8 V LVCMOS Low Slew – Applies to 1.5 V DC Core Voltage
Commercial-Case Conditions: T_J = 70°C, Worst-Case V_{CC} = 1.425 V, Worst-Case V_{CCI} = 1.7 V

Drive Strength	Speed Grade	t_{DOUT}	t_{DP}	t_{DIN}	t_{PY}	t_{PYS}	t_{EOUT}	t_{ZL}	t_{ZH}	t_{LZ}	t_{HZ}	Units
2 mA	STD	0.97	5.11	0.18	0.99	1.43	0.66	5.20	4.48	1.78	1.30	ns
4 mA	STD	0.97	4.31	0.18	0.99	1.43	0.66	4.39	4.04	2.08	2.07	ns
6 mA	STD	0.97	3.78	0.18	0.99	1.43	0.66	3.85	3.63	2.29	2.46	ns
8 mA	STD	0.97	3.78	0.18	0.99	1.43	0.66	3.85	3.63	2.29	2.46	ns

Note: For specific junction temperature and voltage supply levels, refer to Table 2-6 on page 2-6 for derating values.

Table 2-48 • 1.8 V LVCMOS High Slew – Applies to 1.5 V DC Core Voltage
Commercial-Case Conditions: T_J = 70°C, Worst-Case V_{CC} = 1.425 V, Worst-Case V_{CCI} = 1.7 V

Drive Strength	Speed Grade	t_{DOUT}	t_{DP}	t_{DIN}	t_{PY}	t_{PYS}	t_{EOUT}	t_{ZL}	t_{ZH}	t_{LZ}	t_{HZ}	Units
2 mA	STD	0.97	2.21	0.18	0.99	1.43	0.66	2.25	1.86	1.78	1.35	ns
4 mA	STD	0.97	1.97	0.18	0.99	1.43	0.66	2.01	1.64	2.08	2.15	ns
6 mA	STD	0.97	1.82	0.18	0.99	1.43	0.66	1.85	1.53	2.29	2.54	ns
8 mA	STD	0.97	1.82	0.18	0.99	1.43	0.66	1.85	1.53	2.29	2.54	ns

Note: For specific junction temperature and voltage supply levels, refer to Table 2-6 on page 2-6 for derating values.

Applies to 1.2 V DC Core Voltage

Table 2-49 • 1.8 V LVCMOS Low Slew – Applies to 1.2 V DC Core Voltage
Commercial-Case Conditions: T_J = 70°C, Worst-Case V_{CC} = 1.14 V, Worst-Case V_{CCI} = 1.7 V

Drive Strength	Speed Grade	t_{DOUT}	t_{DP}	t_{DIN}	t_{PY}	t_{PYS}	t_{EOUT}	t_{ZL}	t_{ZH}	t_{LZ}	t_{HZ}	Units
2 mA	STD	0.98	5.61	0.19	1.12	1.60	0.67	5.71	4.96	2.18	1.87	ns
4 mA	STD	0.98	4.80	0.19	1.12	1.60	0.67	4.89	4.50	2.49	2.67	ns
6 mA	STD	0.98	4.25	0.19	1.12	1.60	0.67	4.33	4.09	2.71	3.06	ns
8 mA	STD	0.98	4.25	0.19	1.12	1.60	0.67	4.33	4.09	2.71	3.06	ns

Note: For specific junction temperature and voltage supply levels, refer to Table 2-6 on page 2-6 for derating values.

Table 2-50 • 1.8 V LVCMOS High Slew – Applies to 1.2 V DC Core Voltage
Commercial-Case Conditions: T_J = 70°C, Worst-Case V_{CC} = 1.14 V, Worst-Case V_{CCI} = 1.7 V

Drive Strength	Speed Grade	t_{DOUT}	t_{DP}	t_{DIN}	t_{PY}	t_{PYS}	t_{EOUT}	t_{ZL}	t_{ZH}	t_{LZ}	t_{HZ}	Units
2 mA	STD	0.98	2.66	0.19	1.12	1.60	0.67	2.71	2.27	2.18	1.92	ns
4 mA	STD	0.98	2.41	0.19	1.12	1.60	0.67	2.46	2.04	2.49	2.75	ns
6 mA	STD	0.98	2.25	0.19	1.12	1.60	0.67	2.30	1.92	2.70	3.15	ns
8 mA	STD	0.98	2.25	0.19	1.12	1.60	0.67	2.30	1.92	2.70	3.15	ns

Note: For specific junction temperature and voltage supply levels, refer to Table 2-6 on page 2-6 for derating values.

1.5 V LVCMOS (JESD8-11)

Low-Voltage CMOS for 1.5 V is an extension of the LVCMOS standard (JESD8-5) used for general-purpose 1.5 V applications. It uses a 1.5 V input buffer and a push-pull output buffer.

Table 2-51 • Minimum and Maximum DC Input and Output Levels

1.5 V LVCMOS	V_{IL}		V_{IH}		V_{OL}	V_{OH}	I_{OL}	I_{OH}	I_{OSL}	I_{OSH}	I_{IL}	I_{IH}
Drive Strength	Min., V	Max., V	Min., V	Max., V	Max., V	Min., V	mA	mA	Max., mA[1]	Max., mA[1]	µA[2]	µA[2]
2 mA	–0.3	0.35 * V_{CCI}	0.65 * V_{CCI}	1.575	0.25 * V_{CCI}	0.75 * V_{CCI}	2	2	13	16	10	10
4 mA	–0.3	0.35 * V_{CCI}	0.65 * V_{CCI}	1.575	0.25 * V_{CCI}	0.75 * V_{CCI}	4	4	25	33	10	10

Notes:

1. *Currents are measured at high temperature (100°C junction temperature) and maximum voltage.*
2. *Currents are measured at 85°C junction temperature.*
3. *Software default selection highlighted in gray.*

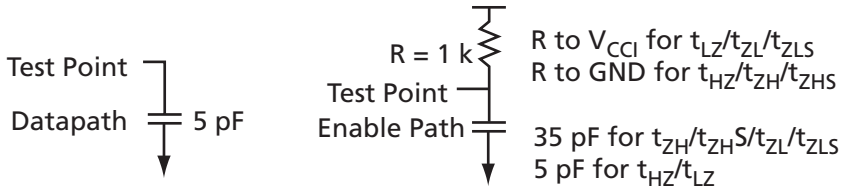


Figure 2-10 • AC Loading

Table 2-52 • AC Waveforms, Measuring Points, and Capacitive Loads

Input LOW (V)	Input HIGH (V)	Measuring Point* (V)	C_{LOAD} (pF)
0	1.5	0.75	5

* *Measuring point = V_{trip}. See Table 2-22 on page 2-20 for a complete table of trip points.*

Timing Characteristics

Applies to 1.5 V DC Core Voltage

Table 2-53 • 1.5 V LVCMOS Low Slew – Applies to 1.5 V DC Core Voltage
Commercial-Case Conditions: T_J = 70°C, Worst-Case V_{CC} = 1.425 V, Worst-Case V_{CCI} = 1.4 V

Drive Strength	Speed Grade	t_{DOUT}	t_{DP}	t_{DIN}	t_{PY}	t_{PYS}	t_{EOUT}	t_{ZL}	t_{ZH}	t_{LZ}	t_{HZ}	Units
2 mA	STD	0.97	5.47	0.18	1.15	1.62	0.66	5.57	4.89	2.13	2.02	ns
4 mA	STD	0.97	4.82	0.18	1.15	1.62	0.66	4.91	4.42	2.37	2.47	ns

Note: For specific junction temperature and voltage supply levels, refer to Table 2-6 on page 2-6 for derating values.

Table 2-54 • 1.5 V LVCMOS High Slew – Applies to 1.5 V DC Core Voltage
Commercial-Case Conditions: T_J = 70°C, Worst-Case V_{CC} = 1.425 V, Worst-Case V_{CCI} = 1.4 V

Drive Strength	Speed Grade	t_{DOUT}	t_{DP}	t_{DIN}	t_{PY}	t_{PYS}	t_{EOUT}	t_{ZL}	t_{ZH}	t_{LZ}	t_{HZ}	Units
2 mA	STD	0.97	2.27	0.18	1.15	1.62	0.66	2.31	1.85	2.13	2.11	ns
4 mA	STD	0.97	2.07	0.18	1.15	1.62	0.66	2.10	1.71	2.37	2.57	ns

Note: For specific junction temperature and voltage supply levels, refer to Table 2-6 on page 2-6 for derating values.

Applies to 1.2 V DC Core Voltage

Table 2-55 • 1.5 V LVCMOS Low Slew – Applies to 1.2 V DC Core Voltage
Commercial-Case Conditions: T_J = 70°C, Worst-Case V_{CC} = 1.14 V, Worst-Case V_{CCI} = 1.4 V

Drive Strength	Speed Grade	t_{DOUT}	t_{DP}	t_{DIN}	t_{PY}	t_{PYS}	t_{EOUT}	t_{ZL}	t_{ZH}	t_{LZ}	t_{HZ}	Units
2 mA	STD	0.98	5.94	0.19	1.26	1.79	0.67	6.05	5.36	2.53	2.58	ns
4 mA	STD	0.98	5.28	0.19	1.26	1.79	0.67	5.38	4.88	2.78	3.04	ns

Note: For specific junction temperature and voltage supply levels, refer to Table 2-6 on page 2-6 for derating values.

Table 2-56 • 1.5 V LVCMOS High Slew – Applies to 1.2 V DC Core Voltage
Commercial-Case Conditions: T_J = 70°C, Worst-Case V_{CC} = 1.14 V, Worst-Case V_{CCI} = 1.4 V

Drive Strength	Speed Grade	t_{DOUT}	t_{DP}	t_{DIN}	t_{PY}	t_{PYS}	t_{EOUT}	t_{ZL}	t_{ZH}	t_{LZ}	t_{HZ}	Units
2 mA	STD	0.98	2.68	0.19	1.26	1.79	0.67	2.73	2.24	2.53	2.67	ns
4 mA	STD	0.98	2.48	0.19	1.26	1.79	0.67	2.52	2.10	2.77	3.14	ns

Note: For specific junction temperature and voltage supply levels, refer to Table 2-6 on page 2-6 for derating values.

1.2 V LVCMOS (JESD8-12A)

Low-Voltage CMOS for 1.2 V complies with the LVCMOS standard JESD8-12A for general purpose 1.2 V applications. It uses a 1.2 V input buffer and a push-pull output buffer.

Table 2-57 • Minimum and Maximum DC Input and Output Levels

1.2 V LVCMOS	V_{IL}		V_{IH}		V_{OL}	V_{OH}	I_{OL}	I_{OH}	I_{OSL}	I_{OSH}	I_{IL}	I_{IH}
Drive Strength	Min., V	Max., V	Min., V	Max., V	Max., V	Min., V	mA	mA	Max., mA[1]	Max., mA[1]	µA[2]	µA[2]
2 mA	–0.3	0.35 * V_{CCI}	0.65 * V_{CCI}	1.26	0.25 * V_{CCI}	0.75 * V_{CCI}	2	2	TBD	TBD	10	10

Notes:

1. *Currents are measured at high temperature (100°C junction temperature) and maximum voltage.*
2. *Currents are measured at 85°C junction temperature.*
3. *Software default selection highlighted in gray.*

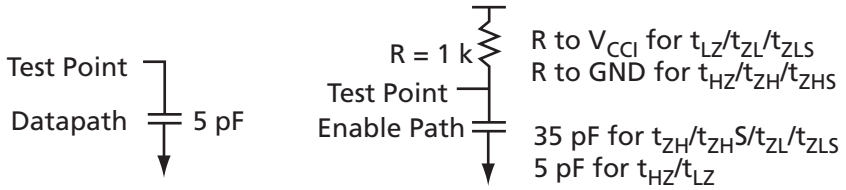


Figure 2-11 • AC Loading

Table 2-58 • AC Waveforms, Measuring Points, and Capacitive Loads

Input LOW (V)	Input HIGH (V)	Measuring Point* (V)	C_{LOAD} (pF)
0	1.2	0.6	5

* *Measuring point = V_{trip}. See Table 2-22 on page 2-20 for a complete table of trip points.*

Timing Characteristics

Applies to 1.2 V DC Core Voltage

Table 2-59 • **1.2 V LVCMOS Low Slew**
Commercial-Case Conditions: T_J = 70°C, Worst-Case V_{CC} = 1.14 V, Worst-Case V_{CCI} = 1.14 V

Drive Strength	Speed Grade	t_{DOUT}	t_{DP}	t_{DIN}	t_{PY}	t_{PYS}	t_{EOUT}	t_{ZL}	t_{ZH}	t_{LZ}	t_{HZ}	Units
2 mA	STD	0.98	8.28	0.19	1.56	2.34	0.67	3.24	2.76	3.00	3.25	ns

Note: For specific junction temperature and voltage supply levels, refer to Table 2-6 on page 2-6 for derating values.

Table 2-60 • **1.2 V LVCMOS High Slew**
Commercial-Case Conditions: T_J = 70°C, Worst-Case V_{CC} = 1.14 V, Worst-Case V_{CCI} = 1.14 V

Drive Strength	Speed Grade	t_{DOUT}	t_{DP}	t_{DIN}	t_{PY}	t_{PYS}	t_{EOUT}	t_{ZL}	t_{ZH}	t_{LZ}	t_{HZ}	Units
2 mA	STD	0.98	2.68	0.19	1.56	2.34	0.67	2.73	2.24	2.53	2.67	ns

Note: For specific junction temperature and voltage supply levels, refer to Table 2-6 on page 2-6 for derating values.

I/O Register Specifications

Fully Registered I/O Buffers with Asynchronous Preset

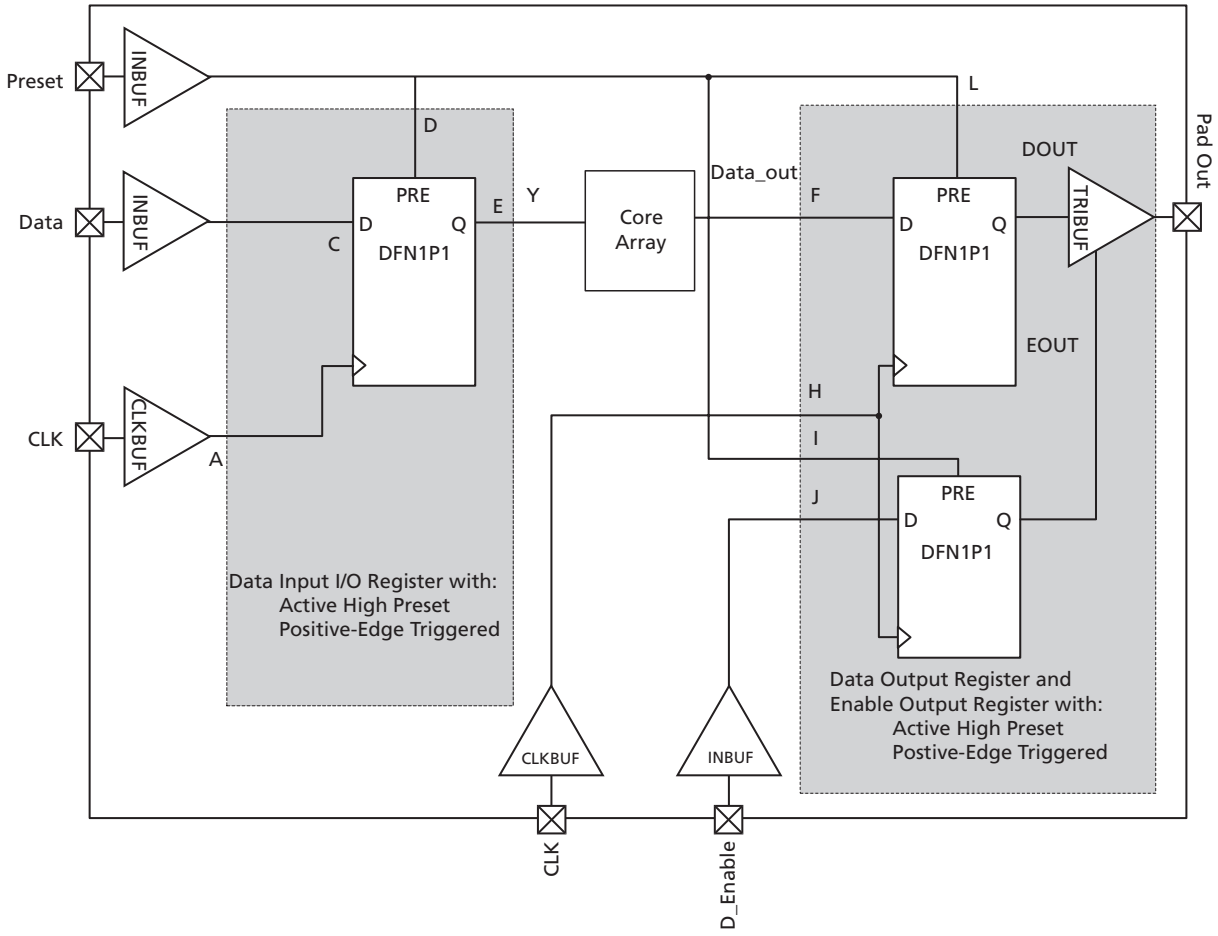


***Figure 2-12* • Timing Model of Registered I/O Buffers with Asynchronous Preset**

Table 2-61 • **Parameter Definition and Measuring Nodes**

Parameter Name	Parameter Definition	Measuring Nodes (from, to)*
t_{OCLKQ}	Clock-to-Q of the Output Data Register	H, DOUT
t_{OSUD}	Data Setup Time for the Output Data Register	F, H
t_{OHD}	Data Hold Time for the Output Data Register	F, H
t_{OPRE2Q}	Asynchronous Preset-to-Q of the Output Data Register	L, DOUT
$t_{OREMPRE}$	Asynchronous Preset Removal Time for the Output Data Register	L, H
$t_{ORECPRE}$	Asynchronous Preset Recovery Time for the Output Data Register	L, H
t_{OECLKQ}	Clock-to-Q of the Output Enable Register	H, EOUT
t_{OESUD}	Data Setup Time for the Output Enable Register	J, H
t_{OEHD}	Data Hold Time for the Output Enable Register	J, H
$t_{OEPRE2Q}$	Asynchronous Preset-to-Q of the Output Enable Register	I, EOUT
$t_{OEREMPRE}$	Asynchronous Preset Removal Time for the Output Enable Register	I, H
$t_{OERECPRE}$	Asynchronous Preset Recovery Time for the Output Enable Register	I, H
t_{ICLKQ}	Clock-to-Q of the Input Data Register	A, E
t_{ISUD}	Data Setup Time for the Input Data Register	C, A
t_{IHD}	Data Hold Time for the Input Data Register	C, A
t_{IPRE2Q}	Asynchronous Preset-to-Q of the Input Data Register	D, E
$t_{IREMPRE}$	Asynchronous Preset Removal Time for the Input Data Register	D, A
$t_{IRECPRE}$	Asynchronous Preset Recovery Time for the Input Data Register	D, A

** See Figure 2-12 on page 2-39 for more information.*

Fully Registered I/O Buffers with Asynchronous Clear

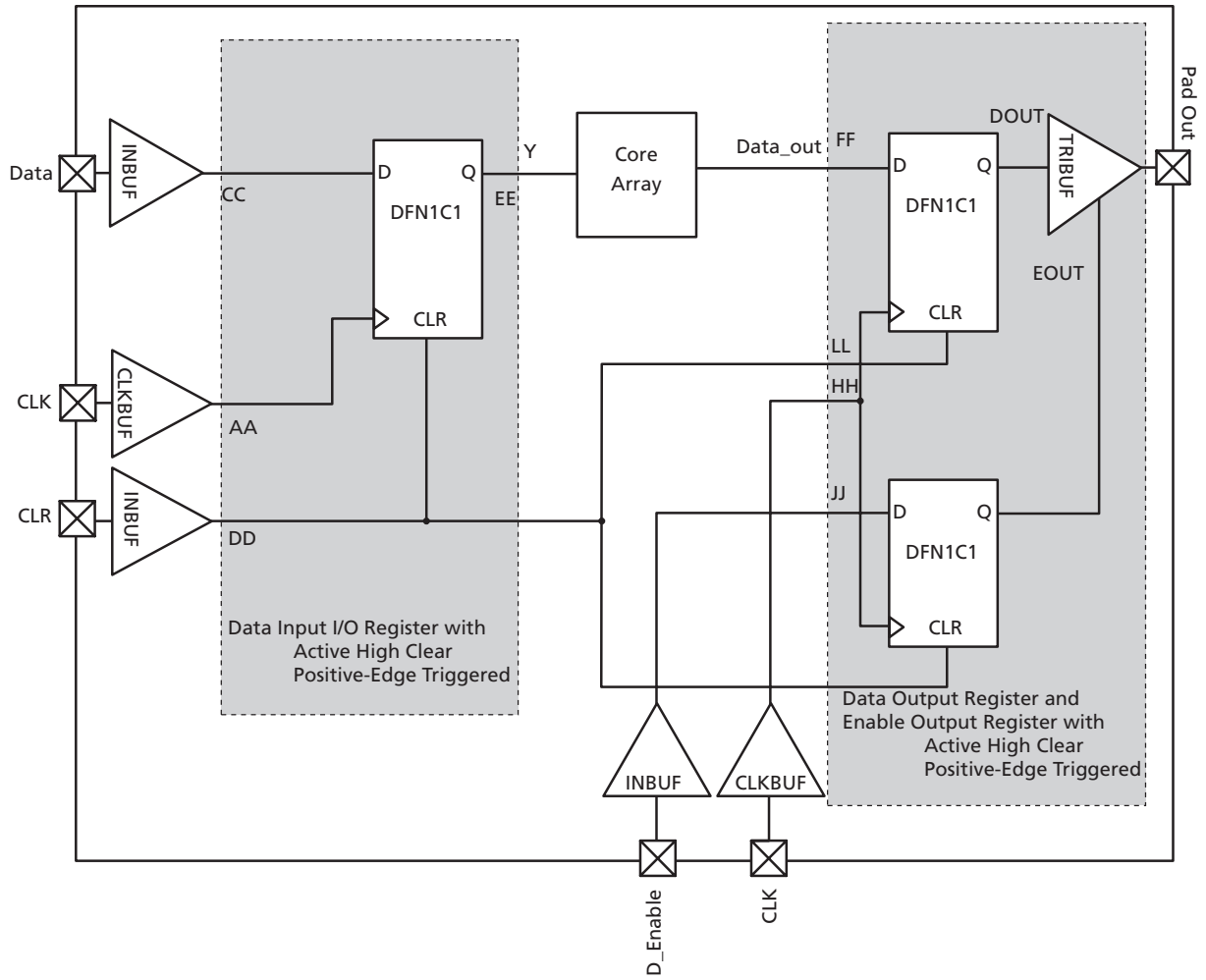


***Figure 2-13* • Timing Model of the Registered I/O Buffers with Asynchronous Clear**

Table 2-62 • **Parameter Definition and Measuring Nodes**

Parameter Name	Parameter Definition	Measuring Nodes (from, to)*
t_{OCLKQ}	Clock-to-Q of the Output Data Register	HH, DOUT
t_{OSUD}	Data Setup Time for the Output Data Register	FF, HH
t_{OHD}	Data Hold Time for the Output Data Register	FF, HH
t_{OCLR2Q}	Asynchronous Clear-to-Q of the Output Data Register	LL, DOUT
$t_{OREMCLR}$	Asynchronous Clear Removal Time for the Output Data Register	LL, HH
$t_{ORECCLR}$	Asynchronous Clear Recovery Time for the Output Data Register	LL, HH
t_{OECLKQ}	Clock-to-Q of the Output Enable Register	HH, EOUT
t_{OESUD}	Data Setup Time for the Output Enable Register	JJ, HH
t_{OEHD}	Data Hold Time for the Output Enable Register	JJ, HH
$t_{OECLR2Q}$	Asynchronous Clear-to-Q of the Output Enable Register	II, EOUT
$t_{OEREMCLR}$	Asynchronous Clear Removal Time for the Output Enable Register	II, HH
$t_{ORECCLR}$	Asynchronous Clear Recovery Time for the Output Enable Register	II, HH
t_{ICLKQ}	Clock-to-Q of the Input Data Register	AA, EE
t_{ISUD}	Data Setup Time for the Input Data Register	CC, AA
t_{IHD}	Data Hold Time for the Input Data Register	CC, AA
t_{ICLR2Q}	Asynchronous Clear-to-Q of the Input Data Register	DD, EE
$t_{IREMCLR}$	Asynchronous Clear Removal Time for the Input Data Register	DD, AA
$t_{IRECCLR}$	Asynchronous Clear Recovery Time for the Input Data Register	DD, AA

** See Figure 2-13 on page 2-41 for more information.*

Input Register

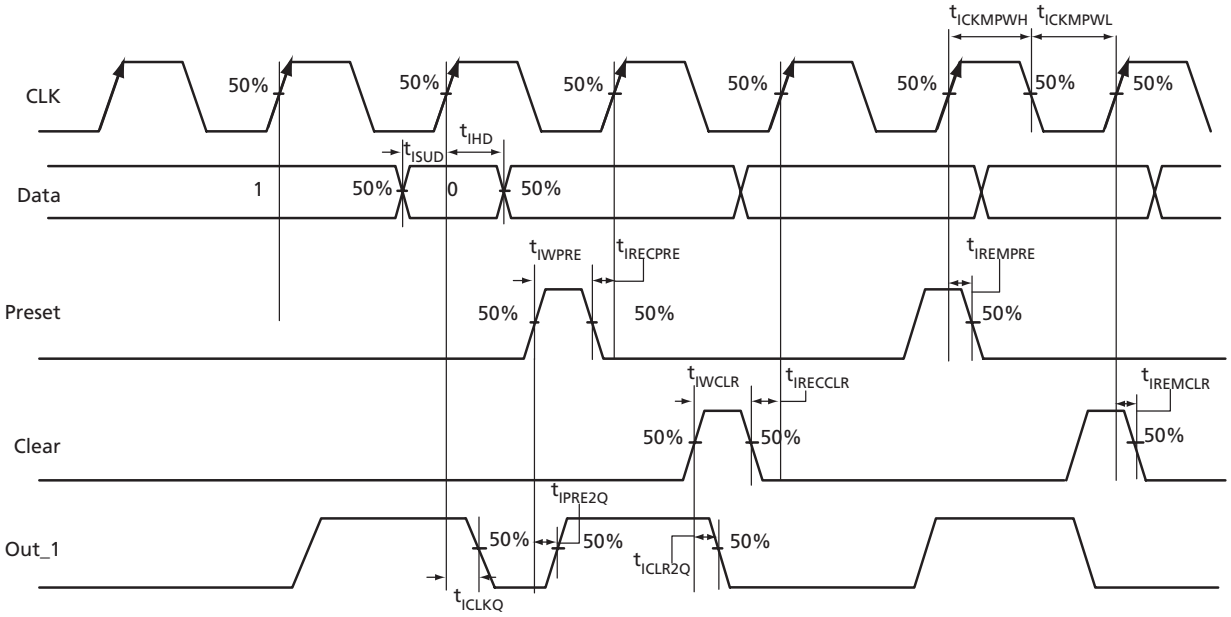


Figure 2-14 • **Input Register Timing Diagram**

Timing Characteristics

1.5 V DC Core Voltage

Table 2-63 • **Input Data Register Propagation Delays**
Commercial-Case Conditions: T_J = 70°C, Worst-Case V_{CC} = 1.425 V

Parameter	Description	Std.	Units
t_{ICLKQ}	Clock-to-Q of the Input Data Register	0.63	ns
t_{ISUD}	Data Setup Time for the Input Data Register	0.18	ns
t_{IHD}	Data Hold Time for the Input Data Register	0.00	ns
t_{ICLR2Q}	Asynchronous Clear-to-Q of the Input Data Register	0.46	ns
t_{IPRE2Q}	Asynchronous Preset-to-Q of the Input Data Register	0.46	ns
$t_{IREMCLR}$	Asynchronous Clear Removal Time for the Input Data Register	0.00	ns
$t_{IRECCLR}$	Asynchronous Clear Recovery Time for the Input Data Register	0.23	ns
$t_{IREMPRE}$	Asynchronous Preset Removal Time for the Input Data Register	0.00	ns
$t_{IRECPRE}$	Asynchronous Preset Recovery Time for the Input Data Register	0.23	ns
t_{IWCLR}	Asynchronous Clear Minimum Pulse Width for the Input Data Register	0.19	ns
t_{IWPRE}	Asynchronous Preset Minimum Pulse Width for the Input Data Register	0.19	ns
$t_{ICKMPWH}$	Clock Minimum Pulse Width HIGH for the Input Data Register	0.28	ns
$t_{ICKMPWL}$	Clock Minimum Pulse Width LOW for the Input Data Register	0.31	ns

Note: For specific junction temperature and voltage supply levels, refer to Table 2-6 on page 2-6 for derating values.

1.2 V DC Core Voltage

Table 2-64 • Input Data Register Propagation Delays
Commercial-Case Conditions: T_J = 70°C, Worst-Case V_{CC} = 1.14 V

Parameter	Description	Std.	Units
t_{ICLKQ}	Clock-to-Q of the Input Data Register	0.99	ns
t_{ISUD}	Data Setup Time for the Input Data Register	0.29	ns
t_{IHD}	Data Hold Time for the Input Data Register	0.00	ns
t_{ICLR2Q}	Asynchronous Clear-to-Q of the Input Data Register	0.68	ns
t_{IPRE2Q}	Asynchronous Preset-to-Q of the Input Data Register	0.68	ns
$t_{IREMCLR}$	Asynchronous Clear Removal Time for the Input Data Register	0.00	ns
$t_{IRECCLR}$	Asynchronous Clear Recovery Time for the Input Data Register	0.24	ns
$t_{IREMPRE}$	Asynchronous Preset Removal Time for the Input Data Register	0.00	ns
$t_{IRECPRE}$	Asynchronous Preset Recovery Time for the Input Data Register	0.24	ns
t_{IWCLR}	Asynchronous Clear Minimum Pulse Width for the Input Data Register	0.19	ns
t_{IWPRE}	Asynchronous Preset Minimum Pulse Width for the Input Data Register	0.19	ns
$t_{ICKMPWH}$	Clock Minimum Pulse Width HIGH for the Input Data Register	0.28	ns
$t_{ICKMPWL}$	Clock Minimum Pulse Width LOW for the Input Data Register	0.31	ns

Note: For specific junction temperature and voltage supply levels, refer to Table 2-7 on page 2-7 for derating values.

Output Register

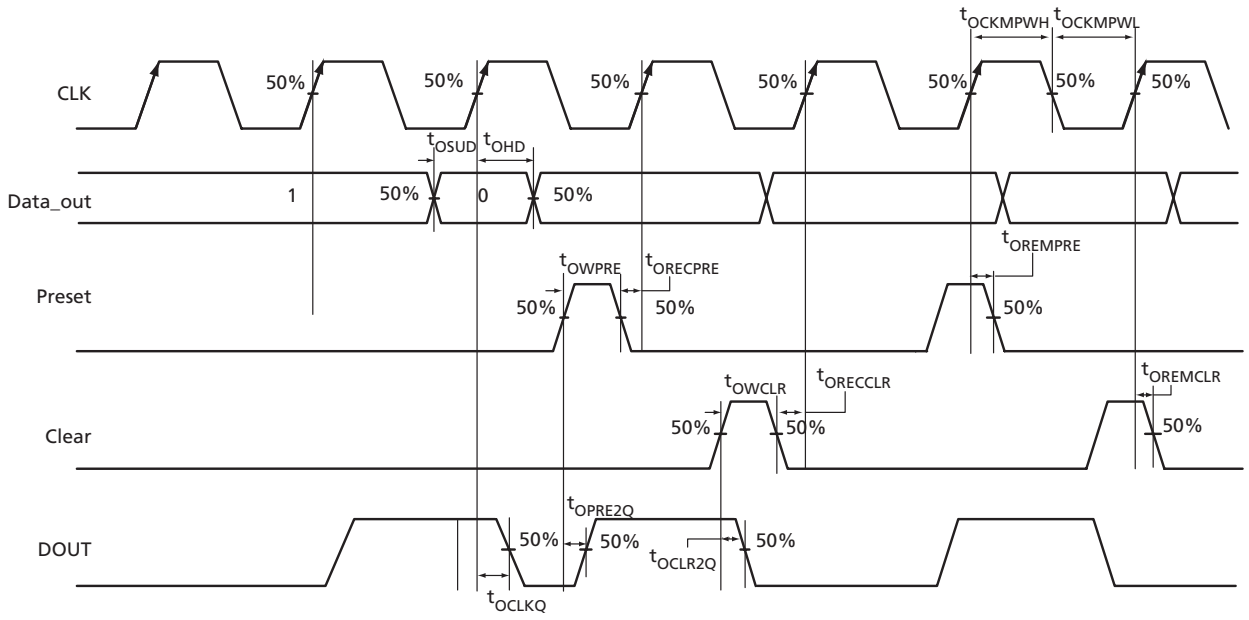


Figure 2-15 • **Output Register Timing Diagram**

Timing Characteristics

1.5 V DC Core Voltage

Table 2-65 • **Output Data Register Propagation Delays**
Commercial-Case Conditions: T_J = 70°C, Worst-Case V_{CC} = 1.425 V

Parameter	Description	Std.	Units
t_{OCLKQ}	Clock-to-Q of the Output Data Register	0.89	ns
t_{OSUD}	Data Setup Time for the Output Data Register	0.18	ns
t_{OHD}	Data Hold Time for the Output Data Register	0.00	ns
t_{OCLR2Q}	Asynchronous Clear-to-Q of the Output Data Register	0.72	ns
t_{OPRE2Q}	Asynchronous Preset-to-Q of the Output Data Register	0.78	ns
$t_{OREMCLR}$	Asynchronous Clear Removal Time for the Output Data Register	0.00	ns
$t_{ORECCLR}$	Asynchronous Clear Recovery Time for the Output Data Register	0.23	ns
$t_{OREMPRE}$	Asynchronous Preset Removal Time for the Output Data Register	0.00	ns
$t_{ORECPRE}$	Asynchronous Preset Recovery Time for the Output Data Register	0.23	ns
t_{OWCLR}	Asynchronous Clear Minimum Pulse Width for the Output Data Register	0.19	ns
t_{OWPRE}	Asynchronous Preset Minimum Pulse Width for the Output Data Register	0.19	ns
$t_{OCKMPWH}$	Clock Minimum Pulse Width HIGH for the Output Data Register	0.28	ns
$t_{OCKMPWL}$	Clock Minimum Pulse Width LOW for the Output Data Register	0.31	ns

Note: For specific junction temperature and voltage supply levels, refer to Table 2-6 on page 2-6 for derating values.

1.2 V DC Core Voltage

Table 2-66 • Output Data Register Propagation Delays
Commercial-Case Conditions: T_J = 70°C, Worst-Case V_{CC} = 1.14 V

Parameter	Description	Std.	Units
t_{OCLKQ}	Clock-to-Q of the Output Data Register	1.37	ns
t_{OSUD}	Data Setup Time for the Output Data Register	0.22	ns
t_{OHD}	Data Hold Time for the Output Data Register	0.00	ns
t_{OCLR2Q}	Asynchronous Clear-to-Q of the Output Data Register	1.05	ns
t_{OPRE2Q}	Asynchronous Preset-to-Q of the Output Data Register	1.14	ns
$t_{OREMCLR}$	Asynchronous Clear Removal Time for the Output Data Register	0.00	ns
$t_{ORECCLR}$	Asynchronous Clear Recovery Time for the Output Data Register	0.24	ns
$t_{OREMPRE}$	Asynchronous Preset Removal Time for the Output Data Register	0.00	ns
$t_{ORECPRE}$	Asynchronous Preset Recovery Time for the Output Data Register	0.24	ns
t_{OWCLR}	Asynchronous Clear Minimum Pulse Width for the Output Data Register	0.19	ns
t_{OWPRE}	Asynchronous Preset Minimum Pulse Width for the Output Data Register	0.19	ns
$t_{OCKMPWH}$	Clock Minimum Pulse Width HIGH for the Output Data Register	0.28	ns
$t_{OCKMPWL}$	Clock Minimum Pulse Width LOW for the Output Data Register	0.31	ns

Note: For specific junction temperature and voltage supply levels, refer to Table 2-7 on page 2-7 for derating values.

Output Enable Register

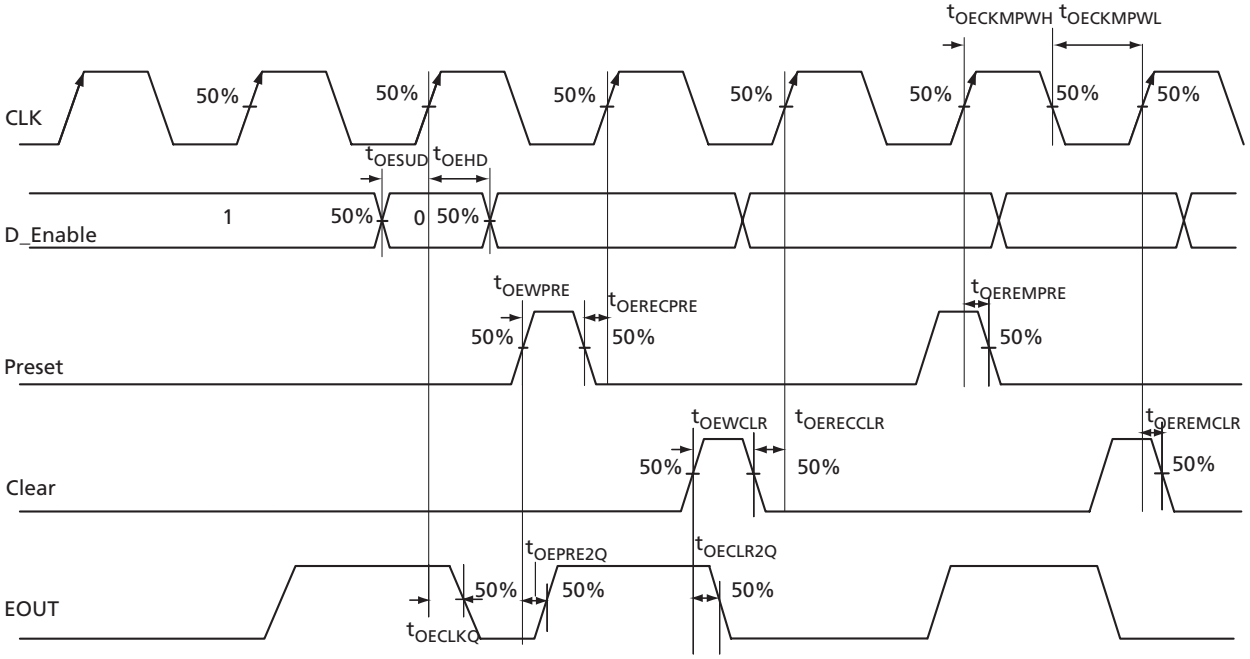


Figure 2-16 • **Output Enable Register Timing Diagram**

Timing Characteristics

1.5 V DC Core Voltage

Table 2-67 • **Output Enable Register Propagation Delays**
Commercial-Case Conditions: T_J = 70°C, Worst-Case V_{CC} = 1.425 V

Parameter	Description	Std.	Units
t_{OECLKQ}	Clock-to-Q of the Output Enable Register	0.91	ns
t_{OESUD}	Data Setup Time for the Output Enable Register	0.18	ns
t_{OEHD}	Data Hold Time for the Output Enable Register	0.00	ns
$t_{OECLR2Q}$	Asynchronous Clear-to-Q of the Output Enable Register	0.74	ns
$t_{OEPRE2Q}$	Asynchronous Preset-to-Q of the Output Enable Register	0.81	ns
$t_{OEREMCLR}$	Asynchronous Clear Removal Time for the Output Enable Register	0.00	ns
$t_{OERECCLR}$	Asynchronous Clear Recovery Time for the Output Enable Register	0.23	ns
$t_{OEREMPRE}$	Asynchronous Preset Removal Time for the Output Enable Register	0.00	ns
$t_{OERECPRE}$	Asynchronous Preset Recovery Time for the Output Enable Register	0.23	ns
t_{OEWCLR}	Asynchronous Clear Minimum Pulse Width for the Output Enable Register	0.19	ns
t_{OEWPRE}	Asynchronous Preset Minimum Pulse Width for the Output Enable Register	0.19	ns
$t_{OECKMPWH}$	Clock Minimum Pulse Width HIGH for the Output Enable Register	0.28	ns
$t_{OECKMPWL}$	Clock Minimum Pulse Width LOW for the Output Enable Register	0.31	ns

Note: For specific junction temperature and voltage supply levels, refer to Table 2-6 on page 2-6 for derating values.

1.2 V DC Core Voltage

Table 2-68 • Output Enable Register Propagation Delays
Commercial-Case Conditions: T_J = 70°C, Worst-Case V_{CC} = 1.14 V

Parameter	Description	Std.	Units
t_{OECLKQ}	Clock-to-Q of the Output Enable Register	1.40	ns
t_{OESUD}	Data Setup Time for the Output Enable Register	0.22	ns
t_{OEHD}	Data Hold Time for the Output Enable Register	0.00	ns
$t_{OECLR2Q}$	Asynchronous Clear-to-Q of the Output Enable Register	1.08	ns
$t_{OEPRE2Q}$	Asynchronous Preset-to-Q of the Output Enable Register	1.19	ns
$t_{OEREMCLR}$	Asynchronous Clear Removal Time for the Output Enable Register	0.00	ns
$t_{ORECCLR}$	Asynchronous Clear Recovery Time for the Output Enable Register	0.24	ns
$t_{OEREMPRE}$	Asynchronous Preset Removal Time for the Output Enable Register	0.00	ns
$t_{ORECPRE}$	Asynchronous Preset Recovery Time for the Output Enable Register	0.24	ns
t_{OEWCLR}	Asynchronous Clear Minimum Pulse Width for the Output Enable Register	0.19	ns
t_{OEWPRE}	Asynchronous Preset Minimum Pulse Width for the Output Enable Register	0.19	ns
$t_{OECKMPWH}$	Clock Minimum Pulse Width HIGH for the Output Enable Register	0.28	ns
$t_{OECKMPWL}$	Clock Minimum Pulse Width LOW for the Output Enable Register	0.31	ns

Note: For specific junction temperature and voltage supply levels, refer to Table 2-7 on page 2-7 for derating values.

VersaTile Characteristics

VersaTile Specifications as a Combinatorial Module

The IGLOO PLUS library offers all combinations of LUT-3 combinatorial functions. In this section, timing characteristics are presented for a sample of the library. For more details, refer to the *Fusion, IGLOOe, and ProASIC3/ E Macro Library Guide*.

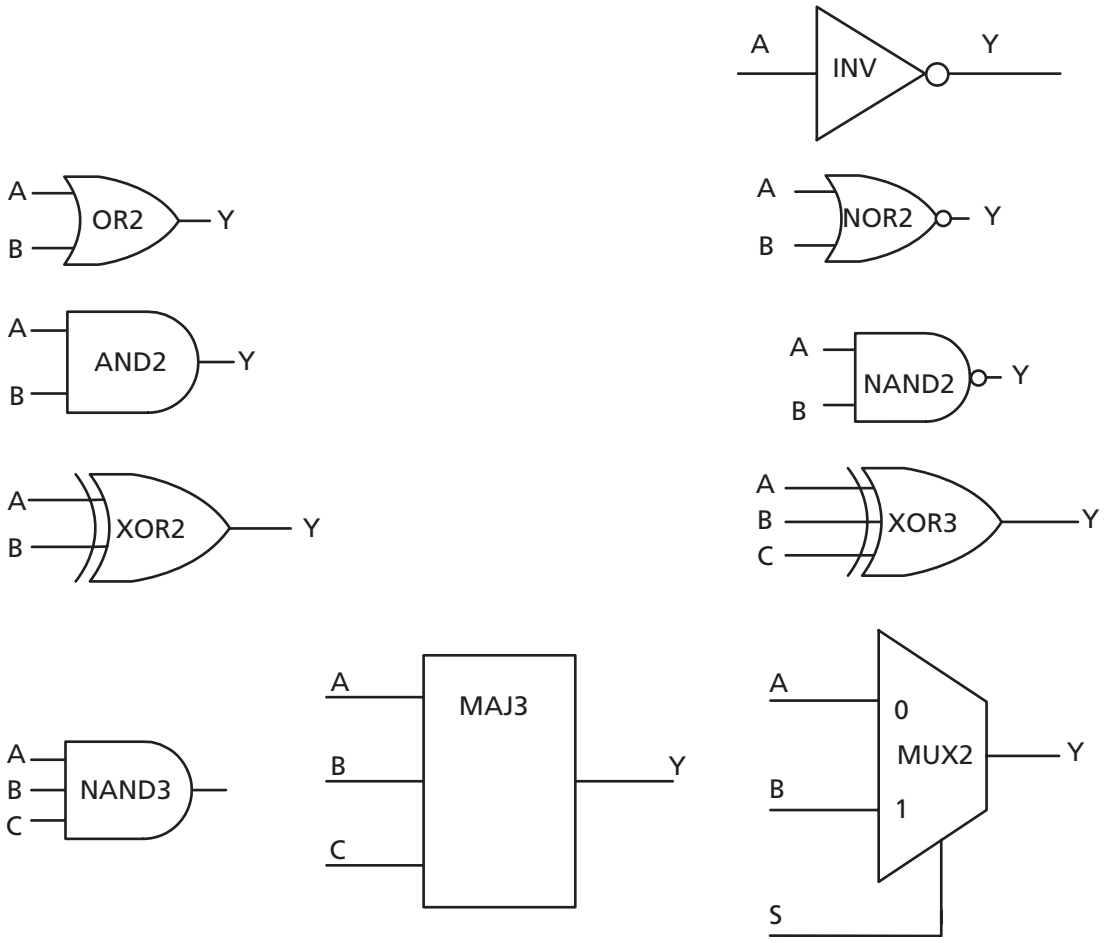


Figure 2-17 • **Sample of Combinatorial Cells**

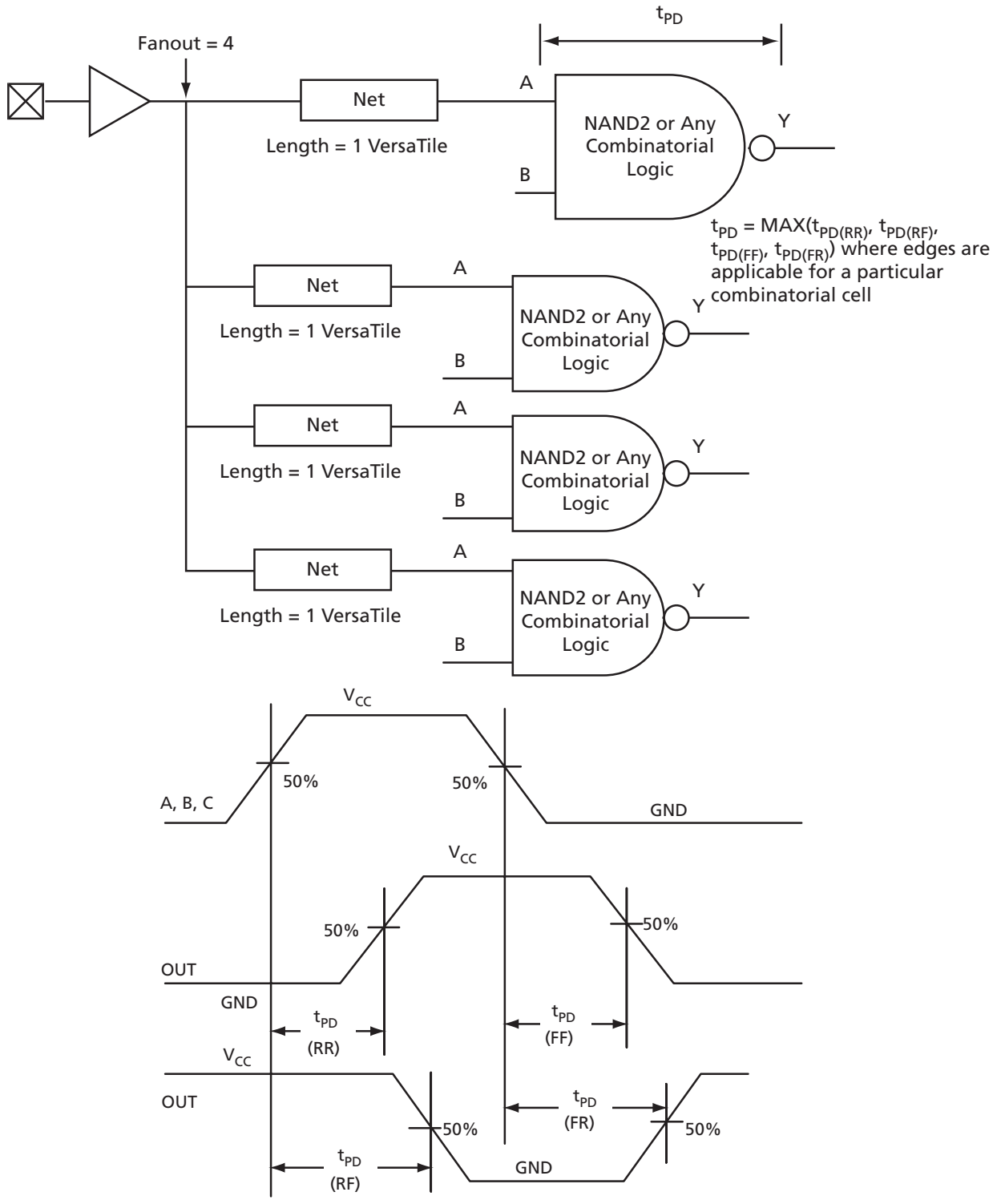


***Figure 2-18* • Timing Model and Waveforms**

Timing Characteristics

1.5 V DC Core Voltage

Table 2-69 • Combinatorial Cell Propagation Delays
Commercial-Case Conditions: T_J = 70°C, Worst-Case V_{CC} = 1.425 V

Combinatorial Cell	Equation	Parameter	Std.	Units
INV	Y = !A	t_{PD}	0.72	ns
AND2	Y = A · B	t_{PD}	0.86	ns
NAND2	Y = !(A · B)	t_{PD}	0.87	ns
OR2	Y = A + B	t_{PD}	0.89	ns
NOR2	Y = !(A + B)	t_{PD}	0.90	ns
XOR2	Y = A ⊕ B	t_{PD}	1.35	ns
MAJ3	Y = MAJ(A, B, C)	t_{PD}	1.33	ns
XOR3	Y = A ⊕ B ⊕ C	t_{PD}	1.98	ns
MUX2	Y = A !S + B S	t_{PD}	1.24	ns
AND3	Y = A · B · C	t_{PD}	1.40	ns

Note: For specific junction temperature and voltage supply levels, refer to Table 2-6 on page 2-6 for derating values.

1.2 V DC Core Voltage

Table 2-70 • Combinatorial Cell Propagation Delays
Commercial-Case Conditions: T_J = 70°C, Worst-Case V_{CC} = 1.14 V

Combinatorial Cell	Equation	Parameter	Std.	Units
INV	Y = !A	t_{PD}	1.27	ns
AND2	Y = A · B	t_{PD}	1.47	ns
NAND2	Y = !(A · B)	t_{PD}	1.52	ns
OR2	Y = A + B	t_{PD}	1.51	ns
NOR2	Y = !(A + B)	t_{PD}	1.57	ns
XOR2	Y = A ⊕ B	t_{PD}	2.28	ns
MAJ3	Y = MAJ(A, B, C)	t_{PD}	2.39	ns
XOR3	Y = A ⊕ B ⊕ C	t_{PD}	3.50	ns
MUX2	Y = A !S + B S	t_{PD}	2.21	ns
AND3	Y = A · B · C	t_{PD}	2.50	ns

Note: For specific junction temperature and voltage supply levels, refer to Table 2-7 on page 2-7 for derating values.

VersaTile Specifications as a Sequential Module

The IGLOO PLUS library offers a wide variety of sequential cells, including flip-flops and latches. Each has a data input and optional enable, clear, or preset. In this section, timing characteristics are presented for a representative sample from the library. For more details, refer to the *Fusion, IGLOOe, and ProASIC3/E Macro Library Guide*.

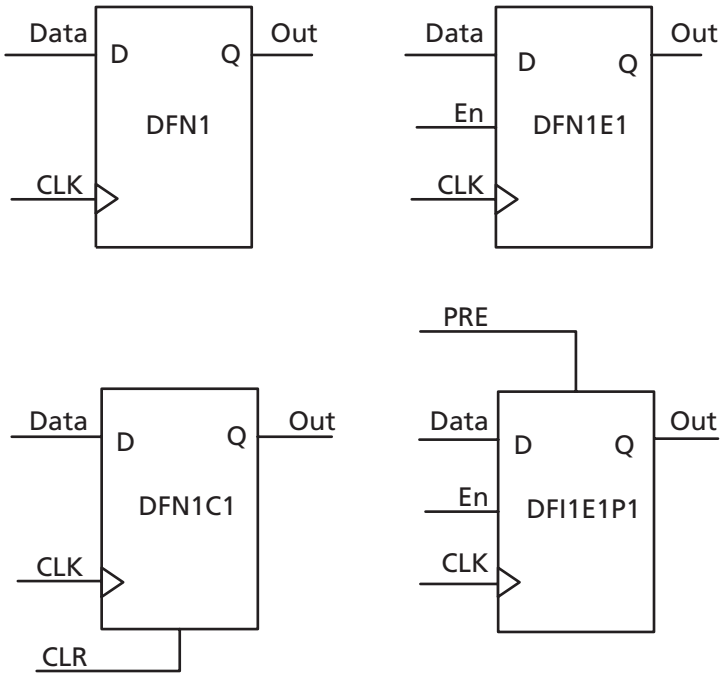


Figure 2-19 • **Sample of Sequential Cells**

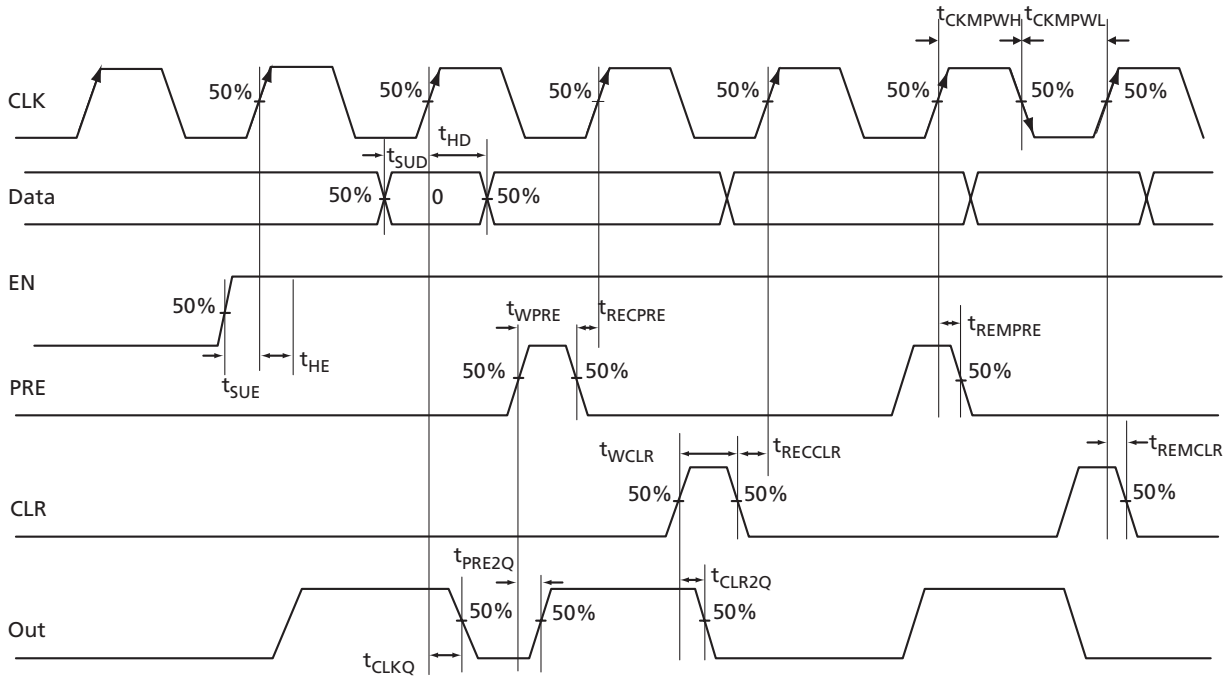


Figure 2-20 • **Timing Model and Waveforms**

Timing Characteristics

1.5 V DC Core Voltage

Table 2-71 • **Register Delays**
Commercial-Case Conditions: T_J = 70°C, Worst-Case V_{CC} = 1.425 V

Parameter	Description	Std.	Units
t_{CLKQ}	Clock-to-Q of the Core Register	0.80	ns
t_{SUD}	Data Setup Time for the Core Register	0.84	ns
t_{HD}	Data Hold Time for the Core Register	0.00	ns
t_{SUE}	Enable Setup Time for the Core Register	0.73	ns
t_{HE}	Enable Hold Time for the Core Register	0.00	ns
t_{CLR2Q}	Asynchronous Clear-to-Q of the Core Register	0.62	ns
t_{PRE2Q}	Asynchronous Preset-to-Q of the Core Register	0.60	ns
t_{REMCLR}	Asynchronous Clear Removal Time for the Core Register	0.00	ns
t_{RECCLR}	Asynchronous Clear Recovery Time for the Core Register	0.23	ns
t_{REMPRE}	Asynchronous Preset Removal Time for the Core Register	0.00	ns
t_{RECPRE}	Asynchronous Preset Recovery Time for the Core Register	0.24	ns
t_{WCLR}	Asynchronous Clear Minimum Pulse Width for the Core Register	0.30	ns
t_{WPRE}	Asynchronous Preset Minimum Pulse Width for the Core Register	0.30	ns
t_{CKMPWH}	Clock Minimum Pulse Width HIGH for the Core Register	0.56	ns
t_{CKMPWL}	Clock Minimum Pulse Width LOW for the Core Register	0.56	ns

Note: For specific junction temperature and voltage supply levels, refer to Table 2-6 on page 2-6 for derating values.

1.2 V DC Core Voltage

Table 2-72 • Register Delays
Commercial-Case Conditions: T_J = 70°C, Worst-Case V_{CC} = 1.14 V

Parameter	Description	Std.	Units
t_{CLKQ}	Clock-to-Q of the Core Register	1.40	ns
t_{SUD}	Data Setup Time for the Core Register	1.35	ns
t_{HD}	Data Hold Time for the Core Register	0.00	ns
t_{SUE}	Enable Setup Time for the Core Register	1.29	ns
t_{HE}	Enable Hold Time for the Core Register	0.00	ns
t_{CLR2Q}	Asynchronous Clear-to-Q of the Core Register	0.89	ns
t_{PRE2Q}	Asynchronous Preset-to-Q of the Core Register	0.87	ns
t_{REMCLR}	Asynchronous Clear Removal Time for the Core Register	0.00	ns
t_{RECCLR}	Asynchronous Clear Recovery Time for the Core Register	0.24	ns
t_{REMPRE}	Asynchronous Preset Removal Time for the Core Register	0.00	ns
t_{RECPRE}	Asynchronous Preset Recovery Time for the Core Register	0.24	ns
t_{WCLR}	Asynchronous Clear Minimum Pulse Width for the Core Register	0.46	ns
t_{WPRE}	Asynchronous Preset Minimum Pulse Width for the Core Register	0.46	ns
t_{CKMPWH}	Clock Minimum Pulse Width HIGH for the Core Register	0.95	ns
t_{CKMPWL}	Clock Minimum Pulse Width LOW for the Core Register	0.95	ns

Note: For specific junction temperature and voltage supply levels, refer to Table 2-7 on page 2-7 for derating values.

Global Resource Characteristics

AGLP125 Clock Tree Topology

Clock delays are device-specific. Figure 2-21 is an example of a global tree used for clock routing. The global tree presented in Figure 2-21 is driven by a CCC located on the west side of the AGLP125 device. It is used to drive all D-flip-flops in the device.

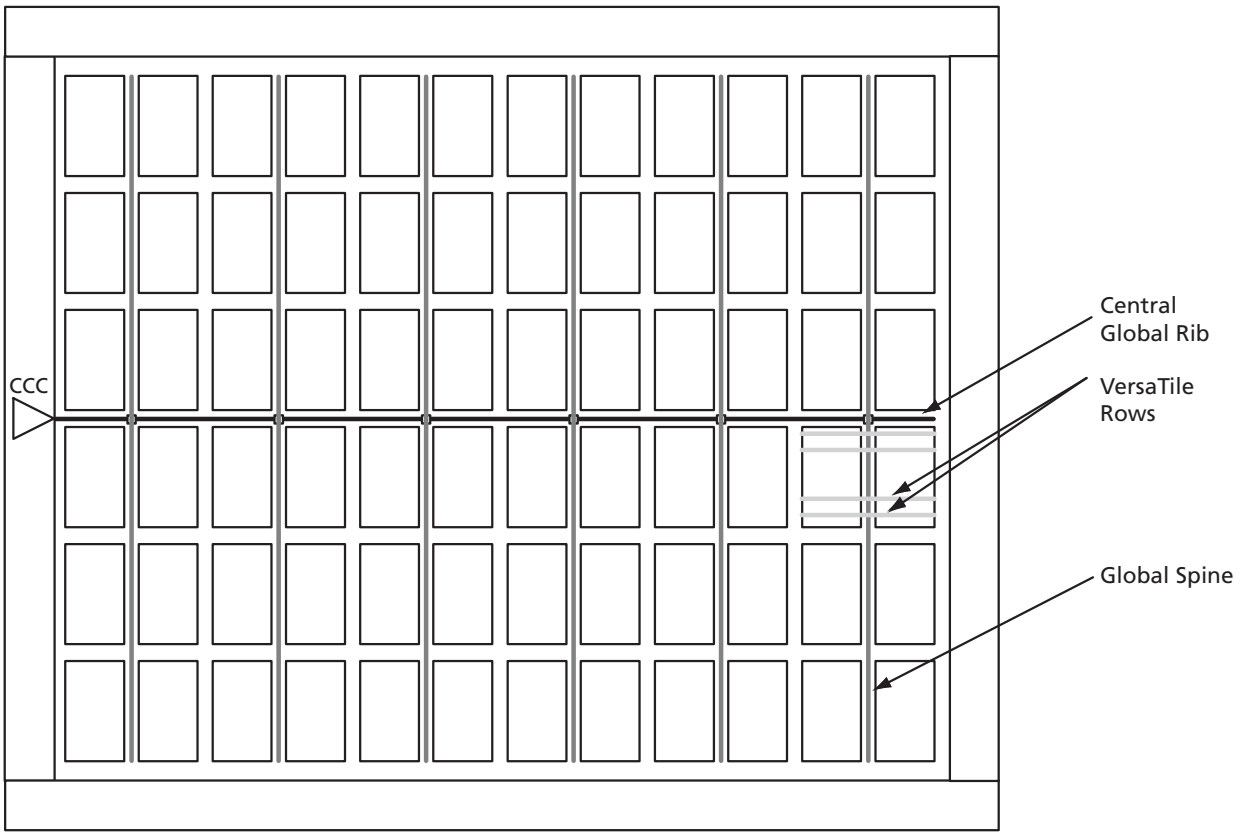


***Figure 2-21* • Example of Global Tree Use in an AGLP125 Device for Clock Routing**

Global Tree Timing Characteristics

Global clock delays include the central rib delay, the spine delay, and the row delay. Delays do not include I/O input buffer clock delays, as these are I/O standard–dependent, and the clock may be driven and conditioned internally by the CCC module. For more details on clock conditioning capabilities, refer to the "Clock Conditioning Circuits" section on page 2-59. Table 2-73 to Table 2-78 on page 2-58 present minimum and maximum global clock delays within each device. Minimum and maximum delays are measured with minimum and maximum loading.

Timing Characteristics

1.5 V DC Core Voltage

Table 2-73 • **AGLP030 Global Resource**
Commercial-Case Conditions: T_J = 70°C, V_{CC} = 1.425 V

Parameter	Description	Std.		Units
		Min.[1]	Max.[2]	
t_{RCKL}	Input LOW Delay for Global Clock	1.21	1.42	ns
t_{RCKH}	Input HIGH Delay for Global Clock	1.23	1.49	ns
$t_{RCKMPWH}$	Minimum Pulse Width HIGH for Global Clock			ns
$t_{RCKMPWL}$	Minimum Pulse Width LOW for Global Clock			ns
t_{RCKSW}	Maximum Skew for Global Clock		0.27	ns
F_{RMAX}	Maximum Frequency for Global Clock			MHz

Notes:

1. *Value reflects minimum load. The delay is measured from the CCC output to the clock pin of a sequential element, located in a lightly loaded row (single element is connected to the global net).*
2. *Value reflects maximum load. The delay is measured on the clock pin of the farthest sequential element, located in a fully loaded row (all available flip-flops are connected to the global net in the row).*
3. *For specific junction temperature and voltage supply levels, refer to Table 2-6 on page 2-6 for derating values.*

Table 2-74 • **AGLP060 Global Resource**
Commercial-Case Conditions: T_J = 70°C, V_{CC} = 1.425 V

Parameter	Description	Std.		Units
		Min.[1]	Max.[2]	
t_{RCKL}	Input LOW Delay for Global Clock	1.32	1.52	ns
t_{RCKH}	Input HIGH Delay for Global Clock	1.34	1.59	ns
$t_{RCKMPWH}$	Minimum Pulse Width HIGH for Global Clock			ns
$t_{RCKMPWL}$	Minimum Pulse Width LOW for Global Clock			ns
t_{RCKSW}	Maximum Skew for Global Clock		0.26	ns
F_{RMAX}	Maximum Frequency for Global Clock			MHz

Notes:

1. *Value reflects minimum load. The delay is measured from the CCC output to the clock pin of a sequential element, located in a lightly loaded row (single element is connected to the global net).*
2. *Value reflects maximum load. The delay is measured on the clock pin of the farthest sequential element, located in a fully loaded row (all available flip-flops are connected to the global net in the row).*
3. *For specific junction temperature and voltage supply levels, refer to Table 2-6 on page 2-6 for derating values.*

Table 2-75 • AGLP125 Global Resource
Commercial-Case Conditions: T_J = 70°C, V_{CC} = 1.425 V

Parameter	Description	Std.		Units
		Min.[1]	Max.[2]	
t_{RCKL}	Input LOW Delay for Global Clock	1.31	1.66	ns
t_{RCKH}	Input HIGH Delay for Global Clock	1.29	1.72	ns
$t_{RCKMPWH}$	Minimum Pulse Width HIGH for Global Clock			ns
$t_{RCKMPWL}$	Minimum Pulse Width LOW for Global Clock			ns
t_{RCKSW}	Maximum Skew for Global Clock		0.43	ns
F_{RMAX}	Maximum Frequency for Global Clock			MHz

Notes:

1. *Value reflects minimum load. The delay is measured from the CCC output to the clock pin of a sequential element, located in a lightly loaded row (single element is connected to the global net).*
2. *Value reflects maximum load. The delay is measured on the clock pin of the farthest sequential element, located in a fully loaded row (all available flip-flops are connected to the global net in the row).*
3. *For specific junction temperature and voltage supply levels, refer to Table 2-6 on page 2-6 for derating values.*

1.2 V DC Core Voltage

Table 2-76 • AGLP030 Global Resource
Commercial-Case Conditions: T_J = 70°C, V_{CC} = 1.14 V

Parameter	Description	Std.		Units
		Min.[1]	Max.[2]	
t_{RCKL}	Input LOW Delay for Global Clock	1.80	2.09	ns
t_{RCKH}	Input HIGH Delay for Global Clock	1.88	2.27	ns
$t_{RCKMPWH}$	Minimum Pulse Width HIGH for Global Clock			ns
$t_{RCKMPWL}$	Minimum Pulse Width LOW for Global Clock			ns
t_{RCKSW}	Maximum Skew for Global Clock		0.39	ns
F_{RMAX}	Maximum Frequency for Global Clock			MHz

Notes:

1. *Value reflects minimum load. The delay is measured from the CCC output to the clock pin of a sequential element, located in a lightly loaded row (single element is connected to the global net).*
2. *Value reflects maximum load. The delay is measured on the clock pin of the farthest sequential element, located in a fully loaded row (all available flip-flops are connected to the global net in the row).*
3. *For specific junction temperature and voltage supply levels, refer to Table 2-7 on page 2-7 for derating values.*

Table 2-77 • AGLP060 Global Resource
Commercial-Case Conditions: T_J = 70°C, V_{CC} = 1.14 V

Parameter	Description	Std.		Units
		Min.[1]	Max.[2]	
t_{RCKL}	Input LOW Delay for Global Clock	2.02	2.30	ns
t_{RCKH}	Input HIGH Delay for Global Clock	2.09	2.46	ns
$t_{RCKMPWH}$	Minimum Pulse Width HIGH for Global Clock			ns
$t_{RCKMPWL}$	Minimum Pulse Width LOW for Global Clock			ns
t_{RCKSW}	Maximum Skew for Global Clock		0.37	ns
F_{RMAX}	Maximum Frequency for Global Clock			MHz

Notes:

1. *Value reflects minimum load. The delay is measured from the CCC output to the clock pin of a sequential element, located in a lightly loaded row (single element is connected to the global net).*
2. *Value reflects maximum load. The delay is measured on the clock pin of the farthest sequential element, located in a fully loaded row (all available flip-flops are connected to the global net in the row).*
3. *For specific junction temperature and voltage supply levels, refer to Table 2-7 on page 2-7 for derating values.*

Table 2-78 • AGLP125 Global Resource
Commercial-Case Conditions: T_J = 70°C, V_{CC} = 1.14 V

Parameter	Description	Std.		Units
		Min.[1]	Max.[2]	
t_{RCKL}	Input LOW Delay for Global Clock	2.08	2.54	ns
t_{RCKH}	Input HIGH Delay for Global Clock	2.15	2.77	ns
$t_{RCKMPWH}$	Minimum Pulse Width HIGH for Global Clock			ns
$t_{RCKMPWL}$	Minimum Pulse Width LOW for Global Clock			ns
t_{RCKSW}	Maximum Skew for Global Clock		0.62	ns
F_{RMAX}	Maximum Frequency for Global Clock			MHz

Notes:

1. *Value reflects minimum load. The delay is measured from the CCC output to the clock pin of a sequential element, located in a lightly loaded row (single element is connected to the global net).*
2. *Value reflects maximum load. The delay is measured on the clock pin of the farthest sequential element, located in a fully loaded row (all available flip-flops are connected to the global net in the row).*
3. *For specific junction temperature and voltage supply levels, refer to Table 2-7 on page 2-7 for derating values.*

Clock Conditioning Circuits

CCC Electrical Specifications

Timing Characteristics

Table 2-79 • IGLOO PLUS CCC/PLL Specification
For IGLOO PLUS V2 or V5 devices, 1.5 V DC Core Supply Voltage

Parameter	Min.	Typ.	Max.	Units
Clock Conditioning Circuitry Input Frequency f_{IN_CCC}	1.5		250	MHz
Clock Conditioning Circuitry Output Frequency f_{OUT_CCC}	0.75		250	MHz
Delay Increments in Programmable Delay Blocks [1, 2]		360		ps
Number of Programmable Values in Each Programmable Delay Block			32	
Input Cycle-to-Cycle Jitter (peak magnitude)			100	MHz
CCC Output Peak-to-Peak Period Jitter F_{CCC_OUT}			1	ns
	Max Peak-to-Peak Period Jitter			
0.75 MHz to 24 MHz	1 Global Network Used	External FB Used	3 Global Networks Used	
24 MHz to 100 MHz	0.50%	0.75%	0.70%	
100 MHz to 250 MHz	1.00%	1.50%	1.20%	
Acquisition Time	2.50%	3.75%	2.75%	
LockControl = 0			300	µs
LockControl = 1			6.0	ms
Tracking Jitter				
LockControl = 0			2.5	
LockControl = 1			1.5	ns
Output Duty Cycle	48.5		51.5	%
Delay Range in Block: Programmable Delay 1 [1, 2, 3]	1.25		15.65	ns
Delay Range in Block: Programmable Delay 2 [1, 2, 3]	0.025		15.65	ns
Delay Range in Block: Fixed Delay [1, 2]		3.5		ns

Notes:

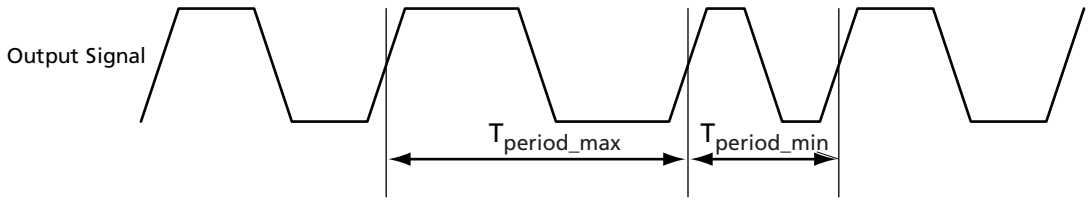
1. *This delay is a function of voltage and temperature. See Table 2-6 on page 2-6 and Table 2-7 on page 2-7 for deratings.*
2. *T_J = 25°C, V_{CC} = 1.5 V*
3. *For definitions of Type 1 and Type 2, refer to the PLL Block Diagram in the Clock Conditioning Circuits in IGLOO and ProASIC3 Devices chapter of the handbook.*
4. *The AGLP030 device does not support PLL.*
5. *Tracking jitter is defined as the variation in clock edge position of PLL outputs with reference to the PLL input clock edge. Tracking jitter does not measure the variation in PLL output period, which is covered by the period jitter parameter.*

Table 2-80 • IGLOO PLUS CCC/PLL Specification
For IGLOO PLUS V2 Devices, 1.2 V DC Core Supply Voltage

Parameter	Min.	Typ.	Max.	Units
Clock Conditioning Circuitry Input Frequency f_{IN_CCC}	1.5		160	MHz
Clock Conditioning Circuitry Output Frequency f_{OUT_CCC}	0.75		160	MHz
Delay Increments in Programmable Delay Blocks [1, 2]		580		ps
Number of Programmable Values in Each Programmable Delay Block			32	
Input Cycle-to-Cycle Jitter (peak magnitude)			60	MHz
CCC Output Peak-to-Peak Period Jitter F_{CCC_OUT}	Max Peak-to-Peak Period Jitter			
	1 Global Network Used	External FB Used	3 Global Networks Used	
0.75 MHz to 24 MHz	0.50%	0.75%	0.70%	
24 MHz to 100 MHz	1.00%	1.50%	1.20%	
100 MHz to 160 MHz	2.50%	3.75%	2.75%	
Acquisition Time				
LockControl = 0			300	µs
LockControl = 1			6.0	ms
Tracking Jitter				
LockControl = 0			4	ns
LockControl = 1			3	ns
Output Duty Cycle	48.5		51.5	%
Delay Range in Block: Programmable Delay 1 [1, 2, 3]	2.3		20.86	ns
Delay Range in Block: Programmable Delay 2 [1, 2, 3]	0.025		20.86	ns
Delay Range in Block: Fixed Delay [1, 2]		5.7		ns

Notes:

1. *This delay is a function of voltage and temperature. See Table 2-6 on page 2-6 and Table 2-7 on page 2-7 for deratings.*
2. *T_J = 25°C, V_{CC} = 1.2 V*
3. *For definitions of Type 1 and Type 2, refer to the PLL Block Diagram in the Clock Conditioning Circuits in IGLOO and ProASIC3 Devices chapter of the handbook.*
4. *The AGLP030 device does not support PLL.*
5. *Tracking jitter is defined as the variation in clock edge position of PLL outputs with reference to PLL input clock edge. Tracking jitter does not measure the variation in PLL output period, which is covered by period jitter parameter.*



Note: Peak-to-peak jitter measurements are defined by $T_{peak\text{-}to\text{-}peak} = T_{period\_max} - T_{period\_min}$.

Figure 2-22 • Peak-to-Peak Jitter Definition

Embedded SRAM and FIFO Characteristics

SRAM

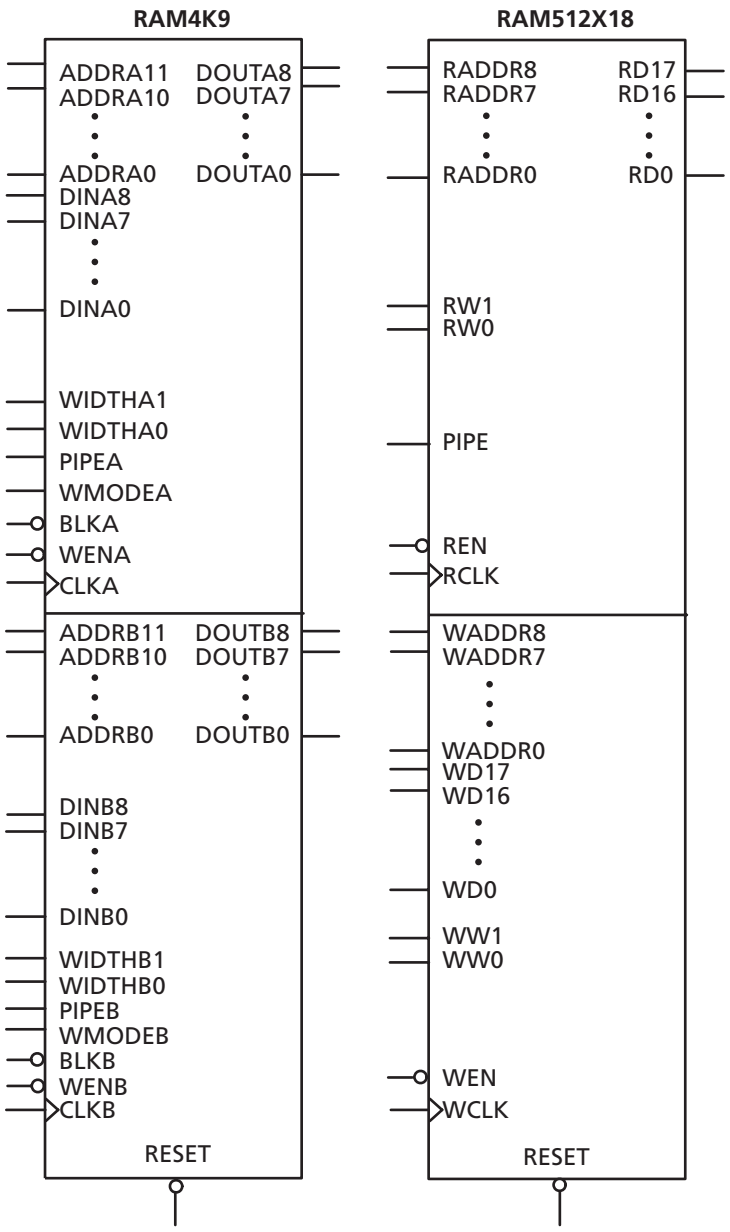


Figure 2-23 • **RAM Models**

Timing Waveforms

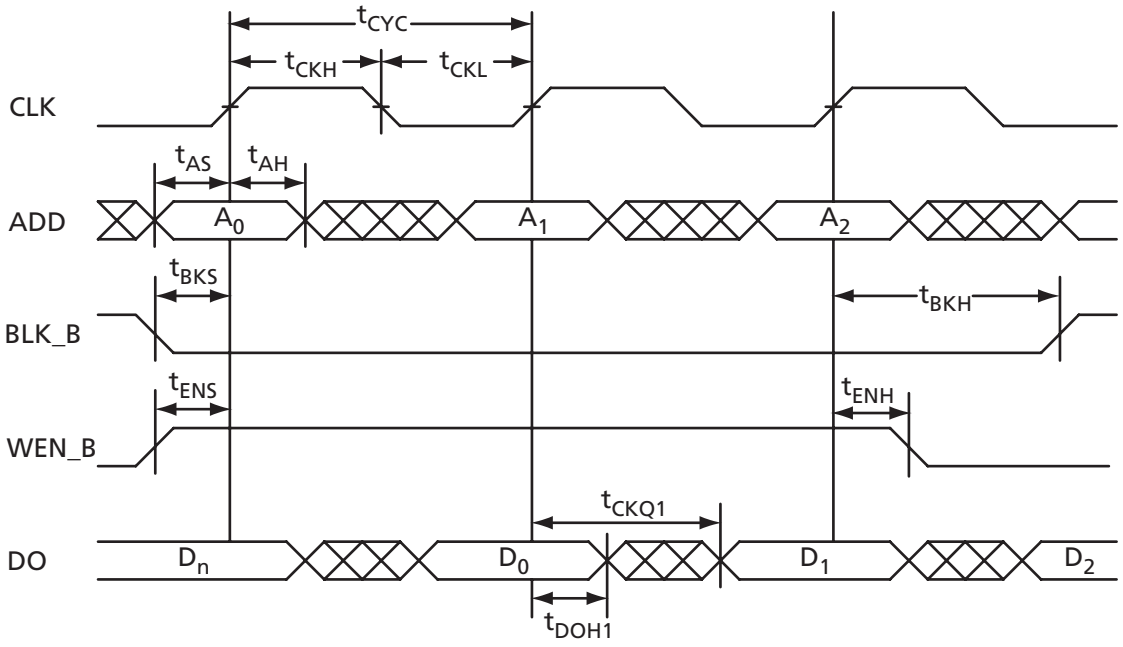


***Figure 2-24* • RAM Read for Pass-Through Output**

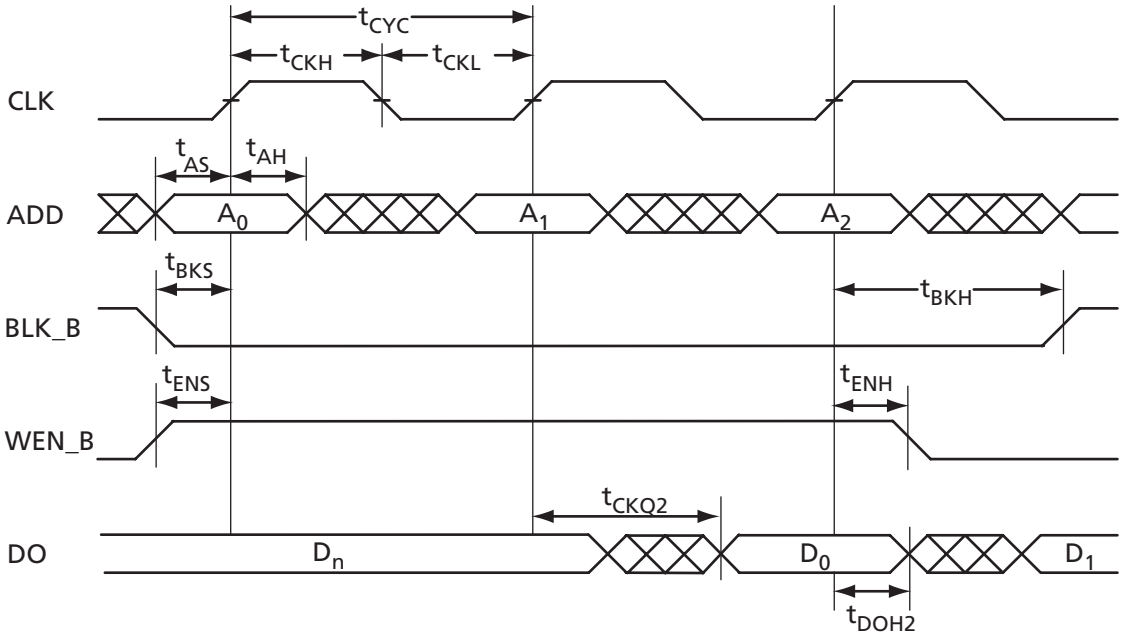


***Figure 2-25* • RAM Read for Pipelined Output**

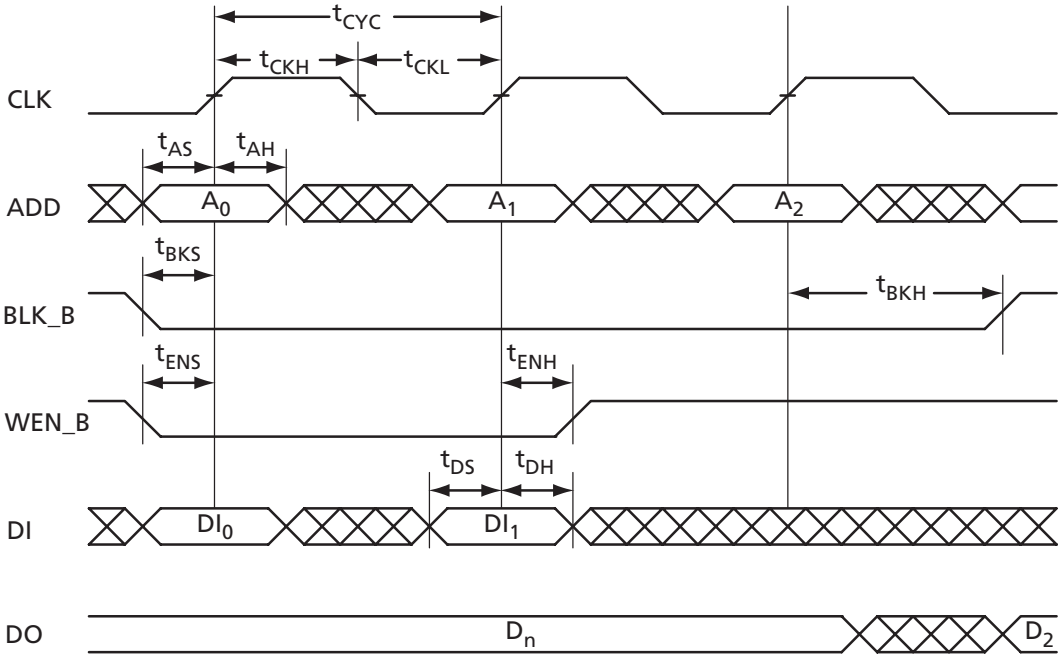


Figure 2-26 • **RAM Write, Output Retained (WMODE = 0)**

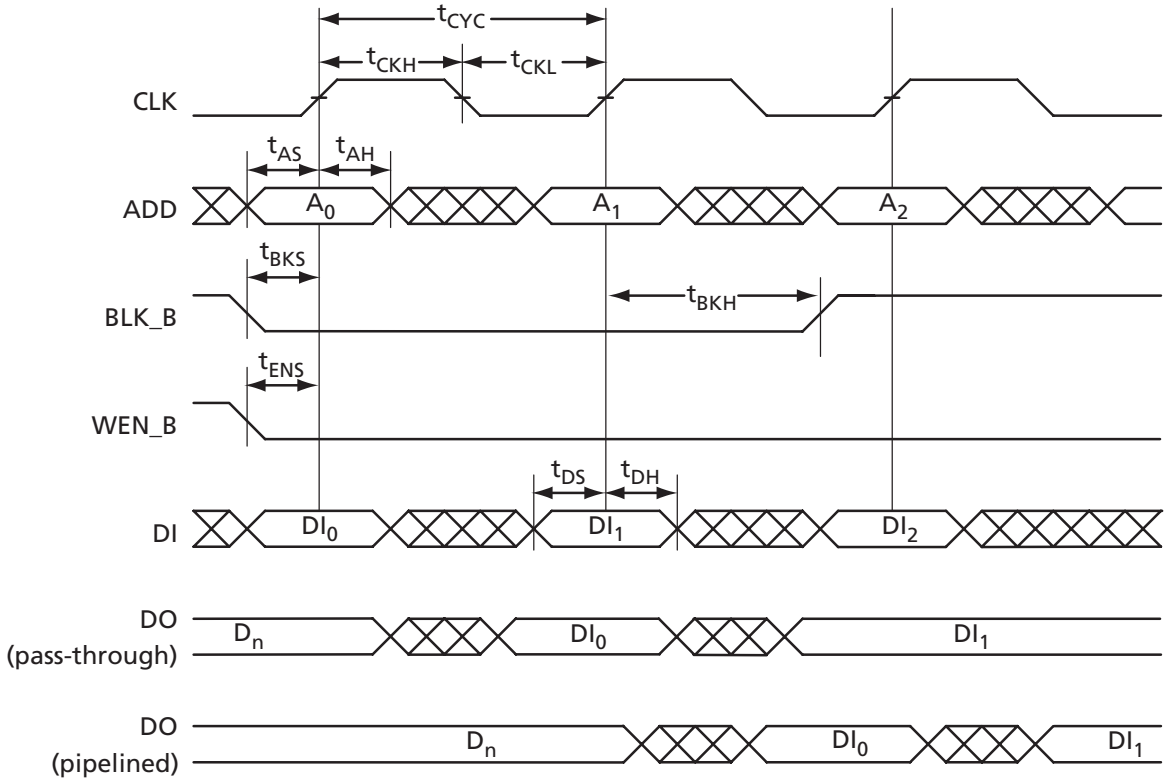


Figure 2-27 • **RAM Write, Output as Write Data (WMODE = 1)**

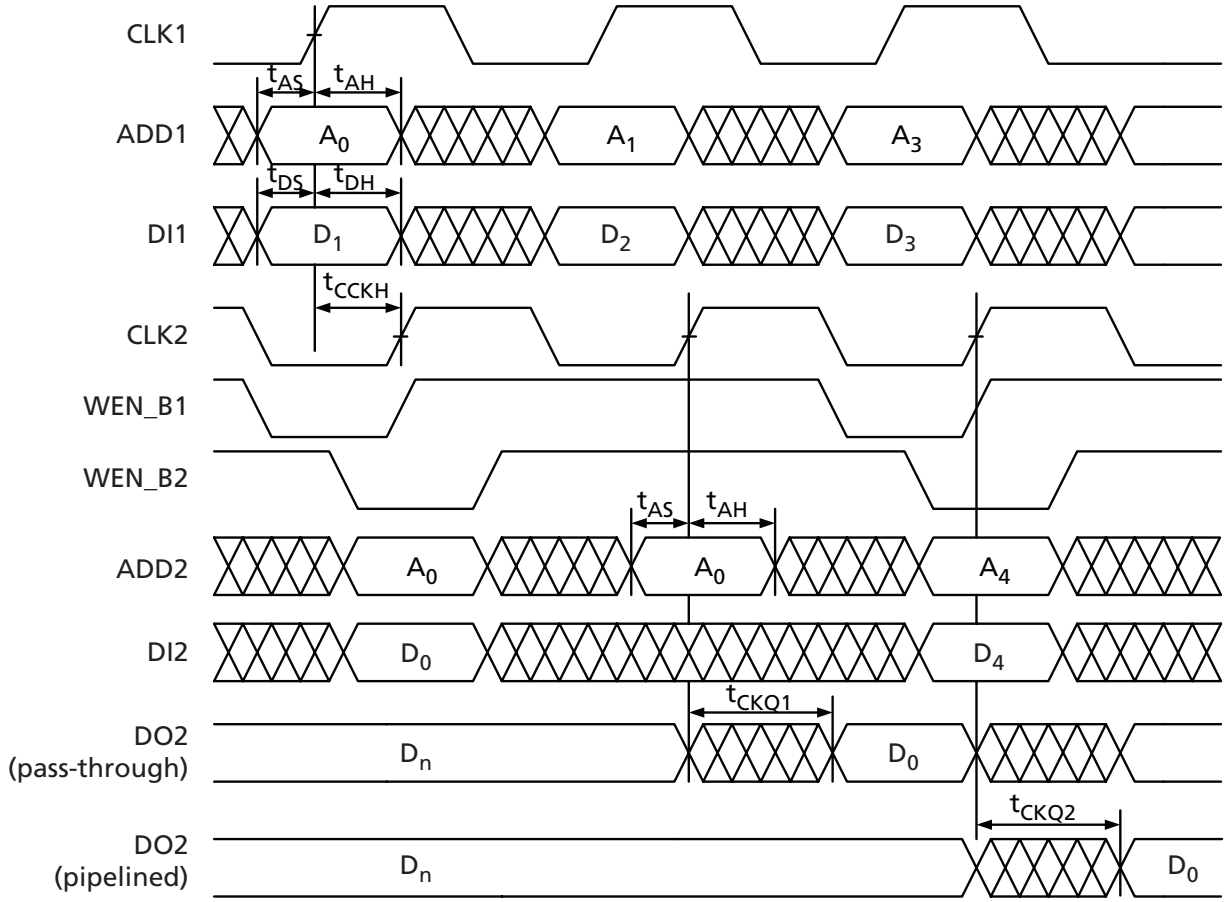


Figure 2-28 • **Write Access after Write onto Same Address**

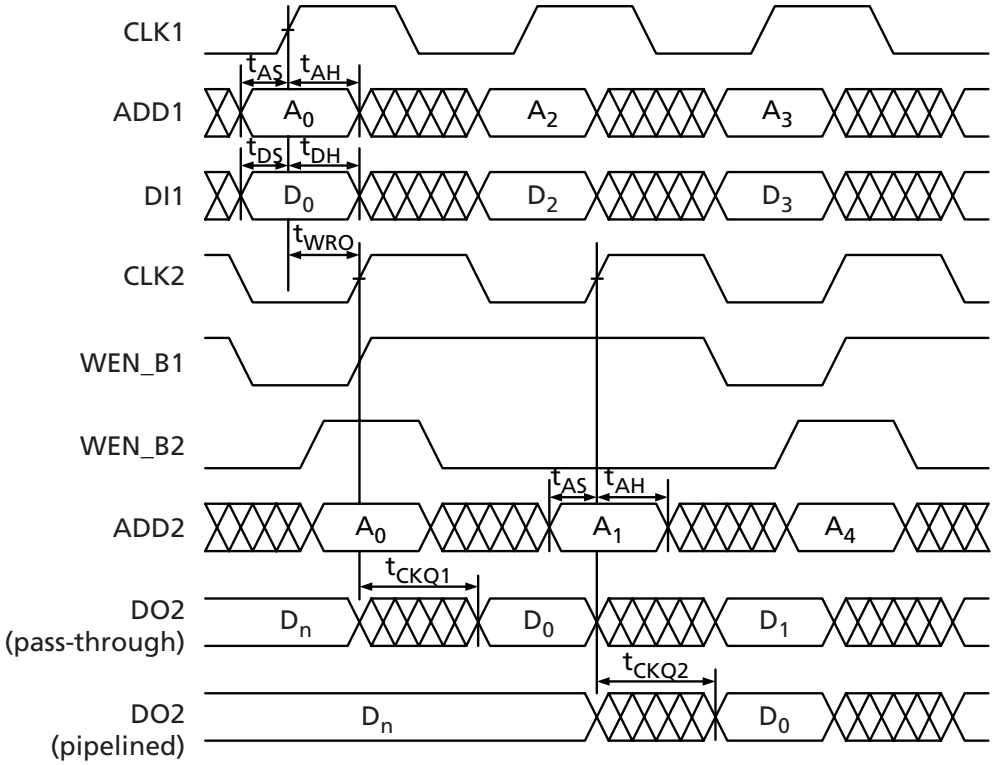


Figure 2-29 • **Read Access after Write onto Same Address**

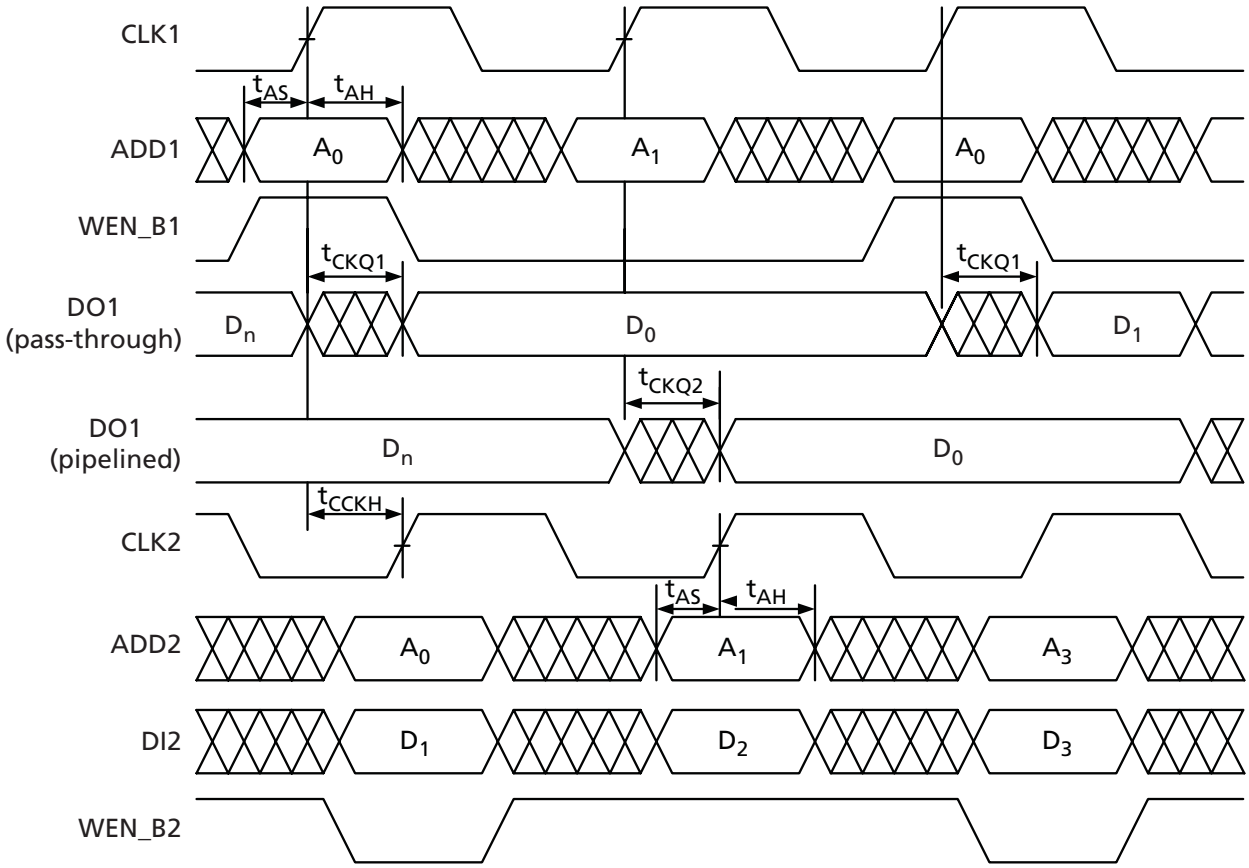


Figure 2-30 • **Write Access after Read onto Same Address**

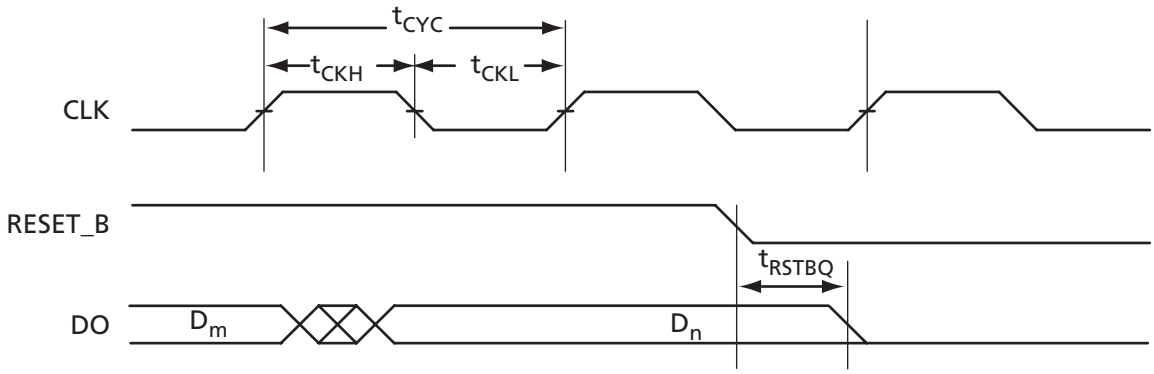


Figure 2-31 • **RAM Reset**

Timing Characteristics

1.5 V DC Core Voltage

Table 2-81 • RAM4K9
Commercial-Case Conditions: T_J = 70°C, Worst-Case V_{CC} = 1.425 V

Parameter	Description	Std.	Units
t_{AS}	Address setup time	0.83	ns
t_{AH}	Address hold time	0.16	ns
t_{ENS}	REN_B, WEN_B setup time	0.81	ns
t_{ENH}	REN_B, WEN_B hold time	0.16	ns
t_{BKS}	BLK_B setup time	1.65	ns
t_{BKH}	BLK_B hold time	0.16	ns
t_{DS}	Input data (DI) setup time	0.71	ns
t_{DH}	Input data (DI) hold time	0.36	ns
t_{CKQ1}	Clock HIGH to new data valid on DO (output retained, WMODE = 0)	3.53	ns
	Clock HIGH to new data valid on DO (flow-through, WMODE = 1)	3.06	ns
t_{CKQ2}	Clock HIGH to new data valid on DO (pipelined)	1.81	ns
t_{WRO}	Address collision clk-to-clk delay for reliable read access after write on same address	TBD	ns
t_{CCKH}	Address collision clk-to-clk delay for reliable write access after write/read on same address	TBD	ns
t_{RSTBQ}	RESET_B LOW to data out LOW on DO (flow-through)	2.06	ns
	RESET_B LOW to data out LOW on DO (pipelined)	2.06	ns
$t_{REMRSTB}$	RESET_B removal	0.61	ns
$t_{RECRSTB}$	RESET_B recovery	3.21	ns
$t_{MPWRSTB}$	RESET_B minimum pulse width	0.68	ns
t_{CYC}	Clock cycle time	6.24	ns
F_{MAX}	Maximum frequency	160	MHz

Note: For specific junction temperature and voltage supply levels, refer to Table 2-6 on page 2-6 for derating values.

Table 2-82 • RAM512X18
Commercial-Case Conditions: T_J = 70°C, Worst-Case V_{CC} = 1.425 V

Parameter	Description	Std.	Units
t_{AS}	Address setup time	0.83	ns
t_{AH}	Address hold time	0.16	ns
t_{ENS}	REN_B, WEN_B setup time	0.73	ns
t_{ENH}	REN_B, WEN_B hold time	0.08	ns
t_{DS}	Input data (DI) setup time	0.71	ns
t_{DH}	Input data (DI) hold time	0.36	ns
t_{CKQ1}	Clock HIGH to new data valid on DO (output retained, WMODE = 0)	4.21	ns
t_{CKQ2}	Clock HIGH to new data valid on DO (pipelined)	1.71	ns
t_{WRO}	Address collision clk-to-clk delay for reliable read access after write on same address	TBD	ns
t_{CCKH}	Address collision clk-to-clk delay for reliable write access after write/read on same address	TBD	ns
t_{RSTBQ}	RESET_B LOW to data out LOW on DO (flow-through)	2.06	ns
	RESET_B LOW to data out LOW on DO (pipelined)	2.06	ns
$t_{REMRSTB}$	RESET_B removal	0.61	ns
$t_{RECRSTB}$	RESET_B recovery	3.21	ns
$t_{MPWRSTB}$	RESET_B minimum pulse width	0.68	ns
t_{CYC}	Clock cycle time	6.24	ns
F_{MAX}	Maximum frequency	160	MHz

Note: For specific junction temperature and voltage supply levels, refer to Table 2-6 on page 2-6 for derating values.

1.2 V DC Core Voltage

Table 2-83 • RAM4K9
Commercial-Case Conditions: T_J = 70°C, Worst-Case V_{CC} = 1.14 V

Parameter	Description	Std.	Units
t_{AS}	Address setup time	1.53	ns
t_{AH}	Address hold time	0.29	ns
t_{ENS}	REN_B, WEN_B setup time	1.50	ns
t_{ENH}	REN_B, WEN_B hold time	0.29	ns
t_{BKS}	BLK_B setup time	3.05	ns
t_{BKH}	BLK_B hold time	0.29	ns
t_{DS}	Input data (DI) setup time	1.33	ns
t_{DH}	Input data (DI) hold time	0.66	ns
t_{CKQ1}	Clock HIGH to new data valid on DO (output retained, WMODE = 0)	6.61	ns
	Clock HIGH to new data valid on DO (flow-through, WMODE = 1)	5.72	ns
t_{CKQ2}	Clock HIGH to new data valid on DO (pipelined)	3.38	ns
t_{WRO}	Address collision clk-to-clk delay for reliable read access after write on same address	TBD	ns
t_{CCKH}	Address collision clk-to-clk delay for reliable write access after write/read on same address	TBD	ns
t_{RSTBQ}	RESET_B LOW to data out LOW on DO (flow-through)	3.86	ns
	RESET_B LOW to data out LOW on DO (pipelined)	3.86	ns
$t_{REMRSTB}$	RESET_B removal	1.12	ns
$t_{RECRSTB}$	RESET_B recovery	5.93	ns
$t_{MPWRSTB}$	RESET_B minimum pulse width	1.18	ns
t_{CYC}	Clock cycle time	10.90	ns
F_{MAX}	Maximum frequency	92	MHz

Note: For specific junction temperature and voltage supply levels, refer to Table 2-7 on page 2-7 for derating values.

Table 2-84 • RAM512X18
Commercial-Case Conditions: T_J = 70°C, Worst-Case V_{CC} = 1.14 V

Parameter	Description	Std.	Units
t_{AS}	Address setup time	1.53	ns
t_{AH}	Address hold time	0.29	ns
t_{ENS}	REN_B, WEN_B setup time	1.36	ns
t_{ENH}	REN_B, WEN_B hold time	0.15	ns
t_{DS}	Input data (DI) setup time	1.33	ns
t_{DH}	Input data (DI) hold time	0.66	ns
t_{CKQ1}	Clock HIGH to new data valid on DO (output retained, WMODE = 0)	7.88	ns
t_{CKQ2}	Clock HIGH to new data valid on DO (pipelined)	3.20	ns
t_{WRO}	Address collision clk-to-clk delay for reliable read access after write on same address	TBD	ns
t_{CCKH}	Address collision clk-to-clk delay for reliable write access after write/read on same address	TBD	ns
t_{RSTBQ}	RESET_B LOW to data out LOW on DO (flow through)	3.86	ns
	RESET_B LOW to data out LOW on DO (pipelined)	3.86	ns
$t_{REMRSTB}$	RESET_B removal	1.12	ns
$t_{RECRSTB}$	RESET_B recovery	5.93	ns
$t_{MPWRSTB}$	RESET_B minimum pulse width	1.18	ns
t_{CYC}	Clock cycle time	10.90	ns
F_{MAX}	Maximum frequency	92	MHz

Note: For specific junction temperature and voltage supply levels, refer to Table 2-7 on page 2-7 for derating values.

FIFO

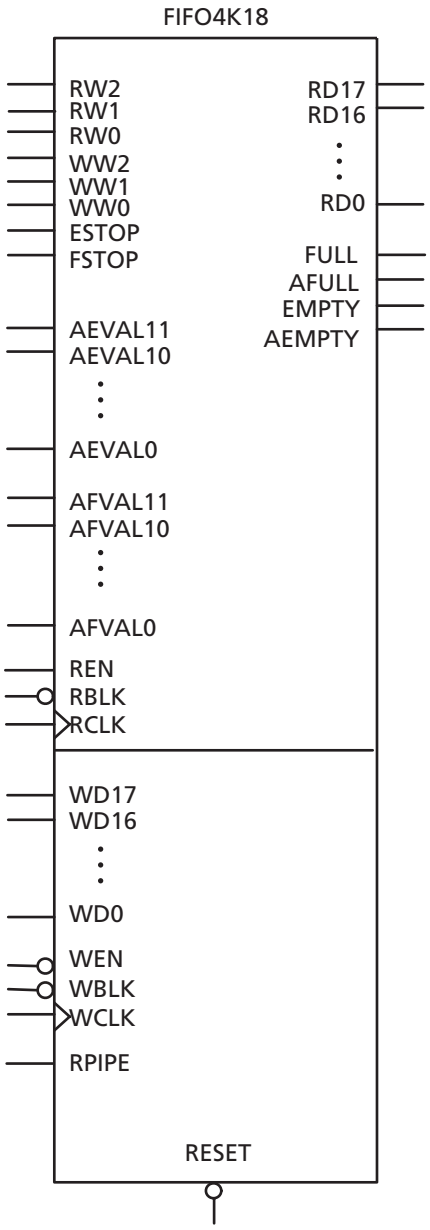


Figure 2-32 • **FIFO Model**

Timing Waveforms

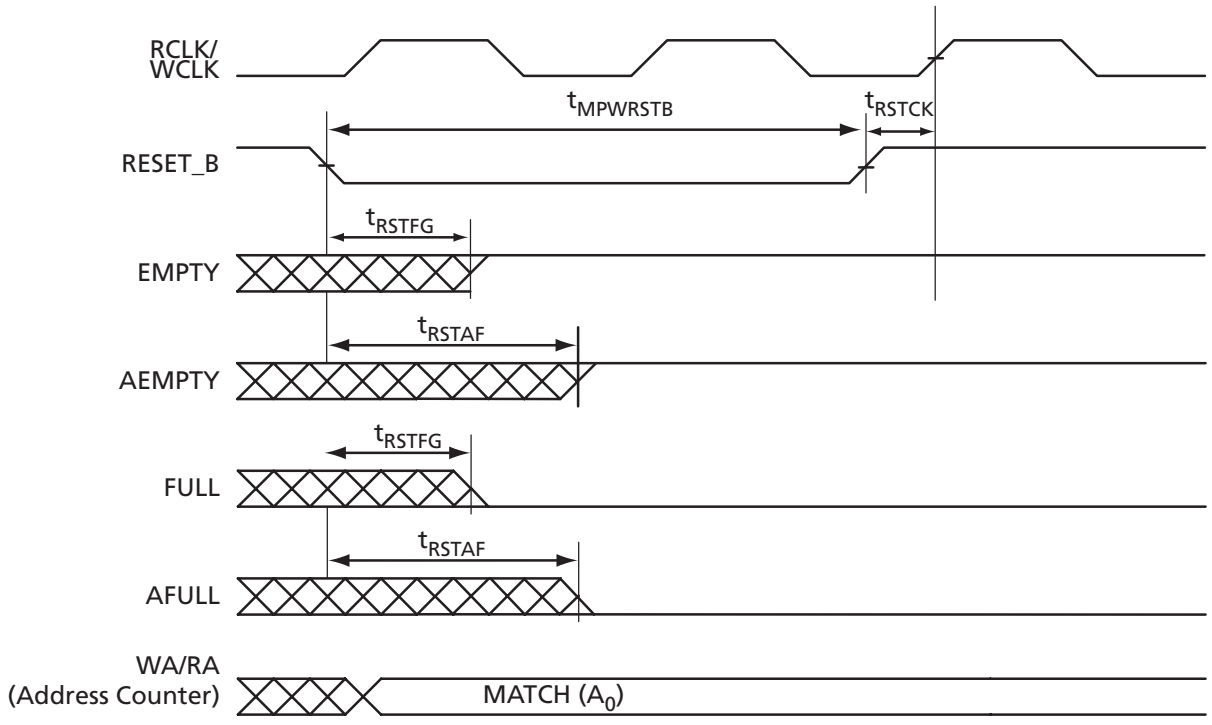


***Figure 2-33* • FIFO Reset**

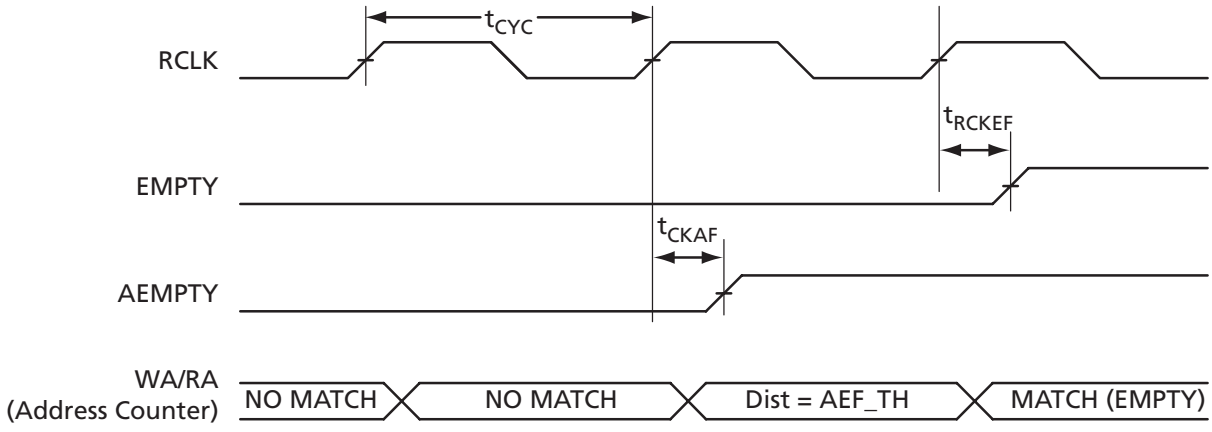


***Figure 2-34* • FIFO EMPTY Flag and AEMPTY Flag Assertion**

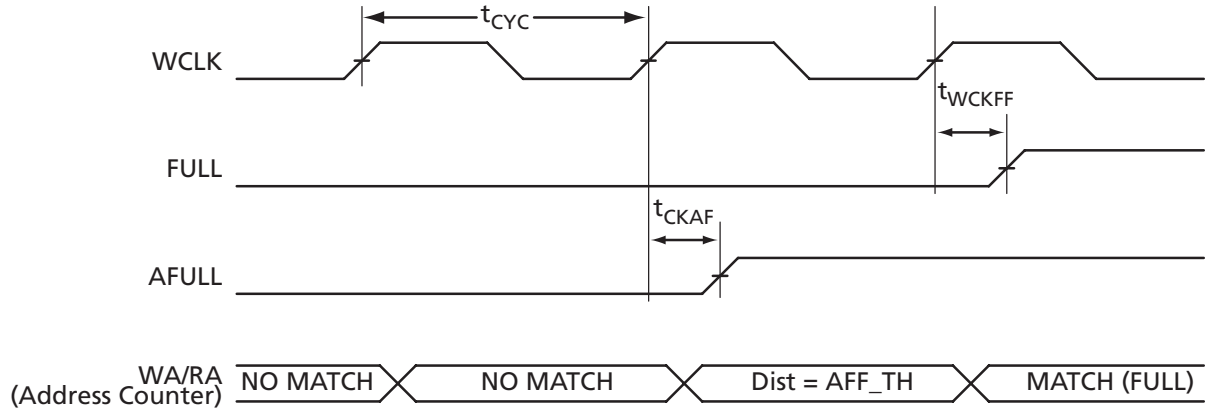


Figure 2-35 • **FIFO FULL Flag and AFULL Flag Assertion**

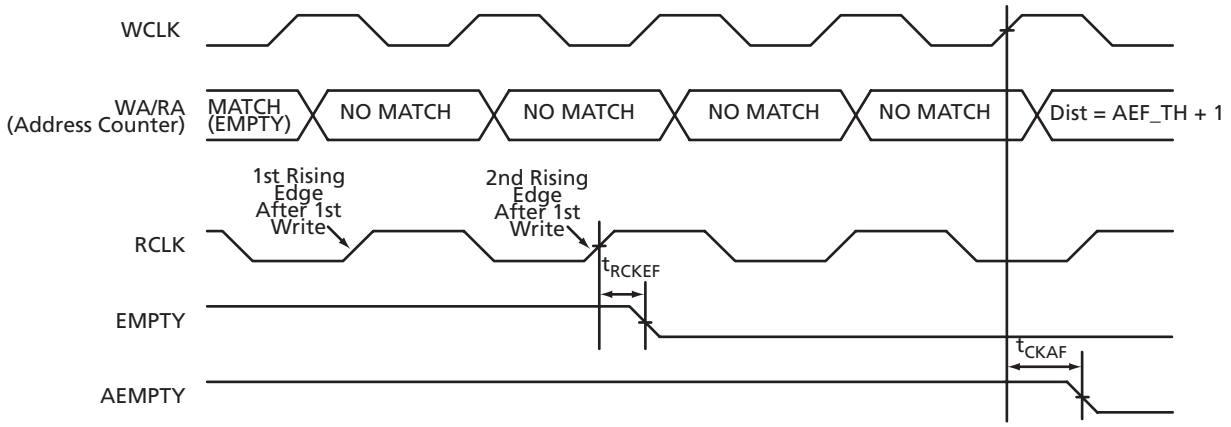


Figure 2-36 • **FIFO EMPTY Flag and AEMPTY Flag Deassertion**

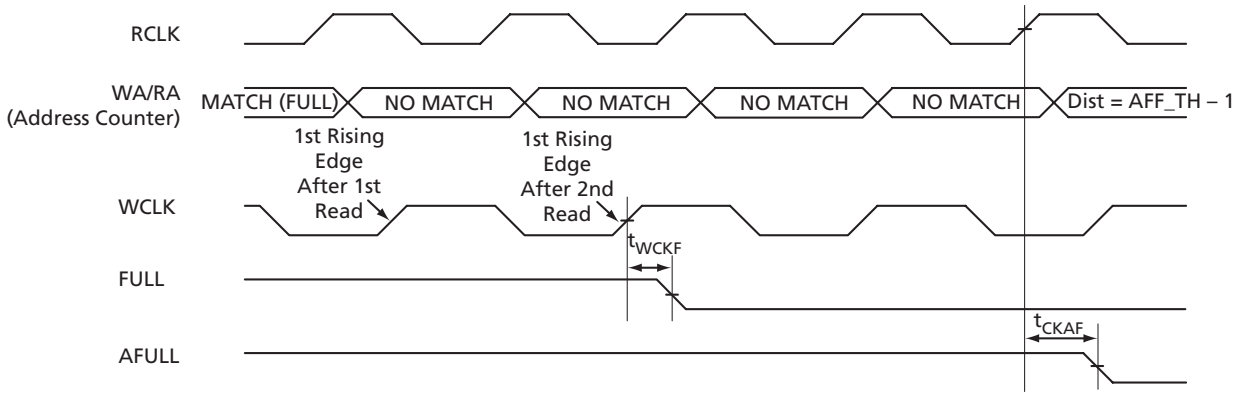


Figure 2-37 • **FIFO FULL Flag and AFULL Flag Deassertion**

Timing Characteristics

1.5 V DC Core Voltage

Table 2-85 • FIFO
Worst Commercial-Case Conditions: T_J = 70°C, V_{CC} = 1.425 V

Parameter	Description	Std.	Units
t_{ENS}	REN_B, WEN_B Setup Time	1.99	ns
t_{ENH}	REN_B, WEN_B Hold Time	0.16	ns
t_{BKS}	BLK_B Setup Time	0.30	ns
t_{BKH}	BLK_B Hold Time	0.00	ns
t_{DS}	Input Data (DI) Setup Time	0.76	ns
t_{DH}	Input Data (DI) Hold Time	0.25	ns
t_{CKQ1}	Clock HIGH to New Data Valid on DO (flow-through)	3.33	ns
t_{CKQ2}	Clock HIGH to New Data Valid on DO (pipelined)	1.80	ns
t_{RCKEF}	RCLK HIGH to Empty Flag Valid	3.53	ns
t_{WCKFF}	WCLK HIGH to Full Flag Valid	3.35	ns
t_{CKAF}	Clock HIGH to Almost Empty/Full Flag Valid	12.85	ns
t_{RSTFG}	RESET_B LOW to Empty/Full Flag Valid	3.48	ns
t_{RSTAF}	RESET_B LOW to Almost Empty/Full Flag Valid	12.72	ns
t_{RSTBQ}	RESET_B LOW to Data Out LOW on DO (flow-through)	2.02	ns
	RESET_B LOW to Data Out LOW on DO (pipelined)	2.02	ns
$t_{REMRSTB}$	RESET_B Removal	0.61	ns
$t_{RECRSTB}$	RESET_B Recovery	3.21	ns
$t_{MPWRSTB}$	RESET_B Minimum Pulse Width	0.68	ns
t_{CYC}	Clock Cycle Time	6.24	ns
F_{MAX}	Maximum Frequency for FIFO	160	MHz

Note: For specific junction temperature and voltage supply levels, refer to Table 2-6 on page 2-6 for derating values.

1.2 V DC Core Voltage

Table 2-86 • FIFO
Worst Commercial-Case Conditions: T_J = 70°C, V_{CC} = 1.14 V

Parameter	Description	Std.	Units
t_{ENS}	REN_B, WEN_B Setup Time	4.13	ns
t_{ENH}	REN_B, WEN_B Hold Time	0.31	ns
t_{BKS}	BLK_B Setup Time	0.30	ns
t_{BKH}	BLK_B Hold Time	0.00	ns
t_{DS}	Input Data (DI) Setup Time	1.56	ns
t_{DH}	Input Data (DI) Hold Time	0.49	ns
t_{CKQ1}	Clock HIGH to New Data Valid on DO (flow-through)	6.80	ns
t_{CKQ2}	Clock HIGH to New Data Valid on DO (pipelined)	3.62	ns
t_{RCKEF}	RCLK HIGH to Empty Flag Valid	7.23	ns
t_{WCKFF}	WCLK HIGH to Full Flag Valid	6.85	ns
t_{CKAF}	Clock HIGH to Almost Empty/Full Flag Valid	26.61	ns
t_{RSTFG}	RESET_B LOW to Empty/Full Flag Valid	7.12	ns
t_{RSTAF}	RESET_B LOW to Almost Empty/Full Flag Valid	26.33	ns
t_{RSTBQ}	RESET_B LOW to Data Out LOW on DO (flow-through)	4.09	ns
	RESET_B LOW to Data Out LOW on DO (pipelined)	4.09	ns
$t_{REMRSTB}$	RESET_B Removal	1.23	ns
$t_{RECRSTB}$	RESET_B Recovery	6.58	ns
$t_{MPWRSTB}$	RESET_B Minimum Pulse Width	1.18	ns
t_{CYC}	Clock Cycle Time	10.90	ns
F_{MAX}	Maximum Frequency for FIFO	92	MHz

Note: For specific junction temperature and voltage supply levels, refer to Table 2-7 on page 2-7 for derating values.

Embedded FlashROM Characteristics

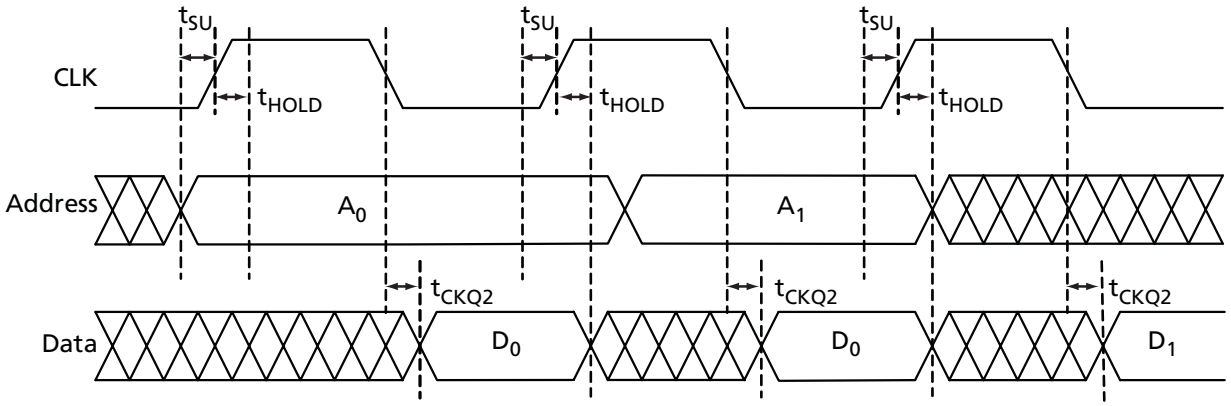


Figure 2-38 • **Timing Diagram**

Timing Characteristics

1.5 V DC Core Voltage

Table 2-87 • **Embedded FlashROM Access Time**
Worst Commercial-Case Conditions: T_J = 70°C, V_{CC} = 1.425 V

Parameter	Description	Std.	Units
t_{SU}	Address Setup Time	0.57	ns
t_{HOLD}	Address Hold Time	0.00	ns
t_{CK2Q}	Clock to Out	33.14	ns
F_{MAX}	Maximum Clock Frequency	15	MHz

1.2 V DC Core Voltage

Table 2-88 • **Embedded FlashROM Access Time**
Worst Commercial-Case Conditions: T_J = 70°C, V_{CC} = 1.14 V

Parameter	Description	Std.	Units
t_{SU}	Address Setup Time	0.59	ns
t_{HOLD}	Address Hold Time	0.00	ns
t_{CK2Q}	Clock to Out	52.04	ns
F_{MAX}	Maximum Clock Frequency	10	MHz

JTAG 1532 Characteristics

JTAG timing delays do not include JTAG I/Os. To obtain complete JTAG timing, add I/O buffer delays to the corresponding standard selected; refer to the I/O timing characteristics in the "User I/O Characteristics" section on page 2-15 for more details.

Timing Characteristics

1.5 V DC Core Voltage

Table 2-89 • JTAG 1532
Commercial-Case Conditions: T_J = 70°C, Worst-Case V_{CC} = 1.425 V

Parameter	Description	Std.	Units
t_{DISU}	Test Data Input Setup Time	1.00	ns
t_{DIHD}	Test Data Input Hold Time	2.00	ns
t_{TMSSU}	Test Mode Select Setup Time	1.00	ns
t_{TMDHD}	Test Mode Select Hold Time	2.00	ns
t_{TCK2Q}	Clock to Q (data out)	8.00	ns
t_{RSTB2Q}	Reset to Q (data out)	25.00	ns
F_{TCKMAX}	TCK Maximum Frequency	15	MHz
$t_{TRSTREM}$	ResetB Removal Time	0.58	ns
$t_{TRSTREC}$	ResetB Recovery Time	0.00	ns
$t_{TRSTMPW}$	ResetB Minimum Pulse	TBD	ns

Note: For specific junction temperature and voltage supply levels, refer to Table 2-6 on page 2-6 for derating values.

1.2 V DC Core Voltage

Table 2-90 • JTAG 1532
Commercial-Case Conditions: T_J = 70°C, Worst-Case V_{CC} = 1.14 V

Parameter	Description	Std.	Units
t_{DISU}	Test Data Input Setup Time	1.50	ns
t_{DIHD}	Test Data Input Hold Time	3.00	ns
t_{TMSSU}	Test Mode Select Setup Time	1.50	ns
t_{TMDHD}	Test Mode Select Hold Time	3.00	ns
t_{TCK2Q}	Clock to Q (data out)	11.00	ns
t_{RSTB2Q}	Reset to Q (data out)	30.00	ns
F_{TCKMAX}	TCK Maximum Frequency	9.00	MHz
$t_{TRSTREM}$	ResetB Removal Time	1.18	ns
$t_{TRSTREC}$	ResetB Recovery Time	0.00	ns
$t_{TRSTMPW}$	ResetB Minimum Pulse	TBD	ns

Note: For specific junction temperature and voltage supply levels, refer to Table 2-6 on page 2-6 for derating values.

Part Number and Revision Date

Part Number 51700102-002-3
Revised October 2008

List of Changes

The following table lists critical changes that were made in the current version of the chapter.

Previous Version	Changes in Current Version (Advance v0.4)	Page
Advance v0.3 (July 2008)	Data was revised significantly in the following tables: Table 2-24 · Summary of I/O Timing Characteristics—Software Default Settings, STD Speed Grade, Commercial-Case Conditions: T_J = 70°C, Worst-Case VCC = 1.425 V, Worst-Case VCCI = 3.0 V, Table 2-25 · Summary of I/O Timing Characteristics—Software Default Settings, STD Speed Grade, Commercial-Case Conditions: T_J = 70°C, Worst-Case VCC = 1.14 V, Worst-Case VCCI = 3.0 V Table 2-43 · 2.5 LVCMOS Low Slew – Applies to 1.2 V DC Core Voltage Table 2-44 · 2.5 V LVCMOS High Slew – Applies to 1.2 V DC Core Voltage	2-21, 2-30
Advance v0.2 (March 2008)	As a result of the Libero IDE v8.4 release, Actel now offers a wide range of core voltage support. The document was updated to change 1.2 V / 1.5 V to 1.2 V to 1.5 V.	N/A
Advance v0.1 (January 2008)	Tables have been updated to reflect default values in the software. The default I/O capacitance is 5 pF. Tables have been updated to include the LVCMOS 1.2 V I/O set.	N/A
	Table note 3 was updated in Table 2-2 · Recommended Operating Conditions [4] to add the sentence, "V_{CCI} should be at the same voltage within a given I/O bank." References to table notes 5, 6, 7, and 8 were added. Reference to table note 3 was removed from V_{PUMP} Operation and placed next to V_{CC}.	2-2
	Table 2-4 · Overshoot and Undershoot Limits 1 was revised to remove "as measured on quiet I/Os" from the title. Table note 2 was revised to remove "estimated SSO density over cycles." Table note 3 was deleted.	2-3
	The table note for Table 2-8 · Quiescent Supply Current (I_{DD}) Characteristics, IGLOO PLUS Flash*Freeze Mode* to remove the sentence stating that values do not include I/O static contribution.	2-7
	The table note for Table 2-9 · Quiescent Supply Current (I_{DD}) Characteristics, IGLOO PLUS Sleep Mode (VCC = 0 V)* was updated to remove V_{JTAG} and V_{CCI} and the statement that values do not include I/O static contribution.	2-7
	The table note for Table 2-10 · Quiescent Supply Current (I_{DD}) Characteristics, IGLOO PLUS Shutdown Mode (VCC, VCCI = 0 V)* was updated to remove the statement that values do not include I/O static contribution.	2-7
	Note 2 of Table 2-11 · Quiescent Supply Current (I_{DD}), No IGLOO PLUS Flash*Freeze Mode1 was updated to include V_{CCPLL}. Table note 4 was deleted.	2-8
	Table 2-12 · Summary of I/O Input Buffer Power (per pin) – Default I/O Software Settings and Table 2-13 · Summary of I/O Output Buffer Power (per pin) – Default I/O Software Settings1 were updated to remove static power. The table notes were updated to reflect that power was measured on V_{CCI}. Table note 2 was added to Table 2-12 · Summary of I/O Input Buffer Power (per pin) – Default I/O Software Settings.	2-8, 2-9

Previous Version	Changes in Current Version (Advance v0.4)	Page
Advance v0.1 (continued)	Table 2-15 · Different Components Contributing to the Static Power Consumption in IGLOO PLUS Devices and Table 2-17 · Different Components Contributing to the Static Power Consumption in IGLOO PLUS Devices were updated to change the definition for P_{DC5} from bank static power to bank quiescent power. Table subtitles were added for Table 2-15 · Different Components Contributing to the Static Power Consumption in IGLOO PLUS Devices, Table 2-16 · Different Components Contributing to Dynamic Power Consumption in IGLOO PLUS Devices, and Table 2-17 · Different Components Contributing to the Static Power Consumption in IGLOO PLUS Devices.	2-10, 2-11
	The "Total Static Power Consumption—P_{STAT}" section was revised.	2-12
	Table 2-31 · Schmitt Trigger Input Hysteresis is new.	2-24

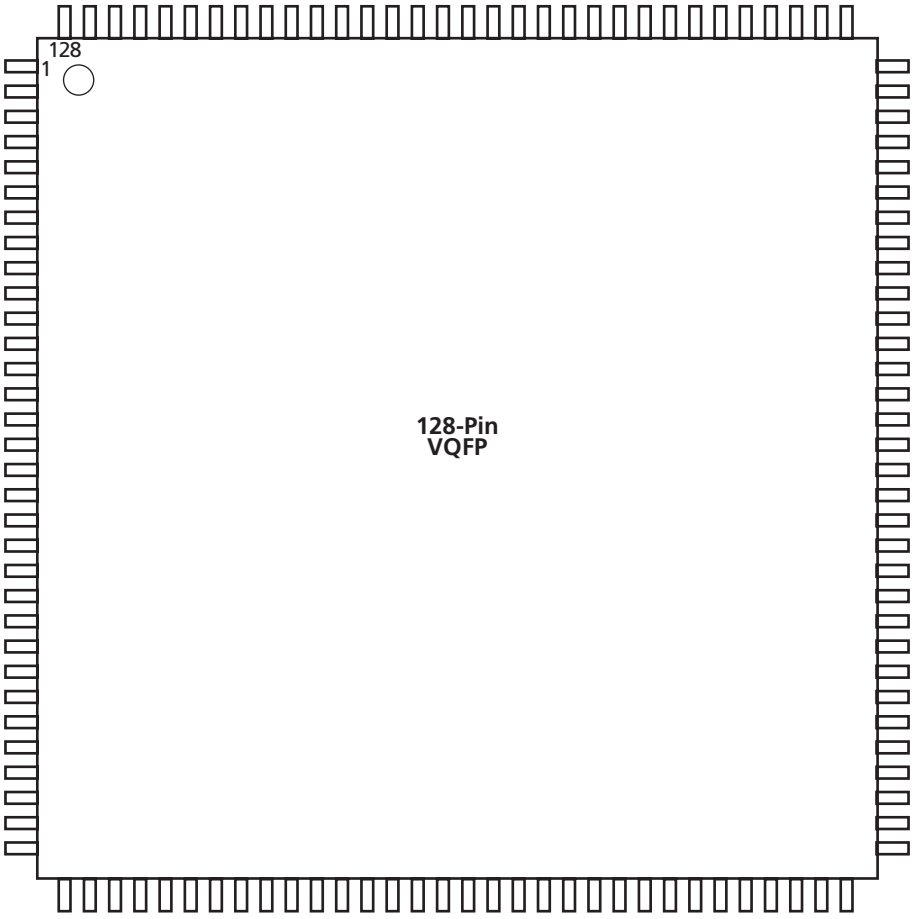
Actel Safety Critical, Life Support, and High-Reliability Applications Policy

The Actel products described in this advance status datasheet may not have completed Actel's qualification process. Actel may amend or enhance products during the product introduction and qualification process, resulting in changes in device functionality or performance. It is the responsibility of each customer to ensure the fitness of any Actel product (but especially a new product) for a particular purpose, including appropriateness for safety-critical, life-support, and other high-reliability applications. Consult Actel's Terms and Conditions for specific liability exclusions relating to life-support applications. A reliability report covering all of Actel's products is available on the Actel website at http://www.actel.com/documents/ORT_Report.pdf. Actel also offers a variety of enhanced qualification and lot acceptance screening procedures. Contact your local Actel sales office for additional reliability information.

Rate This Document

3 – Package Pin Assignments

128-Pin VQFP



Note: This is the bottom view of the package.

Note

For Package Manufacturing and Environmental information, visit the Resource Center at http://www.actel.com/products/solutions/package/docs.aspx.

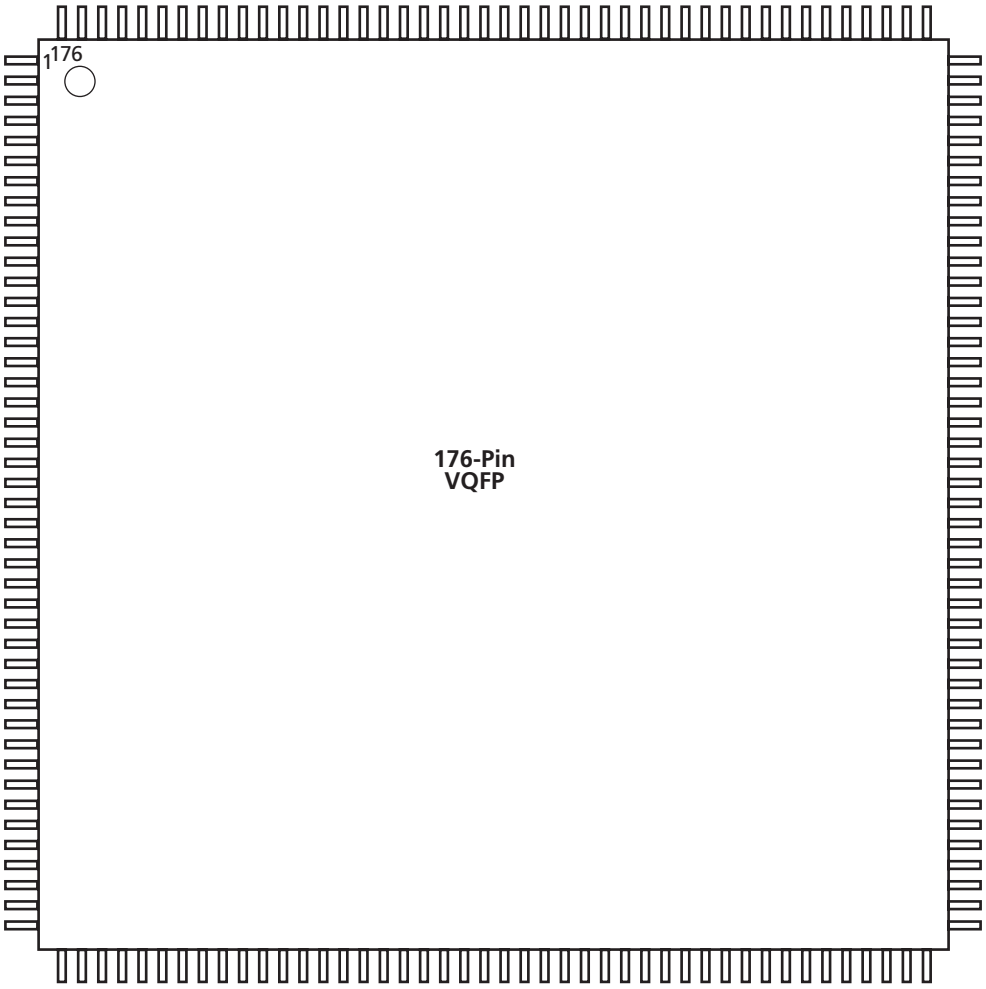
128-Pin VQFP	
Pin Number	AGLP030 Function
1	IO119RSB3
2	IO118RSB3
3	IO117RSB3
4	IO115RSB3
5	IO116RSB3
6	IO113RSB3
7	IO114RSB3
8	GND
9	V_{CCI}B3
10	IO112RSB3
11	IO111RSB3
12	IO110RSB3
13	IO109RSB3
14	GEC0/IO108RSB3
15	GEA0/IO107RSB3
16	GEB0/IO106RSB3
17	V_{CC}
18	IO104RSB3
19	IO103RSB3
20	IO102RSB3
21	IO101RSB3
22	IO100RSB3
23	IO99RSB3
24	GND
25	V_{CCI}B3
26	IO97RSB3
27	IO98RSB3
28	IO95RSB3
29	IO96RSB3
30	IO94RSB3
31	IO93RSB3
32	IO92RSB3
33	IO91RSB2
34	FF/IO90RSB2
35	IO89RSB2
36	IO88RSB2

128-Pin VQFP	
Pin Number	AGLP030 Function
37	IO86RSB2
38	IO84RSB2
39	IO83RSB2
40	GND
41	V_{CCI}B2
42	IO82RSB2
43	IO81RSB2
44	IO79RSB2
45	IO78RSB2
46	IO77RSB2
47	IO75RSB2
48	IO74RSB2
49	V_{CC}
50	IO73RSB2
51	IO72RSB2
52	IO70RSB2
53	IO69RSB2
54	IO68RSB2
55	IO66RSB2
56	IO65RSB2
57	GND
58	V_{CCI}B2
59	IO63RSB2
60	IO61RSB2
61	IO59RSB2
62	TCK
63	TDI
64	TMS
65	V_{PUMP}
66	TDO
67	TRST
68	IO58RSB1
69	V_{JTAG}
70	IO56RSB1
71	IO57RSB1
72	V_{CCI}B1

128-Pin VQFP	
Pin Number	AGLP030 Function
73	GND
74	IO55RSB1
75	IO54RSB1
76	IO53RSB1
77	IO52RSB1
78	IO51RSB1
79	IO50RSB1
80	IO49RSB1
81	V_{CC}
82	GDB0/IO48RSB1
83	GDA0/IO47RSB1
84	GDC0/IO46RSB1
85	IO45RSB1
86	IO44RSB1
87	IO43RSB1
88	IO42RSB1
89	V_{CCI}B1
90	GND
91	IO40RSB1
92	IO41RSB1
93	IO39RSB1
94	IO38RSB1
95	IO37RSB1
96	IO36RSB1
97	IO35RSB0
98	IO34RSB0
99	IO33RSB0
100	IO32RSB0
101	IO30RSB0
102	IO28RSB0
103	IO27RSB0
104	V_{CCI}B0
105	GND
106	IO26RSB0
107	IO25RSB0
108	IO23RSB0

128-Pin VQFP	
Pin Number	**AGLP030 Function**
109	IO22RSB0
110	IO21RSB0
111	IO19RSB0
112	IO18RSB0
113	V_{CC}
114	IO17RSB0
115	IO16RSB0
116	IO14RSB0
117	IO13RSB0
118	IO12RSB0
119	IO10RSB0
120	IO09RSB0
121	V_{CCI}B0
122	GND
123	IO07RSB0
124	IO05RSB0
125	IO03RSB0
126	IO02RSB0
127	IO01RSB0
128	IO00RSB0

176-Pin VQFP



Note: This is the bottom view of the package.

Note

For Package Manufacturing and Environmental information, visit the Resource Center at http://www.actel.com/products/solutions/package/docs.aspx.

176-Pin VQFP	
Pin Number	AGLP060 Function
1	GAA2/IO156RSB3
2	IO155RSB3
3	GAB2/IO154RSB3
4	IO153RSB3
5	GAC2/IO152RSB3
6	GND
7	V_{CCI}B3
8	IO149RSB3
9	IO147RSB3
10	IO145RSB3
11	IO144RSB3
12	IO143RSB3
13	V_{CC}
14	IO141RSB3
15	GFC1/IO140RSB3
16	GFB1/IO138RSB3
17	GFB0/IO137RSB3
18	VCOMPLF
19	GFA1/IO136RSB3
20	V_{CCPLF}
21	GFA0/IO135RSB3
22	GND
23	V_{CCI}B3
24	GFA2/IO134RSB3
25	GFB2/IO133RSB3
26	GFC2/IO132RSB3
27	IO131RSB3
28	IO130RSB3
29	IO129RSB3
30	IO127RSB3
31	IO126RSB3
32	IO125RSB3
33	IO123RSB3
34	IO122RSB3
35	IO121RSB3
36	IO119RSB3

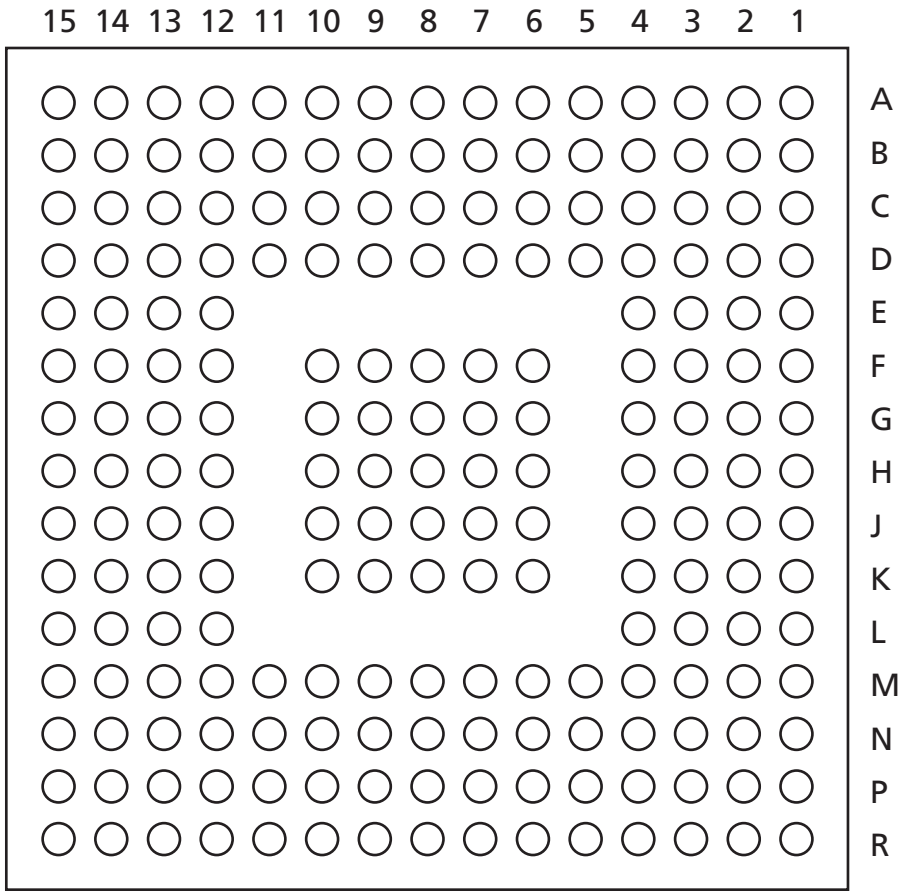
176-Pin VQFP	
Pin Number	AGLP060 Function
37	GND
38	V_{CCI}B3
39	GEC1/IO116RSB3
40	GEB1/IO114RSB3
41	GEC0/IO115RSB3
42	GEB0/IO113RSB3
43	GEA1/IO112RSB3
44	GEA0/IO111RSB3
45	GEA2/IO110RSB2
46	NC
47	FF/GEB2/IO109RSB2
48	GEC2/IO108RSB2
49	IO106RSB2
50	IO107RSB2
51	IO104RSB2
52	IO105RSB2
53	IO102RSB2
54	IO103RSB2
55	GND
56	V_{CCI}B2
57	IO101RSB2
58	IO100RSB2
59	IO99RSB2
60	IO98RSB2
61	IO97RSB2
62	IO96RSB2
63	IO95RSB2
64	IO94RSB2
65	IO93RSB2
66	V_{CC}
67	IO92RSB2
68	IO91RSB2
69	IO90RSB2
70	IO89RSB2
71	IO88RSB2

176-Pin VQFP	
Pin Number	AGLP060 Function
72	IO87RSB2
73	IO86RSB2
74	IO85RSB2
75	IO84RSB2
76	GND
77	V_{CCI}B2
78	IO83RSB2
79	IO82RSB2
80	GDC2/IO80RSB2
81	IO81RSB2
82	GDA2/IO78RSB2
83	GDB2/IO79RSB2
84	NC
85	NC
86	TCK
87	TDI
88	TMS
89	V_{PUMP}
90	TDO
91	TRST
92	V_{JTAG}
93	GDA1/IO76RSB1
94	GDC0/IO73RSB1
95	GDB1/IO74RSB1
96	GDC1/IO72RSB1
97	V_{CCI}B1
98	GND
99	IO70RSB1
100	IO69RSB1
101	IO67RSB1
102	IO66RSB1
103	IO65RSB1
104	IO63RSB1
105	IO62RSB1
106	IO61RSB1
107	GCC2/IO60RSB1

176-Pin VQFP	
Pin Number	AGLP060 Function
108	GCB2/IO59RSB1
109	GCA2/IO58RSB1
110	GCA0/IO57RSB1
111	GCA1/IO56RSB1
112	V_{CCI}B1
113	GND
114	GCB0/IO55RSB1
115	GCB1/IO54RSB1
116	GCC0/IO53RSB1
117	GCC1/IO52RSB1
118	IO51RSB1
119	IO50RSB1
120	V_{CC}
121	IO48RSB1
122	IO47RSB1
123	IO45RSB1
124	IO44RSB1
125	IO43RSB1
126	V_{CCI}B1
127	GND
128	GBC2/IO40RSB1
129	IO39RSB1
130	GBB2/IO38RSB1
131	IO37RSB1
132	GBA2/IO36RSB1
133	GBA1/IO35RSB0
134	NC
135	GBA0/IO34RSB0
136	NC
137	GBB1/IO33RSB0
138	NC
139	GBC1/IO31RSB0
140	GBB0/IO32RSB0
141	GBC0/IO30RSB0
142	IO29RSB0
143	IO28RSB0

176-Pin VQFP	
Pin Number	AGLP060 Function
144	IO27RSB0
145	V_{CCI}B0
146	GND
147	IO26RSB0
148	IO25RSB0
149	IO24RSB0
150	IO23RSB0
151	IO22RSB0
152	IO21RSB0
153	IO20RSB0
154	IO19RSB0
155	IO18RSB0
156	V_{CC}
157	IO17RSB0
158	IO16RSB0
159	IO15RSB0
160	IO14RSB0
161	IO13RSB0
162	IO12RSB0
163	IO11RSB0
164	IO10RSB0
165	IO09RSB0
166	V_{CCI}B0
167	GND
168	IO07RSB0
169	IO08RSB0
170	GAC1/IO05RSB0
171	IO06RSB0
172	GAB1/IO03RSB0
173	GAC0/IO04RSB0
174	GAB0/IO02RSB0
175	GAA1/IO01RSB0
176	GAA0/IO00RSB0

201-Pin CSP



Note: This is the bottom view of the package.

Note

For Package Manufacturing and Environmental information, visit the Resource Center at http://www.actel.com/products/solutions/package/docs.aspx.

201-Pin CSP	
Pin Number	AGLP030 Function
A1	NC
A2	IO04RSB0
A3	IO06RSB0
A4	IO09RSB0
A5	IO11RSB0
A6	IO13RSB0
A7	IO17RSB0
A8	IO18RSB0
A9	IO24RSB0
A10	IO26RSB0
A11	IO27RSB0
A12	IO31RSB0
A13	NC
A14	NC
A15	NC
B1	NC
B2	NC
B3	IO08RSB0
B4	IO05RSB0
B5	IO07RSB0
B6	IO15RSB0
B7	IO14RSB0
B8	IO16RSB0
B9	IO20RSB0
B10	IO22RSB0
B11	IO34RSB0
B12	IO29RSB0
B13	NC
B14	NC
B15	NC
C1	NC
C2	NC
C3	GND
C4	IO00RSB0
C5	IO02RSB0
C6	IO12RSB0

201-Pin CSP	
Pin Number	AGLP030 Function
C7	IO23RSB0
C8	IO19RSB0
C9	IO28RSB0
C10	IO32RSB0
C11	IO35RSB0
C12	NC
C13	GND
C14	IO41RSB1
C15	IO37RSB1
D1	IO117RSB3
D2	IO118RSB3
D3	NC
D4	GND
D5	IO01RSB0
D6	IO03RSB0
D7	IO10RSB0
D8	IO21RSB0
D9	IO25RSB0
D10	IO30RSB0
D11	IO33RSB0
D12	GND
D13	NC
D14	IO36RSB1
D15	IO39RSB1
E1	IO115RSB3
E2	IO114RSB3
E3	NC
E4	NC
E12	NC
E13	NC
E14	GDC0/IO46RSB1
E15	GDB0/IO48RSB1
F1	IO113RSB3
F2	IO116RSB3
F3	IO119RSB3
F4	IO111RSB3

201-Pin CSP	
Pin Number	AGLP030 Function
F6	GND
F7	V_{CC}
F8	V_{CCI}B0
F9	V_{CCI}B0
F10	V_{CCI}B0
F12	NC
F13	NC
F14	IO40RSB1
F15	IO38RSB1
G1	NC
G2	IO112RSB3
G3	IO110RSB3
G4	IO109RSB3
G6	V_{CCI}B3
G7	GND
G8	V_{CC}
G9	GND
G10	GND
G12	NC
G13	NC
G14	IO42RSB1
G15	IO44RSB1
H1	NC
H2	GEB0/IO106RSB3
H3	GEC0/IO108RSB3
H4	NC
H6	V_{CCI}B3
H7	GND
H8	V_{CC}
H9	GND
H10	V_{CCI}B1
H12	IO54RSB1
H13	GDA0/IO47RSB1
H14	IO45RSB1
H15	IO43RSB1
J1	GEA0/IO107RSB3

201-Pin CSP	
Pin Number	AGLP030 Function
J2	IO105RSB3
J3	IO104RSB3
J4	IO102RSB3
J6	V_{CCI}B3
J7	GND
J8	V_{CC}
J9	GND
J10	V_{CCI}B1
J12	NC
J13	NC
J14	IO52RSB1
J15	IO50RSB1
K1	IO103RSB3
K2	IO101RSB3
K3	IO99RSB3
K4	IO100RSB3
K6	GND
K7	V_{CCI}B2
K8	V_{CCI}B2
K9	V_{CCI}B2
K10	V_{CCI}B1
K12	NC
K13	IO57RSB1
K14	IO49RSB1
K15	IO53RSB1
L1	IO96RSB3
L2	IO98RSB3
L3	IO95RSB3
L4	IO94RSB3
L12	NC
L13	NC
L14	IO51RSB1
L15	IO58RSB1
M1	IO93RSB3
M2	IO92RSB3
M3	IO97RSB3

201-Pin CSP	
Pin Number	AGLP030 Function
M4	GND
M5	NC
M6	IO79RSB2
M7	IO77RSB2
M8	IO72RSB2
M9	IO70RSB2
M10	IO61RSB2
M11	IO59RSB2
M12	GND
M13	NC
M14	IO55RSB1
M15	IO56RSB1
N1	NC
N2	NC
N3	GND
N4	NC
N5	IO88RSB2
N6	IO81RSB2
N7	IO75RSB2
N8	IO68RSB2
N9	IO66RSB2
N10	IO65RSB2
N11	IO71RSB2
N12	IO63RSB2
N13	GND
N14	TDO
N15	V_{JTAG}
P1	NC
P2	NC
P3	NC
P4	NC
P5	IO87RSB2
P6	IO86RSB2
P7	IO84RSB2
P8	IO80RSB2
P9	IO74RSB2

201-Pin CSP	
Pin Number	AGLP030 Function
P10	IO73RSB2
P11	IO76RSB2
P12	IO67RSB2
P13	IO64RSB2
P14	V_{PUMP}
P15	TRST
R1	NC
R2	NC
R3	IO91RSB2
R4	FF/IO90RSB2
R5	IO89RSB2
R6	IO83RSB2
R7	IO82RSB2
R8	IO85RSB2
R9	IO78RSB2
R10	IO69RSB2
R11	IO62RSB2
R12	IO60RSB2
R13	TMS
R14	TDI
R15	TCK

201-Pin CSP	
Pin Number	AGLP060 Function
A1	IO150RSB3
A2	GAA0/IO00RSB0
A3	GAC0/IO04RSB0
A4	IO08RSB0
A5	IO11RSB0
A6	IO15RSB0
A7	IO17RSB0
A8	IO18RSB0
A9	IO22RSB0
A10	IO26RSB0
A11	IO29RSB0
A12	GBC1/IO31RSB0
A13	GBA2/IO36RSB1
A14	IO41RSB1
A15	NC
B1	IO151RSB3
B2	GAB2/IO154RSB3
B3	IO06RSB0
B4	IO09RSB0
B5	IO13RSB0
B6	IO10RSB0
B7	IO12RSB0
B8	IO20RSB0
B9	IO23RSB0
B10	IO25RSB0
B11	IO24RSB0
B12	IO27RSB0
B13	IO37RSB1
B14	IO46RSB1
B15	IO42RSB1
C1	IO155RSB3
C2	GAA2/IO156RSB3
C3	GND
C4	GAA1/IO01RSB0
C5	GAB1/IO03RSB0
C6	IO07RSB0

201-Pin CSP	
Pin Number	AGLP060 Function
C7	IO16RSB0
C8	IO21RSB0
C9	IO28RSB0
C10	GBB1/IO33RSB0
C11	GBA1/IO35RSB0
C12	GBB2/IO38RSB1
C13	GND
C14	IO48RSB1
C15	IO39RSB1
D1	IO146RSB3
D2	IO144RSB3
D3	IO148RSB3
D4	GND
D5	GAB0/IO02RSB0
D6	GAC1/IO05RSB0
D7	IO14RSB0
D8	IO19RSB0
D9	GBC0/IO30RSB0
D10	GBB0/IO32RSB0
D11	GBA0/IO34RSB0
D12	GND
D13	GBC2/IO40RSB1
D14	IO51RSB1
D15	IO44RSB1
E1	IO142RSB3
E2	IO149RSB3
E3	IO153RSB3
E4	GAC2/IO152RSB3
E12	IO43RSB1
E13	IO49RSB1
E14	GCC0/IO53RSB1
E15	GCB0/IO55RSB1
F1	IO141RSB3
F2	GFC1/IO140RSB3
F3	IO145RSB3
F4	IO147RSB3

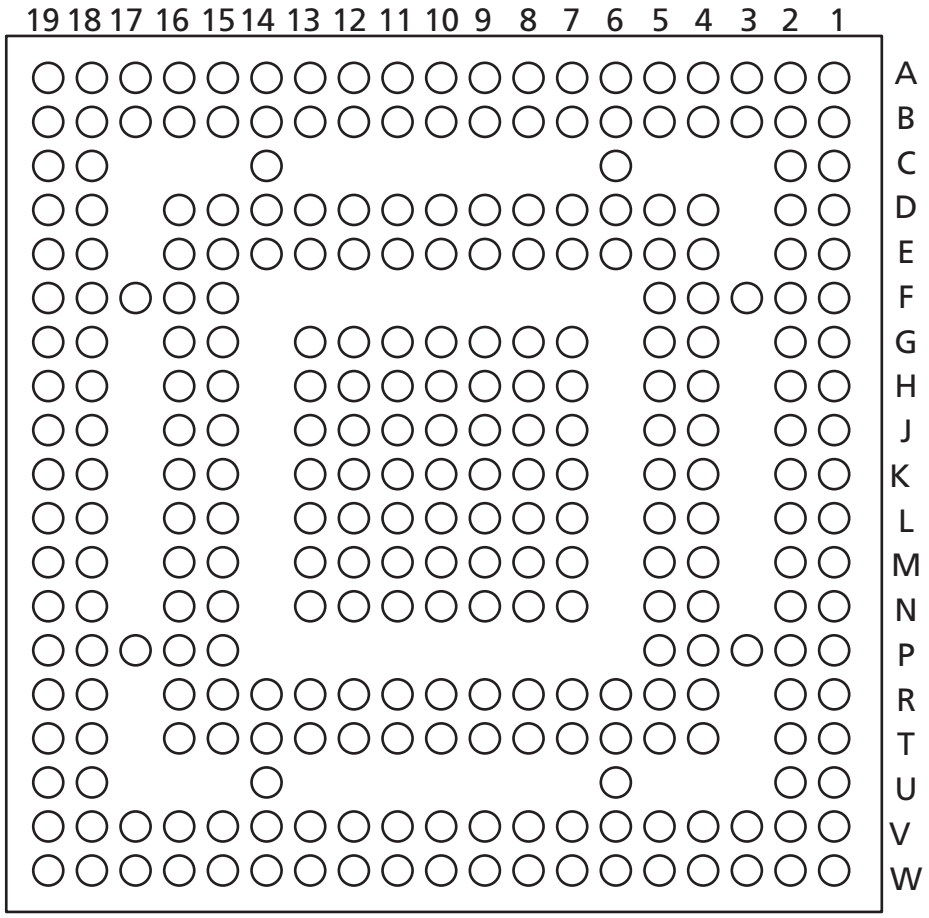
201-Pin CSP	
Pin Number	AGLP060 Function
F6	GND
F7	V_{CC}
F8	V_{CCI}B0
F9	V_{CCI}B0
F10	V_{CCI}B0
F12	IO47RSB1
F13	IO45RSB1
F14	GCC1/IO52RSB1
F15	GCA1/IO56RSB1
G1	VCOMPLF
G2	GFB0/IO137RSB3
G3	GFC0/IO139RSB3
G4	IO143RSB3
G6	V_{CCI}B3
G7	GND
G8	V_{CC}
G9	GND
G10	GND
G12	IO50RSB1
G13	GCB1/IO54RSB1
G14	GCC2/IO60RSB1
G15	GCA2/IO58RSB1
H1	VCCPLF
H2	GFA1/IO136RSB3
H3	GFB1/IO138RSB3
H4	NC
H6	V_{CCI}B3
H7	GND
H8	V_{CC}
H9	GND
H10	V_{CCI}B1
H12	GCB2/IO59RSB1
H13	GCA0/IO57RSB1
H14	IO64RSB1
H15	IO62RSB1
J1	GFA2/IO134RSB3

201-Pin CSP	
Pin Number	AGLP060 Function
J2	GFA0/IO135RSB3
J3	GFB2/IO133RSB3
J4	IO131RSB3
J6	V_{CCI}B3
J7	GND
J8	V_{CC}
J9	GND
J10	V_{CCI}B1
J12	IO61RSB1
J13	IO63RSB1
J14	IO68RSB1
J15	IO66RSB1
K1	IO130RSB3
K2	GFC2/IO132RSB3
K3	IO127RSB3
K4	IO129RSB3
K6	GND
K7	V_{CCI}B2
K8	V_{CCI}B2
K9	V_{CCI}B2
K10	V_{CCI}B1
K12	IO65RSB1
K13	IO67RSB1
K14	IO69RSB1
K15	IO70RSB1
L1	IO126RSB3
L2	IO128RSB3
L3	IO121RSB3
L4	IO123RSB3
L12	GDB1/IO74RSB1
L13	GDC1/IO72RSB1
L14	IO71RSB1
L15	GDC0/IO73RSB1
M1	IO122RSB3
M2	IO124RSB3
M3	IO119RSB3

201-Pin CSP	
Pin Number	AGLP060 Function
M4	GND
M5	IO125RSB3
M6	IO98RSB2
M7	IO96RSB2
M8	IO91RSB2
M9	IO89RSB2
M10	IO82RSB2
M11	GDA2/IO78RSB2
M12	GND
M13	GDA1/IO76RSB1
M14	GDA0/IO77RSB1
M15	GDB0/IO75RSB1
N1	IO117RSB3
N2	IO120RSB3
N3	GND
N4	GEB1/IO114RSB3
N5	IO107RSB2
N6	IO100RSB2
N7	IO94RSB2
N8	IO87RSB2
N9	IO85RSB2
N10	GDC2/IO80RSB2
N11	IO90RSB2
N12	IO84RSB2
N13	GND
N14	TDO
N15	V_{JTAG}
P1	GEC0/IO115RSB3
P2	GEC1/IO116RSB3
P3	GEA0/IO111RSB3
P4	GEA1/IO112RSB3
P5	IO106RSB2
P6	IO105RSB2
P7	IO103RSB2
P8	IO99RSB2
P9	IO93RSB2

201-Pin CSP	
Pin Number	AGLP060 Function
P10	IO92RSB2
P11	IO95RSB2
P12	IO86RSB2
P13	IO83RSB2
P14	V_{PUMP}
P15	TRST
R1	IO118RSB3
R2	GEB0/IO113RSB3
R3	GEA2/IO110RSB2
R4	FF/GEB2/IO109RSB2
R5	GEC2/IO108RSB2
R6	IO102RSB2
R7	IO101RSB2
R8	IO104RSB2
R9	IO97RSB2
R10	IO88RSB2
R11	IO81RSB2
R12	GDB2/IO79RSB2
R13	TMS
R14	TDI
R15	TCK

281-Pin CSP



Note: This is the bottom view of the package.

Note

For Package Manufacturing and Environmental information, visit the Resource Center at http://www.actel.com/products/solutions/package/docs.aspx

281-Pin CSP	
Pin Number	AGLP125 Function
A1	GND
A2	GAB0/IO02RSB0
A3	GAC1/IO05RSB0
A4	IO09RSB0
A5	IO13RSB0
A6	IO15RSB0
A7	IO18RSB0
A8	IO23RSB0
A9	IO25RSB0
A10	V_{CCI}B0
A11	IO33RSB0
A12	IO41RSB0
A13	IO43RSB0
A14	IO46RSB0
A15	IO55RSB0
A16	IO56RSB0
A17	GBC1/IO58RSB0
A18	GBA0/IO61RSB0
A19	GND
B1	GAA2/IO211RSB3
B2	V_{CCI}B0
B3	GAB1/IO03RSB0
B4	GAC0/IO04RSB0
B5	IO11RSB0
B6	GND
B7	IO21RSB0
B8	IO22RSB0
B9	IO28RSB0
B10	IO32RSB0
B11	IO36RSB0
B12	IO39RSB0
B13	IO42RSB0
B14	GND
B15	IO52RSB0
B16	GBC0/IO57RSB0
B17	GBA1/IO62RSB0

281-Pin CSP	
Pin Number	AGLP125 Function
B18	V_{CCI}B1
B19	IO64RSB1
C1	GAB2/IO209RSB3
C2	IO210RSB3
C6	IO12RSB0
C14	IO47RSB0
C18	IO54RSB0
C19	GBB2/IO65RSB1
D1	IO206RSB3
D2	IO208RSB3
D4	GAA0/IO00RSB0
D5	GAA1/IO01RSB0
D6	IO10RSB0
D7	IO17RSB0
D8	IO24RSB0
D9	IO27RSB0
D10	GND
D11	IO31RSB0
D12	IO40RSB0
D13	IO49RSB0
D14	IO45RSB0
D15	GBB0/IO59RSB0
D16	GBA2/IO63RSB1
D18	GBC2/IO67RSB1
D19	IO66RSB1
E1	IO203RSB3
E2	IO205RSB3
E4	IO07RSB0
E5	IO06RSB0
E6	IO14RSB0
E7	IO20RSB0
E8	IO29RSB0
E9	IO34RSB0
E10	IO30RSB0
E11	IO37RSB0
E12	IO38RSB0

281-Pin CSP	
Pin Number	AGLP125 Function
E13	IO48RSB0
E14	GBB1/IO60RSB0
E15	IO53RSB0
E16	IO69RSB1
E18	IO68RSB1
E19	IO71RSB1
F1	IO198RSB3
F2	GND
F3	IO201RSB3
F4	IO204RSB3
F5	IO16RSB0
F15	IO50RSB0
F16	IO74RSB1
F17	IO72RSB1
F18	GND
F19	IO73RSB1
G1	IO195RSB3
G2	IO200RSB3
G4	IO202RSB3
G5	IO08RSB0
G7	GAC2/IO207RSB3
G8	V_{CCI}B0
G9	IO26RSB0
G10	IO35RSB0
G11	IO44RSB0
G12	V_{CCI}B0
G13	IO51RSB0
G15	IO70RSB1
G16	IO75RSB1
G18	GCC0/IO80RSB1
G19	GCB1/IO81RSB1
H1	GFB0/IO191RSB3
H2	IO196RSB3
H4	GFC1/IO194RSB3
H5	GFB1/IO192RSB3
H7	V_{CCI}B3

281-Pin CSP	
Pin Number	**AGLP125 Function**
H8	V_{CC}
H9	V_{CCI}B0
H10	VCC
H11	V_{CCI}B0
H12	V_{CC}
H13	V_{CCI}B1
H15	IO77RSB1
H16	GCB0/IO82RSB1
H18	GCA1/IO83RSB1
H19	GCA2/IO85RSB1
J1	VCOMPLF
J2	GFA0/IO189RSB3
J4	V_{CCPLF}
J5	GFC0/IO193RSB3
J7	GFA2/IO188RSB3
J8	V_{CCI}B3
J9	GND
J10	GND
J11	GND
J12	V_{CCI}B1
J13	GCC1/IO79RSB1
J15	GCA0/IO84RSB1
J16	GCB2/IO86RSB1
J18	IO76RSB1
J19	IO78RSB1
K1	V_{CCI}B3
K2	GFA1/IO190RSB3
K4	GND
K5	IO19RSB0
K7	IO197RSB3
K8	V_{CC}
K9	GND
K10	GND
K11	GND
K12	V_{CC}
K13	GCC2/IO87RSB1

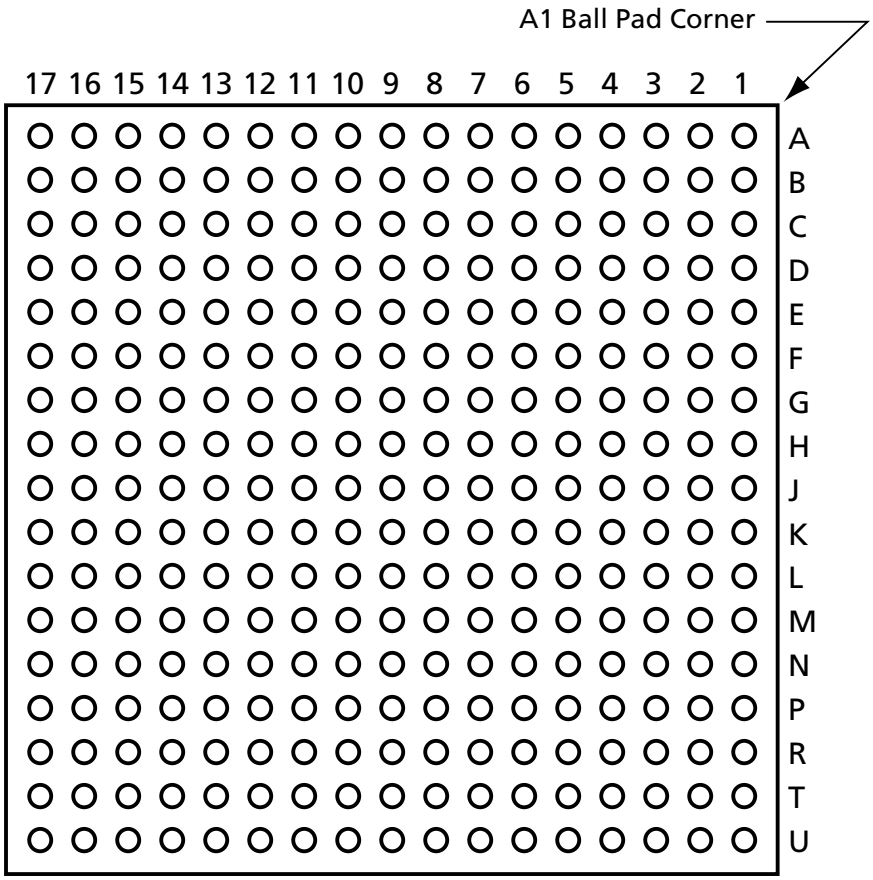
281-Pin CSP	
Pin Number	**AGLP125 Function**
K15	IO89RSB1
K16	GND
K18	IO88RSB1
K19	V_{CCI}B1
L1	GFB2/IO187RSB3
L2	IO185RSB3
L4	GFC2/IO186RSB3
L5	IO184RSB3
L7	IO199RSB3
L8	V_{CCI}B3
L9	GND
L10	GND
L11	GND
L12	V_{CCI}B1
L13	IO95RSB1
L15	IO91RSB1
L16	NC
L18	IO90RSB1
L19	NC
M1	IO180RSB3
M2	IO179RSB3
M4	IO181RSB3
M5	IO183RSB3
M7	V_{CCI}B3
M8	V_{CC}
M9	V_{CCI}B2
M10	V_{CC}
M11	V_{CCI}B2
M12	V_{CC}
M13	V_{CCI}B1
M15	IO122RSB2
M16	IO93RSB1
M18	IO92RSB1
M19	NC
N1	IO178RSB3
N2	IO175RSB3

281-Pin CSP	
Pin Number	**AGLP125 Function**
N4	IO182RSB3
N5	IO161RSB2
N7	GEA2/IO164RSB2
N8	V_{CCI}B2
N9	IO137RSB2
N10	IO135RSB2
N11	IO131RSB2
N12	V_{CCI}B2
N13	V_{PUMP}
N15	IO117RSB2
N16	IO96RSB1
N18	IO98RSB1
N19	IO94RSB1
P1	IO174RSB3
P2	GND
P3	IO176RSB3
P4	IO177RSB3
P5	GEA0/IO165RSB3
P15	IO111RSB2
P16	IO108RSB2
P17	GDC1/IO99RSB1
P18	GND
P19	IO97RSB1
R1	IO173RSB3
R2	IO172RSB3
R4	GEC1/IO170RSB3
R5	GEB1/IO168RSB3
R6	IO154RSB2
R7	IO149RSB2
R8	IO146RSB2
R9	IO138RSB2
R10	IO134RSB2
R11	IO132RSB2
R12	IO130RSB2
R13	IO118RSB2
R14	IO112RSB2

281-Pin CSP	
Pin Number	AGLP125 Function
R15	IO109RSB2
R16	GDA1/IO103RSB1
R18	GDB0/IO102RSB1
R19	GDC0/IO100RSB1
T1	IO171RSB3
T2	GEC0/IO169RSB3
T4	GEB0/IO167RSB3
T5	IO157RSB2
T6	IO158RSB2
T7	IO148RSB2
T8	IO145RSB2
T9	IO143RSB2
T10	GND
T11	IO129RSB2
T12	IO126RSB2
T13	IO125RSB2
T14	IO116RSB2
T15	GDC2/IO107RSB2
T16	TMS
T18	V_{JTAG}
T19	GDB1/IO101RSB1
U1	IO160RSB2
U2	GEA1/IO166RSB3
U6	IO151RSB2
U14	IO121RSB2
U18	TRST
U19	GDA0/IO104RSB1
V1	IO159RSB2
V2	V_{CCI}B3
V3	GEC2/IO162RSB2
V4	IO156RSB2
V5	IO153RSB2
V6	GND
V7	IO144RSB2
V8	IO141RSB2
V9	IO140RSB2

281-Pin CSP	
Pin Number	AGLP125 Function
V10	IO133RSB2
V11	IO127RSB2
V12	IO123RSB2
V13	IO120RSB2
V14	GND
V15	IO113RSB2
V16	GDA2/IO105RSB2
V17	TDI
V18	V_{CCI}B2
V19	TDO
W1	GND
W2	FF/GEB2/IO163RSB2
W3	IO155RSB2
W4	IO152RSB2
W5	IO150RSB2
W6	IO147RSB2
W7	IO142RSB2
W8	IO139RSB2
W9	IO136RSB2
W10	V_{CCI}B2
W11	IO128RSB2
W12	IO124RSB2
W13	IO119RSB2
W14	IO115RSB2
W15	IO114RSB2
W16	IO110RSB2
W17	GDB2/IO106RSB2
W18	TCK
W19	GND

289-Pin CSP



Note: This is the bottom view of the package.

Note

For Package Manufacturing and Environmental information, visit the Resource Center at http://www.actel.com/products/solutions/package/docs.aspx .

289-Pin CSP	
Pin Number	AGLP030 Function
A1	IO03RSB0
A2	NC
A3	NC
A4	GND
A5	IO10RSB0
A6	IO14RSB0
A7	IO16RSB0
A8	IO18RSB0
A9	GND
A10	IO23RSB0
A11	IO27RSB0
A12	NC
A13	NC
A14	GND
A15	NC
A16	NC
A17	IO30RSB0
B1	IO01RSB0
B2	GND
B3	NC
B4	NC
B5	IO07RSB0
B6	NC
B7	V_{CCI}B0
B8	IO17RSB0
B9	IO19RSB0
B10	IO24RSB0
B11	IO28RSB0
B12	V_{CCI}B0
B13	NC
B14	NC
B15	NC
B16	IO31RSB0
B17	GND
C1	NC
C2	IO00RSB0
C3	IO04RSB0
C4	NC

289-Pin CSP	
Pin Number	AGLP030 Function
C5	V_{CCI}B0
C6	IO09RSB0
C7	IO13RSB0
C8	IO15RSB0
C9	IO21RSB0
C10	GND
C11	IO29RSB0
C12	NC
C13	NC
C14	NC
C15	GND
C16	IO34RSB0
C17	NC
D1	NC
D2	IO119RSB3
D3	GND
D4	IO02RSB0
D5	NC
D6	NC
D7	NC
D8	GND
D9	IO20RSB0
D10	IO25RSB0
D11	NC
D12	NC
D13	GND
D14	IO32RSB0
D15	IO35RSB0
D16	NC
D17	NC
E1	V_{CCI}B3
E2	IO114RSB3
E3	IO115RSB3
E4	IO118RSB3
E5	IO05RSB0
E6	NC
E7	IO06RSB0
E8	IO11RSB0

289-Pin CSP	
Pin Number	AGLP030 Function
E9	IO22RSB0
E10	IO26RSB0
E11	V_{CCI}B0
E12	NC
E13	IO33RSB0
E14	IO36RSB1
E15	IO38RSB1
E16	V_{CCI}B1
E17	NC
F1	IO111RSB3
F2	NC
F3	IO116RSB3
F4	V_{CCI}B3
F5	IO117RSB3
F6	NC
F7	NC
F8	IO08RSB0
F9	IO12RSB0
F10	NC
F11	NC
F12	NC
F13	NC
F14	GND
F15	NC
F16	IO37RSB1
F17	IO41RSB1
G1	IO110RSB3
G2	GND
G3	IO113RSB3
G4	NC
G5	NC
G6	NC
G7	GND
G8	GND
G9	V_{CC}
G10	GND
G11	GND
G12	IO40RSB1

289-Pin CSP	
Pin Number	AGLP030 Function
G13	NC
G14	IO39RSB1
G15	IO44RSB1
G16	NC
G17	GND
H1	NC
H2	GEC0/IO108RSB3
H3	NC
H4	IO112RSB3
H5	NC
H6	IO109RSB3
H7	GND
H8	GND
H9	GND
H10	GND
H11	GND
H12	NC
H13	NC
H14	IO45RSB1
H15	V_{CCI}B1
H16	GDB0/IO48RSB1
H17	IO42RSB1
J1	NC
J2	GEA0/IO107RSB3
J3	V_{CCI}B3
J4	IO105RSB3
J5	NC
J6	NC
J7	V_{CC}
J8	GND
J9	GND
J10	GND
J11	V_{CC}
J12	IO50RSB1
J13	IO43RSB1
J14	IO51RSB1
J15	IO52RSB1
J16	GDC0/IO46RSB1

289-Pin CSP	
Pin Number	AGLP030 Function
J17	GDA0/IO47RSB1
K1	GND
K2	GEB0/IO106RSB3
K3	IO102RSB3
K4	IO104RSB3
K5	IO99RSB3
K6	NC
K7	GND
K8	GND
K9	GND
K10	GND
K11	GND
K12	NC
K13	NC
K14	NC
K15	IO53RSB1
K16	GND
K17	IO49RSB1
L1	IO103RSB3
L2	IO101RSB3
L3	NC
L4	GND
L5	NC
L6	NC
L7	GND
L8	GND
L9	V_{CC}
L10	GND
L11	GND
L12	IO58RSB1
L13	IO54RSB1
L14	V_{CCI}B1
L15	NC
L16	NC
L17	NC
M1	NC
M2	V_{CCI}B3
M3	IO100RSB3

289-Pin CSP	
Pin Number	AGLP030 Function
M4	IO98RSB3
M5	IO93RSB3
M6	IO97RSB3
M7	NC
M8	NC
M9	IO71RSB2
M10	NC
M11	IO63RSB2
M12	NC
M13	IO57RSB1
M14	NC
M15	NC
M16	NC
M17	V_{CCI}B1
N1	NC
N2	NC
N3	IO95RSB3
N4	IO96RSB3
N5	GND
N6	NC
N7	IO85RSB2
N8	IO79RSB2
N9	IO77RSB2
N10	V_{CCI}B2
N11	NC
N12	NC
N13	IO59RSB2
N14	NC
N15	GND
N16	IO56RSB1
N17	IO55RSB1
P1	IO94RSB3
P2	NC
P3	GND
P4	NC
P5	NC
P6	IO87RSB2
P7	IO80RSB2

289-Pin CSP	
Pin Number	AGLP030 Function
P8	GND
P9	IO72RSB2
P10	IO67RSB2
P11	IO61RSB2
P12	NC
P13	V_{CCI}B2
P14	NC
P15	IO60RSB2
P16	IO62RSB2
P17	V_{JTAG}
R1	GND
R2	IO91RSB2
R3	NC
R4	NC
R5	NC
R6	V_{CCI}B2
R7	IO83RSB2
R8	IO78RSB2
R9	IO74RSB2
R10	IO70RSB2
R11	GND
R12	NC
R13	NC
R14	NC
R15	NC
R16	TMS
R17	TRST
T1	IO92RSB3
T2	IO89RSB2
T3	NC
T4	GND
T5	NC
T6	IO84RSB2
T7	IO81RSB2
T8	IO76RSB2
T9	V_{CCI}B2
T10	IO69RSB2
T11	IO65RSB2

289-Pin CSP	
Pin Number	AGLP030 Function
T12	IO64RSB2
T13	NC
T14	GND
T15	NC
T16	TDI
T17	TDO
U1	FF/IO90RSB2
U2	GND
U3	NC
U4	IO88RSB2
U5	IO86RSB2
U6	IO82RSB2
U7	GND
U8	IO75RSB2
U9	IO73RSB2
U10	IO68RSB2
U11	IO66RSB2
U12	GND
U13	NC
U14	NC
U15	NC
U16	TCK
U17	V_{PUMP}

289-Pin CSP	
Pin Number	AGLP060 Function
A1	GAB1/IO03RSB0
A2	NC
A3	NC
A4	GND
A5	IO10RSB0
A6	IO14RSB0
A7	IO16RSB0
A8	IO18RSB0
A9	GND
A10	IO23RSB0
A11	IO27RSB0
A12	NC
A13	NC
A14	GND
A15	NC
A16	NC
A17	GBC0/IO30RSB0
B1	GAA1/IO01RSB0
B2	GND
B3	NC
B4	NC
B5	IO07RSB0
B6	NC
B7	V_{CCI}B0
B8	IO17RSB0
B9	IO19RSB0
B10	IO24RSB0
B11	IO28RSB0
B12	VCCIB0
B13	NC
B14	NC
B15	NC
B16	GBC1/IO31RSB0
B17	GND
C1	IO155RSB3
C2	GAA0/IO00RSB0
C3	GAC0/IO04RSB0
C4	NC

289-Pin CSP	
Pin Number	AGLP060 Function
C5	V_{CCI}B0
C6	IO09RSB0
C7	IO13RSB0
C8	IO15RSB0
C9	IO21RSB0
C10	GND
C11	IO29RSB0
C12	NC
C13	NC
C14	NC
C15	GND
C16	GBA0/IO34RSB0
C17	IO39RSB1
D1	IO150RSB3
D2	IO151RSB3
D3	GND
D4	GAB0/IO02RSB0
D5	NC
D6	NC
D7	NC
D8	GND
D9	IO20RSB0
D10	IO25RSB0
D11	NC
D12	NC
D13	GND
D14	GBB0/IO32RSB0
D15	GBA1/IO35RSB0
D16	IO37RSB1
D17	IO42RSB1
E1	V_{CCI}B3
E2	IO147RSB3
E3	GAC2/IO152RSB3
E4	GAA2/IO156RSB3
E5	GAC1/IO05RSB0
E6	NC
E7	IO06RSB0
E8	IO11RSB0

289-Pin CSP	
Pin Number	AGLP060 Function
E9	IO22RSB0
E10	IO26RSB0
E11	V_{CCI}B0
E12	NC
E13	GBB1/IO33RSB0
E14	GBA2/IO36RSB1
E15	GBB2/IO38RSB1
E16	V_{CCI}B1
E17	IO44RSB1
F1	GFC1/IO140RSB3
F2	IO142RSB3
F3	IO149RSB3
F4	V_{CCI}B3
F5	GAB2/IO154RSB3
F6	IO153RSB3
F7	NC
F8	IO08RSB0
F9	IO12RSB0
F10	NC
F11	NC
F12	NC
F13	GBC2/IO40RSB1
F14	GND
F15	IO43RSB1
F16	IO46RSB1
F17	IO45RSB1
G1	GFC0/IO139RSB3
G2	GND
G3	IO144RSB3
G4	IO145RSB3
G5	IO146RSB3
G6	IO148RSB3
G7	GND
G8	GND
G9	VCC
G10	GND
G11	GND
G12	IO48RSB1

289-Pin CSP	
Pin Number	AGLP060 Function
G13	IO41RSB1
G14	IO47RSB1
G15	IO49RSB1
G16	IO50RSB1
G17	GND
H1	V_{COMPLF}
H2	GFB0/IO137RSB3
H3	NC
H4	IO141RSB3
H5	IO143RSB3
H6	GFB1/IO138RSB3
H7	GND
H8	GND
H9	GND
H10	GND
H11	GND
H12	GCC1/IO52RSB1
H13	IO51RSB1
H14	GCA0/IO57RSB1
H15	V_{CCI}B1
H16	GCA2/IO58RSB1
H17	GCC0/IO53RSB1
J1	V_{CCPLF}
J2	GFA1/IO136RSB3
J3	V_{CCI}B3
J4	IO131RSB3
J5	IO130RSB3
J6	IO129RSB3
J7	V_{CC}
J8	GND
J9	GND
J10	GND
J11	V_{CC}
J12	GCB2/IO59RSB1
J13	GCB1/IO54RSB1
J14	IO62RSB1
J15	IO63RSB1
J16	GCB0/IO55RSB1

289-Pin CSP	
Pin Number	AGLP060 Function
J17	GCA1/IO56RSB1
K1	GND
K2	GFA0/IO135RSB3
K3	GFB2/IO133RSB3
K4	IO128RSB3
K5	IO123RSB3
K6	IO125RSB3
K7	GND
K8	GND
K9	GND
K10	GND
K11	GND
K12	IO64RSB1
K13	IO61RSB1
K14	IO66RSB1
K15	IO65RSB1
K16	GND
K17	GCC2/IO60RSB1
L1	GFA2/IO134RSB3
L2	GFC2/IO132RSB3
L3	IO127RSB3
L4	GND
L5	IO121RSB3
L6	GEC1/IO116RSB3
L7	GND
L8	GND
L9	V_{CC}
L10	GND
L11	GND
L12	GDC1/IO72RSB1
L13	GDB1/IO74RSB1
L14	V_{CCI}B1
L15	IO70RSB1
L16	IO68RSB1
L17	IO67RSB1
M1	IO126RSB3
M2	V_{CCI}B3
M3	IO124RSB3

289-Pin CSP	
Pin Number	AGLP060 Function
M4	IO122RSB3
M5	GEB0/IO113RSB3
M6	GEB1/IO114RSB3
M7	NC
M8	NC
M9	IO90RSB2
M10	NC
M11	IO83RSB2
M12	NC
M13	GDA1/IO76RSB1
M14	GDA0/IO77RSB1
M15	IO71RSB1
M16	IO69RSB1
M17	V_{CCI}B1
N1	IO119RSB3
N2	IO120RSB3
N3	GEC0/IO115RSB3
N4	GEA0/IO111RSB3
N5	GND
N6	NC
N7	IO104RSB2
N8	IO98RSB2
N9	IO96RSB2
N10	V_{CCI}B2
N11	NC
N12	NC
N13	GDB2/IO79RSB2
N14	NC
N15	GND
N16	GDB0/IO75RSB1
N17	GDC0/IO73RSB1
P1	IO118RSB3
P2	IO117RSB3
P3	GND
P4	NC
P5	NC
P6	IO106RSB2
P7	IO99RSB2

289-Pin CSP	
Pin Number	AGLP060 Function
P8	GND
P9	IO91RSB2
P10	IO86RSB2
P11	IO81RSB2
P12	NC
P13	V_{CCI}B2
P14	NC
P15	GDA2/IO78RSB2
P16	GDC2/IO80RSB2
P17	V_{JTAG}
R1	GND
R2	GEA2/IO110RSB2
R3	NC
R4	NC
R5	NC
R6	V_{CCI}B2
R7	IO102RSB2
R8	IO97RSB2
R9	IO93RSB2
R10	IO89RSB2
R11	GND
R12	NC
R13	NC
R14	NC
R15	NC
R16	TMS
R17	TRST
T1	GEA1/IO112RSB3
T2	GEC2/IO108RSB2
T3	NC
T4	GND
T5	NC
T6	IO103RSB2
T7	IO100RSB2
T8	IO95RSB2
T9	V_{CCI}B2
T10	IO88RSB2
T11	IO84RSB2

289-Pin CSP	
Pin Number	AGLP060 Function
T12	IO82RSB2
T13	NC
T14	GND
T15	NC
T16	TDI
T17	TDO
U1	FF/GEB2/IO109RSB2
U2	GND
U3	NC
U4	IO107RSB2
U5	IO105RSB2
U6	IO101RSB2
U7	GND
U8	IO94RSB2
U9	IO92RSB2
U10	IO87RSB2
U11	IO85RSB2
U12	GND
U13	NC
U14	NC
U15	NC
U16	TCK
U17	V_{PUMP}

289-Pin CSP	
Pin Number	AGLP125 Function
A1	GAB1/IO03RSB0
A2	IO11RSB0
A3	IO08RSB0
A4	GND
A5	IO19RSB0
A6	IO24RSB0
A7	IO26RSB0
A8	IO30RSB0
A9	GND
A10	IO35RSB0
A11	IO38RSB0
A12	IO40RSB0
A13	IO42RSB0
A14	GND
A15	IO48RSB0
A16	IO54RSB0
A17	GBC0/IO57RSB0
B1	GAA1/IO01RSB0
B2	GND
B3	IO06RSB0
B4	IO13RSB0
B5	IO15RSB0
B6	IO21RSB0
B7	V_{CCI}B0
B8	IO28RSB0
B9	IO31RSB0
B10	IO37RSB0
B11	IO39RSB0
B12	V_{CCI}B0
B13	IO44RSB0
B14	IO46RSB0
B15	IO49RSB0
B16	GBC1/IO58RSB0
B17	GND
C1	IO210RSB3
C2	GAA0/IO00RSB0
C3	GAC0/IO04RSB0
C4	IO09RSB0

289-Pin CSP	
Pin Number	AGLP125 Function
C5	V_{CCI}B0
C6	IO17RSB0
C7	IO23RSB0
C8	IO27RSB0
C9	IO33RSB0
C10	GND
C11	IO43RSB0
C12	IO45RSB0
C13	IO50RSB0
C14	IO52RSB0
C15	GND
C16	GBA0/IO61RSB0
C17	IO68RSB1
D1	IO204RSB3
D2	IO205RSB3
D3	GND
D4	GAB0/IO02RSB0
D5	IO07RSB0
D6	IO10RSB0
D7	IO18RSB0
D8	GND
D9	IO34RSB0
D10	IO41RSB0
D11	IO47RSB0
D12	IO55RSB0
D13	GND
D14	GBB0/IO59RSB0
D15	GBA1/IO62RSB0
D16	IO66RSB1
D17	IO70RSB1
E1	V_{CCI}B3
E2	IO200RSB3
E3	GAC2/IO207RSB3
E4	GAA2/IO211RSB3
E5	GAC1/IO05RSB0
E6	IO12RSB0
E7	IO16RSB0
E8	IO22RSB0

289-Pin CSP	
Pin Number	AGLP125 Function
E9	IO32RSB0
E10	IO36RSB0
E11	V_{CCI}B0
E12	IO56RSB0
E13	GBB1/IO60RSB0
E14	GBA2/IO63RSB1
E15	GBB2/IO65RSB1
E16	V_{CCI}B1
E17	IO73RSB1
F1	GFC1/IO194RSB3
F2	IO196RSB3
F3	IO202RSB3
F4	V_{CCI}B3
F5	GAB2/IO209RSB3
F6	IO208RSB3
F7	IO14RSB0
F8	IO20RSB0
F9	IO25RSB0
F10	IO29RSB0
F11	IO51RSB0
F12	IO53RSB0
F13	GBC2/IO67RSB1
F14	GND
F15	IO75RSB1
F16	IO71RSB1
F17	IO77RSB1
G1	GFC0/IO193RSB3
G2	GND
G3	IO198RSB3
G4	IO203RSB3
G5	IO201RSB3
G6	IO206RSB3
G7	GND
G8	GND
G9	V_{CC}
G10	GND
G11	GND
G12	IO72RSB1

289-Pin CSP	
Pin Number	AGLP125 Function
G13	IO64RSB1
G14	IO69RSB1
G15	IO78RSB1
G16	IO76RSB1
G17	GND
H1	V_{COMPLF}
H2	GFB0/IO191RSB3
H3	IO195RSB3
H4	IO197RSB3
H5	IO199RSB3
H6	GFB1/IO192RSB3
H7	GND
H8	GND
H9	GND
H10	GND
H11	GND
H12	GCC1/IO79RSB1
H13	IO74RSB1
H14	GCA0/IO84RSB1
H15	V_{CCI}B1
H16	GCA2/IO85RSB1
H17	GCC0/IO80RSB1
J1	V_{CCPLF}
J2	GFA1/IO190RSB3
J3	V_{CCI}B3
J4	IO185RSB3
J5	IO183RSB3
J6	IO181RSB3
J7	V_{CC}
J8	GND
J9	GND
J10	GND
J11	V_{CC}
J12	GCB2/IO86RSB1
J13	GCB1/IO81RSB1
J14	IO90RSB1
J15	IO89RSB1
J16	GCB0/IO82RSB1

289-Pin CSP	
Pin Number	AGLP125 Function
J17	GCA1/IO83RSB1
K1	GND
K2	GFA0/IO189RSB3
K3	GFB2/IO187RSB3
K4	IO179RSB3
K5	IO175RSB3
K6	IO177RSB3
K7	GND
K8	GND
K9	GND
K10	GND
K11	GND
K12	IO88RSB1
K13	IO94RSB1
K14	IO95RSB1
K15	IO93RSB1
K16	GND
K17	GCC2/IO87RSB1
L1	GFA2/IO188RSB3
L2	GFC2/IO186RSB3
L3	IO182RSB3
L4	GND
L5	IO173RSB3
L6	GEC1/IO170RSB3
L7	GND
L8	GND
L9	V_{CC}
L10	GND
L11	GND
L12	GDC1/IO99RSB1
L13	GDB1/IO101RSB1
L14	V_{CCI}B1
L15	IO98RSB1
L16	IO92RSB1
L17	IO91RSB1
M1	IO184RSB3
M2	V_{CCI}B3
M3	IO176RSB3

289-Pin CSP	
Pin Number	AGLP125 Function
M4	IO172RSB3
M5	GEB0/IO167RSB3
M6	GEB1/IO168RSB3
M7	IO159RSB2
M8	IO161RSB2
M9	IO135RSB2
M10	IO128RSB2
M11	IO121RSB2
M12	IO113RSB2
M13	GDA1/IO103RSB1
M14	GDA0/IO104RSB1
M15	IO97RSB1
M16	IO96RSB1
M17	V_{CCI}B1
N1	IO180RSB3
N2	IO178RSB3
N3	GEC0/IO169RSB3
N4	GEA0/IO165RSB3
N5	GND
N6	IO156RSB2
N7	IO148RSB2
N8	IO144RSB2
N9	IO137RSB2
N10	V_{CCI}B2
N11	IO119RSB2
N12	IO111RSB2
N13	GDB2/IO106RSB2
N14	IO109RSB2
N15	GND
N16	GDB0/IO102RSB1
N17	GDC0/IO100RSB1
P1	IO174RSB3
P2	IO171RSB3
P3	GND
P4	IO160RSB2
P5	IO157RSB2
P6	IO154RSB2
P7	IO152RSB2

289-Pin CSP	
Pin Number	AGLP125 Function
P8	GND
P9	IO132RSB2
P10	IO125RSB2
P11	IO126RSB2
P12	IO112RSB2
P13	V_{CCI}B2
P14	IO108RSB2
P15	GDA2/IO105RSB2
P16	GDC2/IO107RSB2
P17	V_{JTAG}
R1	GND
R2	GEA2/IO164RSB2
R3	IO158RSB2
R4	IO155RSB2
R5	IO150RSB2
R6	V_{CCI}B2
R7	IO145RSB2
R8	IO141RSB2
R9	IO134RSB2
R10	IO130RSB2
R11	GND
R12	IO118RSB2
R13	IO116RSB2
R14	IO114RSB2
R15	IO110RSB2
R16	TMS
R17	TRST
T1	GEA1/IO166RSB3
T2	GEC2/IO162RSB2
T3	IO153RSB2
T4	GND
T5	IO147RSB2
T6	IO143RSB2
T7	IO140RSB2
T8	IO139RSB2
T9	V_{CCI}B2
T10	IO131RSB2
T11	IO127RSB2

289-Pin CSP	
Pin Number	AGLP125 Function
T12	IO124RSB2
T13	IO122RSB2
T14	GND
T15	IO115RSB2
T16	TDI
T17	TDO
U1	FF/GEB2/IO163RSB2
U2	GND
U3	IO151RSB2
U4	IO149RSB2
U5	IO146RSB2
U6	IO142RSB2
U7	GND
U8	IO138RSB2
U9	IO136RSB2
U10	IO133RSB2
U11	IO129RSB2
U12	GND
U13	IO123RSB2
U14	IO120RSB2
U15	IO117RSB2
U16	TCK
U17	V_{PUMP}

Part Number and Revision Date

Part Number 51700102-003-4
Revised August 2008

List of Changes

The following table lists critical changes that were made in the current version of the chapter.

Previous Version	Changes in Current Version (v1.4)	Page
v1.3 (June 2008)	The "128-Pin VQFP" package drawing and pin table are new.	3-1
	The "176-Pin VQFP" package drawing and pin table are new.	3-4
v1.2 (June 2008)	The "281-Pin CSP" package drawing is new.	3-12
	The "281-Pin CSP" table for the AGLP125 device is new.	3-13
	The "289-Pin CSP" package drawing was incorrect. The graphic was showing the CS281 mechanical drawing and not the CS289 mechanical drawing. This has now been corrected.	3-16
v1.1 (June 2008)	The "289-Pin CSP" table for the AGLP030 device is new.	3-17
v1.0 (January 2008)	The "289-Pin CSP" table for the AGLP060 device is new.	3-20
	The "289-Pin CSP" table for the AGLP125 device is new.	3-23

Datasheet Categories

Categories

In order to provide the latest information to designers, some datasheets are published before data has been fully characterized. Datasheets are designated as "Product Brief," "Advance," "Preliminary," and "Production." The definition of these categories are as follows:

Product Brief

The product brief is a summarized version of a datasheet (advance or production) and contains general product information. This document gives an overview of specific device and family information.

Advance

This version contains initial estimated information based on simulation, other products, devices, or speed grades. This information can be used as estimates, but not for production. This label only applies to the DC and Switching Characteristics chapter of the datasheet and will only be used when the data has not been fully characterized.

Preliminary

The datasheet contains information based on simulation and/or initial characterization. The information is believed to be correct, but changes are possible.

Unmarked (production)

This version contains information that is considered to be final.

Export Administration Regulations (EAR)

The products described in this document are subject to the Export Administration Regulations (EAR). They could require an approved export license prior to export from the United States. An export includes release of product or disclosure of technology to a foreign national inside or outside the United States.

Actel Safety Critical, Life Support, and High-Reliability Applications Policy

The Actel products described in this advance status document may not have completed Actel's qualification process. Actel may amend or enhance products during the product introduction and qualification process, resulting in changes in device functionality or performance. It is the responsibility of each customer to ensure the fitness of any Actel product (but especially a new product) for a particular purpose, including appropriateness for safety-critical, life-support, and other high-reliability applications. Consult Actel's Terms and Conditions for specific liability exclusions relating to life-support applications. A reliability report covering all of Actel's products is available on the Actel website at http://www.actel.com/documents/ORT_Report.pdf. Actel also offers a variety of enhanced qualification and lot acceptance screening procedures. Contact your local Actel sales office for additional reliability information.

Section II – Core Architecture

*Low-Power Flash Technology and Flash*Freeze Mode*

Actel
POWER MATTERS

1 – FPGA Array Architecture in Low-Power Flash Devices

Device Architecture

Advanced Flash Switch

Unlike SRAM FPGAs, the low-power flash devices use a live-at-power-up ISP flash switch as their programming element. Flash cells are distributed throughout the device to provide nonvolatile, reconfigurable programming to connect signal lines to the appropriate VersaTile inputs and outputs. In the flash switch, two transistors share the floating gate, which stores the programming information (Figure 1-1). One is the sensing transistor, which is only used for writing and verification of the floating gate voltage. The other is the switching transistor. The latter is used to connect or separate routing nets, or to configure VersaTile logic. It is also used to erase the floating gate. Dedicated high-performance lines are connected as required using the flash switch for fast, low-skew, global signal distribution throughout the device core. Maximum core utilization is possible for virtually any design. The use of the flash switch technology also removes the possibility of firm errors, which are increasingly common in SRAM-based FPGAs.

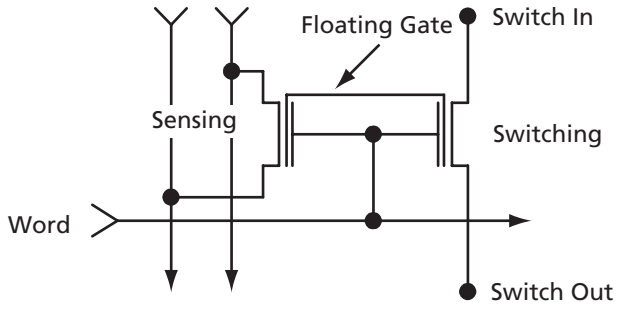


Figure 1-1 • **Flash-Based Switch**

FPGA Array Architecture Support

The low-power flash families listed in Table 1-1 support the architecture features described in this document.

Table 1-1 • **Low-Power Flash Families**

Product Line	Family*	Description
Fusion	Fusion	Mixed-signal FPGA integrating ProASIC®3 FPGA fabric, programmable analog block, support for ARM® Cortex™-M1 soft processors and flash memory into a monolithic device
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 PLUS	IGLOO FPGAs with enhanced I/O capabilities
ProASIC3	ProASIC3	Low-power, high-performance 1.5 V FPGAs
	ProASIC3E	Higher density ProASIC3 FPGAs with six PLLs and additional I/O standards
	ProASIC3L	ProASIC3 FPGAs supporting 1.2 V to 1.5 V with Flash*Freeze technology
	Automotive ProASIC3	ProASIC3 FPGAs qualified for automotive applications
	Military ProASIC3/EL	Military temperature A3PE600L, A3P1000, and A3PE3000L
	RT ProASIC3	Radiation-tolerant RT3PE600L and RT3PE3000L

*Note: *The family names link to the appropriate datasheet, including product brief, DC and switching characteristics, and packaging information.*

IGLOO Terminology

In documentation, the terms IGLOO families and IGLOO devices refer to all of the IGLOO products as listed in Table 1-1. Where the information applies to only one family or limited devices, these exclusions will be explicitly stated.

ProASIC3 Terminology

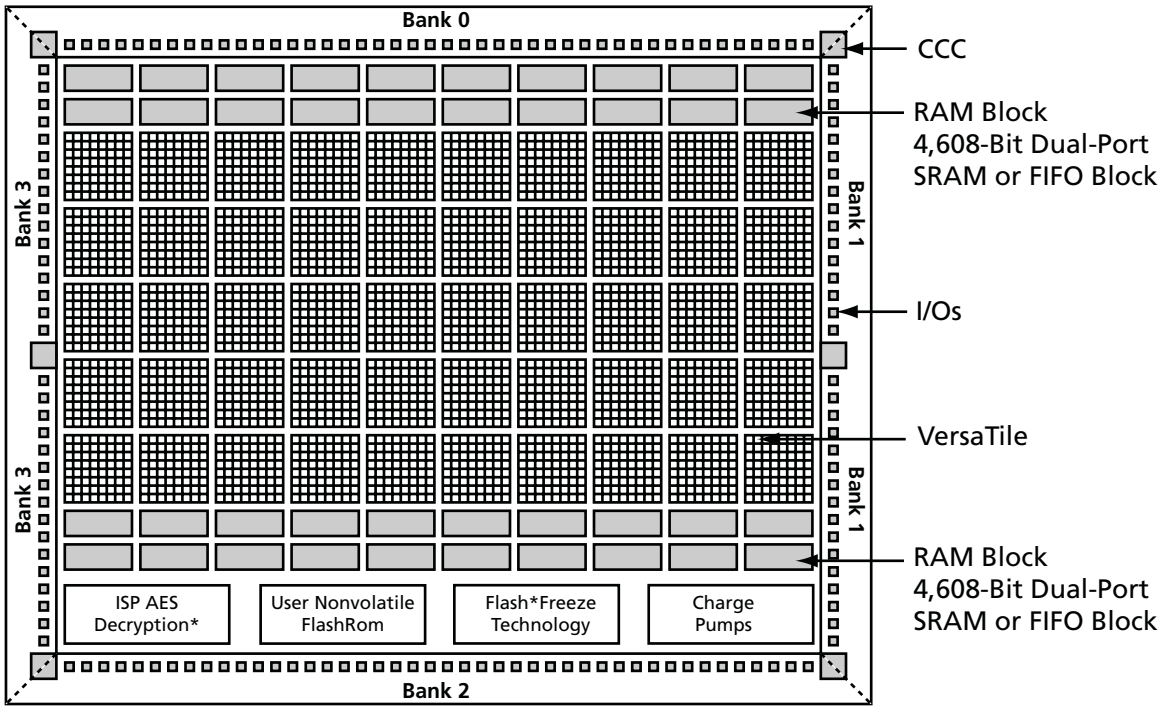
In documentation, the terms ProASIC3 families and ProASIC3 devices refer to all of the ProASIC3 families as listed in Table 1-1. Where the information applies to only one family or limited devices, these exclusions will be explicitly stated.

To further understand the differences between the IGLOO and ProASIC3 families, refer to the *Industry's Lowest Power FPGAs Portfolio*.

Device Overview

The low-power flash devices consist of multiple distinct programmable architectural features (Figure 1-2 on page 1-3 through Figure 1-4 on page 1-4):

- FPGA fabric/core (VersaTiles)
- Routing and clock resources (VersaNets)
- FlashROM
- Dedicated SRAM and/or FIFO
  - 15 k and 30 k gate devices do not support SRAM or FIFO.
  - Automotive devices do not support FIFO operation
- I/O Structures
- Flash*Freeze technology and low-power modes



*Note: Flash*Freeze technology only applies to IGLOO and ProASIC3L families.*

Figure 1-2 • **IGLOO and ProASIC3/L Device Architecture Overview with Four I/O Banks (AGL600 device is shown)**

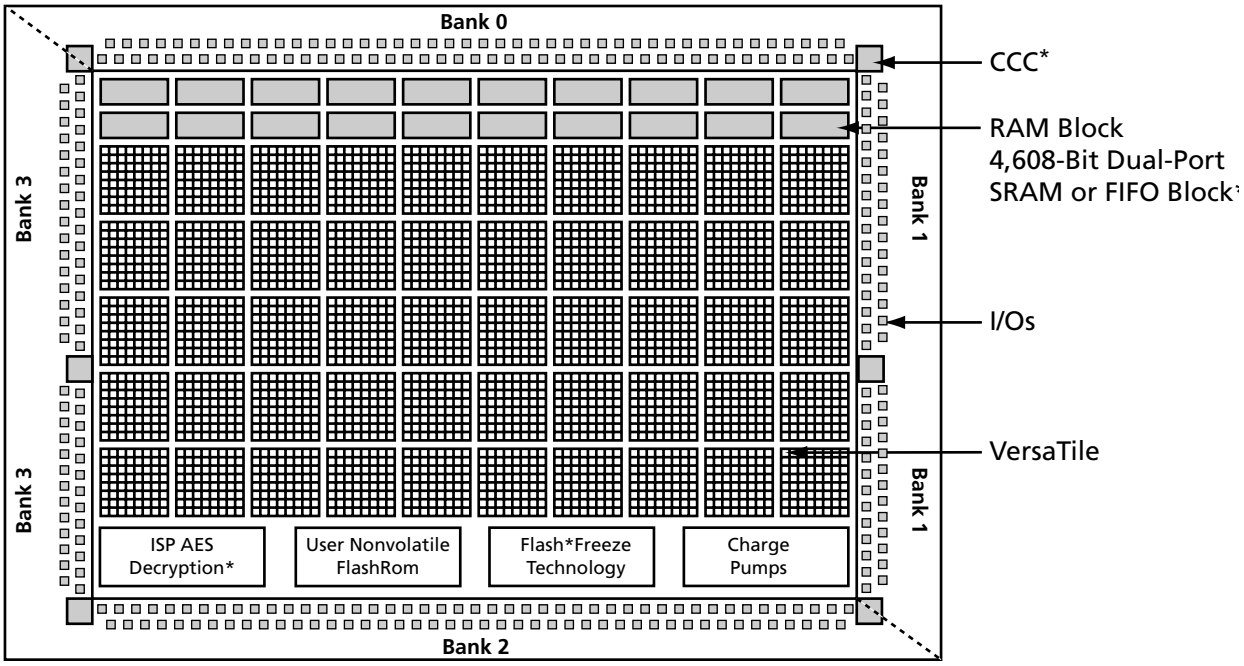
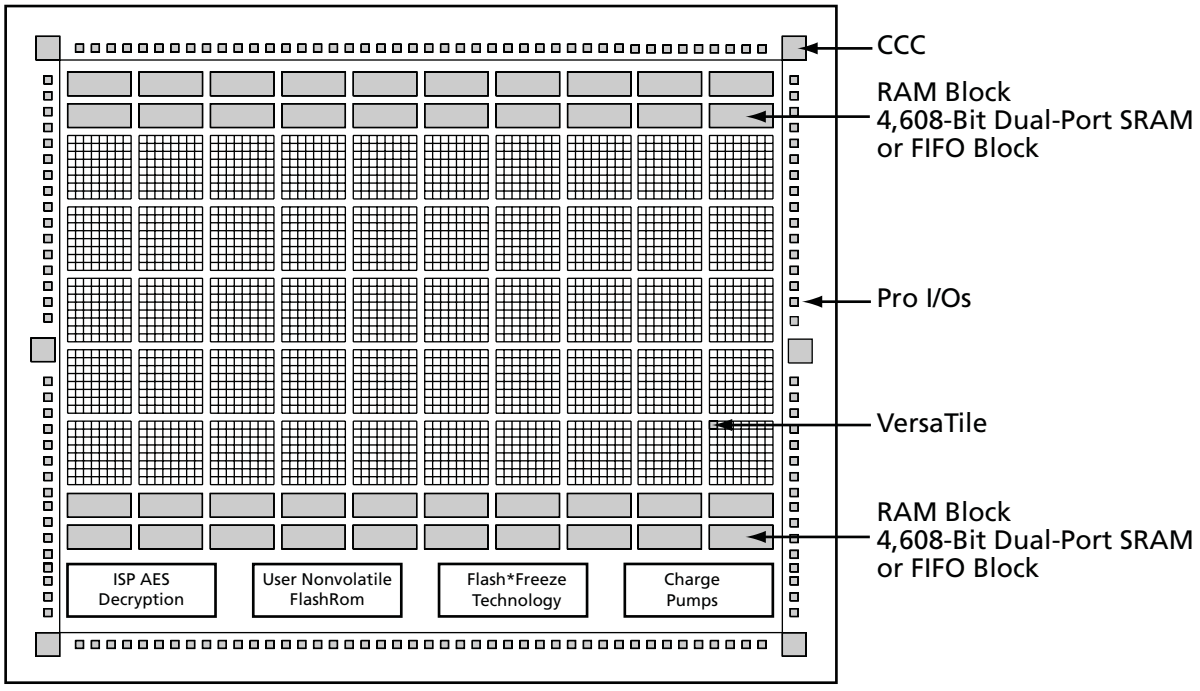


Figure 1-3 • **IGLOO PLUS Device Architecture Overview with Four I/O Banks**



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

Figure 1-4 • **IGLOOe and ProASIC3E Device Architecture Overview (AGLE600 device is shown)**

Core Architecture

VersaTile

The proprietary IGLOO and ProASIC3 device architectures provide granularity comparable to gate arrays. The device core consists of a sea-of-VersaTiles architecture.

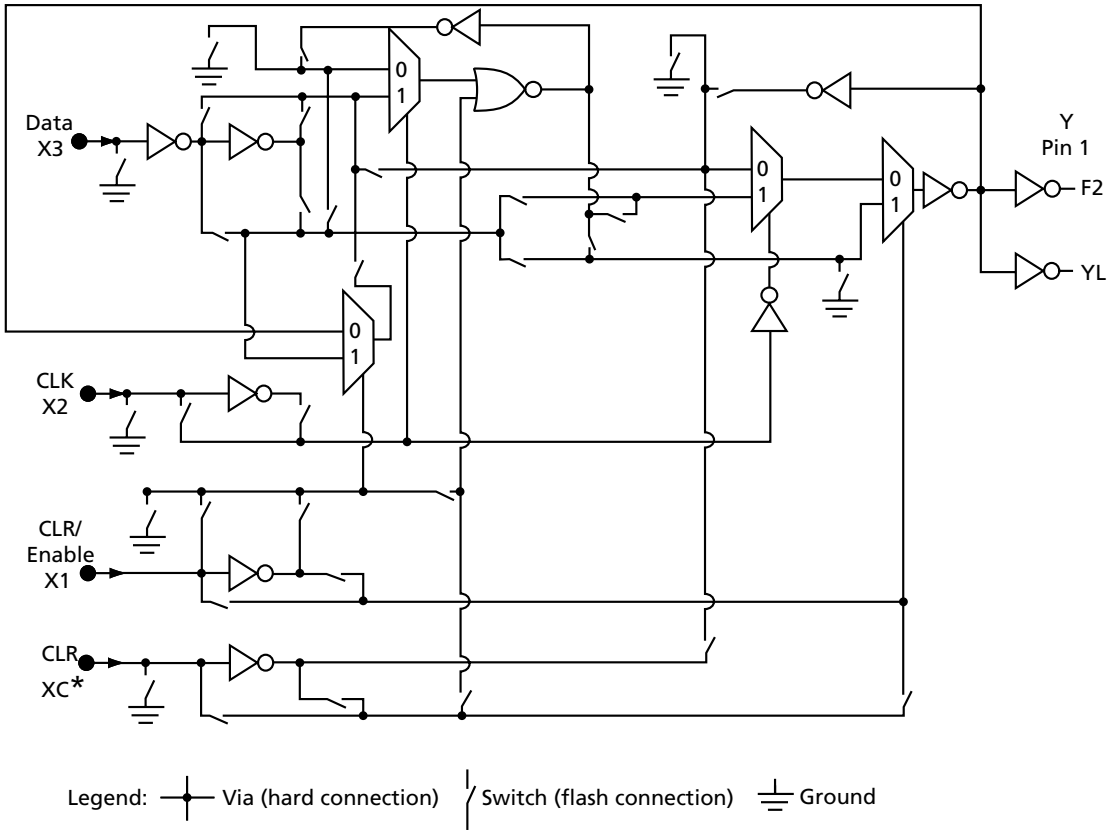
As illustrated in Figure 1-5, there are four inputs in a logic VersaTile cell, and each VersaTile can be configured using the appropriate flash switch connections:

- Any 3-input logic function
- Latch with clear or set
- D-flip-flop with clear or set
- Enable D-flip-flop with clear or set (on a 4th input)

VersaTiles can flexibly map the logic and sequential gates of a design. The inputs of the VersaTile can be inverted (allowing bubble pushing), and the output of the tile can connect to high-speed, very-long-line routing resources. VersaTiles and larger functions can be connected with any of the four levels of routing hierarchy.

When the VersaTile is used as an enable D-flip-flop, SET/CLR is supported by a fourth input. The SET/CLR signal can only be routed to this fourth input over the VersaNet (global) network. However, if, in the user's design, the SET/CLR signal is not routed over the VersaNet network, a compile warning message will be given, and the intended logic function will be implemented by two VersaTiles instead of one.

The output of the VersaTile is F2 when the connection is to the ultra-fast local lines, or YL when the connection is to the efficient long-line or very-long-line resources.



* *This input can only be connected to the global clock distribution network.*

Figure 1-5 • **Low-Power Flash Device Core VersaTile**

Array Coordinates

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

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

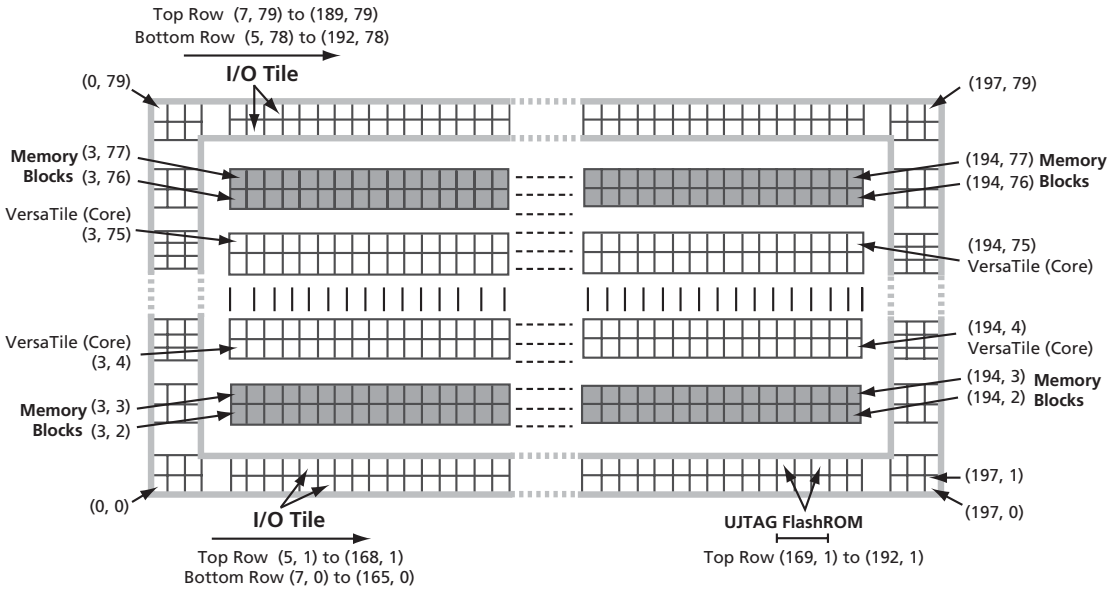
Figure 1-6 on page 1-7 illustrates the array coordinates of a 600 k gate device. For more information on how to use array coordinates for region/placement constraints, see the *Designer User's Guide* or online help (available in the software) for software tools.

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

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

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

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



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 1-6 • **Array Coordinates for AGL600, AGLE600, A3P600, and A3PE600**

Routing Architecture

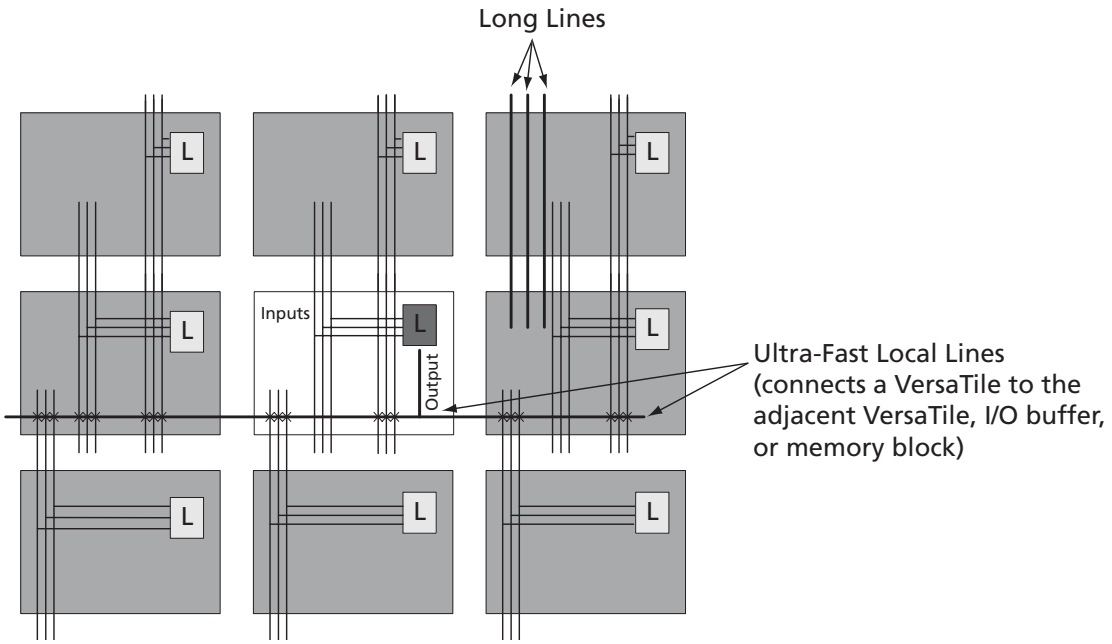
The routing structure of low-power flash devices is designed to provide high performance through a flexible four-level hierarchy of routing resources: ultra-fast local resources; efficient long-line resources; high-speed, very-long-line resources; and the high-performance VersaNet networks.

The ultra-fast local resources are dedicated lines that allow the output of each VersaTile to connect directly to every input of the eight surrounding VersaTiles (Figure 1-7 on page 1-8). The exception to this is that the SET/CLR input of a VersaTile configured as a D-flip-flop is driven only by the VersaTile global network.

The efficient long-line resources provide routing for longer distances and higher-fanout connections. These resources vary in length (spanning one, two, or four VersaTiles), run both vertically and horizontally, and cover the entire device (Figure 1-8 on page 1-9). Each VersaTile can drive signals onto the efficient long-line resources, which can access every input of every VersaTile. Routing software automatically inserts active buffers to limit loading effects.

The high-speed, very-long-line resources, which span the entire device with minimal delay, are used to route very long or high-fanout nets: length ±12 VersaTiles in the vertical direction and length ±16 in the horizontal direction from a given core VersaTile (Figure 1-9 on page 1-9). Very long lines in low-power flash devices have been enhanced over those in previous ProASIC families. This provides a significant performance boost for long-reach signals.

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



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

Figure 1-7 • **Ultra-Fast Local Lines Connected to the Eight Nearest Neighbors**

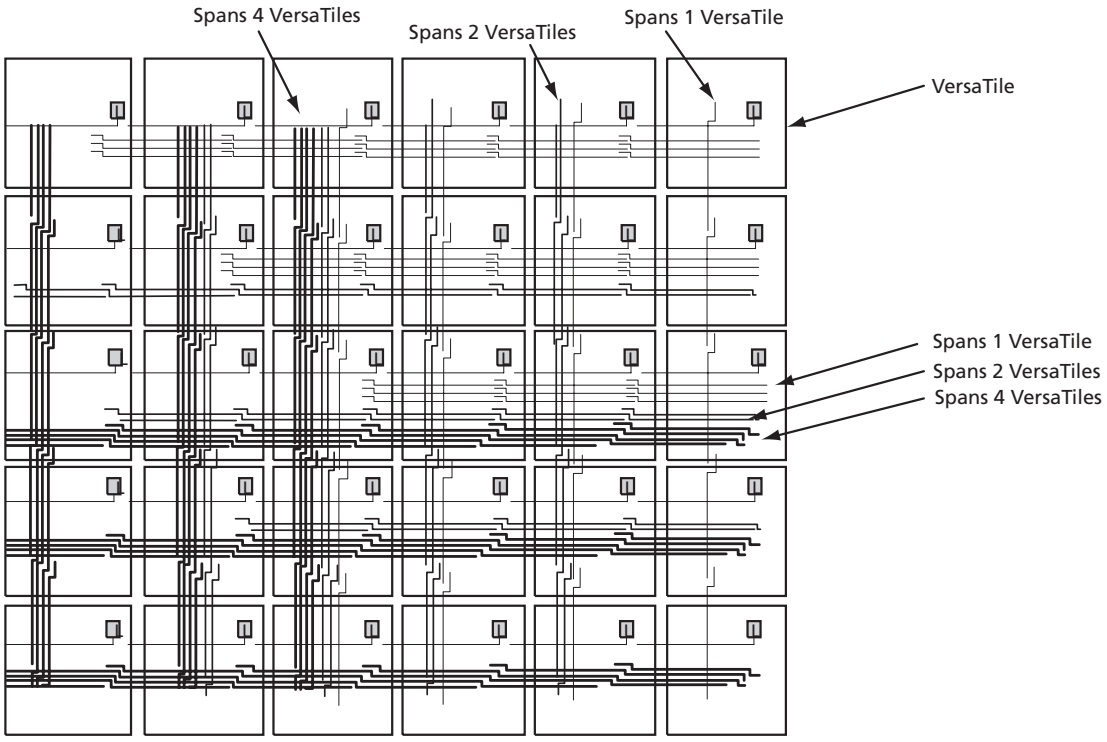


Figure 1-8 • **Efficient Long-Line Resources**

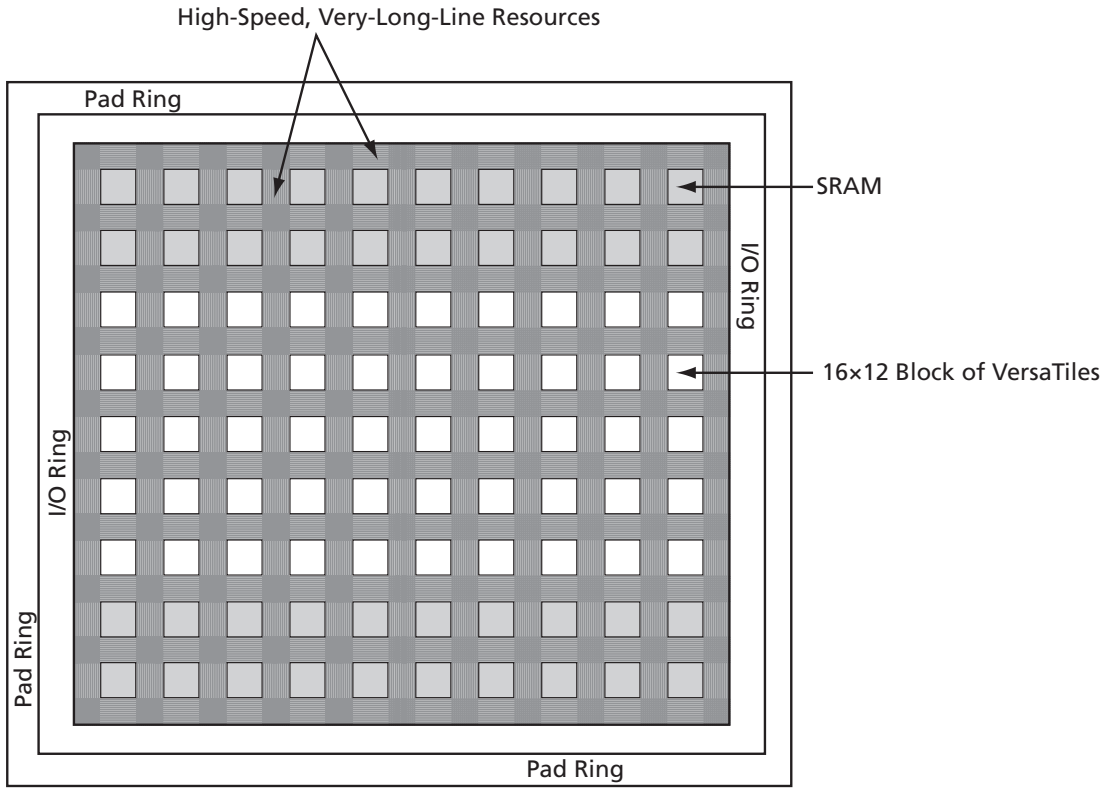


Figure 1-9 • **Very-Long-Line Resources**

Related Documents

Handbook Documents

Global Resources in Actel Low-Power Flash Devices

http://www.actel.com/documents/LPD_Glorbal_HBs.pdf

User's Guides

Designer User's Guide

http://www.actel.com/documents/designer_ug.pdf

Part Number and Revision Date

This document contains content extracted from the Device Architecture section of the datasheet. To improve usability for customers, the device architecture information has now been split into handbook sections, which also include usage information. No technical changes were made to the content unless explicitly listed.

Part Number 51700094-002-3

Revised October 2008

List of Changes

The following table lists critical changes that were made in the current version of the chapter.

Previous Version	Changes in Current Version (v1.3)	Page
v1.2 (June 2008)	The title of this document was changed from "Core Architecture of IGLOO and ProASIC3 Devices" to "FPGA Array Architecture in Low-Power Flash Devices."	1-1
	The "FPGA Array Architecture Support" section was revised to include new families and make the information more concise.	1-2
	Table 1-2 · IGLOO and ProASIC3 Array Coordinates was updated to include Military ProASIC3/EL and RT ProASIC3 devices.	1-6
v1.1 (March 2008)	The following changes were made to the family descriptions in Table 1-1 · Low-Power Flash Families: • ProASIC3L was updated to include 1.5 V. • The number of PLLs for ProASIC3E was changed from five to six.	1-2
v1.0 (January 2008)	Table 1-1 · Low-Power Flash Families and the accompanying text was updated to include the IGLOO PLUS family. The "IGLOO Terminology" section and "Device Overview" section are new.	1-2
	The "Device Overview" section was updated to note that 15 k devices do not support SRAM or FIFO.	1-3
	Figure 1-3 · IGLOO PLUS Device Architecture Overview with Four I/O Banks is new.	1-4
	Table 1-2 · IGLOO and ProASIC3 Array Coordinates was updated to add A3P015 and AGL015.	1-6
	Table 1-3 · IGLOO PLUS Array Coordinates is new.	1-6

2 – Actel's Flash*Freeze Technology and Low-Power Modes

Flash*Freeze Technology and Low-Power Modes

Actel IGLOO,® IGLOO PLUS, ProASIC®3L, and Radiation-Tolerant (RT) ProASIC3 FPGAs with Flash*Freeze technology are designed to meet the most demanding power and area challenges of today's portable electronics products with a reprogrammable, small-footprint, full-featured flash FPGA. The IGLOO, IGLOO PLUS, ProASIC3L, and RT ProASIC3 families offer lower power consumption in static and dynamic modes, utilizing the unique Flash*Freeze technology, than any other FPGA or CPLD.

IGLOO, IGLOO PLUS, ProASIC3L, and RT ProASIC3 devices offer various power-saving modes that enable every system to utilize modes that achieve the lowest total system power. Low Power Active capability (static idle) allows for ultra-low power consumption while the device is operational in the system by maintaining SRAM, registers, I/Os, and logic functions.

Flash*Freeze technology provides an ultra-low-power static mode (Flash*Freeze mode) that retains all SRAM and register information with rapid recovery to Active (operating) mode. IGLOO PLUS has an additional feature when operating in Flash*Freeze mode, allowing it to retain I/O states as well as SRAM and register states. This mechanism enables the user to quickly (within 1 µs) enter and exit Flash*Freeze mode by activating the Flash*Freeze (FF) pin while all power supplies are kept in their original states. In addition, I/Os and clocks connected to the FPGA can still be toggled without impact on device power consumption. While in Flash*Freeze mode, the device retains all core register states and SRAM information. This mode can be configured so that no power is consumed by the I/O banks, clocks, JTAG pins, or PLLs; and the IGLOO and IGLOO PLUS devices consume as little as 5 µW. Actel offers a state management IP core to aid users in gating clocks and managing data before entering Flash*Freeze mode.

This document will guide users in selecting the best low-power mode for their applications, and introduces Actel's Flash*Freeze management IP core.

Actel's Flash Families Support the Flash*Freeze Feature

The low-power flash families listed in Table 2-1 support the Flash*Freeze feature and the functions described in this document.

Table 2-1 • **Low-Power Flash Families**

Product Line	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 PLUS	IGLOO FPGAs with enhanced I/O capabilities
ProASIC3	ProASIC3L	ProASIC3 FPGAs supporting 1.2 V to 1.5 V with Flash*Freeze technology
	Military ProASIC3/EL	Military temperature A3PE600L, A3P1000, and A3PE3000L
	RT ProASIC3	Radiation-tolerant RT3PE600L and RT3PE3000L

*Note: *The family names link to the appropriate datasheet, including product brief, DC and switching characteristics, and packaging information.*

IGLOO Terminology

In documentation, the terms IGLOO families and IGLOO devices refer to all of the IGLOO products as listed in Table 2-1. Where the information applies to only one family or limited devices, these exclusions will be explicitly stated.

ProASIC3 Terminology

In documentation, the terms ProASIC3 families and ProASIC3 devices refer to all of the ProASIC3 families as listed in Table 2-1. Where the information applies to only one family or limited devices, these exclusions will be explicitly stated.

To further understand the differences between the IGLOO and ProASIC3 families, refer to the *Industry's Lowest Power FPGAs Portfolio*.

Low-Power Modes Overview

Table 2-2 summarizes the low-power modes that achieve power consumption reduction when the FPGA or system is idle.

Table 2-2 • **Power Modes Summary**

Mode		V_{CCI}	V_{CC}	Core	Clocks	ULSICC Macro	To Enter Mode	To Resume Operation	Trigger
Active		On	On	On	On	N/A	Initiate clock	None	–
Static	Idle	On	On	On	Off	N/A	Stop clock	Initiate clock	External
	Flash*Freeze type 1	On	On	On	On*	N/A	Assert FF pin	Deassert FF pin	External
	Flash*Freeze type 2	On	On	On	On*	Used to enter Flash*Freeze mode	Assert FF pin and assert LSICC	Deassert FF pin	External
Sleep		On	Off	Off	Off	N/A	Shut down V_{CC}	Turn on V_{CC} supply	External
Shutdown		Off	Off	Off	Off	N/A	Shut down V_{CC} and V_{CCI} supplies	Turn on V_{CC} and V_{CCI} supplies	External

** External clocks can be left toggling while the device is in Flash*Freeze mode. Clocks generated by the embedded PLL will be turned off automatically.*

Static (Idle) Mode

In Static (Idle) mode, none of the clock inputs is switching, and static power is the only power consumed by the device. This mode can be achieved by switching off the incoming clocks to the FPGA, thus benefitting from reduced power consumption. In addition, I/Os draw only minimal leakage current. In this mode, embedded SRAM, I/Os, and registers retain their values so the device can enter and exit this mode just by switching the clocks on or off.

If the device-embedded PLL is used as the clock source, Static (Idle) mode can easily be entered by pulling the PLL POWERDOWN pin LOW (active Low), which will turn off the PLL.

Flash*Freeze Mode

IGLOO, IGLOO PLUS, ProASIC3L, and RT ProASIC3 FPGAs offer an ultra-low static power mode to reduce power consumption while preserving the state of the registers, SRAM contents, and I/O states (IGLOO PLUS only) without switching off any power supplies, inputs, or input clocks.

Flash*Freeze technology enables the user to switch to Flash*Freeze mode within 1 µs, thus simplifying low-power design implementation. The Flash*Freeze (FF) pin (active Low) is a dedicated pin used to enter or exit Flash*Freeze mode directly; or the pin can be routed internally to the FPGA core and state management IP to allow the user's application to decide if and when it is safe to transition to this mode. If the FF pin is not used, it can be used as a regular I/O.

The FF pin has a built-in glitch filter and optional Schmitt trigger (not available for all devices) to prevent entering or exiting Flash*Freeze mode accidentally.

There are two ways to use Flash*Freeze mode. In Flash*Freeze type 1, entering and exiting the mode is exclusively controlled by the assertion and deassertion of the FF pin. This enables an external processor or human interface device to directly control Flash*Freeze mode; however, valid data must be preserved using standard procedures (refer to the "Flash*Freeze Mode Device Behavior" section on page 2-9). In Flash*Freeze mode type 2, entering and exiting the mode is controlled by both the FF pin AND user-defined logic. Flash*Freeze management IP may be used in type 2 mode for clock and data management while entering and exiting Flash*Freeze mode.

Flash*Freeze Type 1: Control by Dedicated Flash*Freeze Pin

Flash*Freeze type 1 is intended for systems where either the device will be reset upon exiting Flash*Freeze mode, or data and clock are managed externally. The device enters Flash*Freeze mode 1 µs after the dedicated FF pin is asserted (active Low), and returns to normal operation when the FF pin is deasserted (High) (Figure 2-1 on page 2-5). In this mode, FF pin assertion or deassertion is the only condition that determines entering or exiting Flash*Freeze mode.

In Actel Libero® Integrated Design Environment (IDE) v8.2 and before, this mode is implemented by enabling Flash*Freeze mode (default setting) in the Compile options of the Actel Designer software. To simplify usage of Flash*Freeze mode, beginning with Libero IDE v8.3, an INBUF_FF I/O macro was introduced. An INBUF_FF I/O buffer must be used to identify the Flash*Freeze input. Actel recommends switching to the new implementation.

In Libero IDE v8.3 and later, the user must manually instantiate the INBUF_FF macro in the top level of the design in order implement Flash*Freeze Type 1, as shown in Figure 2-1 on page 2-5.

Figure 2-1 shows the concept of FF pin control in Flash*Freeze mode type 1.

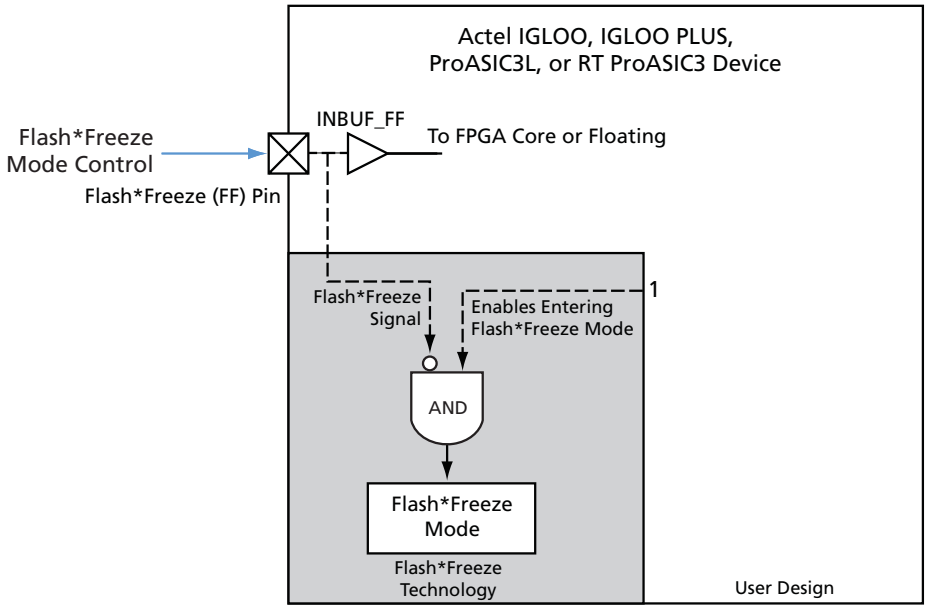


Figure 2-1 • **Flash*Freeze Mode Type 1 – Controlled by the Flash*Freeze Pin**

Figure 2-2 shows the timing diagram for entering and exiting Flash*Freeze mode type 1.

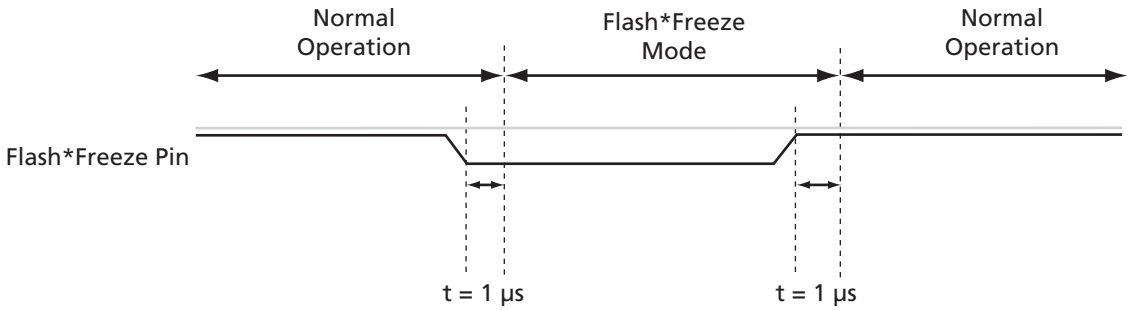


Figure 2-2 • **Flash*Freeze Mode Type 1 – Timing Diagram**

Flash*Freeze Type 2: Control by Dedicated Flash*Freeze Pin and Internal Logic

The device can be made to enter Flash*Freeze mode by activating the FF pin together with Actel's Flash*Freeze management IP core (refer to the "Flash*Freeze Management IP" section on page 2-15 for more information) or user-defined control logic (Figure 2-3) within the FPGA core. This method enables the design to perform important activities before allowing the device to enter Flash*Freeze mode, such as transitioning into a safe state, completing the processing of a critical event. Designers are encouraged to take advantage of Actel's Flash*Freeze Management IP to handle clean entry and exit of Flash*Freeze mode (described later in this document). The device will only enter Flash*Freeze mode when the Flash*Freeze pin is asserted (active Low) and the User Low Static I_{CC} (ULSICC) macro input signal, called the LSICC signal, is asserted (High). One condition is not sufficient to enter Flash*Freeze mode type 2; both the FF pin and LSICC signal must be asserted.

When Flash*Freeze type 2 is implemented in the design, the ULSICC macro needs to be instantiated by the user. There are no functional differences in the device whether the ULSICC macro is instantiated or not, and whether the LSICC signal is asserted or deasserted. The LSICC signal is used only to control entering Flash*Freeze mode. Figure 2-4 on page 2-7 shows the timing diagram for entering and exiting Flash*Freeze mode type 2.

After exiting Flash*Freeze mode type 2 by deasserting the Flash*Freeze pin, the LSICC signal must be deasserted by the user design. This will prevent entering Flash*Freeze mode by asserting the Flash*Freeze pin only.

Refer to Table 2-3 for Flash*Freeze (FF) pin and LSICC signal assertion and deassertion values.

Table 2-3 • **Flash*Freeze Mode Type 1 and Type 2 – Signal Assertion and Deassertion Values**

Signal	Assertion Value	Deassertion Value
Flash*Freeze (FF) pin	Low	High
LSICC signal	High	Low

Notes:

1. *The Flash*Freeze (FF) pin is an active-Low signal, and LSICC is an active-High signal.*
2. *The LSICC signal is used only in Flash*Freeze mode type 2.*

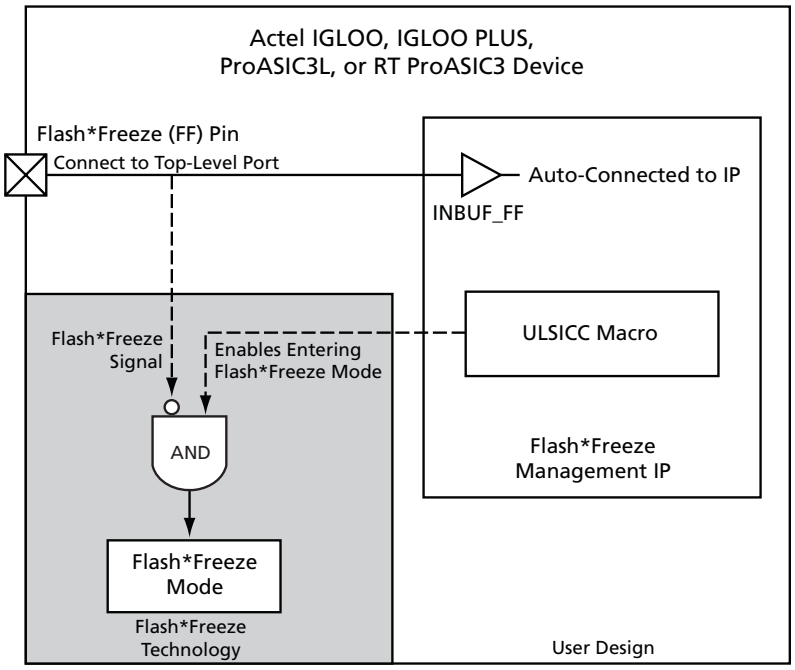


Figure 2-3 • **Flash*Freeze Mode Type 2 – Controlled by Flash*Freeze Pin and Internal Logic (LSICC signal)**

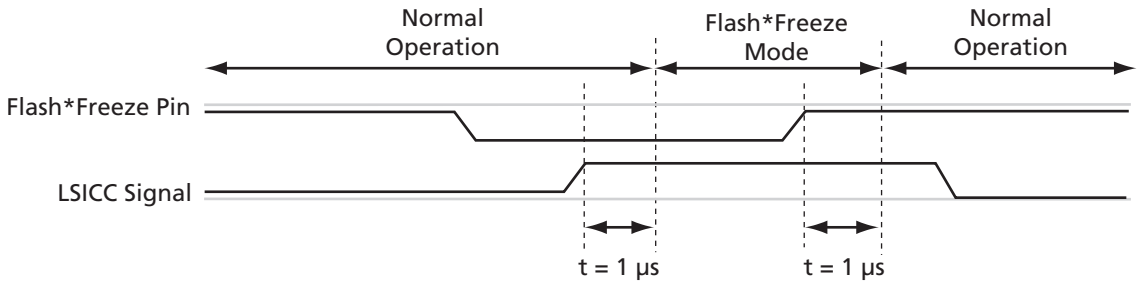


Figure 2-4 • **Flash*Freeze Mode Type 2 – Timing Diagram**

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

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

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

*Note: Refer to Table 2-3 on page 2-6 for Flash*Freeze pin and LSICC signal assertion and deassertion values.*

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

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

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

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

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

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

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

IGLOO PLUS I/O State in Flash*Freeze Mode

In IGLOO PLUS devices, users have multiple options in how to configure I/Os during Flash*Freeze mode:

1. Hold the previous state
2. Set I/O pad to weak pull-up or pull-down
3. Tristate I/O pads

The I/O configuration must be configured by the user in the I/O Attribute Editor or in a PDC constraint file, and can be done on a pin-by-pin basis. The output hold feature will hold the output in the last registered state, using the I/O pad weak pull-up or pull-down resistor when the FF pin is asserted. When inputs are configured with the hold feature enabled, the FPGA core side of the input will hold the last valid state of the input pad before the device entered Flash*Freeze mode. The input pad can be driven to any value, configured as tristate, or configured with the weak pull-up or pull-down I/O pad feature during Flash*Freeze mode without affecting the hold state. If the weak pull-up or pull-down feature is used without the output hold feature, the input and output pads will maintain the configured weak pull-up or pull-down status during Flash*Freeze mode and normal operation. If a fixed weak pull-up or pull-down is defined on an output buffer or as bidirectional in output mode, and a hold state is also defined for the same pin, the pin will be configured in hold state mode during Flash*Freeze mode. During normal operation, the pin will be configured with the predefined weak pull-up or pull-down. Any I/Os that do not use the hold state or I/O pad weak pull-up or pull-down features will be tristated during Flash*Freeze mode and the FPGA core will be driven High by inputs. Inputs that are tristated during Flash*Freeze mode may be left floating without any reliability concern or impact to power consumption.

Table 2-6 shows the I/O pad state based on the configuration and buffer type.

Note that configuring weak pull-up or pull-down for the FF pin is not allowed.

Table 2-6 • **IGLOO PLUS Flash*Freeze Mode (type 1 and type 2)—I/O Pad State**

Buffer Type		Hold State	I/O Pad Weak Pull-Up/-Down	I/O Pad State in Flash*Freeze Mode
Input		Enabled	Enabled	Weak pull-up/pull-down[1]
		Disabled	Enabled	Weak pull-up/pull-down[2]
		Enabled	Disabled	Tristate[1]
		Disabled	Disabled	Tristate[2]
Output		Enabled	"Don't care"	Weak pull to hold state
		Disabled	Enabled	Weak pull-up/pull-down
		Disabled	Disabled	Tristate
Bidirectional / Tristate Buffer	E = 0 (input/tristate)	Enabled	Enabled	Weak pull-up/pull-down[1]
		Disabled	Enabled	Weak pull-up/pull-down[2]
		Enabled	Disabled	Tristate[1]
		Disabled	Disabled	Tristate[2]
	E = 1 (output)	Enabled	"Don't care"	Weak pull to hold state[3]
		Disabled	Enabled	Weak pull-up/pull-down
		Disabled	Disabled	Tristate

Notes:

1. *Internal core logic driven by this input buffer will be set to the value this I/O had when entering Flash*Freeze mode.*
2. *Internal core logic driven by this input buffer will be tied High as long as the device is in Flash*Freeze mode.*
3. *For bidirectional buffers: Internal core logic driven by the input portion of the bidirectional buffer will be set to the hold state.*

Flash*Freeze Mode Device Behavior

*Entering Flash*Freeze Mode*

- Actel IGLOO, IGLOO PLUS, ProASCI3L, and RT ProASIC3 devices are designed and optimized to enter Flash*Freeze mode only when power supplies are stable. If the device is being powered up while the FF pin is asserted (Flash*Freeze mode type 1), or while both FF pin and LSICC signal are asserted (Flash*Freeze mode type 2), the device is expected to enter Flash*Freeze mode within 5 µs after the I/Os and FPGA core have reached their activation levels.
- If the device is already powered up when the FF pin is asserted, the device will enter Flash*Freeze mode within 1 µs (type 1). In Flash*Freeze mode type 2 operation, entering Flash*Freeze mode is completed within 1 µs after both FF pin and LSICC signal are asserted. Exiting Flash*Freeze mode is completed within 1 µs after deasserting the FF pin only.

PLLs

- If an embedded PLL is used, entering Flash*Freeze mode will automatically power down the PLL.
- The PLL output clocks will stop toggling within 1 µs after the assertion of the FF pin in type 1, or after both FF pin and LSICC signal are asserted in type 2. At the same time, I/Os will transition into the state specified in Table 2-6 on page 2-8. The user design must ensure it is safe to enter Flash*Freeze mode.

I/Os and Globals

- While entering Flash*Freeze mode, inputs, globals, and PLLs will enter their Flash*Freeze state asynchronously to each other. As a result, clock and data glitches and narrow pulses may be generated while entering Flash*Freeze mode, as shown in Figure 2-5.

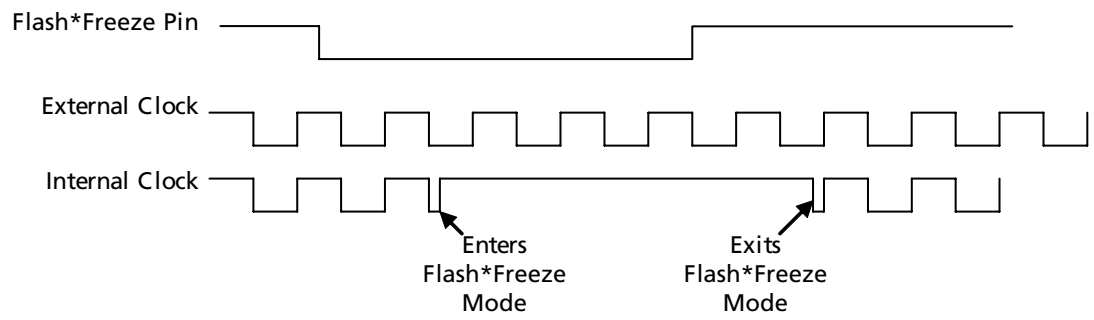


Figure 2-5 • **Narrow Clock Pulses During Flash*Freeze Entrance and Exit**

- I/O banks are not all deactivated simultaneously when entering Flash*Freeze mode. This can cause clocks and inputs to become disabled at different times, resulting in unexpected data being captured.
- Upon entering Flash*Freeze mode, all inputs and globals become tied High internally (except when an input hold state is used on IGLOO PLUS devices). If any of these signals are driven Low or tied Low externally, they will experience a Low to High transition internally when entering Flash*Freeze mode.
- Upon entering type 2 Flash*Freeze mode, ensure the LSICC signal (active High) does not de-assert. This can prevent the device from entering Flash*Freeze mode.
- Asynchronous input to output paths may experience output glitches. For example, on a direct in-to-out path, if the current state is '0' and the input bank turns off first, the input and then the output will transition to '1' before the output enters its Flash*Freeze state. This can be prevented by using latches in asynchronous in-to-out paths.
- The above situations can cause glitches or invalid data to be clocked into and preserved in the device. Refer to the "Flash*Freeze Design Guide" section on page 2-13 for solutions.

During Flash*Freeze Mode

- PLLs are turned off during Flash*Freeze mode.
- I/O pads are configured according to Table 2-5 on page 2-7 and Table 2-6 on page 2-8.
- Inputs and input clocks to the FPGA can toggle without any impact on static power consumption, assuming weak pull-up or pull-down is not selected.
- If weak pull-up or pull-down is selected and the input is driven to the opposite direction, power dissipation will occur.
- Any toggling signals will be charging and discharging the package pin capacitance.
- IGLOO and ProASIC3L outputs will be tristated unless the I/O is configured with weak pull-up or pull-down. The output of the I/O to the FPGA core is logic High regardless of whether the I/O pin is configured with a weak pull-up or pull-down. Refer to Table 2-5 on page 2-7 for more information.
- IGLOO PLUS output behavior will be based on the configuration defined by the user. Refer to Table 2-6 on page 2-8 for a description of output behavior during Flash*Freeze mode.
- The JTAG circuit is active; however, JTAG operations, such as JTAG commands, JTAG bypass, programming, and authentication, cannot be executed. The device must exit Flash*Freeze mode before JTAG commands can be sent. TCK should be static to avoid extra power consumption from the JTAG state machine.
- The FF pin must be externally asserted for the device to stay in Flash*Freeze mode.
- The FF pin is still active; i.e., the pin is used to exit Flash*Freeze mode when deasserted.

Exiting Flash*Freeze Mode

I/Os and Globals

- While exiting Flash*Freeze mode, inputs and globals will exit their Flash*Freeze state asynchronously to each other. As a result, clock and data glitches and narrow pulses may be generated while exiting Flash*Freeze mode, unless clock gating schemes are used.
- I/O banks are not all activated simultaneously when exiting Flash*Freeze mode. This can cause clocks and inputs to become enabled at different times, resulting in unexpected data being captured.
- Upon exiting Flash*Freeze mode, inputs and globals will no longer be tied High internally (does not apply to input hold state on IGLOO PLUS). If any of these signals are driven Low or tied Low externally, they will experience a High-to-Low transition internally when exiting Flash*Freeze mode.
- Applies only to IGLOO PLUS: Output hold state is asynchronously controlled by the signal driving the output buffer (output signal). This ensures a clean, glitch-free transition from hold state to output drive. However, any glitches on the output signal during exit from Flash*Freeze mode may result in glitches on the output pad.
- The above situations can cause glitches or invalid data to be clocked into and preserved in the device. Refer to the "Flash*Freeze Design Guide" on page 2-13 for solutions.

PLLs

- If the embedded PLL is used, the design must allow maximum acquisition time (per device datasheet) for the PLL to acquire the lock signal.

Flash*Freeze Pin Locations

Refer to the *Pin Descriptions* chapter of the handbook for information regarding Flash*Freeze pin location on the available packages. The Flash*Freeze pin location is independent of the device, allowing migration to larger or smaller devices while maintaining the same pin location on the board.

Sleep and Shutdown Modes

Sleep Mode

Actel IGLOO, IGLOO PLUS, ProASIC3L, and RT ProASIC3 FPGAs support Sleep mode when device functionality is not required. In Sleep mode, the FPGA core voltage supply (V_{CC}) is turned off (either grounded or floated) while other power supplies are left on, resulting in the FPGA core being turned off to reduce power consumption. While the device is in Sleep mode, the rest of the system can still be operating and driving the input buffers of the device. The driven inputs do not pull up the internal power planes, and the current draw is limited to minimal leakage current.

Table 2-7 shows the power supply status in Sleep mode. When the V_{CC} power supply is powered off, the corresponding power pin can be left floating or grounded.

Table 2-7 • **Sleep Mode—Power Supply Requirement for IGLOO, IGLOO PLUS, ProASIC3L, and RT ProASIC3 Devices**

Power Supplies	Power Supply State
V_{CC}	Powered off
V_{CCI} = VMV	Powered on
V_{JTAG}	Powered on
V_{PUMP}	Powered on

Refer to the "Power-Up/-Down Behavior" section on page 2-12 for more information about I/O states during Sleep mode and the timing diagram for entering and exiting Sleep mode.

Shutdown Mode

Shutdown mode is supported for all IGLOO PLUS devices, AGL015, AGL030, AGLE600, AGLE3000, and A3PE3000L. Shutdown mode can be used by turning off all power supplies when the device function is not needed. Cold-sparing and hot-insertion features enable these devices to be powered down without turning off the entire system. When power returns, the live-at-power-up feature enables operation of the device after reaching the voltage activation point.

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. Actel recommends that power FETs with low resistance be used to perform the switching action.

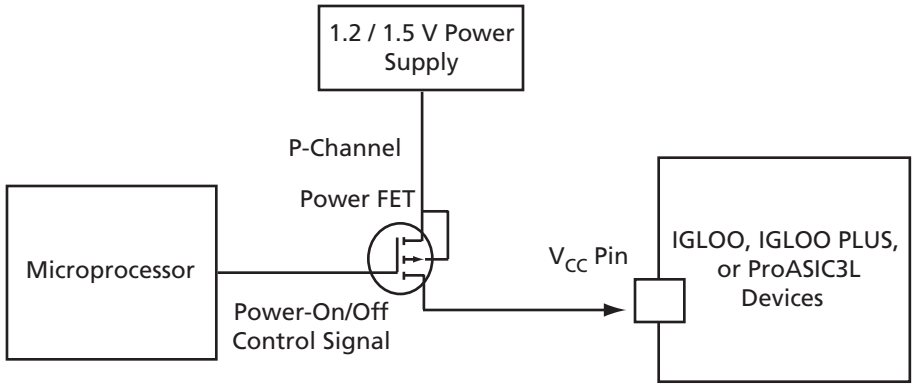


Figure 2-6 • **Controlling Power-On/-Off State Using Microprocessor and Power FET**

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.

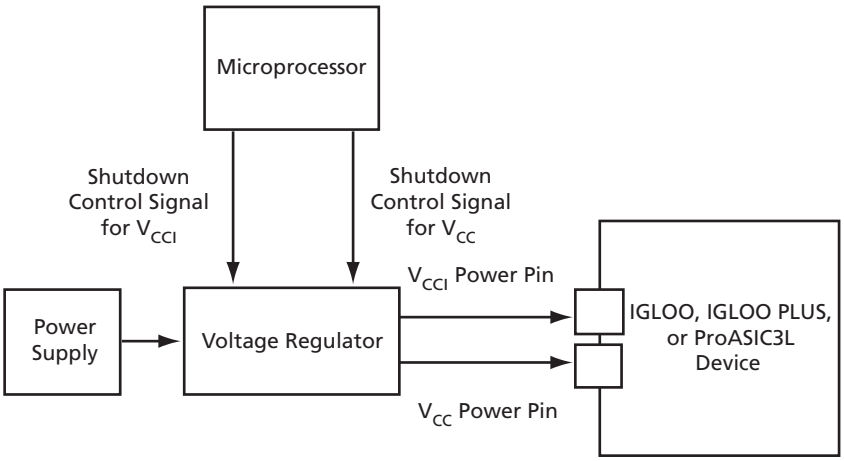


Figure 2-7 • **Controlling Power-On/-Off State Using Microprocessor and Voltage Regulator**

Power-Up/-Down Behavior

By design, all IGLOO, IGLOO 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 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 µs (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 power supply and board-level configurations, the user can easily calculate how long it will take for the core to become inactive or active. For more information, refer to *Power-Up/-Down Behavior of Low-Power Flash Devices*.

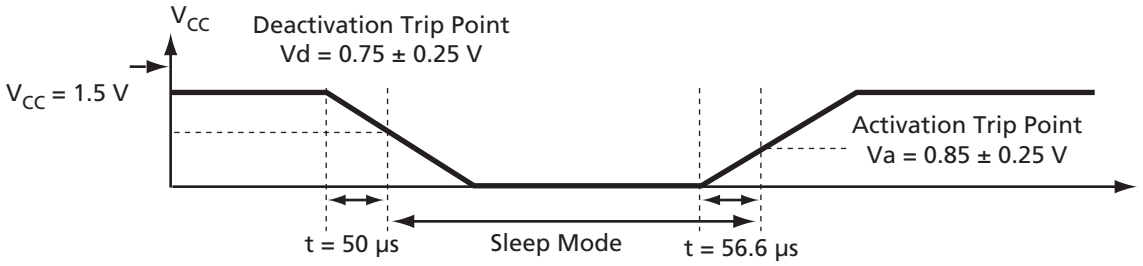


Figure 2-8 • **Entering and Exiting Sleep Mode, Typical Timing Diagram**

Context Save and Restore in Sleep or Shutdown Mode

In Sleep mode or Shutdown mode, the contents of the SRAM, state of the I/Os, and state of the registers are lost when the device is powered off, if no other measure is taken. A low-cost external serial EEPROM can be used to save and restore the contents of the device when entering and exiting Sleep mode or Shutdown mode. In the *Embedded SRAM Initialization Using External Serial EEPROM* application note, detailed information and a reference design are provided for initializing the embedded SRAM using an external serial EEPROM. The user can easily customize the reference design to save and restore the FPGA state when entering and exiting Sleep mode or Shutdown mode. The microcontroller will need to manage this activity; hence, before powering down V_{CC}, the data will be read from the FPGA and stored externally. In a similar way, after the FPGA is powered up, the microcontroller will allow the FPGA to load the data from external memory and restore its original state.

Flash*Freeze Design Guide

This section describes how designers can create reliable designs that use ultra-low power Flash*Freeze modes optimally. The section below provides guidance on how to select the best Flash*Freeze mode for any application. The "Design Solutions" section on page 2-14 gives specific recommendations on how to design and configure clocks, set/reset signals, and I/Os. This section also gives an overview of the design flow and provides details concerning Actel's Flash*Freeze Management IP, which enables clean clock gating and housekeeping. The "Additional Power Conservation Techniques" section on page 2-20 describes board-level considerations for entering and exiting Flash*Freeze mode.

Selecting the Right Flash*Freeze Mode

Both Flash*Freeze modes will bring an FPGA into an ultra-low-power static mode that retains register and SRAM content and sets I/Os to a predetermined configuration. There are two primary differences that distinguish type 2 mode from type 1, and they must be considered when creating a design using Flash*Freeze technology.

First, with type 2 mode, the device has an opportunity to wait for a second signal to enable activation of Flash*Freeze mode. This allows processes to complete prior to deactivating the device, and can be useful to control task completion, data preservation, accidental Flash*Freeze activation, system shutdown, or any other housekeeping function. The second signal may be derived from an external or in-to-out internal source. The second difference between type 1 and type 2 modes is that a design for type 2 mode has an opportunity to cleanly manage clocks and data activity before entering and exiting Flash*Freeze mode. This is particularly important when data preservation is needed, as it ensures valid data is stored prior to entering, and upon exiting, Flash*Freeze mode.

Type 1 Flash*Freeze mode is ideally suited for applications with the following design criteria:

- Entering Flash*Freeze mode is not dependent on any signal other than the external FF pin.
- Internal housekeeping is not required prior to entering Flash*Freeze.
- 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 Actel's Flash*Freeze management IP offers a very low tile count clock and data management solution. Actel's recommendation for most designs is to use type 2 Flash*Freeze mode with Flash*Freeze management IP.

Design Solutions

Clocks

- Actel recommends using a completely synchronous design in Type 2 mode with Flash*Freeze management IP cleanly gate all internal and external clocks. This will prevent narrow pulses upon entrance and exit from Flash*Freeze mode (Figure 2-5 on page 2-9).
- Upon entering Flash*Freeze mode, external clocks become tied off High, internal to the clock pin (unless hold state is used on IGLOO PLUS), and PLLs will be 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, Actel 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, Actel recommends gating them to a known value (preferably '1', to avoid possible narrow pulse upon Flash*Freeze mode exit), and not left 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.

Set/Reset

- Since all I/Os and globals are tied High in Flash*Freeze mode (unless IGLOO PLUS hold state is used), Actel recommends using active Low set/reset at the top-level port. If needed, the signal can be inverted internally.
- If the intention is to always set/reset in Flash*Freeze mode, a self set/reset circuit may be implemented to accomplish this, as shown in Figure 2-9. Configure an active High set/reset input pin so it uses the internal pull-up during Flash*Freeze mode, and drives Low during active mode. When the device exits Flash*Freeze mode, the input will transition from High to Low, releasing the set/reset. Note that this circuit may release set/reset before all outputs become active, since outputs are enabled up to 200 ns after inputs when exiting Flash*Freeze mode.

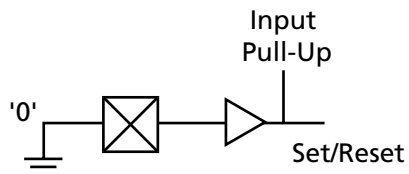


Figure 2-9 • **Flash*Freeze Self-Reset Circuit**

I/Os

- Floating inputs can cause totem pole currents on the input I/O circuitry when the device is in active mode. If inputs will be released (undriven) during Flash*Freeze mode, Actel recommends that they are only released after the device enters Flash*Freeze mode.
- As mentioned earlier, asynchronous input to output paths are subject to possible glitching when entering Flash*Freeze mode. For example, on a direct in-to-out path, if the current state is '0' and the input bank deactivates first, the input and then the output will transition to '1' before the output enters its Flash*Freeze state. This can be prevented by using latches along with Flash*Freeze management IP to gate asynchronous in-to-out paths prior to entering Flash*Freeze mode.

JTAG

- The JTAG state machine is powered, but not active during Flash*Freeze mode.
- TCK should be held in a static state to prevent dynamic power consumption of the JTAG circuit during Flash*Freeze.
- Specific JTAG pin tie-off recommendations suitable for Flash*Freeze mode can be found in the *Pin Descriptions* chapter of the handbook.

ULSICC

- The User Low Static ICC (ULSICC) macro acts as an access point to the hard Flash*Freeze technology block in the device. The ULSICC macro represents a hard, fixed location block in the device. When the LSICC input of the ULSICC macro is driven Low, the Flash*Freeze pin is blocked, and when LSICC is driven High, the Flash*Freeze pin is enabled.
- If the user decides to build his/her own Flash*Freeze type 2 clock and data management logic, note that the LSICC signal on the ULSICC macro is ANDed internally with the Flash*Freeze signal. In order to reliably enter Flash*Freeze, the LSICC signal must remain asserted High while entering and during Flash*Freeze mode.

Flash*Freeze Management IP

One of the key benefits of Actel's Flash*Freeze mode is the ability to preserve the state of all internal registers, SRAM content, and I/Os (IGLOO PLUS only). This feature enables seamless continuation of data processing before and after Flash*Freeze, without the need to reload or reinitialize the FPGA system. Actel's Flash*Freeze management IP, available for type 2 implementation, offers a robust RTL block that ensures clean clock gating of all system clocks before entering and upon exiting Flash*Freeze mode. This IP also gives users the option to perform housekeeping prior to entering Flash*Freeze mode. This section will provide an overview of the Flash*Freeze management IP. Additional information on this IP core can be found in the Libero IDE 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.

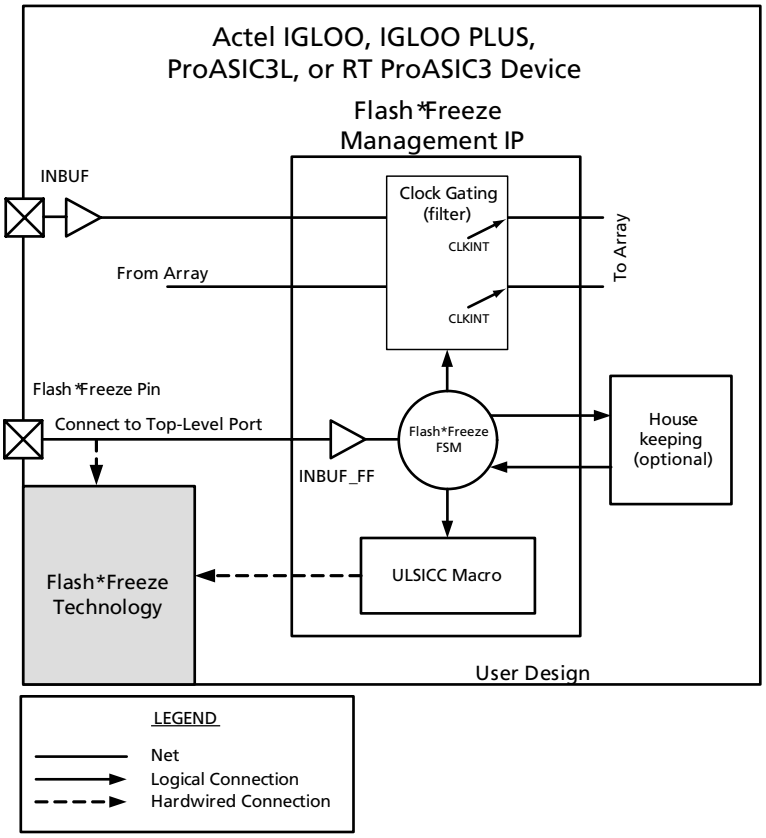


Figure 2-10 • **Flash*Freeze Management IP Block Diagram**

*Flash*Freeze Management FSM*

The Flash*Freeze FSM block is a simple, robust, fully encoded 3-bit state machine that ensures clean entrance 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 2-12 on page 2-18, below. 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 IDE will connect WAIT_HOUSEKEEPING to DONE_HOUSEKEEPING.

Clock Gating Block

Once DONE_HOUSEKEEPING is detected, the FSM will initiate the clock gating circuit by asserting ASSERT_GATE (active Low). ASSERT_GATE is named control_user_clock_net in the IP block. Upon assertion of the ASSERT_GATE signal, the clock will be gated in less than 2 cycles. The clock gating circuit is comprised of a flip-flop, latch, AND gate, and CLKINT, as shown in Figure 2-11. The clock gating block can support gating of up to 17 clocks.

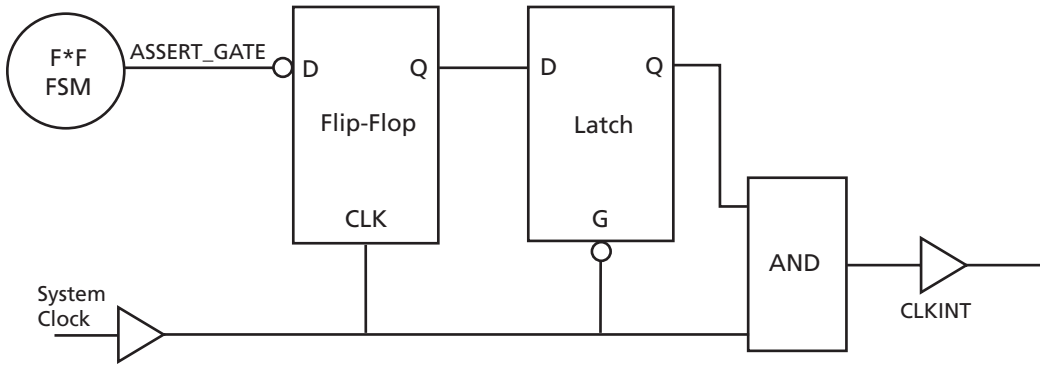


Figure 2-11 • **Clock Gating Circuit**

After initiating the clock gating circuit, the FSM will assert and hold the LSICC signal (active High), feeding the ULSICC macro. This will initiate the 1 µs entrance into Flash*Freeze mode.

Upon deassertion of the Flash*Freeze pin, the FSM will set ASSERT_GATE High. Once the I/O banks become active, the clock will enter the device, and register the ASSERT_GATE signal, cleanly releasing the clock gate.

Design Flow[1]

Actel has developed a convenient and intuitive design flow for configuring and integrating Flash*Freeze technology into an FPGA design. Flash*Freeze type 1 is implemented by instantiating the INBUF_FF macro in the top level of a design. Flash*Freeze type 2 with management IP can be generated by the Libero IDE core generator or SmartGen and instantiated as a single block in the user's design. This single block will include an INBUF_FF macro and the optional Flash*Freeze management IP, which includes the ULSICC macro. If designers do not wish to use this core generator, the INBUF_FF macro and the optional ULSICC macro may be instantiated in the design, and custom Flash*Freeze management IP can be developed by the user. The remainder of this section will cover configuration details of the INBUF_FF macro, the ULSICC macro, and the Flash*Freeze management IP.

Additional information on the tools discussed within this section may be found in the Libero IDE online help.

INBUF_FF

The INBUF_FF macro is a special-purpose input buffer macro that is interpreted downstream in the design flow by Actel's Designer software. When this macro is used, the top-level port will be forced to the dedicated FF pin in the FPGA, and Flash*Freeze mode will be available for use in the device. The following are the design rules for INBUF_FF:

- If INBUF_FF is not used in the design, the device will not be configured to support Flash*Freeze mode.
- When the INBUF_FF macro is used, the FF pin will establish a hardwired connection to the Flash*Freeze technology circuit in the device, as shown in Figure 2-1 on page 2-5, Figure 2-3

1. *This section applies to Libero IDE / Designer v8.3 and later. Actel recommends that designs created in earlier versions of the software be modified to accommodate this flow by instantiating the INBUF_FF macro or the Flash*Freeze management IP. Refer to the Libero IDE / Designer v8.3 release notes and the Libero IDE online help for more information on migrating designs from older software versions.*

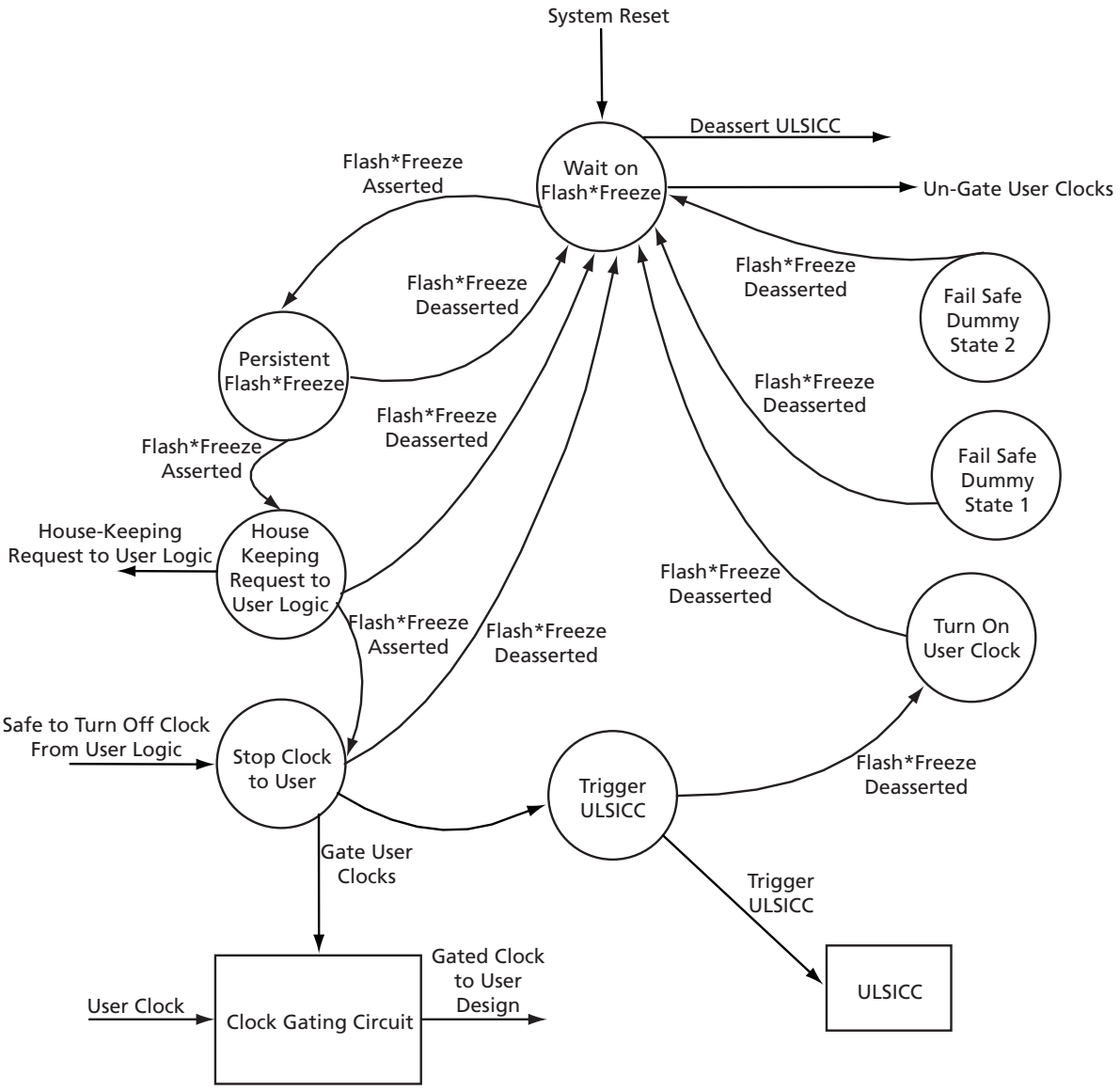


Figure 2-12 • **FSM State Diagram**

on page 2-6, and Figure 2-10 on page 2-16 and described "Flash*Freeze Type 1: Control by Dedicated Flash*Freeze Pin" section on page 2-4.

- 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 V_{CCI} 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 Actel's Designer software. The Schmitt trigger option is only available for IGLOOe, 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 2-6 and Figure 2-10 on page 2-16. 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 IDE (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
```

- There will be added skew and clock insertion delay due to the clock gating circuit. The user should analyze external setup/hold times carefully. The user should also ensure the additional skew across the clock gating filter circuit is accounted for in any paths where the launch register is driven from the filter input clock and captured by a register driven by the gated clock filter output clock.

Power Analysis

SmartPower identifies static and dynamic power consumption problems quickly within a design. It provides a hierarchical view, allowing users to drill down and estimate the power consumption of individual components or events. SmartPower analyzes power consumption for nets, gates, I/Os, memories, clocks, cores, clock domains, power supply rails, peak power during a clock cycle, and switching transitions.

SmartPower generates detailed hierarchical reports of the dynamic power consumption of a design for easy inspection. These reports include design-level power summary, average switching activity, and ambient and junction temperature readings. Input the target clock and data frequencies for a design, and let SmartPower perform a detailed and accurate power analysis. SmartPower supports importing files in the VCD (Value-Change Dump) format as specified in the IEEE 1364 standard. It also supports the Synopsys® Switching Activity Interchange Format (SAIF) standard. Support for these formats lets designers generate switching activity information in a variety of simulators and then import this information directly into SmartPower.

For portable or battery operated applications, a power profile feature enables you to measure power and battery life, based on a sequence of operational modes of the design. In most portable and battery-operated applications, the system is seldom fully "on" 100 percent of the time. "On" is a combination of fully active, standby, sleep, or other functional modes. SmartPower allows users to create a power profile for a design by specifying operational modes and the percent of time the device will run in each of the modes. Power is calculated for each of the modes, and total power is calculated based on the weighted average of all modes.

SmartPower also provides an estimated battery life based on the power profile. The current capacity for a given battery is input and used to estimate the life of the battery. The result is an accurate and realistic indication of battery life.

More information on SmartPower can be found on the Actel website:
http://www.actel.com/products/software/libero/smartpower.aspx.

Additional Power Conservation Techniques

IGLOO and ProASIC3L FPGAs provide many ways to inherently conserve power; however, there are also several design techniques that can be used to reduce power on the board.

- Actel recommends that the designer use the minimum number of I/O banks possible and tie any unused power supplies (such as V_{CCPLL}, V_{CCI}, VMV, and V_{PUMP}) to ground.
- Leave unused I/O ports floating. Unused I/Os are configured by the software as follows:
  - Output buffer is disabled (with tristate value of high impedance)
  - Input buffer is disabled (with tristate value of high impedance)
- Use the lowest available voltage I/O standard, the lowest drive strength, and the slowest slew rate to reduce I/O switching contribution to power consumption.
- Advanced and pro I/O banks may consume slightly higher static current than standard and standard plus banks—avoid using advanced and pro banks whenever practical.
  - The small static power benefit obtained by avoiding advanced or pro I/O banks is usually negligible compared to the benefit of using a low-power I/O standard.
- Deselect RAM blocks that are not being used.
- Only enable read and write ports on RAM blocks when they are needed.
- Gating clocks LOW offers improved static power of RAM blocks.
- Drive the FF port of RAM blocks with the Flash_Freeze_Enabled signal from the Flash*Freeze management IP.
- Drive inputs to the full voltage level so that all transistors are turned on or off completely.
- Avoid using pull-ups and pull-downs on I/Os because these resistors draw some current. Avoid driving resistive loads or bipolar transistors, since these draw a continuous current, thereby adding to the static current.
- When partitioning the design across multiple devices, minimize I/O usage among the devices.

Prototyping for IGLOO and ProASIC3L Devices Using ProASIC3

Prototyping in ProASIC3 does not apply for the IGLOO PLUS family. The IGLOO, ProASIC3L, and ProASIC3 families are architecturally compatible with the exception of the Flash*Freeze technology implementation and the Flash*Freeze pin available on the IGLOO and ProASIC3L devices. For example, the AGL125 user can start prototyping with the A3P125, since the devices share the same features and pinout, except for the Flash*Freeze pin. The pin in that location for ProASIC3 devices is an I/O pin.

If Flash*Freeze mode will be used in the IGLOO or ProASIC3L device, prototyping using the equivalent ProASIC3 device could implement a Flash*Freeze testing pin at the same location in the ProASIC3 package as the actual Flash*Freeze pin on the IGLOO or ProASIC3L device, as listed in Table 2-8.

You can use the Flash*Freeze testing pin on ProASIC3 devices to activate the ULSICC macro internally. Migrating to the IGLOO device is easy, as the timing of the externally controlled ULSICC macro is similar to that of the Flash*Freeze pin. Keep in mind that the activation trigger for the ULSICC macro is active High, whereas the activation trigger for the Flash*Freeze pin is active Low. This means that to emulate entering Flash*Freeze mode using the ULSICC macro in a ProASIC3 device, the external signal should be inverted at the input.

If Flash*Freeze mode will not be used, the Flash*Freeze pin can still be used as a regular user I/O, making it possible to use the same PCB for both the IGLOO and ProASIC3L families. Table 2-8 provides detailed information on prototyping for IGLOO and ProASIC3L families.

Table 2-8 • **Prototyping/Migration Solutions**

Scenario	Board-Level	Software/Designer
Prototype with existing ProASIC3 devices and plan to migrate to IGLOO or ProASIC3L devices to take advantage of ultra-low power consumption. Flash*Freeze feature is not needed.	Since the Flash*Freeze pin is not used, the board layout is the same for all IGLOO and ProASIC3 families. **Note:** Use 1.5 V V_{CC} for IGLOO and ProASIC3. IGLOO and ProASIC3L can support both 1.5 V and 1.2 V core voltage. The 1.2 V core voltage results in lower power consumption.	1. *Open the ProASIC3 ADB file in Designer.* 2. *Export the netlist and PDC files.* 3. *Create a new ADB file targeting the IGLOO or ProASIC3L device and import the files created in step 2.* 4. *Run **Compile** (disable Flash*Freeze mode), run **Layout**, and perform post-layout simulation and timing analysis.*
Prototype with existing ProASIC3 devices and plan to migrate to IGLOO or ProASIC3L devices to take advantage of ultra-low power consumption. Flash*Freeze mode will be used to reduce power consumption when device function is not needed. ULSICC mode is optional.	Users must ensure the equivalent Flash*Freeze pin location is not used as a user I/O. **Note:** Use 1.5 V V_{CC} for IGLOO and ProASIC3. IGLOO and ProASIC3L can support both 1.5 V and 1.2 V core voltage. The 1.2 V core voltage results in lower power consumption.	1. *Open the ProASIC3 ADB file in Designer.* 2. *Export the netlist and PDC files.* 3. *If needed, create a PDC file to define the Flash*Freeze mode I/O state and port name for the Flash*Freeze pin.* 4. *Create a new ADB file targeting the IGLOO or ProASIC3L devices and import the files created in step 2.* 5. *Run **Compile** (enable Flash*Freeze mode), run **Layout**, and perform post-layout simulation and timing analysis.*

Note: Device migration is not supported for IGLOO PLUS devices.

Export Files

To export the netlist and constraint files from Designer, follow the steps below:

1. Export the Netlist in Designer.

   File > **Export** > **Netlist Files** > *.edn

2. Export the PDC file in Designer.

   File > **Export** > **Constraint Files** > *.pdc > select **Complete placement information** (Figure 2-13).

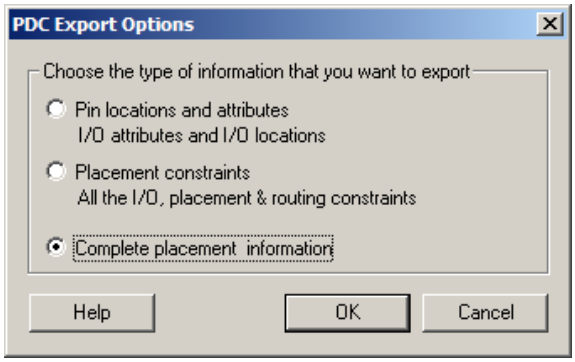


Figure 2-13 • **Export Netlist**

Table 2-9 and Table 2-10 on page 2-23 show the device-to-device and packages compatibility list.

Table 2-9 • **Device Migration—IGLOO Supported Packages in ProASIC3 Devices**

Package	A3P030 AGL030	A3P060 AGL060	A3P125 AGL125	A3P250 AGL250	A3P600 AGL600	A3P1000 AGL1000	A3PE600 AGLE600	A3PE3000 AGLE3000
QN132	✓	✓	✓	✓				
VQ100	✓	✓	✓	✓				
FG144		✓	✓	✓	✓	✓		
CS196		IGLOO and IGLOOe devices only						
FG256				ProASIC3 only	✓	✓	✓	
FG484					✓	✓	✓	✓
FG896								✓

Note: IGLOO PLUS and RT ProASIC3 do not support device migration.

Table 2-10 • **Device Migration—ProASIC3L Supported Packages in ProASIC3 Devices**

Package	A3P250 A3P250L	A3P600 A3P600L	A3P1000 A3P1000L	A3PE3000 A3PE3000L
VQ100	✓			
PQ208	✓	✓	✓	✓
FG144	✓	✓	✓	
FG256	✓	✓	✓	
FG324				ProASIC3L only
FG484		✓	✓	✓
FG896				✓

Note: IGLOO PLUS and RT ProASIC3 do not support device migration.

Conclusion

Actel IGLOO, IGLOO PLUS, ProASIC3L, and RT ProASIC3 family architectures are designed to achieve ultra-low power consumption based on enhanced nonvolatile and live-at-power-up flash-based technology. Power consumption can be reduced further by using Flash*Freeze, Static (Idle), Sleep, and Shutdown power modes. All these features result in a low-power, cost-effective, single-chip solution designed specifically for power-sensitive and battery-operated electronics applications.

Related Documents

Application Notes

Embedded SRAM Initialization Using External Serial EEPROM

http://www.actel.com/documents/EmbeddedSRAMInit_AN.pdf

Handbook Documents

Power-Up/-Down Behavior of Low-Power Flash Devices

http://www.actel.com/documents/LPD_PowerUp_HBs.pdf

Pin Descriptions

http://www.actel.com/documents/LPD_PinDescriptions_HBs.pdf

Part Number and Revision Date

This document contains content extracted from the Device Architecture section of the datasheet, combined with content previously published as an application note describing features and functions of the device. To improve usability for customers, the device architecture information has now been combined with usage information to reduce duplication and possible inconsistencies in published information. No technical changes were made to the datasheet content unless explicitly listed. Changes to the application note content were made only to be consistent with existing datasheet information.

Part Number 51700094-004-5

Revised October 2008

List of Changes

The following table lists critical changes that were made in the current version of the chapter.

Previous Version	Changes in Current Version (v2.1)	Page
v2.0 (October 2008)	The "Flash*Freeze Management FSM" section was updated with the following information: 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.	2-16
v1.3 (June 2008)	The title changed from "Flash*Freeze Technology and Low-Power Modes in IGLOO, IGLOO PLUS, and ProASIC3L Devices" to Actel's Flash*Freeze Technology and Low-Power Modes."	N/A
	The "Actel's Flash Families Support the Flash*Freeze Feature" section was updated.	2-2
	Significant changes were made to this document to support Libero IDE v8.4 and later functionality. RT ProASIC3 device support information is new. In addition to the other major changes, the following tables and figures were updated or are new: Figure 2-3 · Flash*Freeze Mode Type 2 – Controlled by Flash*Freeze Pin and Internal Logic (LSICC signal) – updated Figure 2-5 · Narrow Clock Pulses During Flash*Freeze Entrance and Exit – new Figure 2-10 · Flash*Freeze Management IP Block Diagram– new Figure 2-12 · FSM State Diagram– new Table 2-6 · IGLOO PLUS Flash*Freeze Mode (type 1 and type 2)—I/O Pad State – updated Please review the entire document carefully.	2-6 2-9 2-16 2-18 2-8
v1.2 (March 2008)	The family description for ProASIC3L in Table 2-1 · Low-Power Flash Families was updated to include 1.5 V.	2-2
v1.1 (February 2008)	The part number for this document was changed from 51700094-003-1 to 51700094-004-2.	N/A
	The title of the document was changed to "Flash*Freeze Technology and Low-Power Modes in IGLOO, IGLOO PLUS, and ProASIC3L Devices."	N/A
	The "Flash*Freeze Technology and Low-Power Modes" section was updated to remove the parenthetical phrase, "from 25 µW," in the second paragraph. The following sentence was added to the third paragraph: "IGLOO PLUS has an additional feature when operating in Flash*Freeze mode, allowing it to retain I/O states as well as SRAM and register states."	2-1
	The "Power Conservation Techniques" section was updated to add V_{JTAG} to the parenthetical list of power supplies that should be tied to the ground plane if unused. Additional information was added regarding how the software configures unused I/Os.	2-1
	Table 2-1 · Low-Power Flash Families and the accompanying text was updated to include the IGLOO PLUS family. The "IGLOO Terminology" section and "ProASIC3 Terminology" section are new.	2-2
	The "Flash*Freeze Mode" section was revised to include that I/O states are preserved in Flash*Freeze mode for IGLOO PLUS devices. The last sentence in the second paragraph was changed to, "If the FF pin is not used, it can be used as a regular I/O." The following sentence was added for Flash*Freeze mode type 2: "Exiting the mode is controlled by either the FF pin OR the user-defined LSICC signal."	2-4

Previous Version	Changes in Current Version (v2.1)	Page
v1.1 (continued)	The "Flash*Freeze Type 1: Control by Dedicated Flash*Freeze Pin" section was revised to change instructions for implementing this mode, including instructions for implementation with Libero IDE v8.3.	2-4
	Figure 2-1 · Flash*Freeze Mode Type 1 – Controlled by the Flash*Freeze Pin was updated.	2-5
	The "Refer to Table 2-3 for Flash*Freeze (FF) pin and LSICC signal assertion and deassertion values." section was renamed from "Type 2 Software Implementation."	2-6
	The "Type 2 Software Implementation for Libero IDE v8.3" section is new.	2-6
	Figure 2-3 · Flash*Freeze Mode Type 2 – Controlled by Flash*Freeze Pin and Internal Logic (LSICC signal) was updated.	2-6
	Figure 2-4 · Flash*Freeze Mode Type 2 – Timing Diagram was revised to show deasserting LSICC after the device has exited Flash*Freeze mode.	2-7
	The "IGLOO PLUS I/O State in Flash*Freeze Mode" section was added to include information for IGLOO PLUS devices. Table 2-6 · IGLOO PLUS Flash*Freeze Mode (type 1 and type 2)—I/O Pad State is new.	2-7, 2-8
	The "During Flash*Freeze Mode" section was revised to include a new bullet pertaining to output behavior for IGLOO PLUS. The bullet on JTAG operation was revised to provide more detail.	2-10
	Figure 2-6 · Controlling Power-On/-Off State Using Microprocessor and Power FET and Figure 2-7 · Controlling Power-On/-Off State Using Microprocessor and Voltage Regulator were updated to include IGLOO PLUS.	2-11, 2-12
	The first sentence of the "Shutdown Mode" section was updated to list the devices for which it is supported.	2-11
	The first paragraph of the "Power-Up/-Down Behavior" section was revised. The second sentence was changed to, "The I/Os remain tristated until the last voltage supply (V_{CC} or V_{CCI}) is powered to its activation level." The word "activation" replaced the word "functional." The sentence, "During power-down, device I/Os become tristated once the first power supply (V_{CC} or V_{CCI}) drops below its deactivation voltage level" was revised. The word "deactivation" replaced the word "brownout."	2-12
	The "Prototyping for IGLOO and ProASIC3L Devices Using ProASIC3" section was revised to state that prototyping in ProASIC3 does not apply for the IGLOO PLUS family.	2-21
	Table 2-8 · Prototyping/Migration Solutions, Table 2-9 · Device Migration—IGLOO Supported Packages in ProASIC3 Devices, and Table 2-10 · Device Migration—ProASIC3L Supported Packages in ProASIC3 Devices were updated with a table note stating that device migration is not supported for IGLOO PLUS devices.	2-21, 2-23
	The text following Table 2-10 · Device Migration—ProASIC3L Supported Packages in ProASIC3 Devices was moved to a new section: the "Flash*Freeze Design Guide" section.	2-13
v1.0 (January 2008)	Table 2-1 · Low-Power Flash Families was updated to remove the ProASIC3, ProASIC3E, and Automotive ProASIC3 families, which were incorrectly included.	2-2

Previous Version	Changes in Current Version (v2.1)	Page
51900147-2/5.07	Detailed descriptions of low-power modes are described in the advanced datasheets. This application note was updated to describe how to use the features in an IGLOO/e application.	N/A
	Figure 2-1 · Flash*Freeze Mode Type 1 – Controlled by the Flash*Freeze Pin was updated.	2-5
	Figure 2-2 · Flash*Freeze Mode Type 1 – Timing Diagram is new.	2-5
	Steps 4 and 5 are new in the "Refer to Table 2-3 for Flash*Freeze (FF) pin and LSICC signal assertion and deassertion values." section.	2-6
51900147-1/3.07	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.	2-4
	Figure 2-2 · Flash*Freeze Mode Type 1 – Timing Diagram was updated.	2-5
	Information about ULSICC was added to the "Prototyping for IGLOO and ProASIC3L Devices Using ProASIC3" section.	2-21
51900147-0/8.06	In the "Flash*Freeze Mode" section, "active high" was changed to "active low."	2-4
	The "Prototyping for IGLOO and ProASIC3L Devices Using ProASIC3" section was updated with information concerning the Flash*Freeze pin.	2-21

Global Resources and Clock Conditioning

Actel
POWER MATTERS

3 – Global Resources in Actel Low-Power Flash Devices

Introduction

Actel IGLOO,® Fusion, and ProASIC®3 FPGA devices offer a powerful, low-delay VersaNet global network scheme and have extensive support for multiple clock domains. In addition to the Clock Conditioning Circuits (CCCs) and phase-locked loops (PLLs), there is a comprehensive global clock distribution network called a VersaNet global network. Each logical element (VersaTile) input and output port has access to these global networks. The VersaNet global networks can be used to distribute low-skew clock signals or high-fanout nets. In addition, these highly segmented VersaNet global networks offer users the flexibility to create low-skew local networks using spines. This document describes VersaNet global networks and discusses how to assign signals to these global networks and spines in a design flow. Details concerning low-power flash device PLLs are described in *Clock Conditioning Circuits in IGLOO and ProASIC3 Devices*. This document describes the low-power flash devices' global architecture and uses of these global networks in designs.

Global Architecture

Low-power flash devices offer powerful and flexible control of circuit timing through the use of analog circuitry. Each chip has up to six CCCs, some with PLLs.

- In IGLOOe, ProASIC3EL, and ProASIC3E devices, all CCCs have PLLs—hence, 6 PLLs per device.
- In IGLOO, IGLOO PLUS, ProASIC3L, and ProASIC3 devices, the west CCC contains a PLL core (except in 15 k and 30 k devices).
- In Fusion devices, the west CCC also contains a PLL core. In the two larger devices (AFS600 and AFS1500), the west and east CCCs each contain a PLL.

Each PLL includes delay lines, a phase shifter (0°, 90°, 180°, 270°), and clock multipliers/dividers. Each CCC has all the circuitry needed for the selection and interconnection of inputs to the VersaNet global network. The east and west CCCs each have access to three VersaNet global lines on each side of the chip (six global lines total). The CCCs at the four corners each have access to three quadrant global lines in each quadrant of the chip (except in 15 k gate and 30 k gate devices).

In 15 k and 30 k gate devices, all six VersaNet global lines are driven from three southern I/Os, located toward the east and west sides. Each of these tiles can be configured to select a central I/O on its respective side or an internal routed signal as the input signal. 15 k and 30 k gate devices do not support any clock conditioning circuitry, nor do they contain the VersaNet global network concept of top and bottom spines.

The flexible use of the VersaNet global network allows the designer to address several design requirements. User applications that are clock-resource-intensive can easily route external or gated internal clocks using VersaNet global routing networks. Designers can also drastically reduce delay penalties and minimize resource usage by mapping critical, high-fanout nets to the VersaNet global network.

The following sections give an overview of the VersaNet global network, the structure of the global network, and the clock aggregation feature that enables a design to have very low clock skew using spines.

Global Resource Support in Low-Power Devices

The low-power flash families listed in Table 3-1 support the global resources and the functions described in this document.

Table 3-1 • **Low-Power Flash Families**

Product Line	Family*	Description
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
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 PLUS	IGLOO FPGAs with enhanced I/O capabilities
ProASIC3	ProASIC3	Low-power, high-performance 1.5 V FPGAs
	ProASIC3E	Higher density ProASIC3 FPGAs with six PLLs and additional I/O standards
	ProASIC3L	ProASIC3 FPGAs supporting 1.2 V to 1.5 V with Flash*Freeze technology
	Automotive ProASIC3	ProASIC3 FPGAs qualified for automotive applications
	Military ProASIC3/EL	Military temperature A3PE600L, A3P1000, and A3PE3000L
	RT ProASIC3	Radiation-tolerant RT3PE600L and RT3PE3000L

*Note: *The family names link to the appropriate datasheet, including product brief, DC and switching characteristics, and packaging information.*

IGLOO Terminology

In documentation, the terms IGLOO families and IGLOO devices refer to all of the IGLOO products as listed in Table 3-1. Where the information applies to only one family or limited devices, these exclusions will be explicitly stated.

ProASIC3 Terminology

In documentation, the terms ProASIC3 families and ProASIC3 devices refer to all of the ProASIC3 families as listed in Table 3-1. Where the information applies to only one family or limited devices, these exclusions will be explicitly stated.

To further understand the differences between the IGLOO and ProASIC3 families, refer to the *Industry's Lowest Power FPGAs Portfolio*.

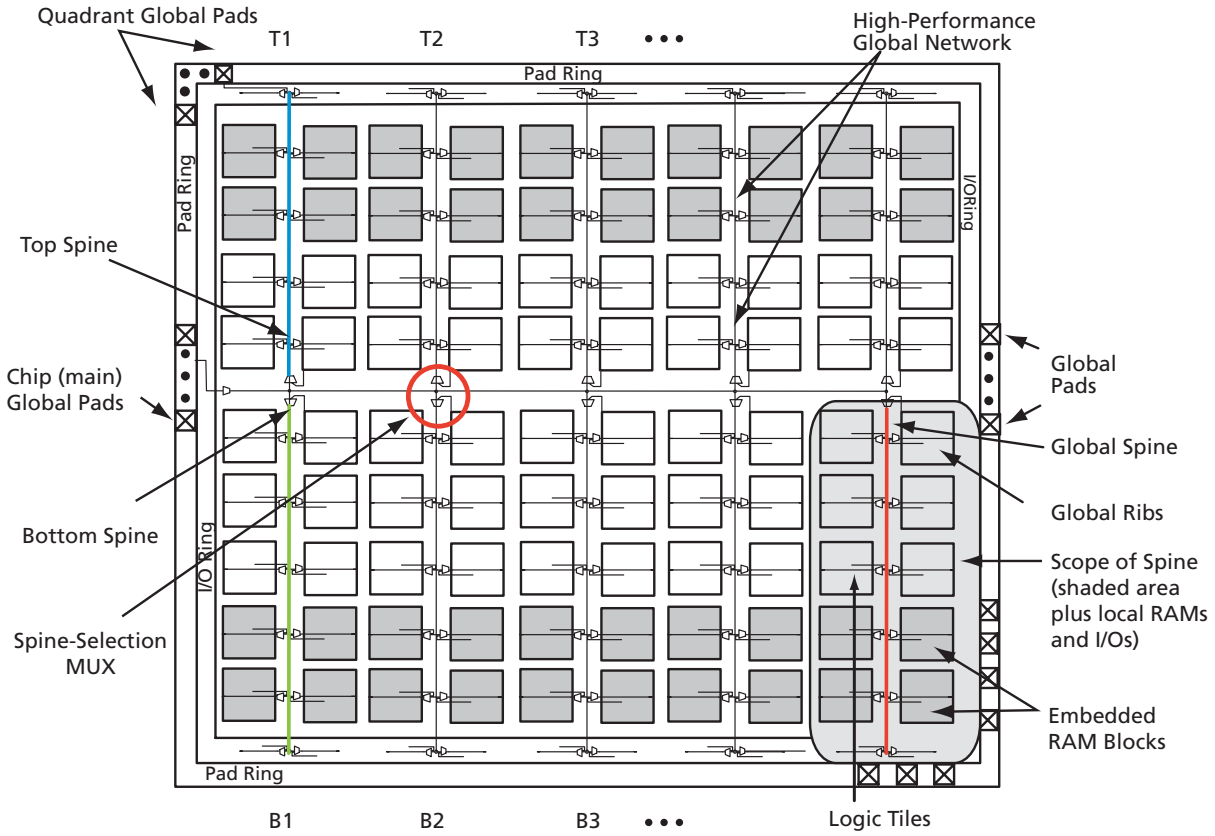
VersaNet Global Network Distribution

One of the architectural benefits of low-power flash architecture is the set of powerful, low-delay VersaNet global networks that can access the VersaTiles, SRAM, and I/O tiles of the device. Each device offers a chip global network with six global lines that are distributed from the center of the FPGA array. In addition, each device, (except the 15 k and 30 k gate device), has four quadrant global networks, each with three regional global line resources. These quadrant global networks can only drive a signal inside their own quadrant. Each core VersaTile has access to nine global line resources—three quadrant and six chip-wide (main) global networks—and a total of 18 globals are available on the device (3 × 4 regional from each quadrant and 6 global).

Figure 3-1 shows simplified device architecture, and Figure 3-2 on page 3-4 shows an overview of the VersaNet global networks.

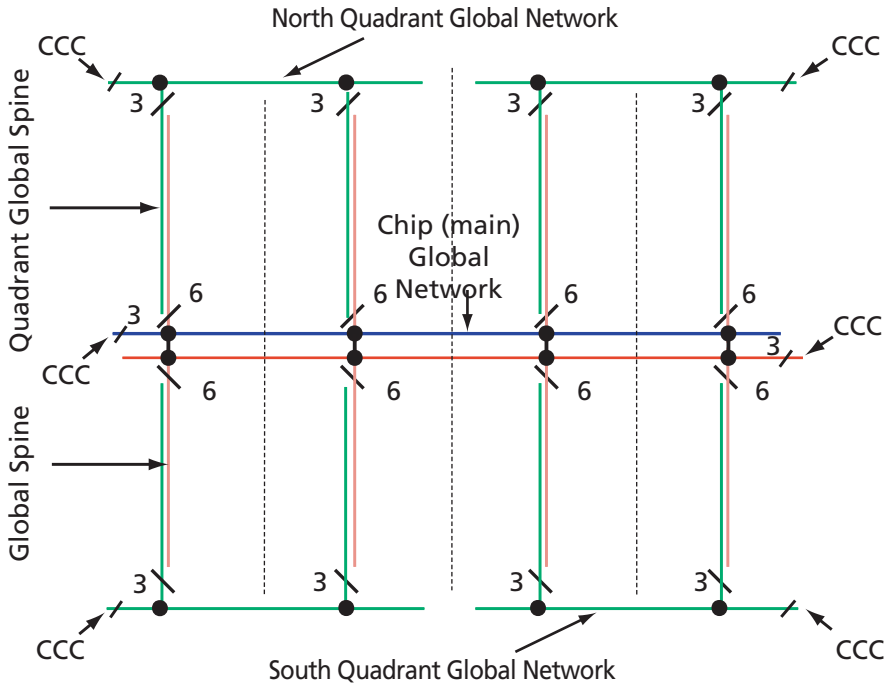
The VersaNet global networks are segmented and consist of VersaNet global networks, spines, global ribs, and global multiplexers (MUXes), as shown in Figure 3-1. The global networks are driven from the global rib at the center of the die or quadrant global networks at the north or south side of the die. The global network uses the MUX trees to access the spine, and the spine uses the clock ribs to access the VersaTile. Access is available to the chip or quadrant global networks and the spines through the global MUXes. Access to the spine using the global MUXes is explained in the "Spine Architecture" section on page 3-4.

These VersaNet global networks offer fast, low-skew routing resources for high-fanout nets, including clock signals. In addition, these highly segmented global networks offer users the flexibility to create low-skew local networks using spines for up to 252 internal/external clocks or other high-fanout nets in low-power flash devices. Optimal usage of these low-skew networks can result in significant improvement in design performance.



Note: Not applicable to 15 k and 30 k gate devices

Figure 3-1 • **Overview of VersaNet Global Network and Device Architecture**



Note: Not applicable to 15 k and 30 k gate devices.

Figure 3-2 • **Simplified VersaNet Global Network**

Spine Architecture

The low-power flash device architecture allows the VersaNet global networks to be segmented. Each of these networks contains spines (the vertical branches of the global network tree) and ribs that can reach all the VersaTiles inside its region. The nine spines available in a vertical column reside in global networks with two separate regions of scope: the quadrant global network, which has three spines, and the chip (main) global network, which has six spines. Note that there are three quadrant spines in each quadrant of the device (except in 15 k and 30 k gate devices). There are four quadrant global network regions per device. In 15 k and 30 k gate devices, there is no quadrant clock network, so there are only six spines in each spine tree. The spines are the vertical branches of the global network tree, shown in Figure 3-2.

Each spine in a vertical column of a chip (main) global network is further divided into two equal-length spine segments: one in the top and one in the bottom half of the die (except in 15 k and 30 k gate devices).

Top and bottom spine segments radiating from the center of a device have the same height. However, just as in the ProASIC^{PLUS}® family, signals assigned only to the top and bottom spine cannot access the middle two rows of the die. The spines for quadrant clock networks do not cross the middle of the die and cannot access the middle two rows of the architecture.

Each spine and its associated ribs cover a certain area of the device (the "scope" of the spine; see Figure 3-2). Each spine is accessed by the dedicated global network MUX tree architecture, which defines how a particular spine is driven—either by the signal on the global network from a CCC, for example, or by another net defined by the user. Details of the chip (main) global network spine-selection MUX are presented in Figure 3-4 on page 3-7. The spine drivers for each spine are located in the middle of the die.

Quadrant spines can be driven from user I/Os on the north and south sides of the die. The ability to drive spines in the quadrant global networks can have a significant effect on system performance for high-fanout inputs to a design. Access to the top quadrant spine regions is from the top of the die, and access to the bottom quadrant spine regions is from the bottom of the die. The A3PE3000 device has 28 clock trees and each tree has nine spines; this flexible global network architecture enables users to map up to 252 different internal/external clocks in an A3PE3000 device.

Table 3-2 • **Globals/Spines/Rows for IGLOO and ProASIC3 Devices**

ProASIC3/ ProASIC3L Devices	IGLOO Devices	Chip Globals	Quadrant Globals (4×3)	Clock Trees	Globals/ Spines per Tree	Total Spines per Device	VersaTiles in Each Tree	Total VersaTiles	Rows in Each Spine
A3P015	AGL015	6	0	1	9	9	384	384	12
A3P030	AGL030	6	0	2	9	18	384	768	12
A3P060	AGL060	6	12	4	9	36	384	1,536	12
A3P125	AGL125	6	12	8	9	72	384	3,072	12
A3P250/L	AGL250	6	12	8	9	72	768	6,144	24
A3P400		6	12	12	9	108	768	9,216	24
A3P600/L	AGL600	6	12	12	9	108	1,152	13,824	36
A3P1000/L	AGL1000	6	12	16	9	144	1,536	24,576	48
A3PE600/L	AGLE600	6	12	12	9	108	1,120	13,440	35
A3PE1500		6	12	20	9	180	1,888	37,760	59
A3PE3000/L	AGLE3000	6	12	28	9	252	2,656	74,368	83

Table 3-3 • **Globals/Spines/Rows for IGLOO PLUS Devices**

IGLOO PLUS Devices	Chip Globals	Quadrant Globals (4×3)	Clock Trees	Globals/ Spines per Tree	Total Spines per Device	VersaTiles in Each Tree	Total VersaTiles	Rows in Each Spine
AGLP030	6	0	2	9	18	384*	792	12
AGLP060	6	12	4	9	36	384*	1,584	12
AGLP125	6	12	8	9	72	384*	3,120	12

*Note: *Clock trees that are located at far left and far right will support more VersaTiles.*

Table 3-4 • **Globals/Spines/Rows for Fusion Devices**

Fusion Device	Chip Globals	Quadrant Globals (4×3)	Clock Trees	Globals/ Spines per Tree	Total Spines per Device	VersaTiles in Each Tree	Total VersaTiles	Rows in Each Spine
AFS090	6	12	6	9	54	384	2,304	12
AFS250	6	12	8	9	72	768	6,144	24
AFS600	6	12	12	9	108	1,152	13,824	36
AFS1500	6	12	20	9	180	1,920	38,400	60

Spine Access

The physical location of each spine is identified by the letter 'T' (top) or 'B' (bottom) and an accompanying number (T*n* or B*n*). The number *n* indicates the horizontal location of the spine; 1 refers to the first spine on the left side of the die. Since there are six chip spines in each spine tree, there are up to six spines available for each combination of 'T' (or 'B') and *n* (for example, six T1 spines). Similarly, there are three quadrant spines available for each combination of 'T' (or 'B') and *n* (for example, four T1 spines), as shown in Figure 3-3 on page 3-6.

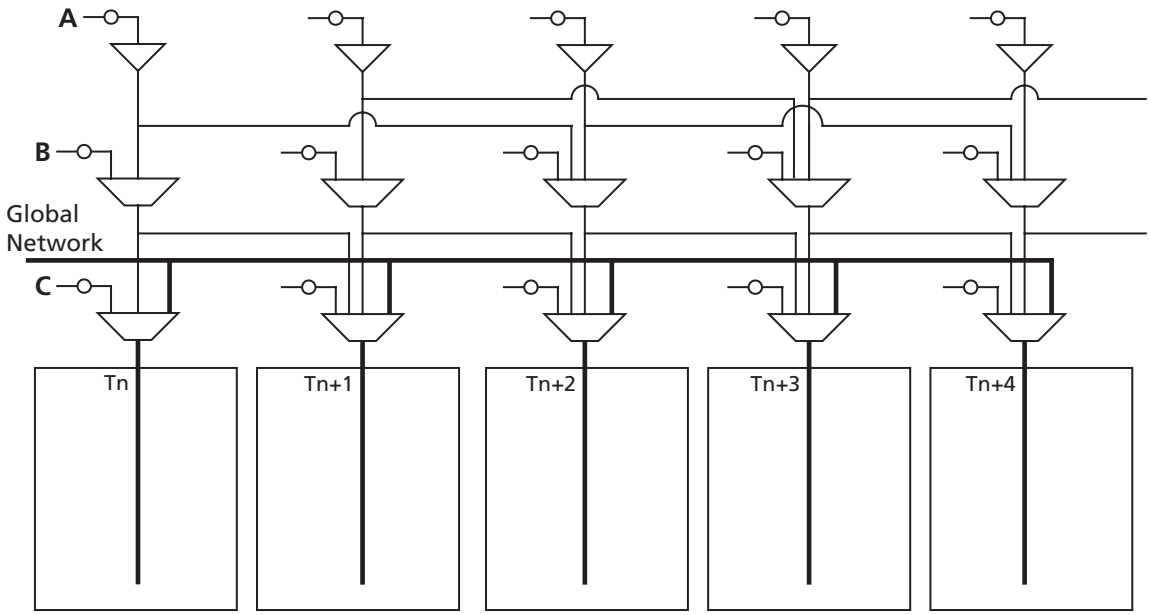


Figure 3-3 • **Chip Global Aggregation**

Spines are also called local clocks, and are accessed by the dedicated global MUX architecture. These MUXes define how a particular spine is driven. Refer to Figure 3-4 on page 3-7 for the global MUX architecture. The MUXes for each chip global spine are located in the middle of the die. Access to the top and bottom chip global spine is available from the middle of the die. There is no control dependency between the top and bottom spines. If a top spine, T1, of a chip global network is assigned to a net, B1 is not wasted and can be used by the global clock network. The signal assigned only to the top or bottom spine cannot access the middle two rows of the architecture. However, if a spine is using the top and bottom at the same time (T1 and B1, for instance), the previous restriction is lifted.

The MUXes for each quadrant global spine are located in the north and south sides of the die. Access to the top and bottom quadrant global spines is available from the north and south sides of the die. Since the MUXes for quadrant spines are located in the north and south sides of the die, you should not try to drive T1 and B1 quadrant spines from the same signal.

Using Clock Aggregation

Clock aggregation allows for multi-spine clock domains to be assigned using hardwired connections, without adding any extra skew. A MUX tree, shown in Figure 3-4, provides the necessary flexibility to allow long lines, local resources, or I/Os to access domains of one, two, or four global spines. Signal access to the clock aggregation system is achieved through long-line resources in the central rib in the center of the die, and also through local resources in the north and south ribs, allowing I/Os to feed directly into the clock system. As Figure 3-5 indicates, this access system is contiguous.

There is no break in the middle of the chip for the north and south I/O VersaNet access. This is different from the quadrant clocks located in these ribs, which only reach the middle of the rib.

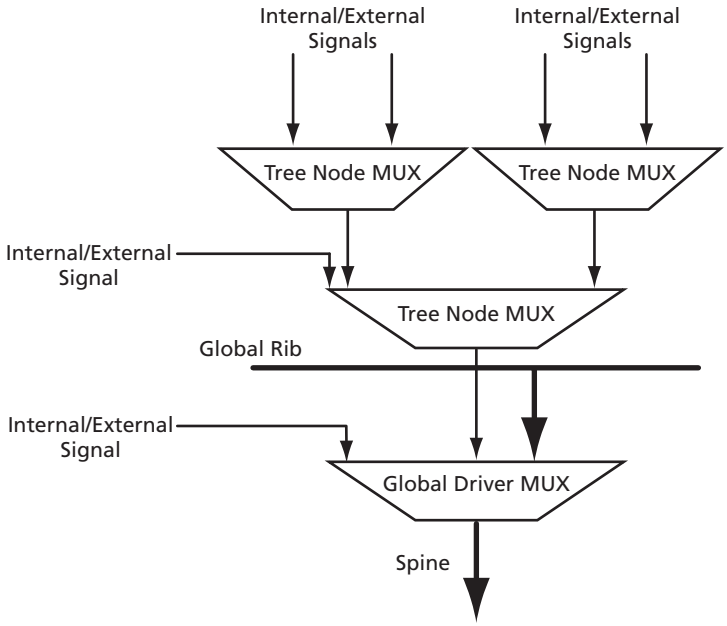


Figure 3-4 • **Spine Selection MUX of Global Tree**

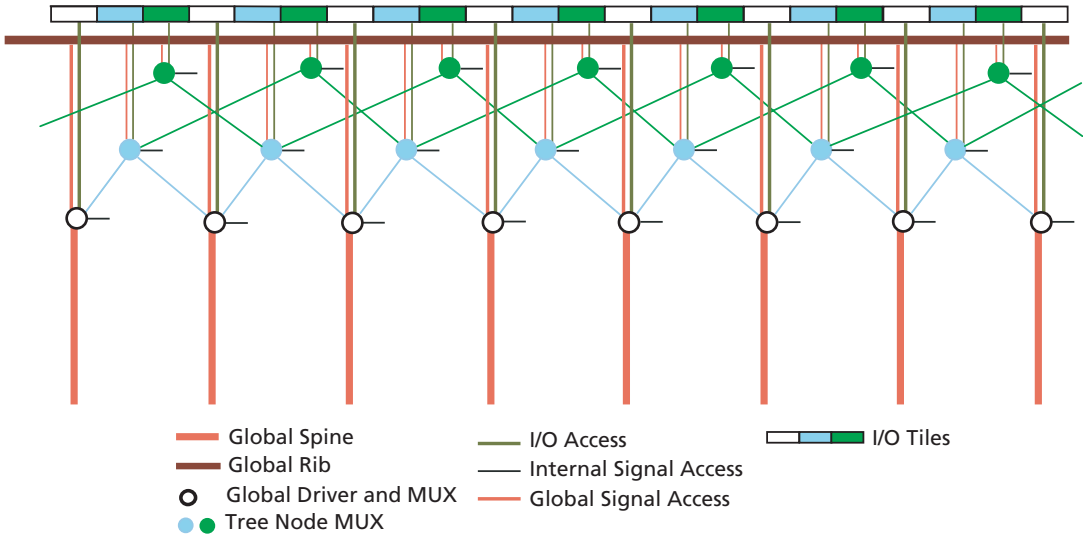


Figure 3-5 • **Clock Aggregation Tree Architecture**

Clock Aggregation Architecture

This clock aggregation feature allows a balanced clock tree, which improves clock skew. The physical regions for clock aggregation are defined from left to right and shift by one spine. For chip global networks, there are three types of clock aggregation available, as shown in Figure 3-6:

- Long lines that can drive up to four adjacent spines
- Long lines that can drive up to two adjacent spines
- Long lines that can drive one spine

There are three types of clock aggregation available for the quadrant spines, as shown in Figure 3-6:

- I/Os or local resources that can drive up to four adjacent spines
- I/Os or local resources that can drive up to two adjacent spines
- I/Os or local resources that can drive one spine
- As an example, A3PE600 and AFS600 devices have twelve spine locations: T1, T2, T3, T4, T5, T6, B1, B2, B3, B4, B5, and B6. Table 3-5 shows the clock aggregation you can have in A3PE600 and AFS600.

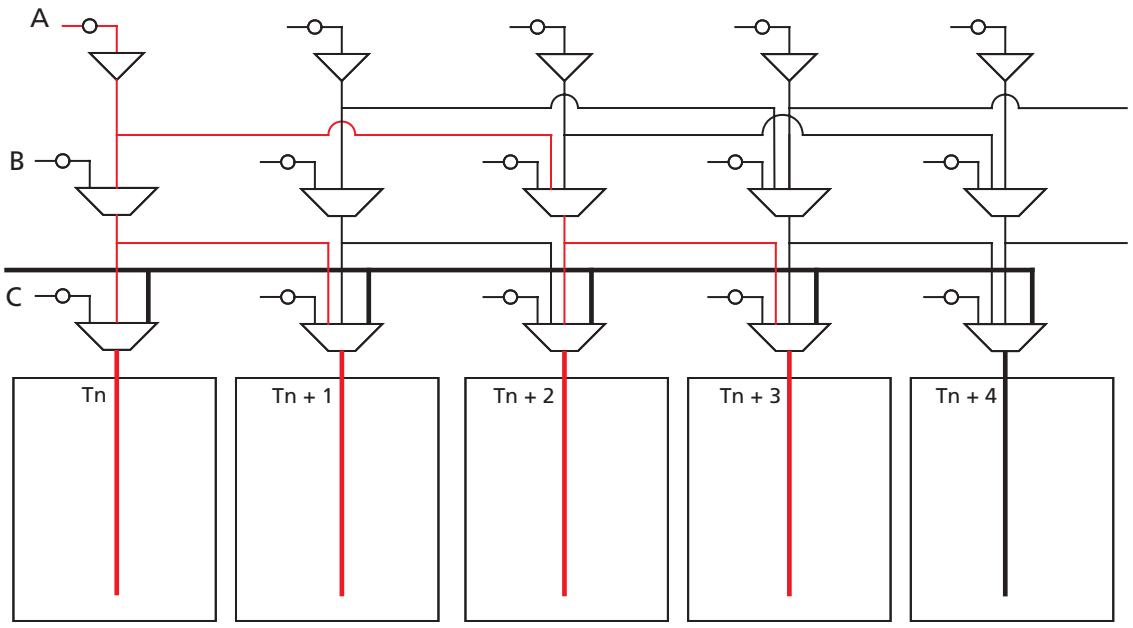


Figure 3-6 • **Four Spines Aggregation**

Table 3-5 • **Spine Aggregation in A3PE600 or AFS600**

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

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

I/O Banks and Global I/Os

The following sections give an overview of naming conventions and other related I/O information.

Naming of Global I/Os

In low-power flash devices, the global I/Os have access to certain clock conditioning circuitry and have direct access to the global network. Additionally, the global I/Os can be used as regular I/Os, since they have identical capabilities to those of regular I/Os. Due to the comprehensive and flexible nature of the I/Os in low-power flash devices, a naming scheme is used to show the details of the I/O. The global I/O uses the generic name Gmn/IOuxwByVz. Refer to the I/O Structure section of the handbook for the device that you are using for more information on this naming convention.

Figure 3-7 represents the global input pins connection to the northwest CCC or northwest quadrant global networks for a low-power flash device. Each global buffer, as well as the PLL reference clock, can be driven from one of the following:

- 3 dedicated single-ended I/Os using a hardwired connection
- 2 dedicated differential I/Os using a hardwired connection
- The FPGA core

Since each bank can have a different I/O standard, the user should be careful to choose the correct global I/O for the design. There are 54 global pins available to access 18 global networks. For the single-ended and voltage-referenced I/O standards, you can use any of these three available I/Os to access the global network. For differential I/O standards such as LVDS and LVPECL, the I/O macro needs to be placed on GAA0 and GAA1 or a similar location. The unassigned global I/Os can be used as regular I/Os. Note that pin names starting with GF and GC are associated with the chip global networks, and GA, GB, GD, and GE are used for quadrant global networks.

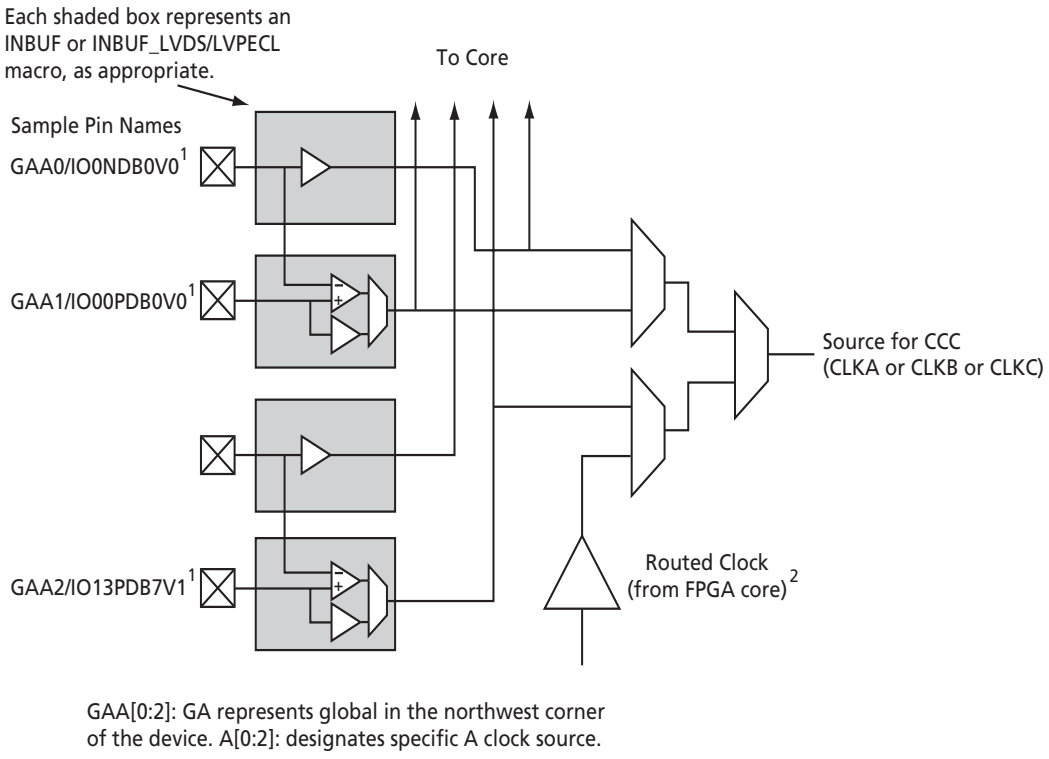


Figure 3-7 • **Global I/O Overview**

Unused Global I/O Configuration

The unused clock inputs behave similarly to the unused Pro I/Os. The Actel Designer software automatically configures the unused global pins as inputs with pull-up resistors if they are not used as regular I/O.

I/O Banks and Global I/O Standards

In low-power flash devices, any I/O or internal logic can be used to drive the global network. However, only the global macro placed at the global pins will use the hardwired connection between the I/O and global network. Global signal (signal driving a global macro) assignment to I/O banks is no different from regular I/O assignment to I/O banks with the exception that you are limited to the pin placement location available. Only global signals compatible with both the V_{CCI} and V_{REF} standards can be assigned to the same bank.

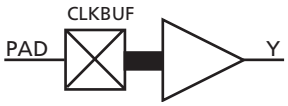
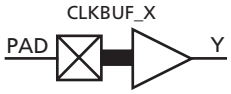
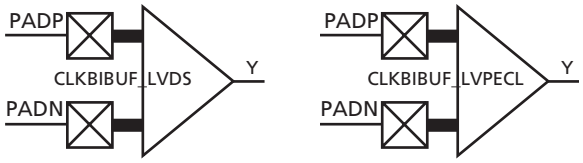
Design Recommendations

The following sections provide design flow recommendations for using a global network in a design.

Global Macros and I/O Standards

Low-power flash devices have six chip global networks and four quadrant clock networks. However, the same clock macros are used for assigning signals to chip globals and quadrant globals. Depending on the clock macro placement or assignment in the Physical Design Constraint (PDC) file or MultiView Navigator (MVN), the signal will use the chip global network or quadrant network. Table 3-6 on page 3-11 lists the clock macros available for low-power flash devices. Refer to the *IGLOO, Fusion and ProASIC3 Macro Library Guide* for details.

Table 3-6 • **Clock Macros**

Macro Name	Description	Symbol
CLKBUF	Input macro for Clock Network	
CLKBUF_x	Input macro for Clock Network with specific I/O standard	
CLKBUF_LVDS/ LVPECL	LVDS or LVPECL input macro for Clock Network	
CLKINT	Internal clock interface	
CLKBIBUF	Bidirectional macro with input dedicated to routed Clock Network	

Use these available macros to assign a signal to the global network. In addition to these global macros, PLL and CLKDLY macros can also drive the global networks. Use I/O–standard–specific clock macros (CLKBUF_x) to instantiate a specific I/O standard for the global signals. Table 3-7 shows the list of these I/O–standard–specific macros. Note that if you use these I/O–standard–specific clock macros, you cannot change the I/O standard later in the design stage. If you use the regular CLKBUF macro, you can use MVN or the PDC file in Designer to change the I/O standard. The default I/O standard for CLKBUF is LVTTL in the current Actel Libero® Integrated Design Environment (IDE) and Designer software.

Table 3-7 • **I/O Standards within CLKBUF**

Name	Description
CLKBUF_LVCMOS5	LVCMOS clock buffer with 5.0 V CMOS voltage level
CLKBUF_LVCMOS33	LVCMOS clock buffer with 3.3 V CMOS voltage level
CLKBUF_LVCMOS25	LVCMOS clock buffer with 2.5 V CMOS voltage level[1]
CLKBUF_LVCMOS18	LVCMOS clock buffer with 1.8 V CMOS voltage level
CLKBUF_LVCMOS15	LVCMOS clock buffer with 1.5 V CMOS voltage level
CLKBUF_LVCMOS12	LVCMOS clock buffer with 1.2 V CMOS voltage level
CLKBUF_PCI	PCI clock buffer
CLKBUF_PCIX	PCIX clock buffer
CLKBUF_GTL25	GTL clock buffer with 2.5 V CMOS voltage level[1]
CLKBUF_GTL33	GTL clock buffer with 3.3 V CMOS voltage level[1]

Notes:

1. *Supported in only the IGLOOe, ProASIC3E, AFS600, and AFS1500 devices*
2. *By default, the CLKBUF macro uses the 3.3 V LVTTL I/O technology.*

Table 3-7 • **I/O Standards within CLKBUF (continued)**

Name	Description
CLKBUF_GTLP25	GTL+ clock buffer with 2.5 V CMOS voltage level[1]
CLKBUF_GTLP33	GTL+ clock buffer with 3.3 V CMOS voltage level[1]
CLKBUF_ HSTL _I	HSTL Class I clock buffer[1]
CLKBUF_ HSTL _II	HSTL Class II clock buffer[1]
CLKBUF_SSTL2_I	SSTL2 Class I clock buffer[1]
CLKBUF_SSTL2_II	SSTL2 Class II clock buffer[1]
CLKBUF_SSTL3_I	SSTL3 Class I clock buffer[1]
CLKBUF_SSTL3_II	SSTL3 Class II clock buffer[1]

Notes:

1. *Supported in only the IGLOOe, ProASIC3E, AFS600, and AFS1500 devices*
2. *By default, the CLKBUF macro uses the 3.3 V LVTTL I/O technology.*

The current synthesis tool libraries only infer the CLKBUF or CLKINT macros in the netlist. All other global macros must be instantiated manually into your HDL code. The following is an example of CLKBUF_LVCMOS25 global macro instantiations that you can copy and paste into your code:

VHDL

```
component clkbuf_lvcmos25
  port (pad : in std_logic; y : out std_logic);
end component

begin
-- concurrent statements
u2 : clkbuf_lvcmos25 port map (pad =>  ext_clk, y => int_clk);
end
```

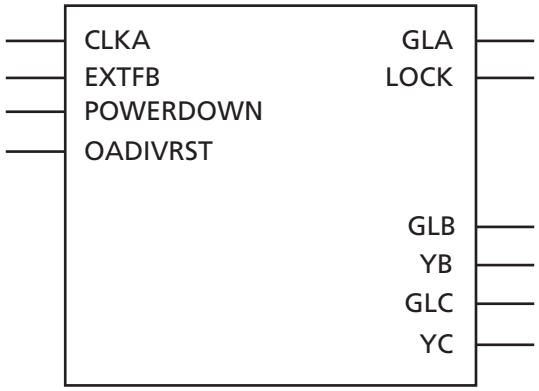
Verilog

```
module design (_______);

input _____;
output _______;

clkbuf_lvcmos25 u2 (.y(int_clk), .pad(ext_clk);

endmodule
```

Using Global Macros in Synplicity

The Synplify® synthesis tool automatically inserts global buffers for nets with high fanout during synthesis. By default, Synplicity® puts six global macros (CLKBUF or CLKINT) in the netlist, including any global instantiation or PLL macro. Synplify always honors your global macro instantiation. If you have a PLL (only primary output is used) in the design, Synplify adds five more global buffers in the netlist. Synplify uses the following global counting rule to add global macros in the netlist:

1. CLKBUF: 1 global buffer
2. CLKINT: 1 global buffer
3. CLKDLY: 1 global buffer
4. PLL: 1 to 3 global buffers
   - GLA, GLB, GLC, YB, and YC are counted as 1 buffer.
   - GLB or YB is used or both are counted as 1 buffer.
   - GLC or YC is used or both are counted as 1 buffer.



Note: OAVDIVRST exists only in the Fusion PLL.

Figure 3-8 • **PLLs in Low-Power Flash Devices**

You can use the syn_global_buffers attribute in Synplify to specify a maximum number of global macros to be inserted in the netlist. This can also be used to restrict the number of global buffers inserted. In the Synplicity 8.1 version, a new attribute, syn_global_minfanout, has been added for low-power flash devices. This enables you to promote only the high-fanout signal to global. However, be aware that you can only have six signals assigned to chip global networks, and the rest of the global signals should be assigned to quadrant global networks. So, if the netlist has 18 global macros, the remaining 12 global macros should have fanout that allows the instances driven by these globals to be placed inside a quadrant.

Global Promotion and Demotion Using PDC

The HDL source file or schematic is the preferred place for defining which signals should be assigned to a clock network using clock macro instantiation. This method is preferred because it is guaranteed to be honored by the synthesis tools and Designer software and stop any replication on this net by the synthesis tool. Note that a signal with fanout may have logic replication if it is not promoted to global during synthesis. In that case, the user cannot promote that signal to global using PDC. See Synplicity Help for details on using this attribute. To help you with global management, Designer allows you to promote a signal to a global network or demote a global macro to a regular macro from the user netlist using the compile options and/or PDC commands.

The following are the PDC constraints you can use to promote a signal to a global network:

1. PDC syntax to promote a regular net to a chip global clock:

   ```
   assign_global_clock –net netname
   ```

   The following will happen during promotion of a regular signal to a global network:
   - If the net is external, the net will be driven by a CLKINT inserted automatically by Compile.
   - The I/O macro will not be changed to CLKBUF macros.
   - If the net is an internal net, the net will be driven by a CLKINT inserted automatically by Compile.
2. PDC syntax to promote a net to a quadrant clock:

   ```
   assign_local_clock –net netname –type quadrant UR|UL|LR|LL
   ```

   This follows the same rule as the chip global clock network.

The following PDC command demotes the clock nets to regular nets.

```
unassign_global_clock -net netname
```

The following will happen during demotion of a global signal to regular nets:

- CLKBUF_x becomes INBUF_x; CLKINT is removed from the netlist.
- The essential global macro, such as the output of the Clock Conditioning Circuit, cannot be demoted.
- No automatic buffering will happen.

Since no automatic buffering happens when a signal is demoted, this net may have a high delay due to large fanout. This may have a negative effect on the quality of the results. Actel recommends that the automatic global demotion only be used on small-fanout nets. Use clock networks for high-fanout nets to improve timing and routability.

Spine Assignment

The low-power flash device architecture allows the global networks to be segmented and used as clock spines. These spines, also called local clocks, enable the use of PDC or MVN to assign a signal to a spine.

PDC syntax to promote a net to a spine/local clock:

```
assign_local_clock –net netname –type [quadrant|chip] Tn|Bn|Tn:Bm
```

If the net is driven by a clock macro, Designer automatically demotes the clock net to a regular net before it is assigned to a spine. Nets driven by a PLL or CLKDLY macro cannot be assigned to a local clock.

When assigning a signal to a spine or quadrant global network using PDC (pre-compile), the Designer software will legalize the shared instances. The number of shared instances to be legalized can be controlled by compile options. If these networks are created in MVN (only quadrant globals can be created), no legalization is done (as it is post-compile). Designer does not do legalization between non-clock nets.

As an example, consider two nets, net_clk and net_reset, driving the same flip-flop. The following PDC constraints are used:

```
assign_local_clock –net net_clk –type chip T3
assign_local_clock –net net_reset –type chip T1:T2
```

During Compile, Designer adds a buffer in the reset net and places it in the T1 or T2 region, and places the flip-flop in the T3 spine region (Figure 3-9).

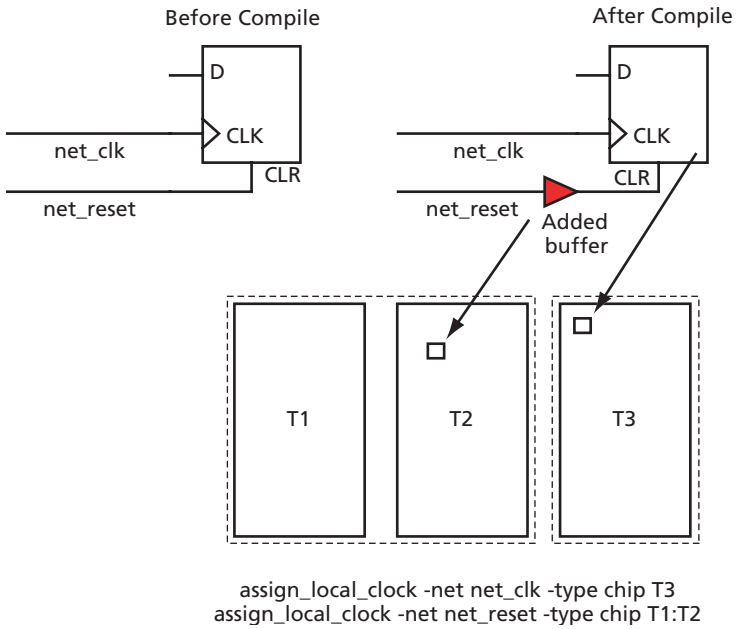


Figure 3-9 • **Adding a Buffer for Shared Instances**

You can control the maximum number of shared instances allowed for the legalization to take place using the Compile Option dialog box shown in Figure 3-10. Refer to Libero IDE / Designer online help for details on the Compile Option dialog box. A large number of shared instances most likely indicates a floorplanning problem that you should address.

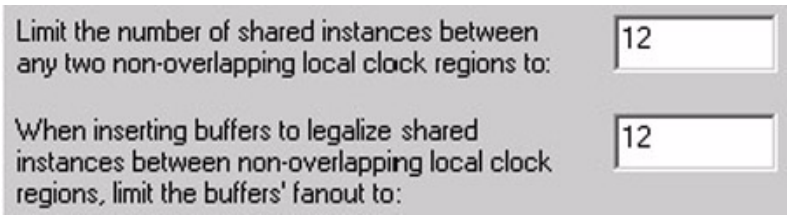


***Figure 3-10* • Shared Instances in the Compile Option Dialog Box**

Designer Flow for Global Assignment

To achieve the desired result, pay special attention to global management during synthesis and place-and-route. The current Synplify tool does not insert more than six global buffers in the netlist by default. Thus, the default flow will not assign any signal to the quadrant global network. However, you can use attributes in Synplify and increase the default global macro assignment in the netlist. Designer v6.2 supports automatic quadrant global assignment, which was not available in Designer v6.1. Layout will make the choice to assign the correct signals to global. However, you can also utilize PDC and perform manual global assignment to overwrite any automatic assignment. The following step-by-step suggestions guide you in the layout of your design and help you improve timing in Designer:

1. Run Compile and check the Compile report. The Compile report has global information in the "Device Utilization" section that describes the number of chip and quadrant signals in the design. A "Net Report" section describes chip global nets, quadrant global nets, local clock nets, a list of nets listed by fanout, and net candidates for local clock assignment. Review this information. Note that YB or YC are counted as global only when they are used in isolation; if you use YB only and not GLB, this net is not shown in the global/quadrant nets report. Instead, it appears in the Global Utilization report.
2. If some signals have a very high fanout and are candidates for global promotion, promote those signals to global using the compile options or PDC commands. Figure 3-11 on page 3-16 shows the Globals Management section of the compile options. Select **Promote regular nets whose fanout is greater than** and enter a reasonable value for fanouts.

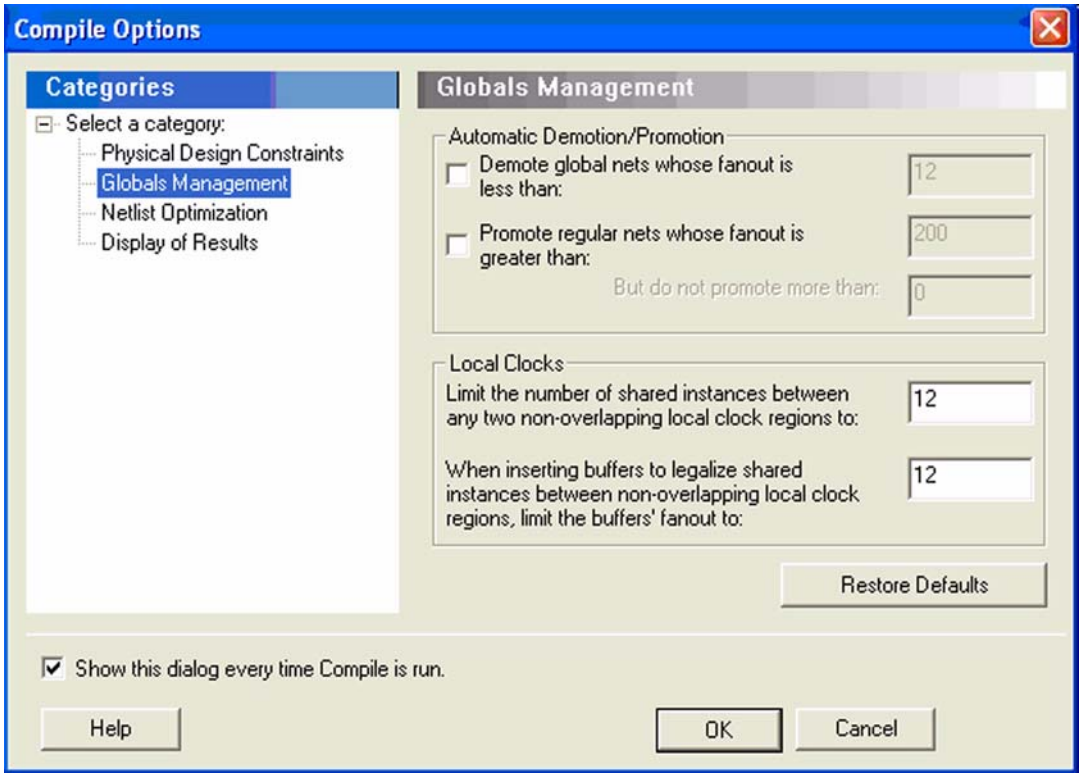


Figure 3-11 • **Globals Management GUI in Designer**

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 local clocks for the signals that do not need to go to the whole chip but should have low skew. This local clocks 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 3-6 on page 3-8, 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.

Simple Design Example

Consider a design consisting of six building blocks (shift registers) and targeted for an A3PE600-PQ208 (Figure 3-9 on page 3-14). The example design consists of two PLLs (PLL1 has GLA only; PLL2 has both GLA and GLB), a global reset (ACLR), an enable (EN_ALL), and three external clock domains (QCLK1, QCLK2, and QCLK3) driving the different blocks of the design. Note that the PQ208 package only has two PLLs (which access the chip global network). Because of fanout, the global reset and enable signals need to be assigned to the chip global resources. There is only one free chip global for the remaining global (QCLK1, QCLK2, QCLK3). Place two of these signals on the quadrant global resource. The design example demonstrates manually assignment of QCLK1 and QCLK2 to the quadrant global using the PDC command.

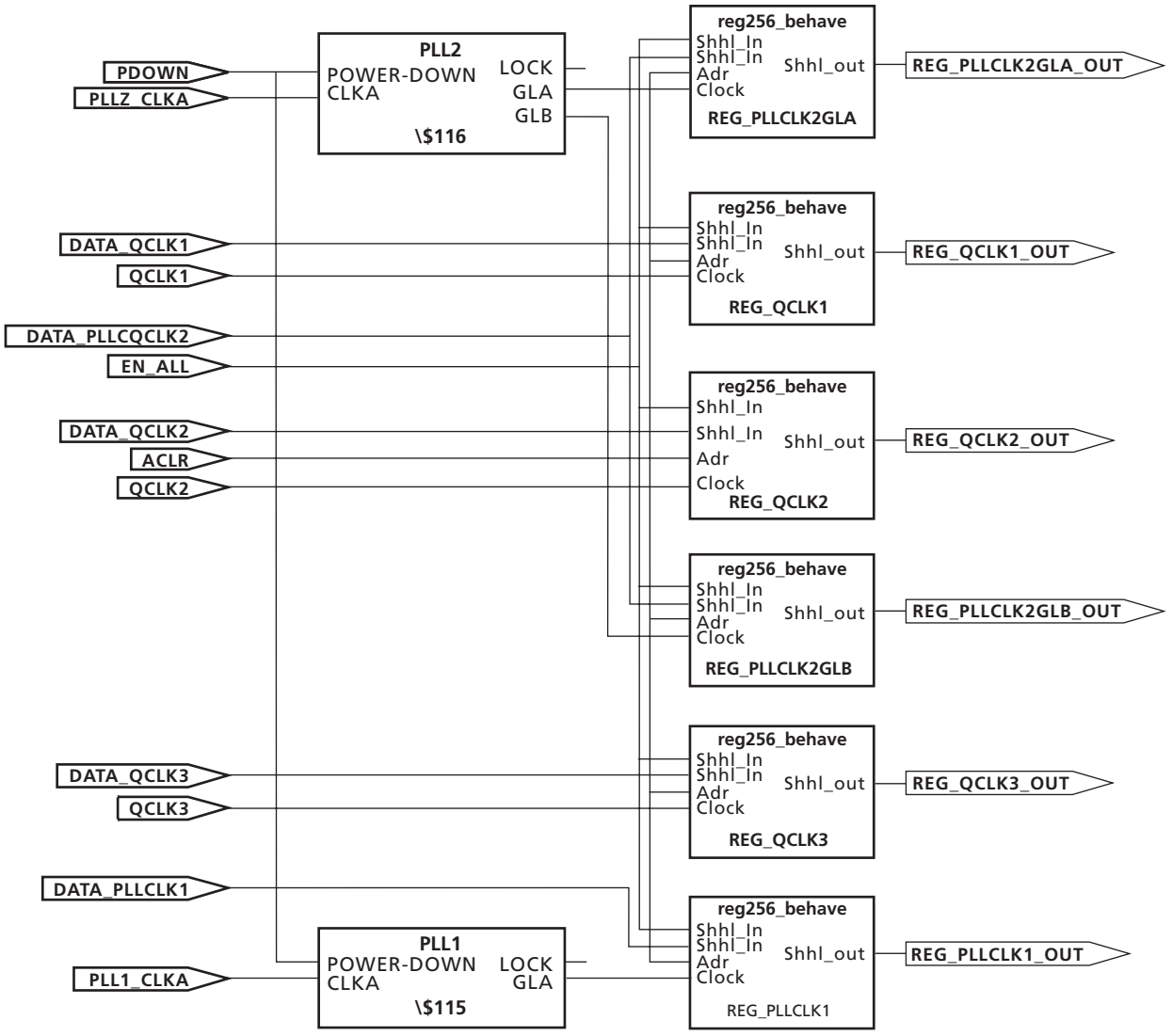


Figure 3-12 • **Block Diagram of the Global Management Example Design**

Step 1

Run Synthesis with default options. The Synplicity log shows the following device utilization:

Cell usage:

	cell count	area	count*area
DFN1E1C1	1536	2.0	3072.0
BUFF	278	1.0	278.0
INBUF	10	0.0	0.0
VCC	9	0.0	0.0
GND	9	0.0	0.0
OUTBUF	6	0.0	0.0
CLKBUF	3	0.0	0.0
PLL	2	0.0	0.0
TOTAL	1853		3350.0

Step 2

Run Compile with the **Promote regular nets whose fanout is greater than** option selected in Designer; you will see the following in the Compile report:

```
Device utilization report:
==========================
CORE               Used:   1536  Total:  13824   (11.11%)
IO (W/ clocks)     Used:     19  Total:    147   (12.93%)
Differential IO    Used:      0  Total:     65    (0.00%)
GLOBAL             Used:      8  Total:     18   (44.44%)
PLL                Used:      2  Total:      2  (100.00%)
RAM/FIFO           Used:      0  Total:     24    (0.00%)
FlashROM           Used:      0  Total:      1    (0.00%)
.....................
The following nets have been assigned to a global resource:
Fanout  Type           Name
-------------------------
1536    INT_NET        Net   : EN_ALL_c
                       Driver: EN_ALL_pad_CLKINT
                       Source: AUTO PROMOTED
1536    SET/RESET_NET  Net   : ACLR_c
                       Driver: ACLR_pad_CLKINT
                       Source: AUTO PROMOTED
256     CLK_NET        Net   : QCLK1_c
                       Driver: QCLK1_pad_CLKINT
                       Source: AUTO PROMOTED
256     CLK_NET        Net   : QCLK2_c
                       Driver: QCLK2_pad_CLKINT
                       Source: AUTO PROMOTED
256     CLK_NET        Net   : QCLK3_c
                       Driver: QCLK3_pad_CLKINT
                       Source: AUTO PROMOTED
256     CLK_NET        Net   : $1N14
                       Driver: $1I5/Core
                       Source: ESSENTIAL
256     CLK_NET        Net   : $1N12
                       Driver: $1I6/Core
                       Source: ESSENTIAL
256     CLK_NET        Net   : $1N10
                       Driver: $1I6/Core
                       Source: ESSENTIAL
```

Designer will promote five more signals to global due to high fanout. There are eight signals assigned to global networks.

During Layout, Designer will assign two of the signals to quadrant global locations.

Step 3 (optional)

You can also assign the QCLK1_c and QCLK2_c nets to quadrant regions using the following PDC commands:

```
assign_local_clock –net QCLK1_c  –type quadrant UL
assign_local_clock –net QCLK2_c  –type quadrant LL
```

Step 4

Import this PDC with the netlist and run Compile again. You will see the following in the Compile report:

```
The following nets have been assigned to a global resource:
Fanout  Type          Name
--------------------------
1536    INT_NET       Net   : EN_ALL_c
                      Driver: EN_ALL_pad_CLKINT
                      Source: AUTO PROMOTED
1536    SET/RESET_NET Net   : ACLR_c
                      Driver: ACLR_pad_CLKINT
                      Source: AUTO PROMOTED
256     CLK_NET       Net   : QCLK3_c
                      Driver: QCLK3_pad_CLKINT
                      Source: AUTO PROMOTED
256     CLK_NET       Net   : $1N14
                      Driver: $1I5/Core
                      Source: ESSENTIAL
256     CLK_NET       Net   : $1N12
                      Driver: $1I6/Core
                      Source: ESSENTIAL
256     CLK_NET       Net   : $1N10
                      Driver: $1I6/Core
                      Source: ESSENTIAL
The following nets have been assigned to a quadrant clock resource using PDC:
Fanout  Type          Name
--------------------------
256     CLK_NET       Net   : QCLK1_c
                      Driver: QCLK1_pad_CLKINT
                      Region: quadrant_UL
256     CLK_NET       Net   : QCLK2_c
                      Driver: QCLK2_pad_CLKINT
                      Region: quadrant_LL
```

Step 5

Run Layout.

Global Management in PLL Design

This section describes the legal global network connections to PLLs in the low-power flash devices. For detailed information on using PLLs, refer to *Clock Conditioning Circuits in IGLOO and ProASIC3 Devices*. Actel recommends that you use the dedicated global pins to directly drive the reference clock input of the associated PLL for reduced propagation delays and clock distortion. However, low-power flash devices offer the flexibility to connect other signals to reference clock inputs. Each PLL is associated with three global networks (Figure 3-7 on page 3-9). There are some limitations, such as when trying to use the global and PLL at the same time:

- If you use a PLL with only primary output, you can still use the remaining two free global networks.
- If you use three globals associated with a PLL location, you cannot use the PLL on that location.
- If the YB or YC output is used standalone, it will occupy one global, even though this signal does not go to the global network.

Using Spines of Occupied Global Networks

When a signal is assigned to a global network, the flash switches are programmed to set the MUX select lines (explained in the "Clock Aggregation Architecture" section on page 3-8) to drive the spines of that network with the global net. However, if the global net is restricted from reaching into the scope of a spine, the MUX drivers of that spine are available for other high-fanout or critical signals (Figure 3-13).

For example, if you want to limit the CLK1_c signal to the left half of the chip and want use the right side of the same global network for CLK2_c, you can add the following PDC commands:

```
define_region -name region1 -type inclusive 0 0 34 29
assign_net_macros region1 CLK1_c
assign_local_clock –net CLK2_c –type chip B2
```

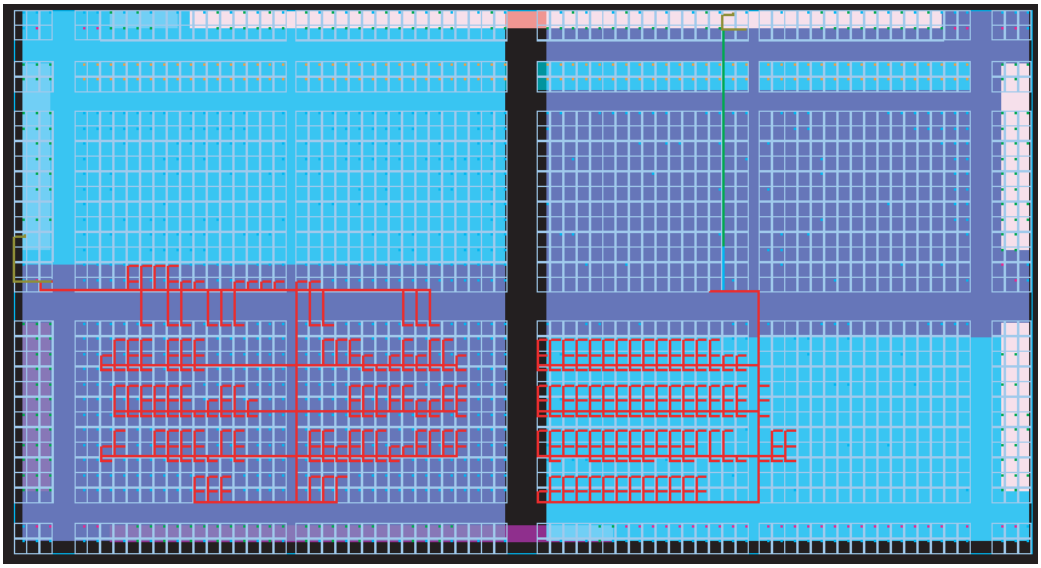


***Figure 3-13* • Design Example Using Spines of Occupied Global Networks**

Conclusion

IGLOO, Fusion, and ProASIC3 devices contain 18 global networks: 6 chip global networks and 12 quadrant global networks. These global networks can be segmented into local low-skew networks called spines. The spines provide low-skew networks for the high-fanout signals of a design. These allow you up to 252 different internal/external clocks in an A3PE3000 device. This document describes the architecture for the global network, plus guidelines and methodologies in assigning signals to globals and spines.

Related Documents

Handbook Documents

Clock Conditioning Circuits in IGLOO and ProASIC3 Devices
http://www.actel.com/LPD_CCC_HBs.pdf
I/O Structures in IGLOO PLUS Devices
http://www.actel.com/documents/IGLOOPLUS_IO_HBs.pdf
I/O Structures in IGLOO and ProASIC3 Devices
http://www.actel.com/documents/IGLOO_PA3_IO_HBs.pdf
I/O Structures in IGLOOe and ProASIC3E Devices
http://www.actel.com/documents/IGLOOe_PA3E_IO_HBs.pdf

User's Guides

IGLOO, Fusion, and ProASIC3 Macro Library Guide
http://www.actel.com/documents/pa3_libguide_ug.pdf

Part Number and Revision Date

This document contains content extracted from the Device Architecture section of the datasheet, combined with content previously published as an application note describing features and functions of the device. To improve usability for customers, the device architecture information has now been combined with usage information, to reduce duplication and possible inconsistencies in published information. No technical changes were made to the datasheet content unless explicitly listed. Changes to the application note content were made only to be consistent with existing datasheet information.

Part Number 51700094-005-3
Revised October 2008

List of Changes

The following table lists critical changes that were made in the current version of the chapter.

Previous Version	Changes in Current Version (v1.3)	Page
v1.2 (June 2008)	A third bullet was added to the beginning of the "Global Architecture" section: In Fusion devices, the west CCC also contains a PLL core. In the two larger devices (AFS600 and AFS1500), the west and east CCCs each contain a PLL.	3-1
	The "Global Resource Support in Low-Power Devices" section was revised to include new families and make the information more concise.	3-2
	Table 3-2 · Globals/Spines/Rows for IGLOO and ProASIC3 Devices was updated to include A3PE600/L in the device column.	3-5
	Table note 1 was revised in Table 3-7 · I/O Standards within CLKBUF to include AFS600 and AFS1500.	3-11
v1.1 (March 2008)	The following changes were made to the family descriptions in Table 3-1 · Low-Power Flash Families: • ProASIC3L was updated to include 1.5 V. • The number of PLLs for ProASIC3E was changed from five to six.	3-2
v1.0 (January 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."	3-1
	Table 3-1 · Low-Power Flash Families and the accompanying text was updated to include the IGLOO PLUS family. The "IGLOO Terminology" section and "ProASIC3 Terminology" section are new.	3-2
	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 3-2 · Simplified VersaNet Global Network were updated to include mention of 15 k gate devices.	3-3, 3-4
	Table 3-2 · 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.	3-5
	Table 3-3 · Globals/Spines/Rows for IGLOO PLUS Devices is new.	3-5
	CLKBUF_LVCMOS12 was added to Table 3-7 · I/O Standards within CLKBUF.	3-11
	The "Handbook Documents" section was updated to include the three different I/O Structures chapters for ProASIC3 and IGLOO device families.	3-21
51900087-1/3.05	Figure 3-2 · Simplified VersaNet Global Network was updated.	3-4
	The "Naming of Global I/Os" section was updated.	3-9
	The "Using Global Macros in Synplicity" section was updated.	3-12
	The "Global Promotion and Demotion Using PDC" section was updated.	3-13
	The "Designer Flow for Global Assignment" section was updated.	3-15
	The "Simple Design Example" section was updated.	3-17
51900087-0/1.05	Table 3-2 · Globals/Spines/Rows for IGLOO and ProASIC3 Devices was updated.	3-5

4 – Clock Conditioning Circuits in Low-Power Flash Devices and Mixed-Signal FPGAs

Introduction

This document outlines the following device information: Clock Conditioning Circuits (CCC) features, PLL core specifications, functional descriptions, software configuration information, detailed usage information, recommended board-level considerations, and other considerations concerning clock conditioning circuits and global networks in low-power flash devices or mixed-signal FPGAs.

Overview of Clock Conditioning Circuitry

In Fusion, IGLOO®, and ProASIC®3 devices, the CCCs are used to implement frequency division, frequency multiplication, phase shifting, and delay operations. The CCCs are available in six chip locations—each of the four chip corners and the middle of the east and west chip sides. For device-specific variations, refer to the "Device-Specific Layout" section on page 4-16.

The CCC is composed of the following:

- PLL core
- 3 phase selectors
- 6 programmable delays and 1 fixed delay that advances/delays phase
- 5 programmable frequency dividers that provide frequency multiplication/division (not shown in Figure 4-5 on page 4-10 because they are automatically configured based on the user's required frequencies)
- 1 dynamic shift register that provides CCC dynamic reconfiguration capability

Figure 4-1 provides a simplified block diagram of the physical implementation of the building blocks in each of the CCCs.

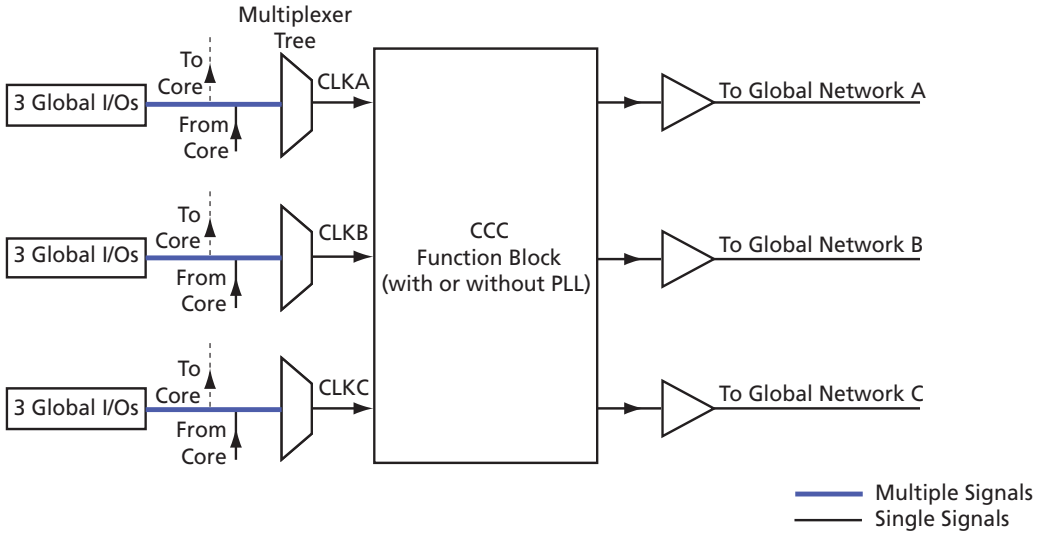


Figure 4-1 • **Overview of the CCCs Offered in Fusion, IGLOO, and ProASIC3**

Each CCC can implement up to three independent global buffers (with or without programmable delay) or a PLL function (programmable frequency division/multiplication, phase shift, and delays) with up to three global outputs. Unused global outputs of a PLL can be used to implement independent global buffers, up to a maximum of three global outputs for a given CCC.

CCC Programming

The CCC block is fully configurable, either via flash configuration bits set in the programming bitstream or through an asynchronous interface. This asynchronous dedicated shift register interface is dynamically accessible from inside the low-power flash devices to permit parameter changes, such as PLL divide ratios and delays, during device operation.

To increase the versatility and flexibility of the clock conditioning system, the CCC configuration is determined either by the user during the design process, with configuration data being stored in flash memory as part of the device programming procedure, or by writing data into a dedicated shift register during normal device operation.

This latter mode allows the user to dynamically reconfigure the CCC without the need for core programming. The shift register is accessed through a simple serial interface. Refer to *UJTAG Applications in Actel's Low-Power Flash Devices* or the application note *Using Global Resources in Actel Fusion Devices*.

Global Resources

Low-power flash and mixed signal devices provide three global routing networks (GLA, GLB, and GLC) for each of the CCC locations. There are potentially many I/O locations; each global I/O location can be chosen from only one of three possibilities. This is controlled by the multiplexer tree circuitry in each global network. Once the I/O location is selected, the user has the option to utilize the CCCs before the signals are connected to the global networks. The CCC in each location (up to six) has the same structure, so generating the CCC macros is always done with an identical software GUI. The CCCs in the corner locations drive the quadrant global networks, and the CCCs in the middle of the east and west chip sides drive the chip global networks. The quadrant global networks span only a quarter of the device, while the chip global networks span the entire device. For more details on global resources offered in low-power flash devices, refer to *Global Resources in Actel Low-Power Flash Devices*.

A global buffer can be placed in any of the three global locations (CLKA-GLA, CLKB-GLB, or CLKC-GLC) of a given CCC. A PLL macro uses the CLKA CCC input to drive its reference clock. It uses the GLA and, optionally, the GLB and GLC global outputs to drive the global networks. A PLL macro can also drive the YB and YC regular core outputs. The GLB (or GLC) global output cannot be reused if the YB (or YC) output is used. Refer to the "PLL Macro Signal Descriptions" section on page 4-8 for more information.

Each global buffer, as well as the PLL reference clock, can be driven from one of the following:

- 3 dedicated single-ended I/Os using a hardwired connection
- 2 dedicated differential I/Os using a hardwired connection
- The FPGA core

CCC Support in Actel's Low-Power Flash Devices

The low-power flash families listed in Table 4-1 support the CCC feature and the functions described in this document.

Table 4-1 • **Low-Power Flash Families**

Product Line	Family*	Description
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
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 PLUS	IGLOO FPGAs with enhanced I/O capabilities
ProASIC3	ProASIC3	Low-power, high-performance 1.5 V FPGAs
	ProASIC3E	Higher density ProASIC3 FPGAs with six PLLs and additional I/O standards
	ProASIC3L	ProASIC3 FPGAs supporting 1.2 V to 1.5 V with Flash*Freeze technology
	Automotive ProASIC3	ProASIC3 FPGAs qualified for automotive applications
	Military ProASIC3/EL	Military temperature A3PE600L, A3P1000, and A3PE3000L
	RT ProASIC3	Radiation-tolerant RT3PE600L and RT3PE3000L

*Note: *The family names link to the appropriate datasheet, including product brief, DC and switching characteristics, and packaging information.*

IGLOO Terminology

In documentation, the terms IGLOO families and IGLOO devices refer to all of the IGLOO products as listed in Table 4-1. Where the information applies to only one family or limited devices, these exclusions will be explicitly stated.

ProASIC3 Terminology

In documentation, the terms ProASIC3 families and ProASIC3 devices refer to all of the ProASIC3 families as listed in Table 4-1. Where the information applies to only one family or limited devices, these exclusions will be explicitly stated.

To further understand the differences between the IGLOO and ProASIC3 families, refer to the *Industry's Lowest Power FPGAs Portfolio*.

Global Buffers with No Programmable Delays

Access to the global / quadrant global networks can be configured directly from the global I/O buffer, bypassing the CCC functional block (as indicated by the dotted lines in Figure 4-1 on page 4-1). Internal signals driven by the FPGA core can use the global / quadrant global networks by connecting via the routed clock input of the multiplexer tree.

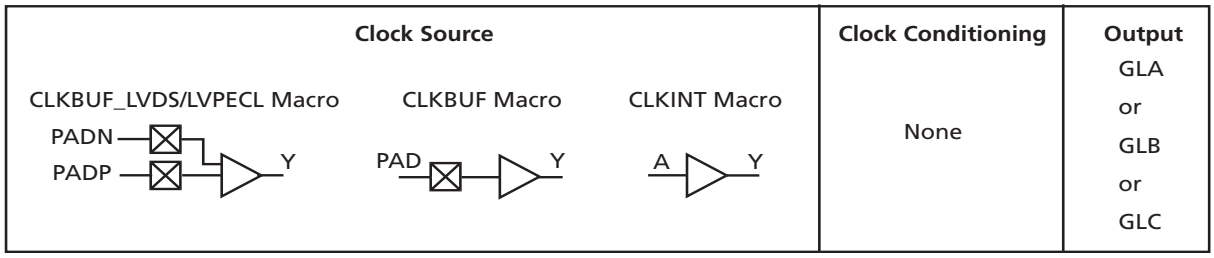
There are many specific CLKBUF macros supporting the wide variety of single-ended I/O inputs (CLKBUF) and differential I/O standards (CLKBUF_LVDS/LVPECL) in the low-power flash families. They are used when connecting global I/Os directly to the global/quadrant networks.

When an internal signal needs to be connected to the global/quadrant network, the CLKINT macro is used to connect the signal to the routed clock input of the network's MUX tree.

To utilize direct connection from global I/Os or from internal signals to the global/quadrant networks, CLKBUF, CLKBUF_LVPECL/LVDS, and CLKINT macros are used (Figure 4-2).

- The CLKBUF and CLKBUF_LVPECL/LVDS/B-LVDS/M-LVDS macros are composite macros that include an I/O macro driving a global buffer, which uses a hardwired connection.
- The CLKBUF, CLKBUF_LVPECL/LVDS/B-LVDS/M-LVDS, and CLKINT macros are pass-through clock sources and do not use the PLL or provide any programmable delay functionality.
- The CLKINT macro provides a global buffer function driven internally by the FPGA core.

The available CLKBUF macros are described in the *IGLOO, Fusion, and ProASIC3 Macro Library Guide*.



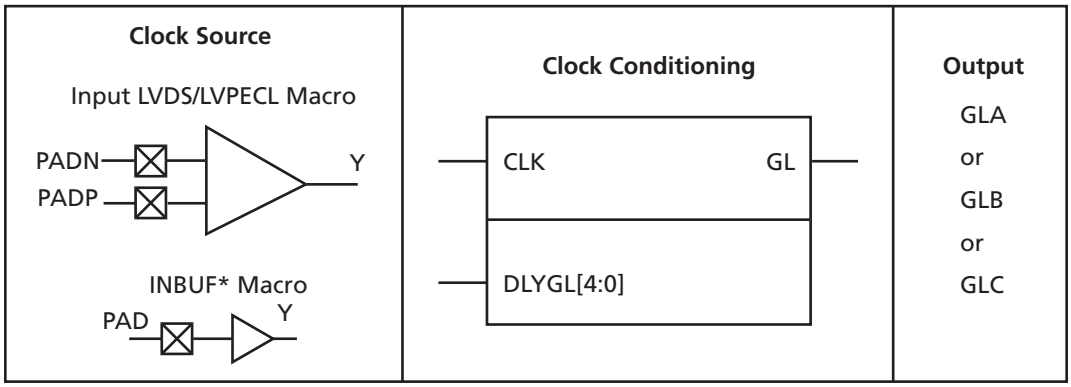
Note: The CLKDLY macro uses programmable delay element type 2 (refer to "CLKDLY Macro Usage" section on page 4-5).

Figure 4-2 • **CCC Options: Global Buffers with No Programmable Delay**

Global Buffer with Programmable Delay

Clocks requiring clock adjustments can utilize the programmable delay cores before connecting to the global / quadrant global networks. A maximum of 18 CCC global buffers can be instantiated in a device—three per CCC and up to six CCCs per device.

Each CCC functional block contains a programmable delay element for each of the global networks (up to three), and users can utilize these features by using the corresponding macro (Figure 4-3 on page 4-5).



Note: For INBUF driving a PLL macro or CLKDLY macro, the I/O will be hard-routed to the CCC; i.e., will be placed by software to a dedicated Global I/O.*

Figure 4-3 • **CCC Options: Global Buffers with Programmable Delay**

The CLKDLY macro is a pass-through clock source that does not use the PLL, but provides the ability to delay the clock input using a programmable delay. The CLKDLY macro takes the selected clock input and adds a user-defined delay element. This macro generates an output clock phase shift from the input clock.

The CLKDLY macro can be driven by an INBUF* macro to create a composite macro, where the I/O macro drives the global buffer (with programmable delay) using a hardwired connection. In this case, the software will automatically place the dedicated global I/O in the appropriate locations. Many specific INBUF macros support the wide variety of single-ended and differential I/O standards supported by the low-power flash family. The available INBUF macros are described in the *IGLOO, Fusion, and ProASIC3 Macro Library Guide*.

The CLKDLY macro can be driven directly from the FPGA core. The CLKDLY macro can also be driven from an I/O that is routed through the FPGA regular routing fabric. In this case, users must instantiate a special macro, PLLINT, to differentiate the clock input driven by the hardwired I/O connection.

The visual CLKDLY configuration in the SmartGen area of the Actel Libero® Integrated Design Environment (IDE) and Designer tools allows the user to select the desired amount of delay and configures the delay elements appropriately. SmartGen also allows the user to select the input clock source. SmartGen will automatically instantiate the special macro, PLLINT, when needed.

CLKDLY Macro Signal Descriptions

The CLKDLY macro supports one input and one output. Each signal is described in Table 4-2.

Table 4-2 • **Input and Output Description of the CLKDLY Macro**

Signal	Name	I/O	Description
CLK	Reference Clock	Input	Reference clock input for PLL core Input clock for primary output clock, GLA
GL	Global Output	Output	Primary output clock to respective global/quadrant clock networks

CLKDLY Macro Usage

When a CLKDLY macro is used in a CCC location, the programmable delay element is used to allow the clock delays to go to the global network. In addition, the user can bypass the PLL in a CCC location integrated with a PLL, but use the programmable delay that is associated with the global network by instantiating the CLKDLY macro. The same is true when using programmable delay elements in a CCC location with no PLLs (the user needs to instantiate the CLKDLY macro). There is no difference between the programmable delay elements used for the PLL and the CLKDLY macro.

The CCC will be configured to use the programmable delay elements in accordance with the macro instantiated by the user.

As an example, if the PLL is not used in a particular CCC location, the designer is free to specify up to three CLKDLY macros in the CCC, each of which can have its own input frequency and delay adjustment options. If the PLL core is used, assuming output to only one global clock network, the other two global clock networks are free to be used by either connecting directly from the global inputs or connecting from one or two CLKDLY macros for programmable delay.

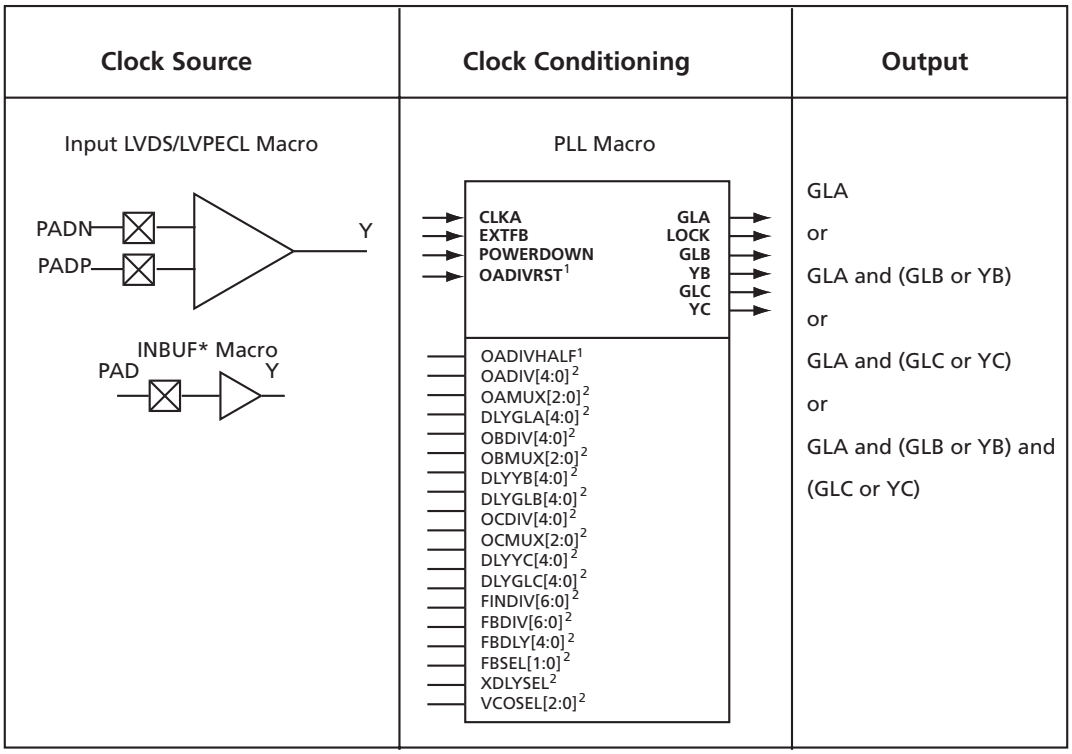
The programmable delay elements are shown in the block diagram of the PLL block shown in Figure 4-5 on page 4-10. Note that any CCC locations with no PLL present contain only the programmable delay blocks going to the global networks (labeled "Programmable Delay Type 2"). Refer to the "Clock Delay Adjustment" section on page 4-24 for a description of the programmable delay types used for the PLL. Also refer to Table 4-13 on page 4-30 for Programmable Delay Type 1 step delay values, and Table 4-14 on page 4-31 for Programmable Delay Type 2 step delay values. CCC locations with a PLL present can be configured to utilize only the programmable delay blocks (Programmable Delay Type 2) going to the global networks A, B, and C.

Global network A can be configured to use only the programmable delay element (bypassing the PLL) if the PLL is not used in the design. Figure 4-5 on page 4-10 shows a block diagram of the PLL, where the programmable delay elements are used for the global networks (Programmable Delay Type 2).

Global Buffers with PLL function

Clocks requiring frequency synthesis or clock adjustments can utilize the PLL core before connecting to the global / quadrant global networks. A maximum of 18 CCC global buffers can be instantiated in a device—three per CCC and up to six CCCs per device. Each PLL core can generate up to three global/quadrant clocks, while a clock delay element provides one.

The PLL functionality of the clock conditioning block is supported by the PLL macro.



Notes:

1. *For Fusion only.*
2. *Refer to the IGLOO, Fusion, and ProASIC3 Macro Library Guide for more information.*
3. *For INBUF* driving a PLL macro or CLKDLY macro, the I/O will be hard-routed to the CCC; i.e., will be placed by software to a dedicated Global I/O.*

Figure 4-4 • **CCC Options: Global Buffers with PLL**

The PLL macro provides five derived clocks (three independent) from a single reference clock. The PLL macro also provides power-down input and lock output signals. The additional inputs shown on the macro are configuration settings, which are configured through the use of SmartGen. For manual setting of these bits refer to the *IGLOO, Fusion, and ProASIC3 Macro Library Guide* for details.

Figure 4-5 on page 4-10 illustrates the various clock output options and delay elements.

PLL Macro Signal Descriptions

The PLL macro supports two inputs and up to six outputs. Table 4-3 gives a description of each signal.

Table 4-3 • **Input and Output Signals of the PLL Block**

Signal	Name	I/O	Description
CLKA	Reference Clock	Input	Reference clock input for PLL core; Input clock for primary output clock, GLA
OADIVRST	Reset Signal for the Output Divider A	Input	For Fusion only. OADIVRST can be used when you bypass the PLL core (i.e., OAMUX = 001). The purpose of the OADIVRST signals is to reset the output of the final clock divider in order to synchronize it with the input to that divider when the PLL is bypassed. The signal is active on a low to high transition. The signal must be low for at least one divider input. If PLL core is used, this signal is "don't care" and the internal circuitry will generate the reset signal for the synchronization purpose.
OADIVHALF	Output A Division by Half	Input	For Fusion only. Active high. Division by half feature. This feature can only be used when users bypass the PLL core (i.e., OAMUX = 001) and the RC Oscillator (RCOSC) drives the CLKA input. This can be used to divide the 100 MHz RC oscillator by a factor of 1.5, 2.5, 3.5, 4.5 ... 14.5). Refer to Table 4-17 on page 4-32 for more information.
EXTFB	External Feedback	Input	Allows an external signal to be compared to a reference clock in the PLL core's phase detector.
POWERDOWN	Power Down	Input	Active low input that selects power-down mode and disables the PLL. With the POWERDOWN signal asserted, the PLL core sends 0 V signals on all of the outputs.
GLA	Primary Output	Output	Primary output clock to respective global/quadrant clock networks
GLB	Secondary 1 Output	Output	Secondary 1 output clock to respective global/quadrant clock networks
YB	Core 1 Output	Output	Core 1 output clock to local routing network
GLC	Secondary 2 Output	Output	Secondary 2 output clock to respective global/quadrant clock networks
YC	Core 2 Output	Output	Core 2 output clock to local routing network
LOCK	PLL Lock Indicator	Output	Active-high signal indicating that steady-state lock has been achieved between CLKA and the PLL feedback signal

Input Clock

The inputs to the input reference clock (CLKA) of the PLL can come from global input pins, regular I/O pins, or internally from the core. For Fusion families, the input reference clock can also be from the embedded RC oscillator or crystal oscillator.

Global Output Clocks

GLA (Primary), GLB (Secondary 1), and GLC (Secondary 2) are the outputs of Global Multiplexer 1, Global Multiplexer 2, and Global Multiplexer 3, respectively. These signals (GLx) can be used to drive the high-speed global and quadrant networks of the low-power flash devices.

A global multiplexer block consists of the input routing for selecting the input signal for the GLx clock and the output multiplexer, as well as delay elements associated with that clock.

Core Output Clocks

YB and YC are known as Core Outputs and can be used to drive internal logic without using global network resources. This is especially helpful when global network resources must be conserved and utilized for other timing-critical paths.

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

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

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

PLL Macro Block Diagram

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

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

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

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

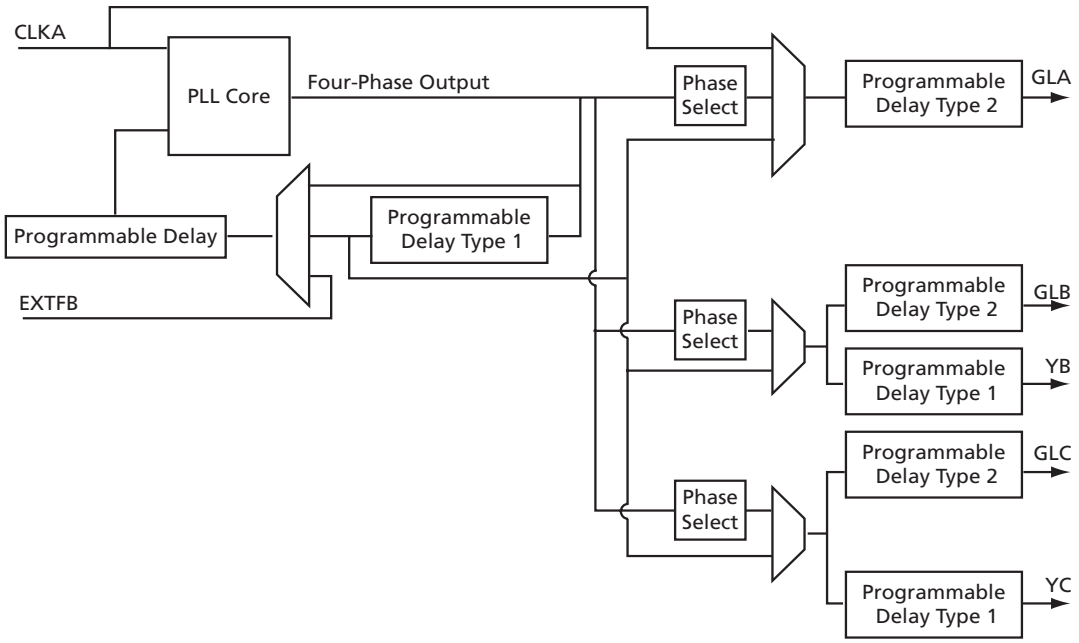
1. By an INBUF* macro to create a composite macro, where the I/O macro drives the global buffer (with programmable delay) using a hardwired connection. In this case, the I/O must be placed in one of the dedicated global I/O locations.
2. Driven directly from the FPGA core.
3. From an I/O that is routed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

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

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

SmartGen also allows the user to select the various delays and phase shift values necessary to adjust the phases between the reference clock (CLKA) and the derived clocks (GLA, GLB, GLC, YB, and YC).

SmartGen allows the user to select the input clock source. SmartGen automatically instantiates the special macro, PLLINT, when needed.



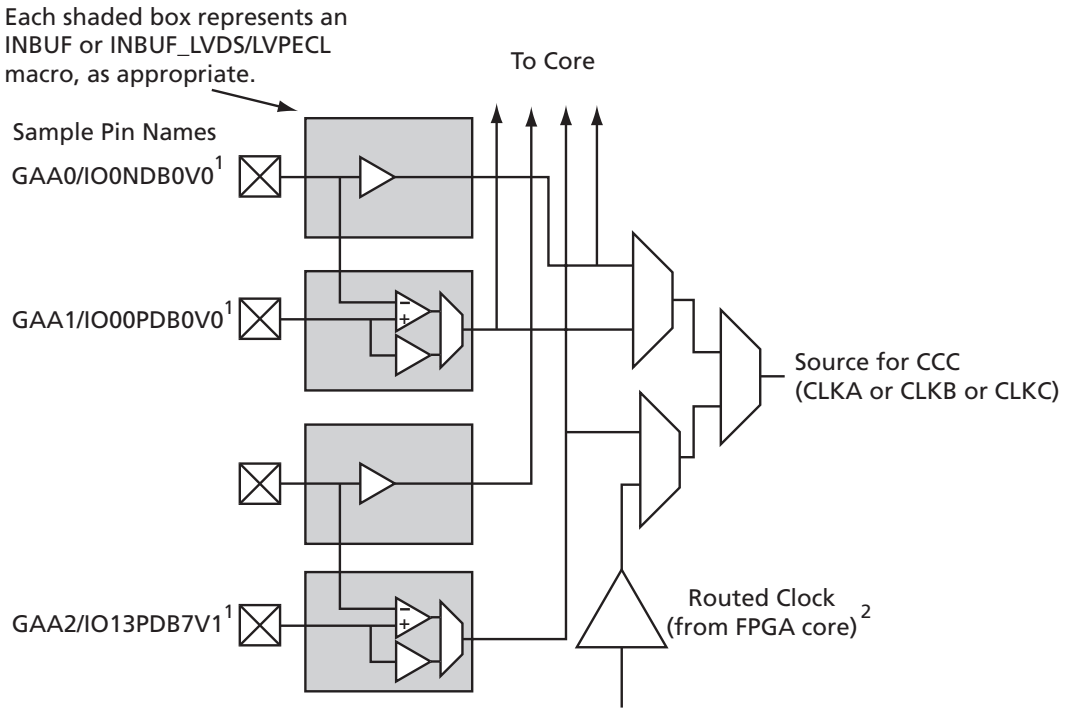
Note: Clock divider and clock multiplier blocks are not shown in this figure or in SmartGen. They are automatically configured based on the user's required frequencies.

Figure 4-5 • **CCC with PLL Block**

Global Input Selections

Low-power flash devices provide the flexibility of choosing one of the three global input pad locations available to connect to a CCC functional block or to a global / quadrant global network. Figure 4-6 on page 4-11 shows the detailed architecture of each global input structure. If the single-ended I/O standard is chosen, there is flexibility to choose one of the global input pads (the first, second, and fourth input). Once chosen, the other I/O locations are used as regular I/Os. If the differential I/O standard is chosen, the first and second inputs are considered as paired, and the third input is paired with a regular I/O.

The user then has the choice of selecting one of the two sets to be used as the clock input source to the CCC functional block. There is also the option to allow an internal clock signal to feed the global network or the CCC functional block. A multiplexer tree selects the appropriate global input for routing to the desired location. Note that the global I/O pads do not need to feed the global network; they can also be used as regular I/O pads.



Notes:

1. *Represents the global input pins. Globals have direct access to the clock conditioning block and are not routed via the FPGA fabric. Refer to User I/O Naming Conventions in I/O Structures in IGLOO and ProASIC3 Devices.*
2. *Instantiate the routed clock source input as follows:*
   a) Connect the output of a logic element to the clock input of a PLL, CLKDLY, or CLKINT macro.
   b) Do not place a clock source I/O (INBUF or INBUF_LVPECL/LVDS/B-LVDS/M-LVDS/DDR) in a relevant global pin location.

Figure 4-6 • **Clock Input Sources Including CLKBUF, CLKBUF_LVDS/LVPECL, and CLKINT**

Each global buffer, as well as the PLL reference clock, can be driven from one of the following:

- 3 dedicated single-ended I/Os using a hardwired connection
- 2 dedicated differential I/Os using a hardwired connection
- The FPGA core

Since the architecture of the devices varies as size increases, the following list details I/O types supported for globals:

IGLOO and ProASIC3

- LVDS-, B-LVDS-, and M-LVDS–based clock sources are only available on 250 k gate devices and above.
- 65 k and 125 k gate devices support single-ended clock sources only.
- 15 k and 30 k gate devices support these inputs for CCC only and do not contain a PLL.

Fusion

- AFS600 and AFS1500: All single-ended, differential, and voltage-referenced I/O standards (Pro I/O).
- AFS090 and AFS250: All single-ended and differential I/O standards.

Clock Sources for PLL and CLKDLY Macros

The input reference clock (CLKA for a PLL macro, CLK for a CLKDLY macro) can be accessed from different sources via the associated clock multiplexer tree. Each CCC has the option of choosing the source of the input clock from one of the following:

- Hardwired I/O
- External I/O
- Core Logic
- RC Oscillator (Fusion only)
- Crystal Oscillator (Fusion only)

The SmartGen macro builder tool allows users to easily create the PLL and CLKDLY macros with the desired settings. Actel strongly recommends using SmartGen be used to generate the CCC macros.

Hardwired I/O Clock Source

Hardwired I/O refers to global input pins that are hardwired to the multiplexer tree, which directly accesses the CCC global buffers. These global input pins have designated pin locations and are indicated with the I/O naming convention *Gmn* (*m* refers to any one of the positions where the PLL core is available, and *n* refers to any one of the three global input MUXes and the pin number of the associated global location, *m*). Choosing this option provides the benefit of directly connecting to the CCC reference clock input, which provides less delay. See Figure 4-7 for an example illustration of the connections, shown in red. If a CLKDLY macro is initiated to utilize the programmable delay element of the CCC, the clock input can be placed at one of nine dedicated global input pin locations. In other words, if Hardwired I/O is chosen as the input source, the user can decide to place the input pin in one of the GmA0, GmA1, GmA2, GmB0, GmB1, GmB2, GmC0, GmC1, or GmC2 locations of the low-power flash devices. When a PLL macro is used to utilize the PLL core in a CCC location, the clock input of the PLL can only be connected to one of three GmA* global pin locations: GmA0, GmA1, or GmA2.

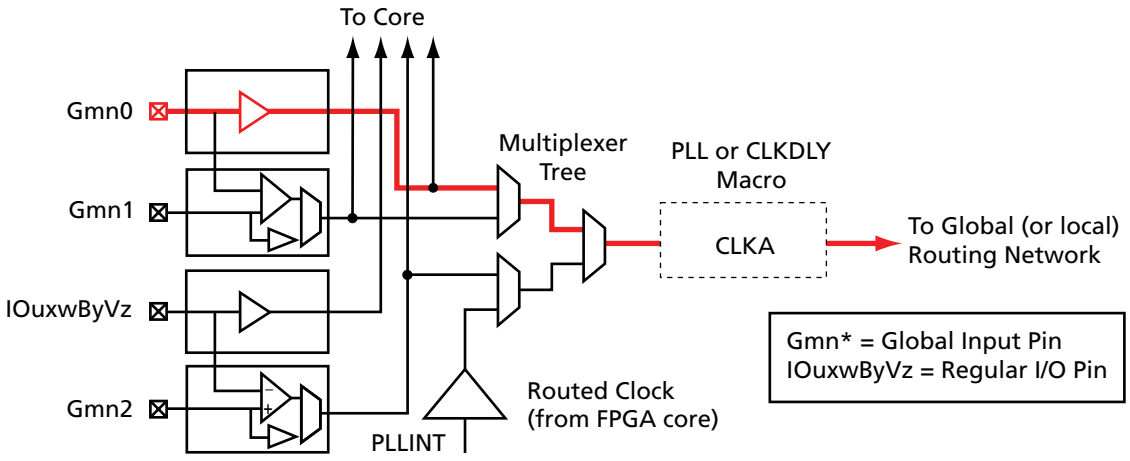


Figure 4-7 • **Illustration of Hardwired I/O (global input pins) Usage**

External I/O Clock Source

External I/O refers to regular I/O pins. The clock source is instantiated with one of the various INBUF options and accesses the CCCs via internal routing. The user has the option of assigning this input to any of the I/Os labeled with the I/O convention *IOuxwByVz*. *Refer to User I/O Naming Conventions in I/O Structures in IGLOO and ProASIC3 Devices and for Fusion, refer to the Fusion Mixed-Signal Programmable System Chip datasheet for more information.* Figure 4-8 gives a brief explanation of external I/O usage. Choosing this option provides the freedom of selecting any user I/O location but introduces additional delay because the signal connects to the routed clock input through internal routing before connecting to the CCC reference clock input.

For the External I/O option, the routed signal would be instantiated with a PLLINT macro before connecting to the CCC reference clock input. This instantiation is conveniently done automatically by SmartGen when this option is selected. Actel recommends using the SmartGen tool to generate the CCC macro. The instantiation of the PLLINT macro results in the use of the routed clock input of the I/O to connect to the PLL clock input. If not using SmartGen, manually instantiate a PLLINT macro before the PLL reference clock to indicate that the regular I/O driving the PLL reference clock should be used (see Figure 4-8 for an example illustration of the connections, shown in red).

In the above two options, the clock source must be instantiated with one of the various INBUF macros. The reference clock pins of the CCC functional block core macros must be driven by regular input macros (INBUFs), not clock input macros.

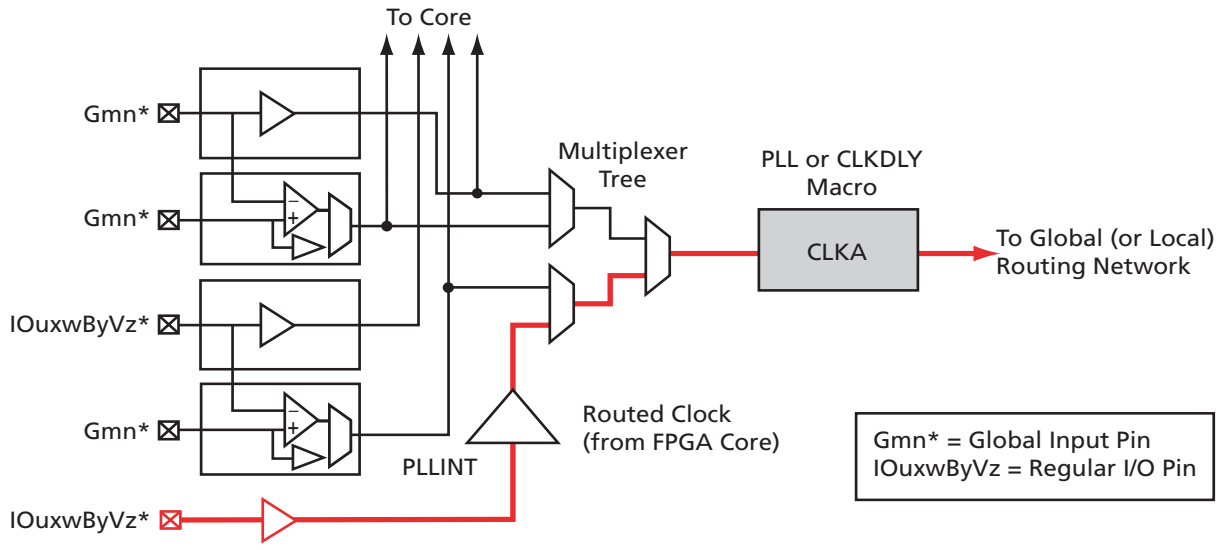


Figure 4-8 • **Illustration of External I/O Usage**

For Fusion devices, the input reference clock can also be from the embedded RC oscillator and crystal oscillator. In this case, the CCC configuration is the same as the hardwired I/O clock source, and users are required to instantiate the RC oscillator or crystal oscillator macro and connect its output to the input reference clock of the CCC block.

Core Logic Clock Source

Core logic refers to internal routed nets. Internal routed signals access the CCC via the FPGA Core Fabric. Similar to the External I/O option, whenever the clock source comes internally from the core itself, the routed signal is instantiated with a PLLINT macro before connecting to the CCC clock input (see Figure 4-9 for an example illustration of the connections, shown in red).

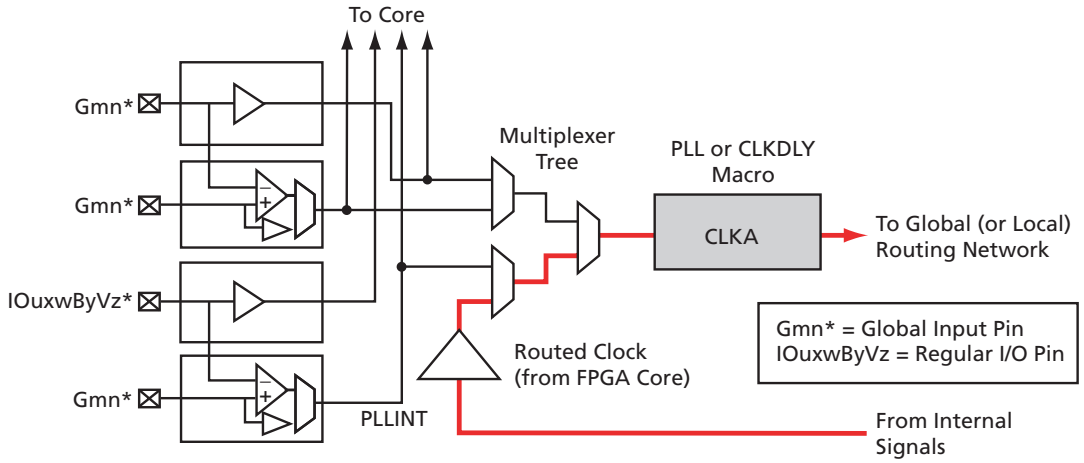


Figure 4-9 • **Illustration of Core Logic Usage**

For Fusion devices, the input reference clock can also be from the embedded RC oscillator and crystal oscillator. In this case, the CCC configuration is the same as the hardwired I/O clock source, and users are required to instantiate the RC oscillator or crystal oscillator macro and connect its output to the input reference clock of the CCC block.

Available I/O Standards

Table 4-4 • **Available I/O Standards within CLKBUF and CLKBUF_LVDS/LVPECL Macros**

CLKBUF_LVCMOS5
CLKBUF_LVCMOS33 [1]
CLKBUF_LVCMOS25 [2]
CLKBUF_LVCMOS18
CLKBUF_LVCMOS15
CLKBUF_PCI
CLKBUF_PCIX [3]
CLKBUF_GTL25 [2,3]
CLKBUF_GTL33 [2,3]
CLKBUF_GTLP25 [2,3]
CLKBUF_GTLP33 [2,3]
CLKBUF_HSTL_I [2,3]
CLKBUF_HSTL_II [2,3]
CLKBUF_SSTL3_I [2,3]
CLKBUF_SSTL3_II [2,3]
CLKBUF_SSTL2_I [2,3]
CLKBUF_SSTL2_II [2,3]
CLKBUF_LVDS [4]
CLKBUF_LVPECL

Notes:

1. *By default, the CLKBUF macro uses 3.3 V LVTTL I/O technology. For more details, refer to the* IGLOO, Fusion, and ProASIC3 Macro Library Guide.
2. *I/O standards only supported in ProASIC3E and IGLOOe families.*
3. *I/O standards only supported in the following Fusion devices: AFS600 and AFS1500.*
4. *B-LVDS and M-LVDS standards are supported by CLKBUF_LVDS.*

Global Synthesis Constraints

The Synplify® synthesis tool, by default, allows six clocks in a design for Fusion, IGLOO, and ProASIC3. When more than six clocks are needed in the design, a user synthesis constraint attribute, syn_global_buffers, can be used to control the maximum number of clocks (up to 18) that can be inferred by the synthesis engine.

High-fanout nets will be inferred with clock buffers and/or internal clock buffers. If the design consists of CCC global buffers, they are included in the count of clocks in the design.

The subsections below discuss the clock input source (global buffers with no programmable delays) and the clock conditioning functional block (global buffers with programmable delays and/or PLL function) in detail.

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/ProASIC3L	**IGLOO/IGLOO PLUS**			
A3P015	AGL015	All	0	2
A3P030	AGL030/AGLP030	All	0	2
A3P060	AGL060/AGLP060	All	1	5
A3P125	AGL125/AGLP125	All	1	5
A3P250/L	AGL250	All	1	5
A3P400		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

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

Clock Conditioning Circuits with Integrated PLLs

Each of the CCCs with integrated PLLs includes the following:

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

Clock Conditioning Circuits without Integrated PLLs

Each of the simplified CCCs without integrated PLLs in the low-power flash families is composed of the following:

- 3 global multiplexer blocks that steer signals from the global pads and the programmable delay elements onto the global networks
- 3 programmable delay elements to provide time delay adjustments

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 4-10.

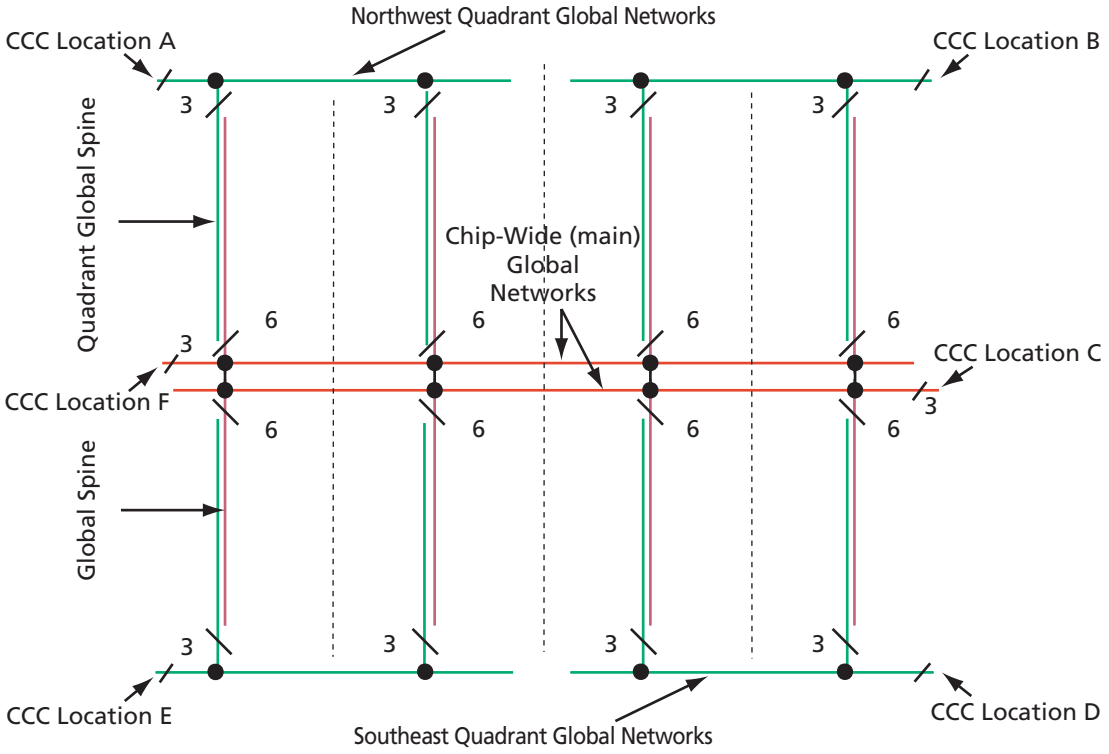


Figure 4-10 • **Global Network Architecture**

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

In Figure 4-13 on page 4-20 through Figure 4-14 on page 4-20, 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."

IGLOO and ProASIC3 CCC Locations

In all IGLOO and ProASIC3 devices (except 15 k and 30 k gate devices, which do not contain PLLs), six CCCs are located in the same positions as the IGLOOe and ProASIC3E CCCs. Only one of the CCCs has an integrated PLL and is located in the middle of the west (middle left) side of the device. The other five CCCs are simplified CCCs and are located in the four corners and the middle of the east side of the device (Figure 4-11).

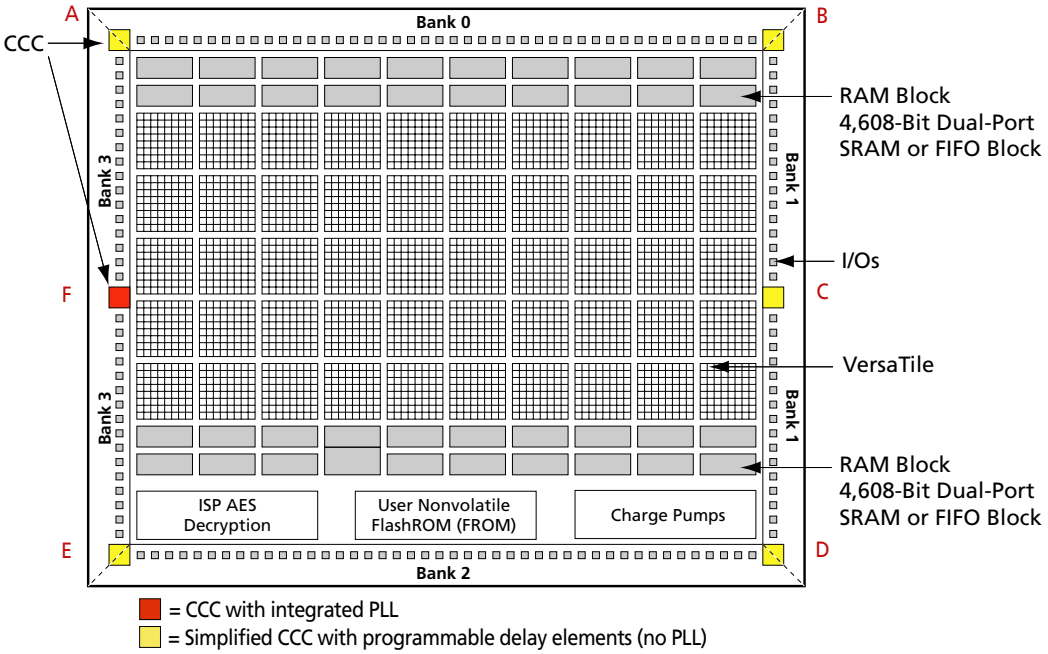


***Figure 4-11* • CCC Locations in IGLOO and ProASIC3 Family Devices (except 15 k and 30 k gate devices)**

Note that the 15 k and 30 k gate devices do not support PLL features. This device has only six global I/O buffers: three on the east and three on the west, located in the middle of the side of the device. A CCC functional block is available in each of the two locations, for a total of two CCCs. No quadrant global networks are present in the four corners of this device (Figure 4-12).

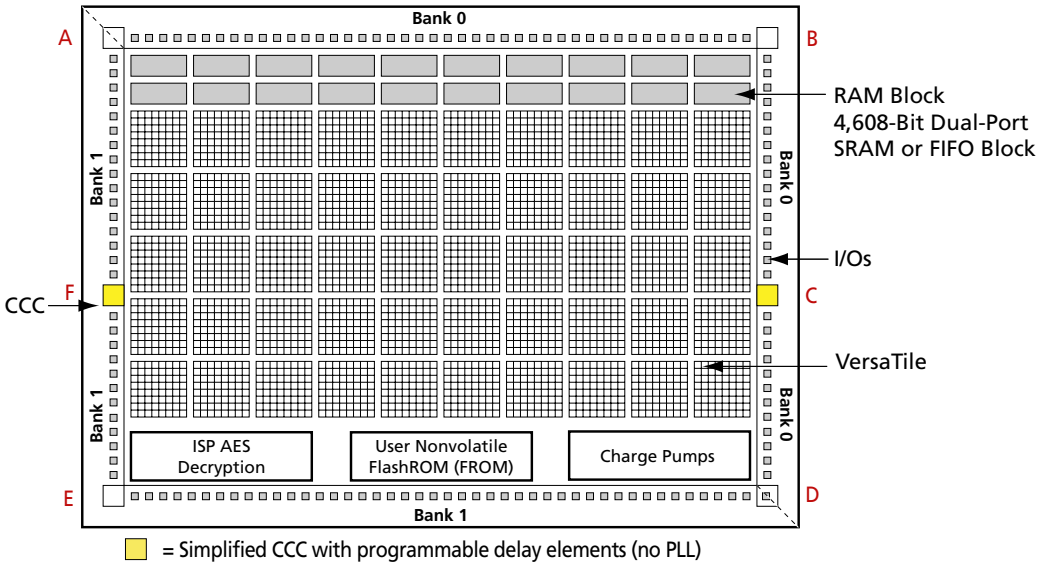


***Figure 4-12* • CCC Locations in the 15 k and 30 k Gate Devices**

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 4-13).

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 4-14).

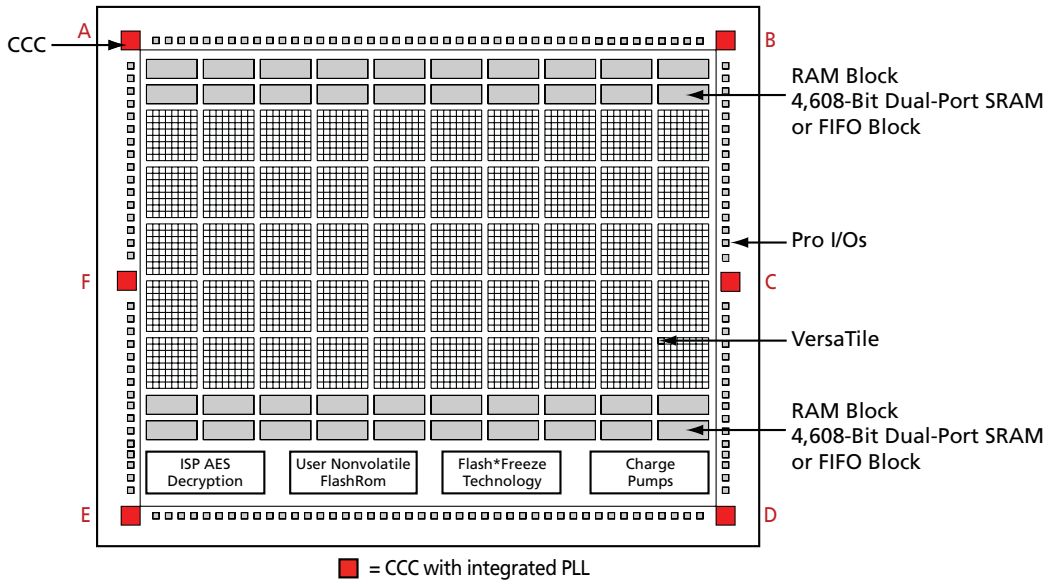


Figure 4-13 • **CCC Locations in IGLOOe and ProASIC3E Family Devices (except PQFP-208 package)**

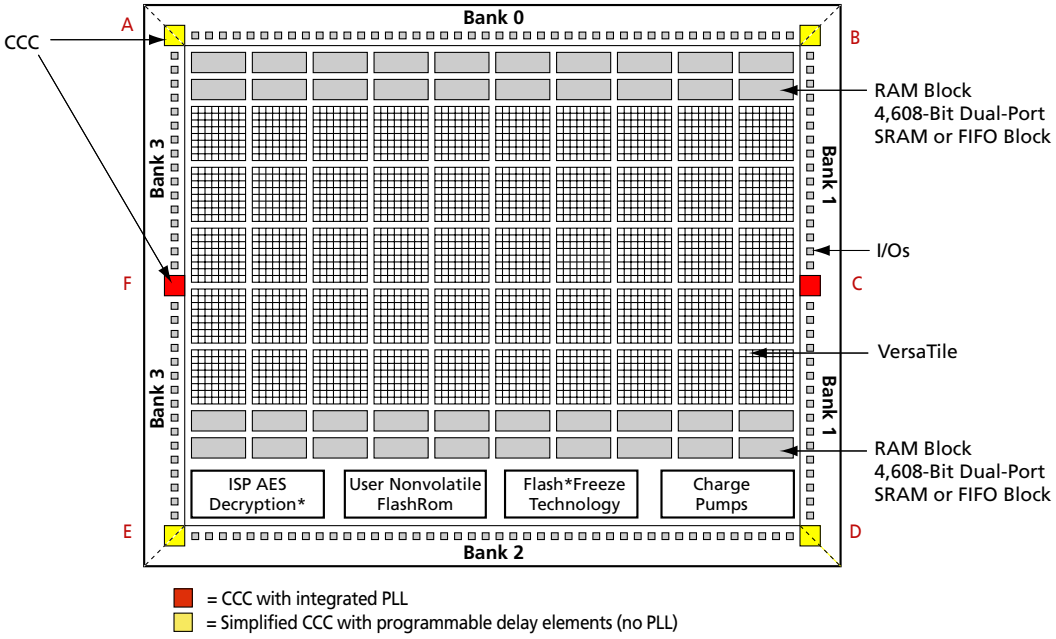


Figure 4-14 • **CCC Locations in ProASIC3E Family Devices (PQFP-208 package)**

Fusion CCC Locations

Fusion 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 and Figure 4-16). The device can have one integrated PLL in the middle of the west side of the device or two integrated PLLs in the middle of the east and west sides of the device (middle right and middle left).

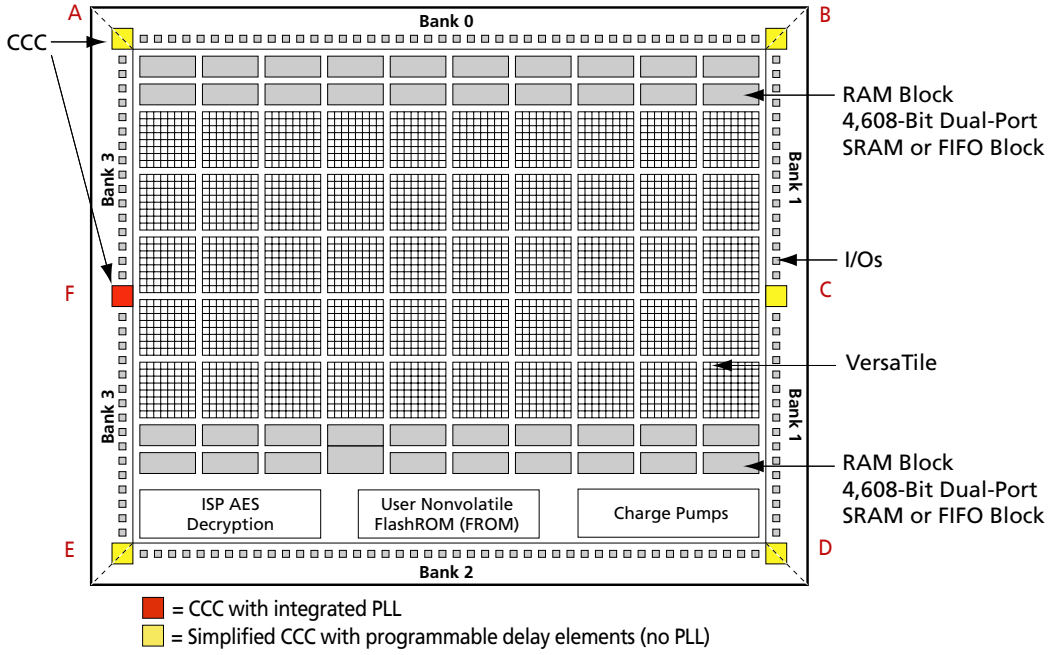


Figure 4-15 • **CCC Locations in Fusion Family Devices (AFS090, AFS250, M1AFS250)**

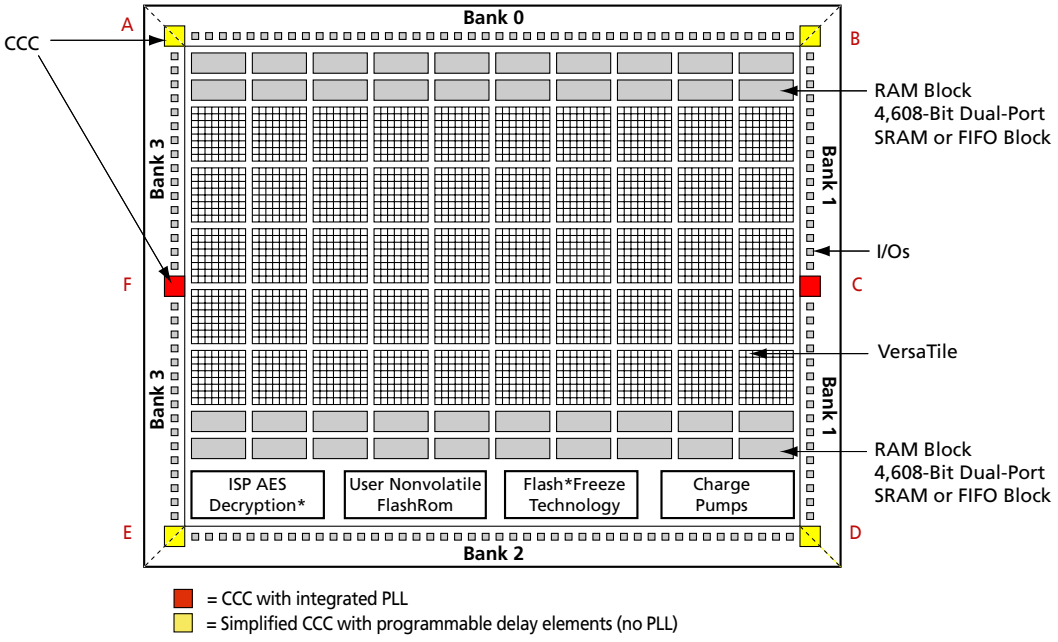


Figure 4-16 • **CCC Locations in Fusion Family Devices (except AFS090, AFS250, M1AFS250)**

PLL Core Specifications

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

Loop Bandwidth

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

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

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

PLL Core Operating Principles

This section briefly describes the basic principles of PLL operation. The PLL core is composed of a phase detector (PD), a low-pass filter (LPF), and a four-phase voltage-controlled oscillator (VCO). Figure 4-17 illustrates a basic single-phase PLL core with a divider and delay in the feedback path.

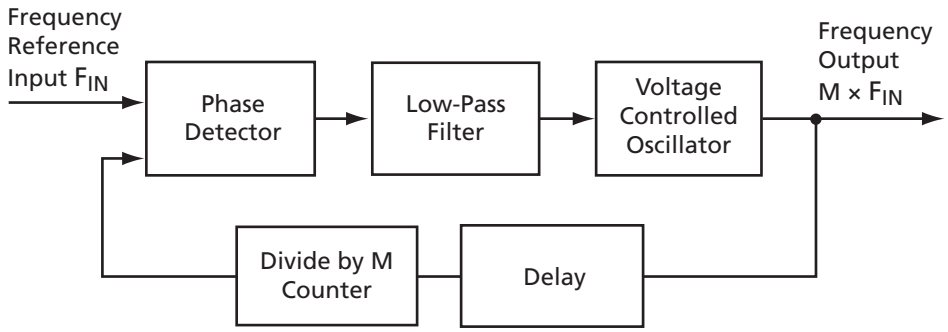


Figure 4-17 • **Simplified PLL Core with Feedback Divider and Delay**

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

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

Consider Figure 4-17 with the feedback path bypassing the divider and delay elements. If the LPF steadily applies a voltage to the VCO such that the output frequency is identical to the input frequency, this steady-state condition is known as lock. Note that the input and output phases are also identical. The PLL core sets a LOCK output signal HIGH to indicate this condition.

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

Now suppose the feedback divider is inserted in the feedback path. As the division factor M (shown in Figure 4-18) is increased, the average phase difference increases. The average phase difference will cause the VCO to increase its frequency until the output signal is phase-identical to the input after undergoing division. In other words, lock in both frequency and phase is achieved when the output frequency is M times the input. Thus, clock division in the feedback path results in multiplication at the output.

A similar argument can be made when the delay element is inserted into the feedback path. To achieve steady-state lock, the VCO output signal will be delayed by the input period *less* the feedback delay. For periodic signals, this is equivalent to time-advancing the output clock by the feedback delay.

Another key parameter of a PLL system is the acquisition time. Acquisition time is the amount of time it takes for the PLL to achieve lock (i.e., phase-align the feedback signal with the input reference clock). For example, suppose there is no voltage applied to the VCO, allowing it to operate at its free-running frequency. Should an input reference clock suddenly appear, a lock would be established within the maximum acquisition time.

Functional Description

This section provides detailed descriptions of PLL block functionality: clock dividers and multipliers, clock delay adjustment, phase adjustment, and dynamic PLL configuration.

Clock Dividers and Multipliers

The PLL block contains five programmable dividers. Figure 4-18 shows a simplified PLL block.

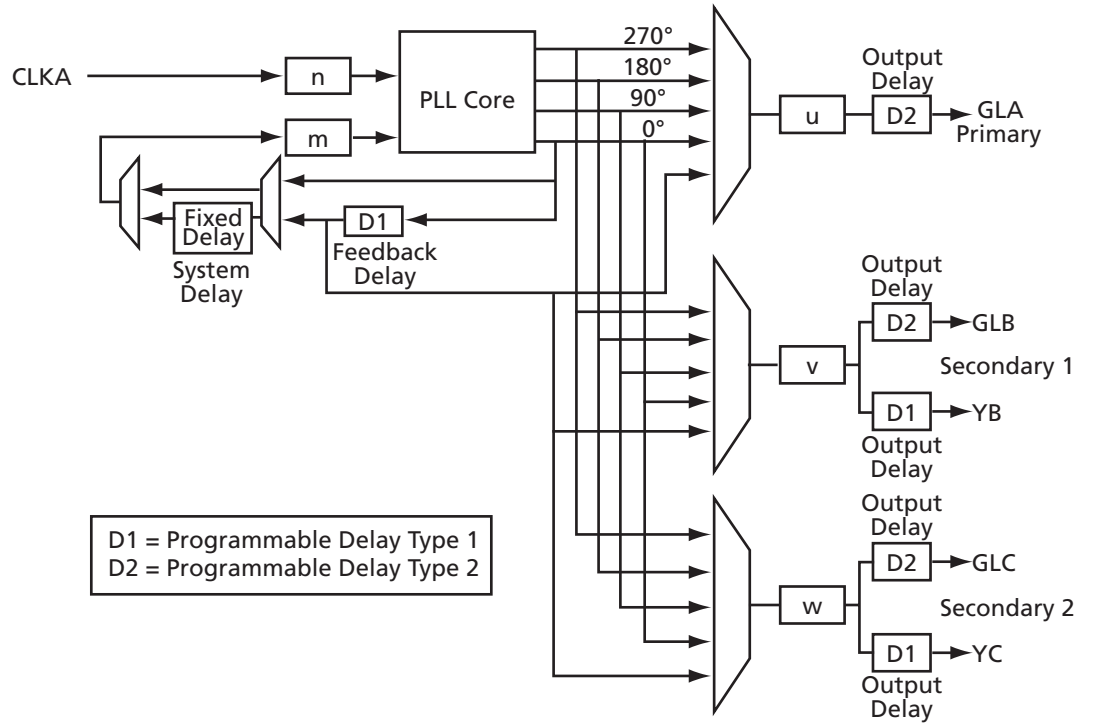


***Figure 4-18* • PLL Block Diagram**

Dividers *n* and *m* (the input divider and feedback divider, respectively) provide integer frequency division factors from 1 to 128. The output dividers *u*, *v*, and *w* provide integer division factors from 1 to 32. Frequency scaling of the reference clock CLKA is performed according to the following formulas:

$$f_{GLA} = f_{CLKA} \times m / (n \times u) – \text{GLA Primary PLL Output Clock}$$

EQ 4-1

$$f_{GLB} = f_{YB} = f_{CLKA} \times m / (n \times v) – \text{GLB Secondary 1 PLL Output Clock(s)}$$

EQ 4-2

$$f_{GLC} = f_{YC} = f_{CLKA} \times m / (n \times w) – \text{GLC Secondary 2 PLL Output Clock(s)}$$

EQ 4-3

SmartGen provides a user-friendly method of generating the configured PLL netlist, which includes automatically setting the division factors to achieve the closest possible match to the requested frequencies. Since the five output clocks share the *n* and *m* dividers, the achievable output frequencies are interdependent and related according to the following formula:

$$f_{GLA} = f_{GLB} \times (v / u) = f_{GLC} \times (w / u)$$

EQ 4-4

Clock Delay Adjustment

There are a total of seven configurable delay elements implemented in the PLL architecture.

Two of the delays are located in the feedback path, entitled System Delay and Feedback Delay. System Delay provides a fixed delay of 2 ns (typical), and Feedback Delay provides selectable delay values from 0.6 ns to 5.56 ns in 160 ps increments (typical). For PLLs, delays in the feedback path will effectively advance the output signal from the PLL core with respect to the reference clock. Thus, the System and Feedback delays generate negative delay on the output clock. Additionally, each of these delays can be independently bypassed if necessary.

The remaining five delays perform traditional time delay and are located at each of the outputs of the PLL. Besides the fixed global driver delay of 0.755 ns for each of the global networks, the global multiplexer outputs (GLA, GLB, and GLC) each feature an additional selectable delay value from 0.025 ns to 0.76 ns in the first step, and then to 5.56 ns in 160 ps increments. The additional YB and YC signals have access to a selectable delay from 0.6 ns to 5.56 ns in 160 ps increments (typical). This is the same delay value as the CLKDLY macro. It is similar to CLKDLY, which bypasses the PLL core just to take advantage of the phase adjustment option with the delay value.

The following parameters must be taken into consideration to achieve minimum delay at the outputs (GLA, GLB, GLC, YB, and YC) relative to the reference clock: routing delays from the PLL core to CCC outputs, core outputs and global network output delays, and the feedback path delay. The feedback path delay acts as a time advance of the input clock and will offset any delays introduced beyond the PLL core output. The routing delays are determined from back-annotated simulation and are configuration-dependent.

Phase Adjustment

The output from the PLL core can be phase-adjusted with respect to the reference input clock, CLKA. The user can select a 0°, 90°, 180°, or 270° phase shift independently for each of the outputs YA, GLB/YB, and GLC/YC. Note that each of these phase-adjusted signals might also undergo further frequency division and/or time adjustment via the remaining dividers and delays located at the outputs of the PLL.

Dynamic PLL Configuration

The CCCs can be configured both statically and dynamically.

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

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

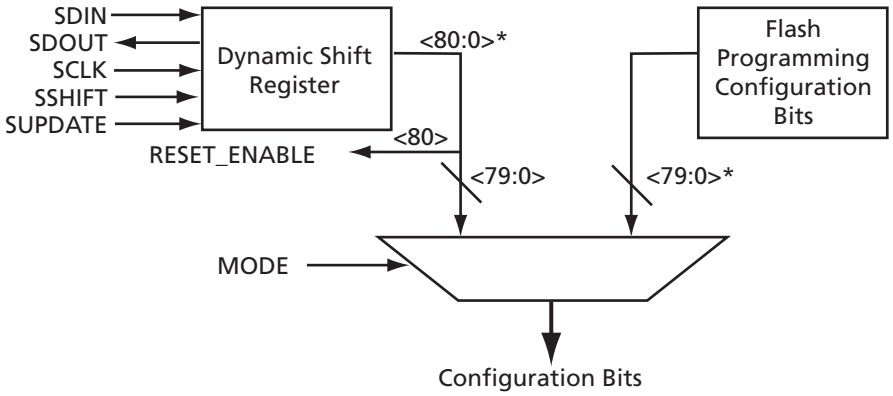
Flash Configuration Bits

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

Dynamic Shift Register Outputs

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

Figure 4-19 illustrates a simplified block diagram of the MUX architecture in the CCCs.



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

***Figure 4-19* • The CCC Configuration MUX Architecture**

The selection between the flash configuration bits and the bits from the configuration register is made using the MODE signal shown in Figure 4-19. If the MODE signal is logic HIGH, the dynamic shift register configuration bits are selected. There are 81 control bits to configure the different functions of the CCC.

Each group of control bits is assigned a specific location in the configuration shift register. For a list of the 81 configuration bits (C[80:0]) in the CCC and a description of each, refer to "PLL Configuration Bits Description" on page 4-27. The configuration register can be serially loaded with the new configuration data and programmed into the CCC using the following ports:

- SDIN: The configuration bits are serially loaded into a shift register through this port. The LSB of the configuration data bits should be loaded first.
- SDOUT: The shift register contents can be shifted out (LSB first) through this port using the shift operation.
- SCLK: This port should be driven by the shift clock.
- SSHIFT: The active-high shift enable signal should drive this port. The configuration data will be shifted into the shift register if this signal is HIGH. Once SSHIFT goes LOW, the data shifting will be halted.
- SUPDATE: The SUPDATE signal is used to configure the CCC with the new configuration bits when shifting is complete.

To access the configuration ports of the shift register (SDIN, SDOUT, SSHIFT, etc.), the user should instantiate the CCC macro in his design with appropriate ports. Actel recommends that users choose SmartGen to generate the CCC macros with the required ports for dynamic reconfiguration.

Users must familiarize themselves with the architecture of the CCC core and its input, output, and configuration ports to implement the desired delay and output frequency in the CCC structure. Figure 4-20 shows a model of the CCC with configurable blocks and switches.

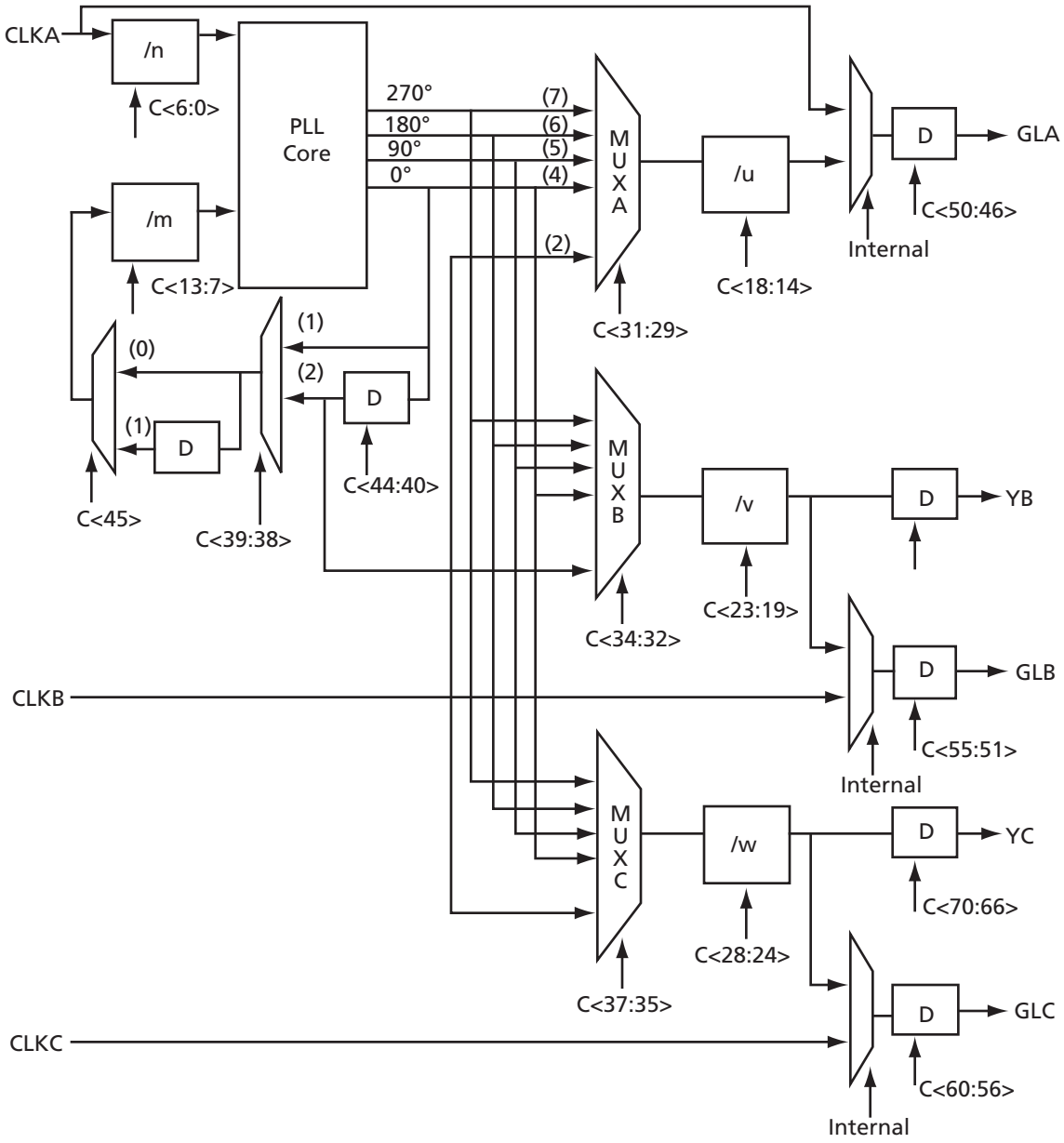


Figure 4-20 • **CCC Block Control Bits – Graphical Representation of Assignments**

Loading the Configuration Register

The most important part of CCC dynamic configuration is to load the shift register properly with the configuration bits. There are different ways to access and load the configuration shift register:

- JTAG interface
- Logic core
- Specific I/O tiles

JTAG Interface

The JTAG interface requires no additional I/O pins. The JTAG TAP controller is used to control the loading of the CCC configuration shift register.

Low-power flash devices provide a user interface macro between the JTAG pins and the device core logic. This macro is called UJTAG. A user should instantiate the UJTAG macro in his design to access the configuration register ports via the JTAG pins.

For more information on CCC dynamic reconfiguration using UJTAG, refer to *UJTAG Applications in Actel's Low-Power Flash Devices.*

Logic Core

If the logic core is employed, the user must design a module to provide the configuration data and control the shifting and updating of the CCC configuration shift register. In effect, this is a user-designed TAP controller, which requires additional chip resources.

Specific I/O Tiles

If specific I/O tiles are used for configuration, the user must provide the external equivalent of a TAP controller. This does not require additional core resources but does use pins.

Shifting the Configuration Data

To enter a new configuration, all 81 bits must shift in via SDIN. After all bits are shifted, SSHIFT must go LOW and SUPDATE HIGH to enable the new configuration. For simulation purposes, bits <71:73> and <77:80> are "don't cares."

The SUPDATE signal must be LOW during any clock cycle where SSHIFT is active. After SUPDATE is asserted, it must go back to the LOW state until a new update is required.

PLL Configuration Bits Description

Table 4-7 • **Configuration Bit Descriptions for the CCC Blocks**

Config. Bits	Signal	Name	Description
<88:87>	GLMUXCFG [1:0][1]	NGMUX configuration	The configuration bits specify the input clocks to the NGMUX (refer to Table 4-16 on page 4-31).[2]
86	OCDIVHALF[1]	Division by half	When the PLL is bypassed, the 100 MHz RC oscillator can be divided by the divider factor in Table 4-17 on page 4-32).
85	OBDIVHALF[1]	Division by half	When the PLL is bypassed, the 100 MHz RC oscillator can be divided by certain 0.5 factor (Refer to Table 4-17 on page 4-32).
84	OADIVHALF[1]	Division by half	When the PLL is bypassed, the 100 MHz RC oscillator can be divided by certain 0.5 factor (Refer to Table 4-15 on page 4-31).
83	RXCSEL[1]	CLKC input selection	Select the CLKC input clock source between RC oscillator and crystal oscillator (refer to Table 4-15 on page 4-31).[2]
82	RXBSEL[1]	CLKB input selection	Select the CLKB input clock source between RC oscillator and crystal oscillator (refer to Table 4-15 on page 4-31).[2]

Notes:

1. *The <88:81> configuration bits are only for the Fusion dynamic CCC.*
2. *This value depends on the input clock source, so Layout must complete before these bits can be set. After completing Layout in Designer, generate the "CCC_Configuration" report by choosing* ***Tools*** *>* ***Report*** *>* ***CCC_Configuration****. The report contains the appropriate settings for these bits.*

Table 4-7 • **Configuration Bit Descriptions for the CCC Blocks (continued)**

Config. Bits	Signal	Name	Description
81	RXASEL[1]	CLKA input selection	Select the CLKA input clock source between RC oscillator and crystal oscillator (refer to Table 4-15 on page 4-31).[2]
80	RESETEN	Reset Enable	Enables (active high) the synchronization of PLL output dividers after dynamic reconfiguration (SUPDATE). The Reset Enable signal is READ-ONLY and should not be modified via dynamic reconfiguration.
79	DYNCSEL	Clock Input C Dynamic Select	Configures clock input C to be sent to GLC for dynamic control.[2]
78	DYNBSEL	Clock Input B Dynamic Select	Configures clock input B to be sent to GLB for dynamic control.[2]
77	DYNASEL	Clock Input A Dynamic Select	Configures clock input A for dynamic PLL configuration.[2]
<76:74>	VCOSEL[2:0]	VCO Gear Control	Three-bit VCO Gear Control for four frequency ranges (refer to Table 4-18 on page 4-32 and Table 4-19 on page 4-32).
73	STATCSEL	MUX Select on Input C	MUX selection for clock input C[2]
72	STATBSEL	MUX Select on Input B	MUX selection for clock input B[2]
71	STATASEL	MUX Select on Input A	MUX selection for clock input A[2]
<70:66>	DLYC[4:0]	YC Output Delay	Sets the output delay value for YC.
<65:61>	DLYB[4:0]	YB Output Delay	Sets the output delay value for YB.
<60:56>	DLYGLC[4:0]	GLC Output Delay	Sets the output delay value for GLC.
<55:51>	DLYGLB[4:0]	GLB Output Delay	Sets the output delay value for GLB.
<50:46>	DLYGLA[4:0]	Primary Output Delay	Primary, GLA Output Delay
45	XDLYSEL	System Delay Select	When selected, inserts System Delay in the feedback path in Figure 4-18 on page 4-23.
<44:40>	FBDLY[4:0]	Feedback Delay	Sets the feedback delay value for the feedback element in Figure 4-18 on page 4-23.
<39:38>	FBSEL[1:0]	Primary Feedback Delay Select	Controls the feedback MUX: no delay, include programmable delay element, or use external feedback.
<37:35>	OCMUX[2:0]	Secondary 2 Output Select	Selects from the VCO's four phase outputs for GLC/YC.
<34:32>	OBMUX[2:0]	Secondary 1 Output Select	Selects from the VCO's four phase outputs for GLB/YB.
<31:29>	OAMUX[2:0]	GLA Output Select	Selects from the VCO's four phase outputs for GLA.

Notes:

1. *The <88:81> configuration bits are only for the Fusion dynamic CCC.*
2. *This value depends on the input clock source, so Layout must complete before these bits can be set. After completing Layout in Designer, generate the "CCC_Configuration" report by choosing* ***Tools*** *>* ***Report*** *>* ***CCC_Configuration****. The report contains the appropriate settings for these bits.*

Table 4-7 • Configuration Bit Descriptions for the CCC Blocks (continued)

Config. Bits	Signal	Name	Description
<28:24>	OCDIV[4:0]	Secondary 2 Output Divider	Sets the divider value for the GLC/YC outputs. Also known as divider *w* in Figure 4-18 on page 4-23. The divider value will be OCDIV[4:0] + 1.
<23:19>	OBDIV[4:0]	Secondary 1 Output Divider	Sets the divider value for the GLB/YB outputs. Also known as divider *v* in Figure 4-18 on page 4-23. The divider value will be OBDIV[4:0] + 1.
<18:14>	OADIV[4:0]	Primary Output Divider	Sets the divider value for the GLA output. Also known as divider *u* in Figure 4-18 on page 4-23. The divider value will be OADIV[4:0] + 1.
<13:7>	FBDIV[6:0]	Feedback Divider	Sets the divider value for the PLL core feedback. Also known as divider *m* in Figure 4-18 on page 4-23. The divider value will be FBDIV[6:0] + 1.
<6:0>	FINDIV[6:0]	Input Divider	Input Clock Divider (/n). Sets the divider value for the input delay on CLKA. The divider value will be FINDIV[6:0] + 1.

Notes:

1. *The <88:81> configuration bits are only for the Fusion dynamic CCC.*
2. *This value depends on the input clock source, so Layout must complete before these bits can be set. After completing Layout in Designer, generate the "CCC_Configuration" report by choosing **Tools** > **Report** > **CCC_Configuration**. The report contains the appropriate settings for these bits.*

Table 4-8 to Table 4-14 on page 4-31 provide descriptions of the configuration data for the configuration bits.

Table 4-8 • Input Clock Divider, FINDIV[6:0] (/n)

FINDIV<6:0> State	Divisor	New Frequency Factor
0	1	1.00000
1	2	0.50000
⋮	⋮	⋮
127	128	0.0078125

Table 4-9 • Feedback Clock Divider, FBDIV[6:0] (/m)

FBDIV<6:0> State	Divisor	New Frequency Factor
0	1	1
1	2	2
⋮	⋮	⋮
127	128	128

Table 4-10 • Output Frequency Dividers
A Output Divider, OADIV <4:0> (/u);
B Output Divider, OBDIV <4:0> (/v);
C Output Divider, OCDIV <4:0> (/w)

OADIV<4:0>; OBDIV<4:0>; CDIV<4:0> State	Divisor	New Frequency Factor
0	1	1.00000
1	2	0.50000
⋮	⋮	⋮
31	32	0.03125

Table 4-11 • MUXA, MUXB, MUXC

OAMUX<2:0>; OBMUX<2:0>; OCMUX<2:0> State	MUX Input Selected
0	None. Six-input MUX and PLL are bypassed. Clock passes only through global MUX and goes directly into HC ribs.
1	Not available
2	PLL feedback delay line output
3	Not used
4	PLL VCO 0° phase shift
5	PLL VCO 90° phase shift
6	PLL VCO 180° phase shift
7	PLL VCO 270° phase shift

Table 4-12 • 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 4-13 • 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 4-14 • **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 4-15 • **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
RXBSEL	**DYNCSEL**	**Source of CLKC**
1	0	RC Oscillator
1	1	Crystal Oscillator

Table 4-16 • **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

Table 4-17 • **Fusion Dynamic CCC Division by Half Configuration**

OADIVHALF / OBDIVHALF / OCDIVHALF	OADIV<4:0> / OBDIV<4:0> / OCDIV<4:0> (in decimal)	Divider Factor	Input Clock Frequency	Output Clock Frequency (MHz)
1	2	1.5	100 MHz RC Oscillator	66.7
	4	2.5		40.0
	6	3.5		28.6
	8	4.5		22.2
	10	5.5		18.2
	12	6.5		15.4
	14	7.5		13.3
	16	8.5		11.8
	18	9.5		10.5
	20	10.5		9.5
	22	11.5		8.7
	24	12.5		8.0
	26	13.5		7.4
	28	14.5		6.9
0	0-31	1-32	Other Clock Sources	Depends on other dividers setting

Table 4-18 • **Configuration Bit <76:75> / VCOSEL<2:1> Selection for All Families**

Voltage	VCOSEL[2:1]							
	00		01		10		11	
	Min. (MHz)	Max. (MHz)	Min. (MHz)	Max. (MHz)	Min. (MHz)	Max. (MHz)	Min. (MHz)	Max. (MHz)
IGLOO and IGLOO PLUS								
1.2 V ± 5%	24	35	30	70	60	140	135	160
1.5 V ± 5%	24	43.75	30	87.5	60	175	135	250
ProASIC3L, RT ProASIC3, and Military ProASIC3/L								
1.2 V ± 5%	24	35	30	70	60	140	135	250
1.5 V ± 5%	24	43.75	30	70	60	175	135	350
ProASIC3 and Fusion								
1.5 V ± 5%	24	43.75	33.75	87.5	67.5	175	135	350

Table 4-19 • **Configuration Bit <74> / VCOSEL<0> Selection for All Families**

VCOSEL[0]	Description
0	Fast PLL lock acquisition time with high tracking jitter. Refer to the corresponding datasheet for specific value and definition.
1	Slow PLL lock acquisition time with low tracking jitter. Refer to the corresponding datasheet for specific value and definition.

Software Configuration

SmartGen automatically generates the desired CCC functional block by configuring the control bits, and allows the user to select two CCC modes: Static PLL and Delayed Clock (CLKDLY).

Static PLL Configuration

The newly implemented Visual PLL Configuration Wizard feature provides the user a quick and easy way to configure the PLL with the desired settings (Figure 4-21). The user can invoke SmartGen to set the parameters and generate the netlist file with the appropriate flash configuration bits set for the CCCs. As mentioned in "PLL Macro Block Diagram" on page 4-9, the input reference clock CLKA can be configured to be driven by Hardwired I/O, External I/O, or Core Logic. The user enters the desired settings for all the parameters (output frequency, output selection, output phase adjustment, clock delay, feedback delay, and system delay). Notice that the actual values (divider values, output frequency, delay values, and phase) are shown to aid the user in reaching the desired design frequency in real time. These values are typical-case data. Best- and worst-case data can be observed through static timing analysis in SmartTime within Designer.

For dynamic configuration, the CCC parameters are defined using either the external JTAG port or an internally defined serial interface via the built-in dynamic shift register. This feature provides the ability to compensate for changes in the external environment.

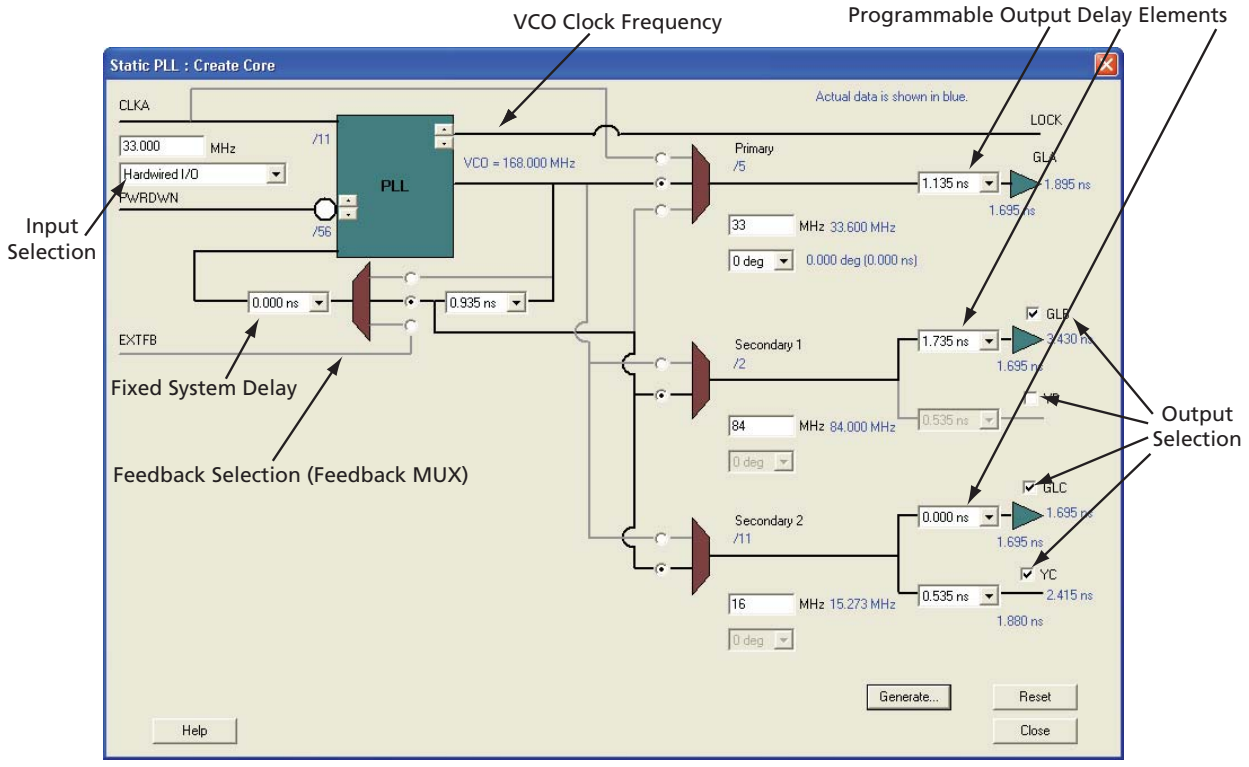


Figure 4-21 • **Visual PLL Configuration Wizard**

Feedback Configuration

The PLL provides both internal and external feedback delays. Depending on the configuration, various combinations of feedback delays can be achieved.

Internal Feedback Configuration

This configuration essentially sets the feedback multiplexer to route the VCO output of the PLL core as the input to the feedback of the PLL. The feedback signal can be processed with the fixed system and the adjustable feedback delay, as shown in Figure 4-22. The dividers are automatically configured by SmartGen based on the user input.

Indicated below is the System Delay pull-down menu. The System Delay can be bypassed by setting it to 0. When set, it adds a 2 ns delay to the feedback path (which results in delay advancement of the output clock by 2 ns).

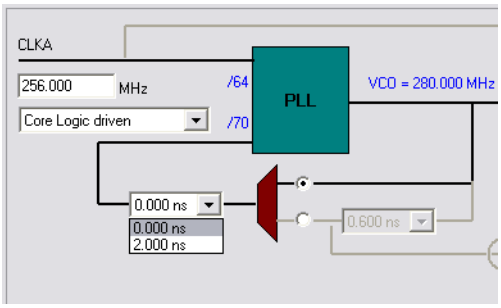


***Figure 4-22* • Internal Feedback with Selectable System Delay**

Figure 4-23 shows the controllable Feedback Delay. If set properly in conjunction with the fixed System Delay, the total output delay can be advanced significantly.

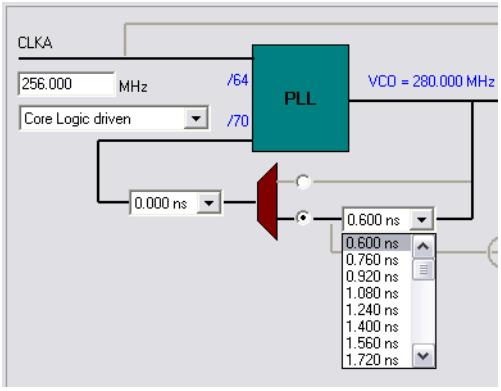


***Figure 4-23* • Internal Feedback with Selectable Feedback Delay**

External Feedback Configuration

For certain applications, such as those requiring generation of PCB clocks that must be matched with existing board delays, it is useful to implement an external feedback, EXTFB. The Phase Detector of the PLL core will receive CLKA and EXTFB as inputs. EXTFB may be processed by the fixed System Delay element as well as the *M* divider element. The EXTFB option is currently not supported.

After setting all the required parameters, users can generate one or more PLL configurations with HDL or EDIF descriptions by clicking the **Generate** button. SmartGen gives the option of saving session results and messages in a log file:

```
****************
Macro Parameters
****************

Name                            : test_pll
Family                          : ProASIC3E
Output Format                   : VHDL
Type                            : Static PLL
Input Freq(MHz)                 : 10.000
CLKA Source                     : Hardwired I/O
Feedback Delay Value Index      : 1
Feedback Mux Select             : 2
XDLY Mux Select                 : No
Primary Freq(MHz)               : 33.000
Primary PhaseShift              : 0
Primary Delay Value Index       : 1
Primary Mux Select              : 4
Secondary1 Freq(MHz)            : 66.000
Use GLB                         : YES
Use YB                          : YES
GLB Delay Value Index           : 1
YB Delay Value Index            : 1
Secondary1 PhaseShift           : 0
Secondary1 Mux Select           : 4
Secondary2 Freq(MHz)            : 101.000
Use GLC                         : YES
Use YC                          : NO
GLC Delay Value Index           : 1
YC Delay Value Index            : 1
Secondary2 PhaseShift           : 0
Secondary2 Mux Select           : 4

…
…
…

Primary Clock frequency 33.333
Primary Clock Phase Shift 0.000
Primary Clock Output Delay from CLKA 0.180

Secondary1 Clock frequency 66.667
Secondary1 Clock Phase Shift 0.000
Secondary1 Clock Global Output Delay from CLKA 0.180
Secondary1 Clock Core Output Delay from CLKA 0.625

Secondary2 Clock frequency 100.000
Secondary2 Clock Phase Shift 0.000
Secondary2 Clock Global Output Delay from CLKA 0.180
```

Below is an example Verilog HDL description of a legal PLL core configuration generated by SmartGen:

```
module test_pll(POWERDOWN,CLKA,LOCK,GLA);
```

```
input POWERDOWN, CLKA;
output  LOCK, GLA;

    wire VCC, GND;

    VCC VCC_1_net(.Y(VCC));
    GND GND_1_net(.Y(GND));
    PLL Core(.CLKA(CLKA), .EXTFB(GND), .POWERDOWN(POWERDOWN),
        .GLA(GLA), .LOCK(LOCK), .GLB(), .YB(), .GLC(), .YC(),
        .OADIV0(GND), .OADIV1(GND), .OADIV2(GND), .OADIV3(GND),
        .OADIV4(GND), .OAMUX0(GND), .OAMUX1(GND), .OAMUX2(VCC),
        .DLYGLA0(GND), .DLYGLA1(GND), .DLYGLA2(GND), .DLYGLA3(GND)
        , .DLYGLA4(GND), .OBDIV0(GND), .OBDIV1(GND), .OBDIV2(GND),
        .OBDIV3(GND), .OBDIV4(GND), .OBMUX0(GND), .OBMUX1(GND),
        .OBMUX2(GND), .DLYYB0(GND), .DLYYB1(GND), .DLYYB2(GND),
        .DLYYB3(GND), .DLYYB4(GND), .DLYGLB0(GND), .DLYGLB1(GND),
        .DLYGLB2(GND), .DLYGLB3(GND), .DLYGLB4(GND), .OCDIV0(GND),
        .OCDIV1(GND), .OCDIV2(GND), .OCDIV3(GND), .OCDIV4(GND),
        .OCMUX0(GND), .OCMUX1(GND), .OCMUX2(GND), .DLYYC0(GND),
        .DLYYC1(GND), .DLYYC2(GND), .DLYYC3(GND), .DLYYC4(GND),
        .DLYGLC0(GND), .DLYGLC1(GND), .DLYGLC2(GND), .DLYGLC3(GND)
        , .DLYGLC4(GND), .FINDIV0(VCC), .FINDIV1(GND), .FINDIV2(
       VCC), .FINDIV3(GND), .FINDIV4(GND), .FINDIV5(GND),
        .FINDIV6(GND), .FBDIV0(VCC), .FBDIV1(GND), .FBDIV2(VCC),
        .FBDIV3(GND), .FBDIV4(GND), .FBDIV5(GND), .FBDIV6(GND),
        .FBDLY0(GND), .FBDLY1(GND), .FBDLY2(GND), .FBDLY3(GND),
        .FBDLY4(GND), .FBSEL0(VCC), .FBSEL1(GND), .XDLYSEL(GND),
        .VCOSEL0(GND), .VCOSEL1(GND), .VCOSEL2(GND));
    defparam Core.VCOFREQUENCY = 33.000;
endmodule
```

The "PLL Configuration Bits Description" section on page 4-27 provides descriptions of the PLL configuration bits for completeness. The configuration bits are shown as busses only for purposes of illustration. They will actually be broken up into individual pins in compilation libraries and all simulation models. For example, the FBSEL[1:0] bus will actually appear as pins FBSEL1 and FBSEL0. The setting of these select lines for the static PLL configuration is performed by the software and is completely transparent to the user.

Dynamic PLL Configuration

To generate a dynamically reconfigurable CCC, the user should select **Dynamic CCC** in the configuration section of the SmartGen GUI (Figure 4-24). This will generate both the CCC core and the configuration shift register / control bit MUX.

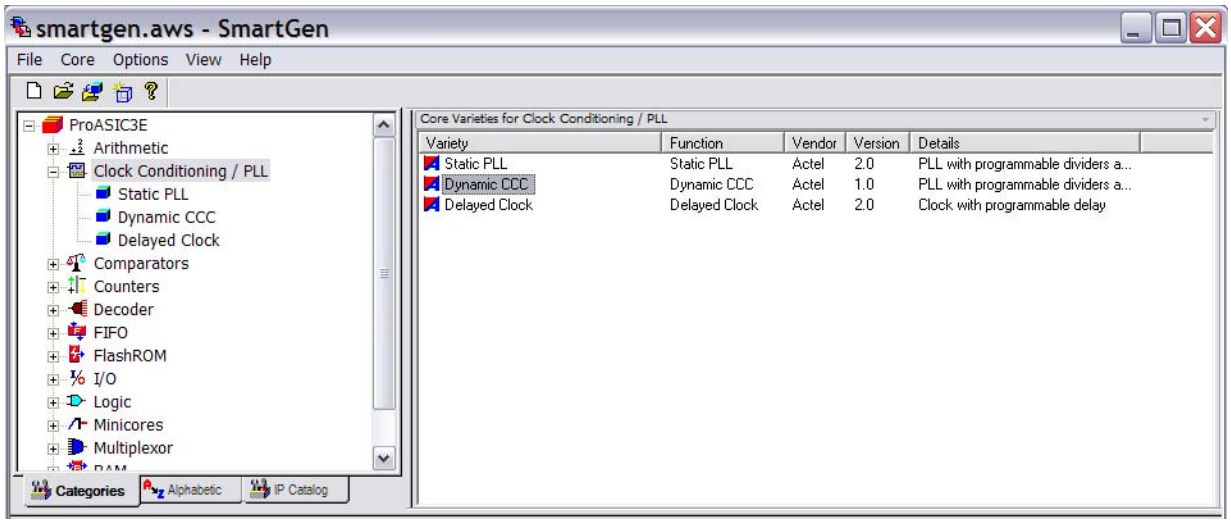


Figure 4-24 • **SmartGen GUI**

Even if dynamic configuration is selected in SmartGen, the user must still specify the static configuration data for the CCC (Figure 4-25). The specified static configuration is used whenever the MODE signal is set to LOW and the CCC is required to function in the static mode. The static configuration data can be used as the default behavior of the CCC where required.

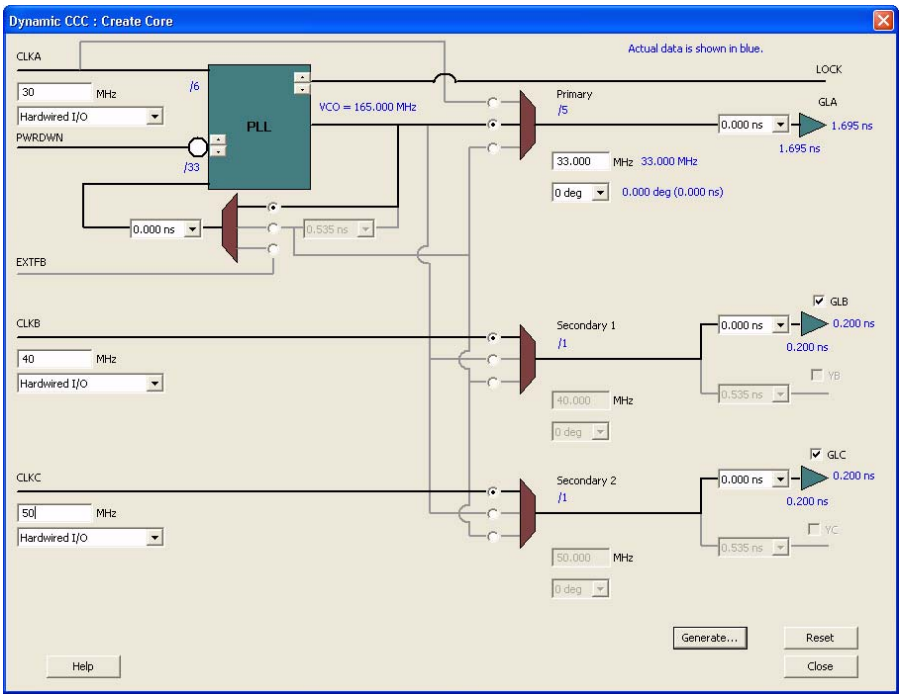


Figure 4-25 • **Dynamic CCC Configuration in SmartGen**

When SmartGen is used to define the configuration that will be shifted in via the serial interface, SmartGen prints out the values of the 81 configuration bits. For ease of use, several configuration bits are automatically inferred by SmartGen when the dynamic PLL core is generated; however, <71:73> (STATASEL, STATBSEL, STATCSEL) and <77:79> (DYNASEL, DYNBSEL, DYNCSEL) depend on the input clock source of the corresponding CCC. Users must first run Layout in Designer to determine the exact setting for these ports. After Layout is complete, generate the "CCC_Configuration" report by choosing **Tools** > **Reports** > **CCC_Configuration** in the Designer software. Refer to "PLL Configuration Bits Description" on page 4-27 for descriptions of the PLL configuration bits. For simulation purposes, bits <71:73> and <78:80> are "don't cares." Therefore, it is strongly suggested that SmartGen be used to generate the correct configuration bit settings for the dynamic PLL core.

After setting all the required parameters, users can generate one or more PLL configurations with HDL or EDIF descriptions by clicking the **Generate** button. SmartGen gives the option of saving session results and messages in a log file:

```
****************
Macro Parameters
****************

Name                            : dyn_pll_hardio
Family                          : ProASIC3E
Output Format                   : VERILOG
Type                            : Dynamic CCC
Input Freq(MHz)                 : 30.000
CLKA Source                     : Hardwired I/O
Feedback Delay Value Index      : 1
Feedback Mux Select             : 1
XDLY Mux Select                 : No
Primary Freq(MHz)               : 33.000
Primary PhaseShift              : 0
Primary Delay Value Index       : 1
Primary Mux Select              : 4
Secondary1 Freq(MHz)            : 40.000
Use GLB                         : YES
Use YB                          : NO
GLB Delay Value Index           : 1
YB Delay Value Index            : 1
Secondary1 PhaseShift           : 0
Secondary1 Mux Select           : 0
Secondary1 Input Freq(MHz)      : 40.000
CLKB Source                     : Hardwired I/O
Secondary2 Freq(MHz)            : 50.000
Use GLC                         : YES
Use YC                          : NO
GLC Delay Value Index           : 1
YC Delay Value Index            : 1
Secondary2 PhaseShift           : 0
Secondary2 Mux Select           : 0
Secondary2 Input Freq(MHz)      : 50.000
CLKC Source                     : Hardwired I/O

Configuration Bits:
FINDIV[6:0]     0000101
FBDIV[6:0]      0100000
OADIV[4:0]      00100
OBDIV[4:0]      00000
OCDIV[4:0]      00000
OAMUX[2:0]      100
OBMUX[2:0]      000
OCMUX[2:0]      000
FBSEL[1:0]      01
FBDLY[4:0]      00000
XDLYSEL         0
DLYGLA[4:0]     00000
```

```
DLYGLB[4:0]     00000
DLYGLC[4:0]     00000
DLYYB[4:0]      00000
DLYYC[4:0]      00000
VCOSEL[2:0]     100

Primary Clock Frequency 33.000
Primary Clock Phase Shift 0.000
Primary Clock Output Delay from CLKA 1.695

Secondary1 Clock Frequency 40.000
Secondary1 Clock Phase Shift 0.000
Secondary1 Clock Global Output Delay from CLKB 0.200

Secondary2 Clock Frequency 50.000
Secondary2 Clock Phase Shift 0.000
Secondary2 Clock Global Output Delay from CLKC 0.200

#####################################
# Dynamic Stream Data
#####################################
-------------------------------------
|NAME    |SDIN     |VALUE   |TYPE     |
-------------------------------------
|FINDIV  |[6:0]    |0000101 |EDIT     |
|FBDIV   |[13:7]   |0100000 |EDIT     |
|OADIV   |[18:14]  |00100   |EDIT     |
|OBDIV   |[23:19]  |00000   |EDIT     |
|OCDIV   |[28:24]  |00000   |EDIT     |
|OAMUX   |[31:29]  |100     |EDIT     |
|OBMUX   |[34:32]  |000     |EDIT     |
|OCMUX   |[37:35]  |000     |EDIT     |
|FBSEL   |[39:38]  |01      |EDIT     |
|FBDLY   |[44:40]  |00000   |EDIT     |
|XDLYSEL |[45]     |0       |EDIT     |
|DLYGLA  |[50:46]  |00000   |EDIT     |
|DLYGLB  |[55:51]  |00000   |EDIT     |
|DLYGLC  |[60:56]  |00000   |EDIT     |
|DLYYB   |[65:61]  |00000   |EDIT     |
|DLYYC   |[70:66]  |00000   |EDIT     |
|STATASEL|[71]     |X       |MASKED   |
|STATBSEL|[72]     |X       |MASKED   |
|STATCSEL|[73]     |X       |MASKED   |
|VCOSEL  |[76:74]  |100     |EDIT     |
|DYNASEL |[77]     |X       |MASKED   |
|DYNBSEL |[78]     |X       |MASKED   |
|DYNCSEL |[79]     |X       |MASKED   |
|RESETEN |[80]     |1       |READONLY |
```

Below is the resultant Verilog HDL description of a legal dynamic PLL core configuration generated by SmartGen:

```
module dyn_pll_macro(POWERDOWN, CLKA, LOCK, GLA, GLB, GLC, SDIN, SCLK, SSHIFT, SUPDATE,
  MODE, SDOUT, CLKB, CLKC);

input POWERDOWN, CLKA;
output  LOCK, GLA, GLB, GLC;
input  SDIN, SCLK, SSHIFT, SUPDATE, MODE;
output  SDOUT;
input  CLKB, CLKC;

  wire VCC, GND;

  VCC VCC_1_net(.Y(VCC));
  GND GND_1_net(.Y(GND));
```

```
  DYNCCC Core(.CLKA(CLKA), .EXTFB(GND), .POWERDOWN(POWERDOWN), .GLA(GLA), .LOCK(LOCK),
    .CLKB(CLKB), .GLB(GLB), .YB(), .CLKC(CLKC), .GLC(GLC), .YC(), .SDIN(SDIN),
    .SCLK(SCLK), .SSHIFT(SSHIFT), .SUPDATE(SUPDATE), .MODE(MODE), .SDOUT(SDOUT),
    .OADIV0(GND), .OADIV1(GND), .OADIV2(VCC), .OADIV3(GND), .OADIV4(GND), .OAMUX0(GND),
    .OAMUX1(GND), .OAMUX2(VCC), .DLYGLA0(GND), .DLYGLA1(GND), .DLYGLA2(GND),
    .DLYGLA3(GND), .DLYGLA4(GND), .OBDIV0(GND), .OBDIV1(GND), .OBDIV2(GND),
    .OBDIV3(GND), .OBDIV4(GND), .OBMUX0(GND), .OBMUX1(GND), .OBMUX2(GND), .DLYYB0(GND),
    .DLYYB1(GND), .DLYYB2(GND), .DLYYB3(GND), .DLYYB4(GND), .DLYGLB0(GND),
    .DLYGLB1(GND), .DLYGLB2(GND), .DLYGLB3(GND), .DLYGLB4(GND), .OCDIV0(GND),
    .OCDIV1(GND), .OCDIV2(GND), .OCDIV3(GND), .OCDIV4(GND), .OCMUX0(GND), .OCMUX1(GND),
    .OCMUX2(GND), .DLYYC0(GND), .DLYYC1(GND), .DLYYC2(GND), .DLYYC3(GND), .DLYYC4(GND),
    .DLYGLC0(GND), .DLYGLC1(GND), .DLYGLC2(GND), .DLYGLC3(GND), .DLYGLC4(GND),
    .FINDIV0(VCC), .FINDIV1(GND), .FINDIV2(VCC), .FINDIV3(GND), .FINDIV4(GND),
    .FINDIV5(GND), .FINDIV6(GND), .FBDIV0(GND), .FBDIV1(GND), .FBDIV2(GND),
    .FBDIV3(GND), .FBDIV4(GND), .FBDIV5(VCC), .FBDIV6(GND), .FBDLY0(GND), .FBDLY1(GND),
    .FBDLY2(GND), .FBDLY3(GND), .FBDLY4(GND), .FBSEL0(VCC), .FBSEL1(GND),
    .XDLYSEL(GND), .VCOSEL0(GND), .VCOSEL1(GND), .VCOSEL2(VCC));
  defparam Core.VCOFREQUENCY = 165.000;

endmodule
```

Delayed Clock Configuration

The CLKDLY macro can be generated with the desired delay and input clock source (Hardwired I/O, External I/O, or Core Logic), as in Figure 4-26.

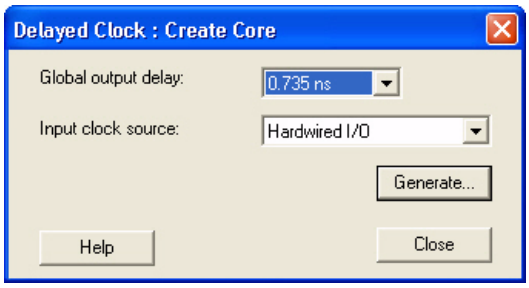


Figure 4-26 • **Delayed Clock Configuration Dialog Box**

After setting all the required parameters, users can generate one or more PLL configurations with HDL or EDIF descriptions by clicking the **Generate** button. SmartGen gives the option of saving session results and messages in a log file:

```
***************
Macro Parameters
***************

Name                            : delay_macro
Family                          : ProASIC3
Output Format                   : Verilog
Type                            : Delayed Clock
Delay Index                     : 2
CLKA Source                     : Hardwired I/O

Total Clock Delay = 0.935 ns.

The resultant CLKDLY macro Verilog netlist is as follows:

module delay_macro(GL,CLK);

output GL;
input  CLK;
```

```
  wire VCC, GND;

  VCC VCC_1_net(.Y(VCC));
  GND GND_1_net(.Y(GND));
  CLKDLY Inst1(.CLK(CLK), .GL(GL), .DLYGL0(VCC), .DLYGL1(GND), .DLYGL2(VCC),
    .DLYGL3(GND), .DLYGL4(GND));

endmodule
```

Detailed Usage Information

Clock Frequency Synthesis

Deriving clocks of various frequencies from a single reference clock is known as frequency synthesis. The PLL has an input frequency range from 1.5 to 350 MHz. This frequency is automatically divided down to a range between 1.5 MHz and 5.5 MHz by input dividers (not shown Figure 4-17 on page 4-22) between PLL macro inputs and PLL phase detector inputs. The VCO output is capable of an output range from 24 to 350 MHz. With dividers before the input to the PLL core and following the VCO outputs, the VCO output frequency can be divided to provide the final frequency range from 0.75 to 350 MHz. Using SmartGen, the dividers are automatically set to achieve the closest possible matches to the specified output frequencies.

Users should be cautious when selecting the desired PLL input and output frequencies and the I/O buffer standard used to connect to the PLL input and output clocks. Depending on the I/O standards used for the PLL input and output clocks, the I/O frequencies have different maximum limits. Refer to the family datasheets for specifications of maximum I/O frequencies for supported I/O standards. Desired PLL input or output frequencies will not be achieved if the selected frequencies are higher than the maximum I/O frequencies allowed by the selected I/O standards. Users should be careful when selecting the I/O standards used for PLL input and output clocks. Performing post-layout simulation can help detect this type of error, which will be identified with pulse width violation errors. Users are strongly encouraged to perform post-layout simulation to ensure the I/O standard used can provide the desired PLL input or output frequencies. Users can also choose to cascade PLLs together to achieve the high frequencies needed for their applications. Details of cascading PLLs are discussed in the "Cascading CCCs" section on page 4-46.

In SmartGen, the actual generated frequency (under typical operating conditions) will be displayed beside the requested output frequency value. This provides the ability to determine the exact frequency that can be generated by SmartGen, in real time. The log file generated by SmartGen is a useful tool in determining how closely the requested clock frequencies match the user specifications. For example, assume a user specifies 101 MHz as one of the secondary output frequencies. If the best output frequency that could be achieved were 100 MHz, the log file generated by SmartGen would indicate the actual generated frequency.

Simulation Verification

The integration of the generated PLL and CLKDLY modules is similar to any VHDL component or Verilog module instantiation in a larger design; i.e., there is no special requirement that users need to take into account to successfully synthesize their designs.

For simulation purposes, users need to refer to the VITAL or Verilog library that includes the functional description and associated timing parameters. Refer to the Software Tools section of the Actel website to obtain the family simulation libraries. If Actel Designer is installed, these libraries are stored in the following locations:

<Designer_Installation_Directory>\lib\vtl\95\proasic3.vhd

<Designer_Installation_Directory>\lib\vtl\95\proasic3e.vhd

<Designer_Installation_Directory>\lib\vlog\ proasic3.v

<Designer_Installation_Directory>\lib\vlog\ proasic3e.v

For Libero IDE users, there is no need to compile the simulation libraries, as they are conveniently pre-compiled in the Model*Sim*® Actel simulation tool.

The following is an example of a PLL configuration utilizing the clock frequency synthesis and clock delay adjustment features. The steps include generating the PLL core with SmartGen, performing simulation for verification with Model*Sim*, and performing static timing analysis with SmartTime in Designer.

Parameters of the example PLL configuration:

Input Frequency – 20 MHz

Primary Output Requirement – 20 MHz with clock advancement of 3.02 ns

Secondary 1 Output Requirement – 40 MHz with clock delay of 2.515 ns

Figure 4-27 shows the SmartGen settings. Notice that the overall delays are calculated automatically, allowing the user to adjust the delay elements appropriately to obtain the desired delays.

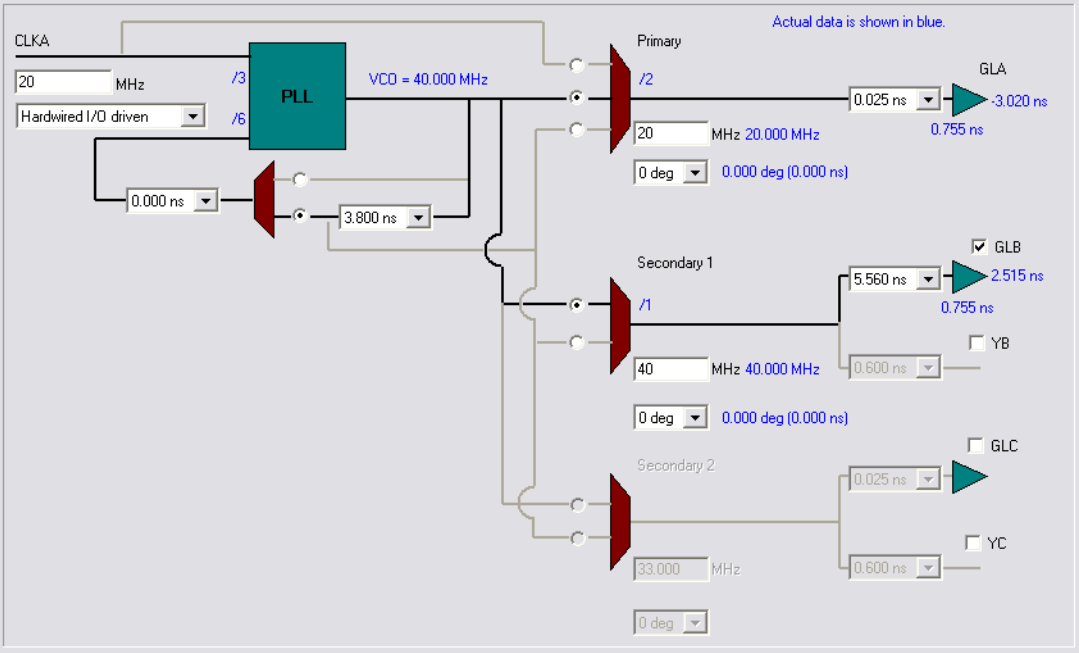


Figure 4-27 • **SmartGen Settings**

After confirming the correct settings, generate a structural netlist of the PLL and verify PLL core settings by checking the log file:

```
Name                              : test_pll_delays
Family                            : ProASIC3E
Output Format                     : VHDL
Type                              : Static PLL
Input Freq(MHz)                   : 20.000
CLKA Source                       : Hardwired I/O
Feedback Delay Value Index        : 21
Feedback Mux Select               : 2
XDLY Mux Select                   : No
Primary Freq(MHz)                 : 20.000
Primary PhaseShift                : 0
Primary Delay Value Index         : 1
Primary Mux Select                : 4
Secondary1 Freq(MHz)              : 40.000
Use GLB                           : YES
Use YB                            : NO
```

```
…
…
…
Primary Clock frequency 20.000
Primary Clock Phase Shift 0.000
Primary Clock Output Delay from CLKA -3.020

Secondary1 Clock frequency 40.000
Secondary1 Clock Phase Shift 0.000
Secondary1 Clock Global Output Delay from CLKA 2.515
```

Next, perform simulation in Model*Sim* to verify the correct delays. Figure 4-28 shows the simulation results. The delay values match those reported in the SmartGen PLL Wizard.

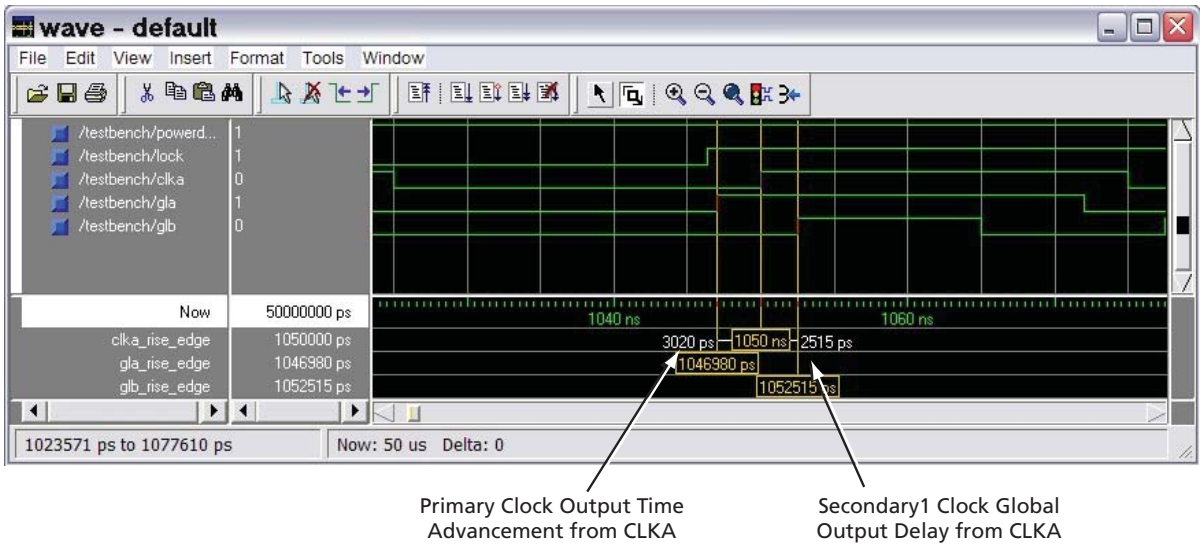


Figure 4-28 • **Model*Sim* Simulation Results**

The timing can also be analyzed using SmartTime in Designer. The user should import the synthesized netlist to Designer, perform Compile and Layout, and then invoke SmartTime. Go to **Tools** > **Options** and change the maximum delay operating conditions to **Typical Case**. Then expand the Clock-to-Out paths of GLA and GLB and the individual components of the path delays are shown. The path of GLA is shown in Figure 4-29 on page 4-44 displaying the same delay value.

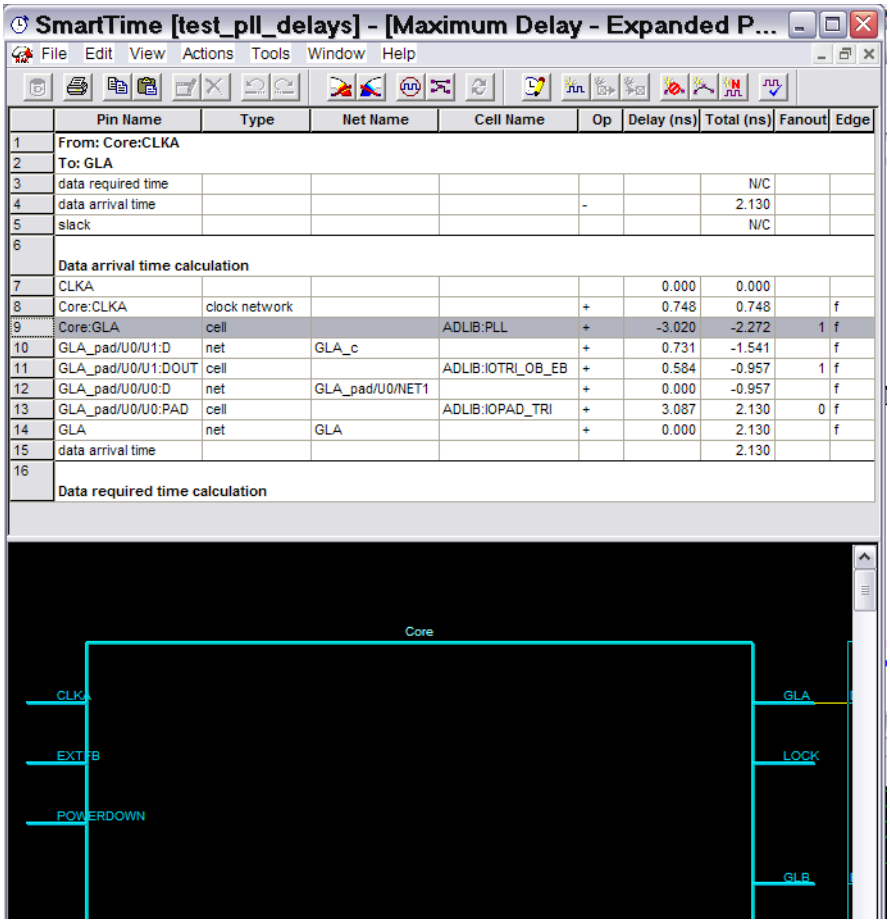


Figure 4-29 • **Static Timing Analysis Using SmartTime**

Place-and-Route Stage Considerations

Several considerations must be noted to properly place the CCC macros for layout.

For CCCs with clock inputs configured with the Hardwired I/O–Driven option:

- PLL macros must have the clock input pad coming from one of the GmA* locations.
- CLKDLY macros must have the clock input pad coming from one of the Global I/Os.

If a PLL with a Hardwired I/O input is used at a CCC location and a Hardwired I/O–Driven CLKDLY macro is used at the same CCC location, the clock input of the CLKDLY macro must be chosen from one of the GmB* or GmC* pin locations. If the PLL is not used or is an External I/O–Driven or Core Logic–Driven PLL, the clock input of the CLKDLY macro can be sourced from the GmA*, GmB*, or GmC* pin locations.

For CCCs with clock inputs configured with the External I/O–Driven option, the clock input pad can be assigned to any regular I/O location (IO******** pins). Note that since global I/O pins can also be used as regular I/Os, regardless of CCC function (CLKDLY or PLL), clock inputs can also be placed in any of these I/O locations.

By default, the Designer layout engine will place global nets in the design at one of the six chip globals. When the number of globals in the design is greater than six, the Designer layout engine will automatically assign additional globals to the quadrant global networks of the low-power flash devices. If the user wishes to decide which global signals should be assigned to chip globals (six available) and which to the quadrant globals (three per quadrant for a total of 12 available), the assignment can be achieved with PinEditor, ChipPlanner, or by importing a placement

constraint file. Layout will fail if the global assignments are not allocated properly. See the "Physical Constraints for Quadrant Clocks" section for information on assigning global signals to the quadrant clock networks.

Promoted global signals will be instantiated with CLKINT macros to drive these signals onto the global network. This is automatically done by Designer when the Auto-Promotion option is selected. If the user wishes to assign the signals to the quadrant globals instead of the default chip globals, this can done by using ChipPlanner, by declaring a physical design constraint (PDC), or by importing a PDC file.

Physical Constraints for Quadrant Clocks

If it is necessary to promote global clocks (CLKBUF, CLKINT, PLL, CLKDLY) to quadrant clocks, the user can define PDCs to execute the promotion. PDCs can be created using PDC commands (pre-compile) or the MultiView Navigator (MVN) interface (post-compile). The advantage of using the PDC flow over the MVN flow is that the Compile stage is able to automatically promote any regular net to a global net before assigning it to a quadrant. There are three options to place a quadrant clock using PDC commands:

- Place a clock core (not hardwired to an I/O) into a quadrant clock location.
- Place a clock core (hardwired to an I/O) into an I/O location (set_io) or an I/O module location (set_location) that drives a quadrant clock location.
- Assign a net driven by a regular net or a clock net to a quadrant clock using the following command:

  ```
  assign_local_clock -net <net name> -type quadrant <quadrant clock region>
  ```

  where

  `<net name>` is the name of the net assigned to the local user clock region.

  `<quadrant clock region>` defines which quadrant the net should be assigned to. Quadrant clock regions are defined as UL (upper left), UR (upper right), LL (lower left), and LR (lower right).

Note: If the net is a regular net, the software inserts a CLKINT buffer on the net.

For example:

```
assign_local_clock -net localReset -type quadrant UR
```

Keep in mind the following when placing quadrant clocks using MultiView Navigator:

Hardwired I/O–Driven CCCs

- Find the associated clock input port under the Ports tab, and place the input port at one of the Gmn* locations using PinEditor or I/O Attribute Editor, as shown in Figure 4-30.

***Figure 4-30* • Port Assignment for a CCC with Hardwired I/O Clock Input**

- Use quadrant global region assignments by finding the clock net associated with the CCC macro under the Nets tab and creating a quadrant global region for the net, as shown in Figure 4-31.

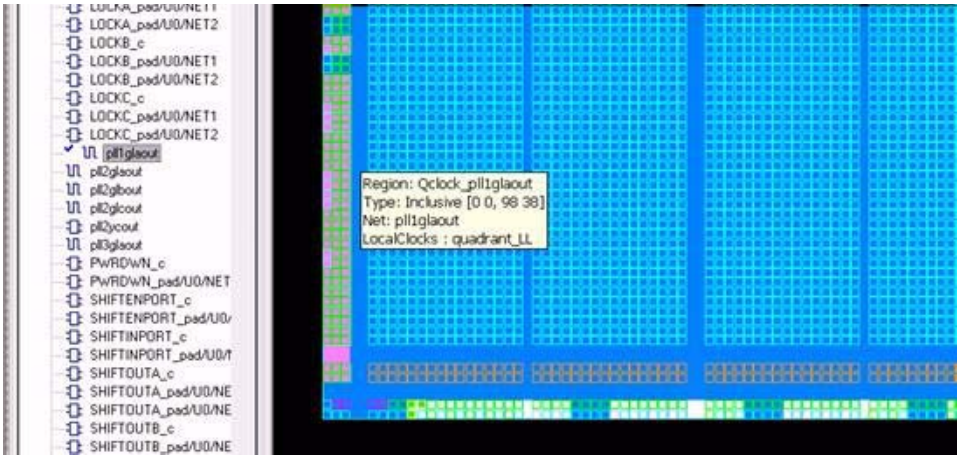


Figure 4-31 • **Quadrant Clock Assignment for a Global Net**

External I/O–Driven CCCs

The above-mentioned recommendation for proper layout techniques will ensure the correct assignment. It is possible that, especially with External I/O–Driven CCC macros, placement of the CCC macro in a desired location may not be achieved. For example, assigning an input port of an External I/O–Driven CCC near a particular CCC location does not guarantee global assignments to the desired location. This is because the clock inputs of External I/O–Driven CCCs can be assigned to any I/O location; therefore, it is possible that the CCC connected to the clock input will be routed to a location other than the one closest to the I/O location, depending on resource availability and placement constraints.

Clock Placer

The clock placer is a placement engine for low-power flash devices that places global signals on the chip global and quadrant global networks. Based on the clock assignment constraints for the chip global and quadrant global clocks, it will try to satisfy all constraints, as well as creating quadrant clock regions when necessary. If the clock placer fails to create the quadrant clock regions for the global signals, it will report an error and stop Layout.

The user must ensure that the constraints set to promote clock signals to quadrant global networks are valid.

Cascading CCCs

The CCCs in low-power flash devices can be cascaded. Cascading CCCs can help achieve more accurate PLL output frequency results than those achievable with a single CCC. In addition, this technique is useful when the user application requires the output clock of the PLL to be a multiple of the reference clock by an integer greater than the maximum feedback divider value of the PLL (divide by 128) to achieve the desired frequency.

For example, the user application may require a 280 MHz output clock using a 2 MHz input reference clock, as shown in Figure 4-32 on page 4-47.

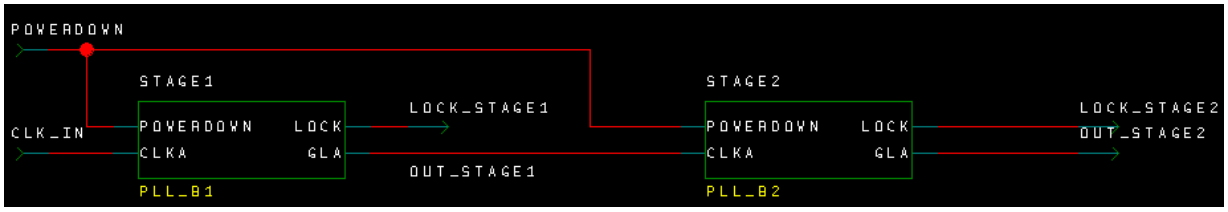


Figure 4-32 • **Cascade PLL Configuration**

Using internal feedback, we know from EQ 4-1 on page 4-24 that the maximum achievable output frequency from the primary output is

$$f_{GLA} = f_{CLKA} \times m / (n \times u) = 2\ MHz \times 128 / (1 \times 1) = 256\ MHz$$

EQ 4-5

Figure 4-33 shows the settings of the initial PLL. When configuring the initial PLL, specify the input to be either Hardwired I/O–Driven or External I/O–Driven. This generates a netlist with the initial PLL routed from an I/O. Do not specify the input to be Core Logic–Driven, as this prohibits the connection from the I/O pin to the input of the PLL.

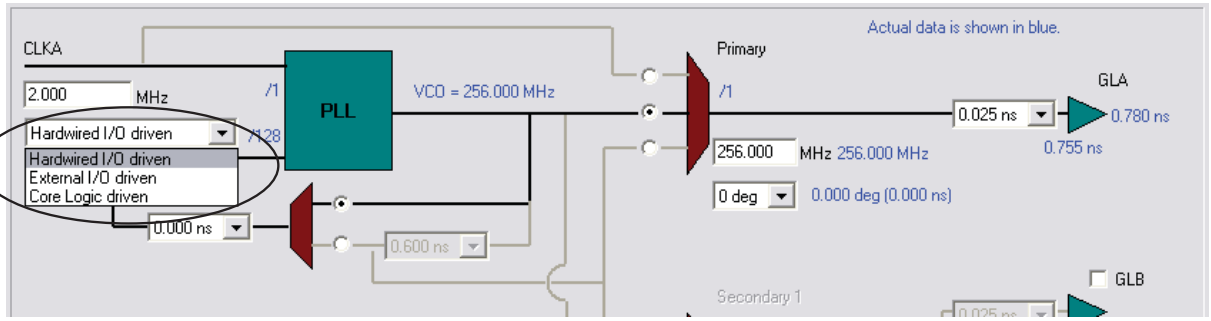


Figure 4-33 • **First-Stage PLL Showing Input of 2 MHz and Output of 256 MHz**

A second PLL can be connected serially to achieve the required frequency. EQ 4-1 on page 4-24 to EQ 4-3 on page 4-24 are extended as follows:

$$f_{GLA2} = f_{GLA} \times m_2 / (n_2 \times u_2) = f_{CLKA1} \times m_1 \times m_2 / (n_1 \times u_1 \times n_2 \times u_2) \text{ – Primary PLL Output Clock}$$

EQ 4-6

$$f_{GLB2} = f_{YB2} = f_{CLKA1} \times m_1 \times m_2 / (n_1 \times n_2 \times v_1 \times v_2) \text{ – Secondary 1 PLL Output Clock(s)}$$

EQ 4-7

$$f_{GLC2} = f_{YC2} = f_{CLKA1} \times m_1 \times m_2 / (n_1 \times n_2 \times w_1 \times w_2) \text{ – Secondary 2 PLL Output Clock(s)}$$

EQ 4-8

In the example, the final output frequency (f_{output}) from the primary output of the second PLL will be as follows (EQ 4-9):

$$f_{output} = f_{GLA2} = f_{GLA} \times m_2 / (n_2 \times u_2) = 256\ MHz \times 70 / (64 \times 1) = 280\ MHz$$

EQ 4-9

Figure 4-34 on page 4-48 shows the settings of the second PLL. When configuring the second PLL (or any subsequent-stage PLLs), specify the input to be Core Logic–Driven. This generates a netlist with the second PLL routed internally from the core. Do not specify the input to be Hardwired I/O–Driven or External I/O–Driven, as these options prohibit the connection from the output of the first PLL to the input of the second PLL.

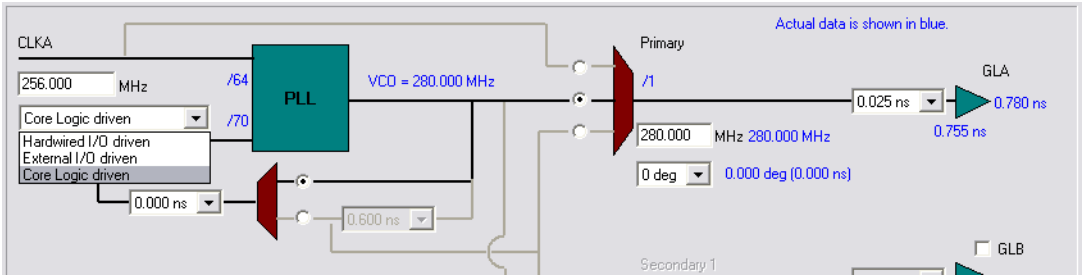


Figure 4-34 • **Second-Stage PLL Showing Input of 256 MHz from First Stage and Final Output of 280 MHz**

Figure 4-35 shows the simulation results, where the first PLL's output period is 3.9 ns (~256 MHz), and the stage 2 (final) output period is 3.56 ns (~280 MHz).

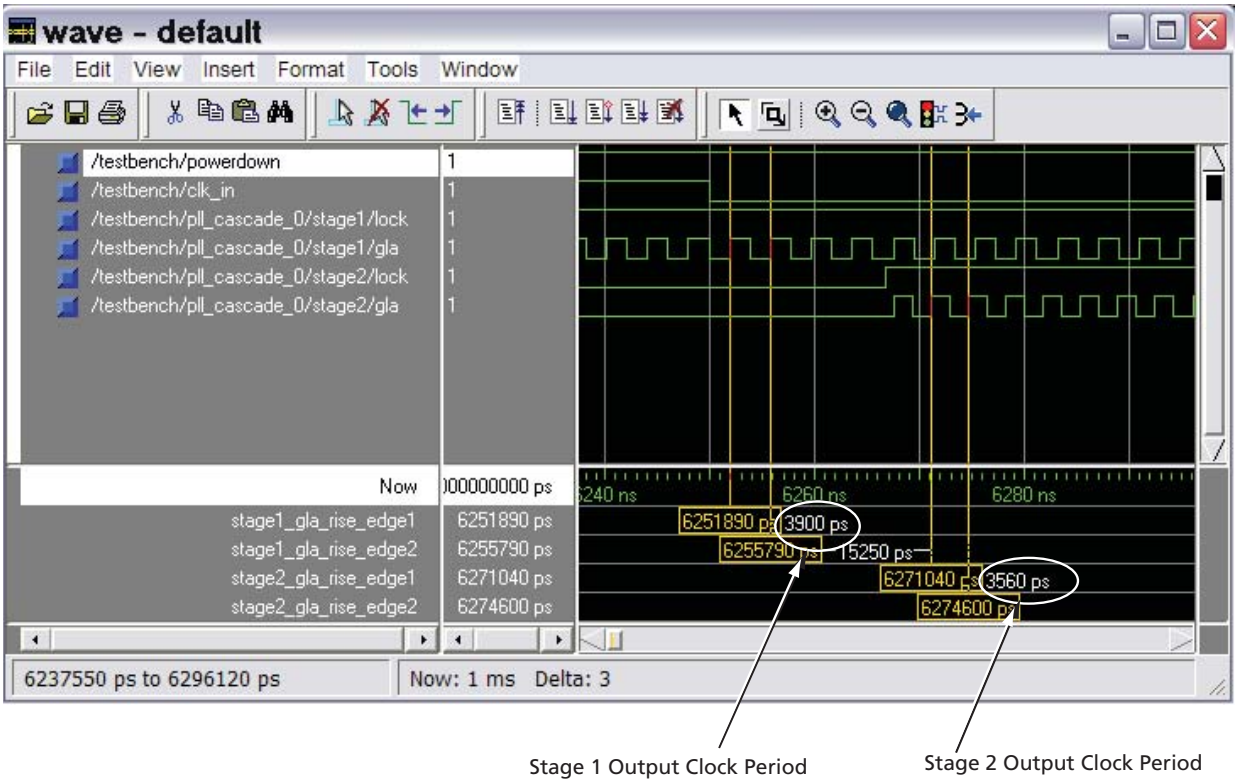


Figure 4-35 • **Model*Sim* Simulation Results**

Recommended Board-Level Considerations

The power to the PLL core is supplied by $V_{CCPLA/B/C/D/E/F}$ (V_{CCPLx}), and the associated ground connections are supplied by $V_{COMPLA/B/C/D/E/F}$ (V_{COMPLx}). When the PLLs are not used, the Actel Designer place-and-route tool automatically disables the unused PLLs to lower power consumption. The user should tie unused V_{CCPLx} and V_{COMPLx} pins to ground. Optionally, the PLL can be turned on/off during normal device operation via the POWERDOWN port (see Table 4-3 on page 4-8).

PLL Power Supply Decoupling Scheme

The PLL core is designed to tolerate noise levels on the PLL power supply as specified in the datasheets. When operated within the noise limits, the PLL will meet the output peak-to-peak jitter specifications specified in the datasheets. User applications should always ensure the PLL power supply is powered from a noise-free or low-noise power source.

However, in situations where the PLL power supply noise level is higher than the tolerable limits, various decoupling schemes can be designed to suppress noise to the PLL power supply. An example is provided in Figure 4-36. The V_{CCPLx} and V_{COMPLx} pins correspond to the PLL analog power supply and ground.

Actel strongly recommends that two ceramic capacitors (10 nF in parallel with 100 nF) be placed close to the power pins (less than 1 inch away). A third generic 10 µF electrolytic capacitor is recommended for low-frequency noise and should be placed farther away due to its large physical size. Actel recommends that a 6.8 µH inductor be placed between the supply source and the capacitors to filter out any low-/medium- and high-frequency noise. In addition, the PCB layers should be controlled so the V_{CCPLx} and V_{COMPLx} planes have the minimum separation possible, thus generating a good-quality RF capacitor.

For more recommendations, refer to the *Board-Level Considerations* application note.

Recommended 100 nF capacitor:

- Producer BC Components, type X7R, 100 nF, 16 V
- BC Components part number: 0603B104K160BT
- Digi-Key part number: BC1254CT-ND
- Digi-Key part number: BC1254TR-ND

Recommended 10 nF capacitor:

- Surface-mount ceramic capacitor
- Producer BC Components, type X7R, 10 nF, 50 V
- BC Components part number: 0603B103K500BT
- Digi-Key part number: BC1252CT-ND
- Digi-Key part number: BC1252TR-ND

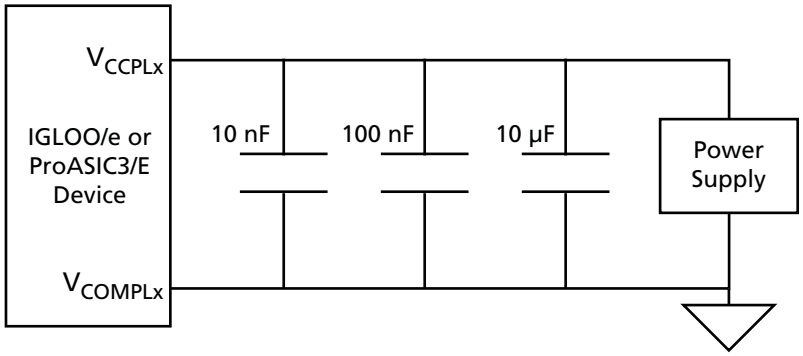


***Figure 4-36* • Decoupling Scheme for One PLL (should be replicated for each PLL used)**

Conclusion

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

Related Documents

Application Notes

Board-Level Considerations

http://www.actel.com/documents/BoardLevelCons_AN.pdf

Handbook Documents

UJTAG Applications in Actel's Low-Power Flash Devices

http://www.actel.com/documents/LPD_UJTAG_HBs.pdf

Global Resources in Actel Low-Power Flash Devices

http://www.actel.com/documents/LPD_Global_HBs.pdf

User I/O Naming Conventions in I/O Structures in IGLOO and ProASIC3 Devices

http://www.actel.com/documents/IGLOO_PA3_IO_HBs.pdf

User's Guides

IGLOO, Fusion, and ProASIC3 Macro Library Guide

http://www.actel.com/documents/pa3_libguide_ug.pdf

Part Number and Revision Date

This document contains content extracted from the Device Architecture section of the datasheet, combined with content previously published as an application note describing features and functions of the device. To improve usability for customers, the device architecture information has now been combined with usage information, to reduce duplication and possible inconsistencies in published information. No technical changes were made to the datasheet content unless explicitly listed. Changes to the application note content were made only to be consistent with existing datasheet information.

Part Number 51700094-006-3

Revised October 2008

List of Changes

The following table lists critical changes that were made in the current version of the chapter.

Previous Version	Changes in the Current Version (v1.3)	Page
v1.3 (June 2008)	This document was updated to include Fusion and RT ProASIC3 device information. Please review the document very carefully.	N/A
	The "CCC Support in Actel's Low-Power Flash Devices" section was updated.	4-3
	In the "Global Buffer with Programmable Delay" section, the following sentence was changed from: "In this case, the I/O must be placed in one of the dedicated global I/O locations." To "In this case, the software will automatically place the dedicated global I/O in the appropriate locations."	4-4
	Figure 4-4 · CCC Options: Global Buffers with PLL was updated to include OADIVRST and OADIVHALF.	4-7
	In Figure 4-5 · CCC with PLL Block "fixed delay" was changed to "programmable delay".	4-7
	Table 4-3 · Input and Output Signals of the PLL Block was updated to include OADIVRST and OADIVHALF descriptions.	4-8
	Table 4-7 · Configuration Bit Descriptions for the CCC Blocks was updated to include configuration bits 88 to 81. Note 2 is new. In addition, the description for bit <76:74> was updated.	4-27
	Table 4-15 · Fusion Dynamic CCC Clock Source Selection and Table 4-16 · Fusion Dynamic CCC NGMUX Configuration are new.	4-31
	Table 4-17 · Fusion Dynamic CCC Division by Half Configuration and Table 4-18 · Configuration Bit <76:75> / VCOSEL<2:1> Selection for All Families are new.	4-32
v1.1 (March 2008)	The following changes were made to the family descriptions in Table 4-1 · Overview of the CCCs Offered in Fusion, IGLOO, and ProASIC3: • ProASIC3L was updated to include 1.5 V. • The number of PLLs for ProASIC3E was changed from five to six.	4-1
v1.0 (January 2008)	Table 4-1 · Low-Power Flash Families and the associated text were updated to include the IGLOO PLUS family. The "IGLOO Terminology" section and "ProASIC3 Terminology" section are new.	4-3
	The "Global Input Selections" section was updated to include 15 k gate devices as supported I/O types for globals, for CCC only.	4-10
	Table 4-5 · Number of CCCs by Device Size and Package was revised to include ProASIC3L, IGLOO PLUS, A3P015, AGL015, AGLP030, AGLP060, and AGLP125.	4-16
	The "IGLOO and ProASIC3 CCC Locations" section was revised to include 15 k gate devices in the exception statements, as they do not contain PLLs.	4-18
51900133-0/5.06	Information about unlocking the PLL was removed from the "Dynamic PLL Configuration" section.	4-24
	In the "Dynamic PLL Configuration" section, information was added about running Layout and determining the exact setting of the ports.	4-37
	In Table 4-7 · Configuration Bit Descriptions for the CCC Blocks, the following bits were updated to delete "transport to the user" and reference the footnote at the bottom of the table: 79 to 71.	4-27

Embedded Memories

Actel
POWER MATTERS

5 – FlashROM in Actel's Low-Power Flash Devices

Introduction

The Fusion, IGLOO,® and ProASIC®3 families of low-power flash-based devices have a dedicated nonvolatile FlashROM memory of 1,024 bits, which provides a unique feature in the FPGA market. The FlashROM can be read, modified, and written using the JTAG (or UJTAG) interface. It can be read but not modified from the FPGA core. Only low-power flash devices contain on-chip user nonvolatile memory (NVM).

Architecture of User Nonvolatile FlashROM

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

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

		Byte Number in Bank / 4 LSB of ADDR (READ)															
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bank Number 3 MSB of ADDR (READ)	7																
	6																
	5																
	4																
	3																
	2																
	1																
	0																

Figure 5-1 • **FlashROM Architecture**

FlashROM Support in Low-Power Devices

The low-power flash families listed in Table 5-1 support the FlashROM feature and the functions described in this document.

Table 5-1 • **Low-Power Flash Families**

Product Line	Family*	Description
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
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 PLUS	IGLOO FPGAs with enhanced I/O capabilities
ProASIC3	ProASIC3	Low-power, high-performance 1.5 V FPGAs
	ProASIC3E	Higher density ProASIC3 FPGAs with six PLLs and additional I/O standards
	ProASIC3L	ProASIC3 FPGAs supporting 1.2 V to 1.5 V with Flash*Freeze technology
	Automotive ProASIC3	ProASIC3 FPGAs qualified for automotive applications
	Military ProASIC3/EL	Military temperature A3PE600L, A3P1000, and A3PE3000L
	RT ProASIC3	Radiation-tolerant RT3PE600L and RT3PE3000L

*Note: *The family names link to the appropriate datasheet, including product brief, DC and switching characteristics, and packaging information.*

IGLOO Terminology

In documentation, the terms IGLOO families and IGLOO devices refer to all of the IGLOO products as listed in Table 5-1. Where the information applies to only one family or limited devices, these exclusions will be explicitly stated.

ProASIC3 Terminology

In documentation, the terms ProASIC3 families and ProASIC3 devices refer to all of the ProASIC3 families as listed in Table 5-1. Where the information applies to only one family or limited devices, these exclusions will be explicitly stated.

To further understand the differences between the IGLOO and ProASIC3 families, refer to the *Industry's Lowest Power FPGAs Portfolio*.

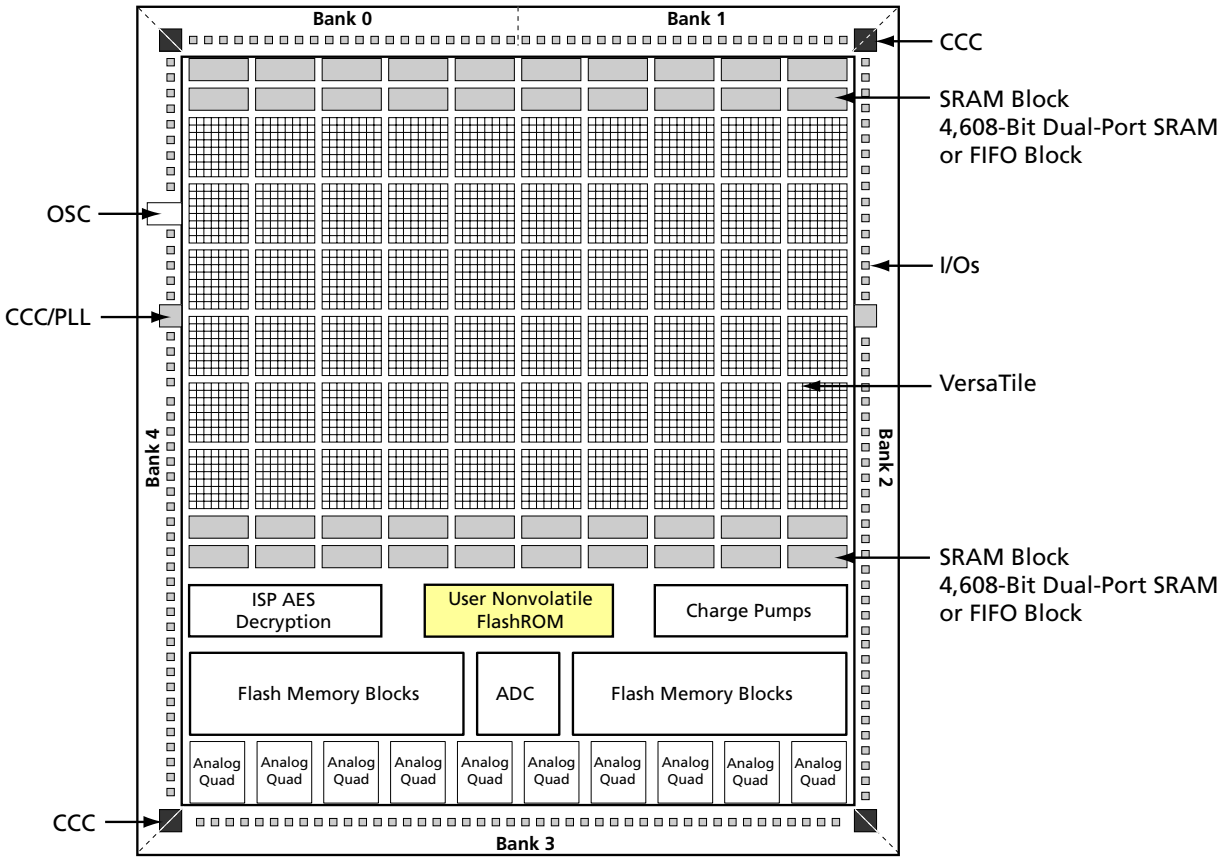


Figure 5-2 • **Fusion Device Architecture Overview (AFS600)**

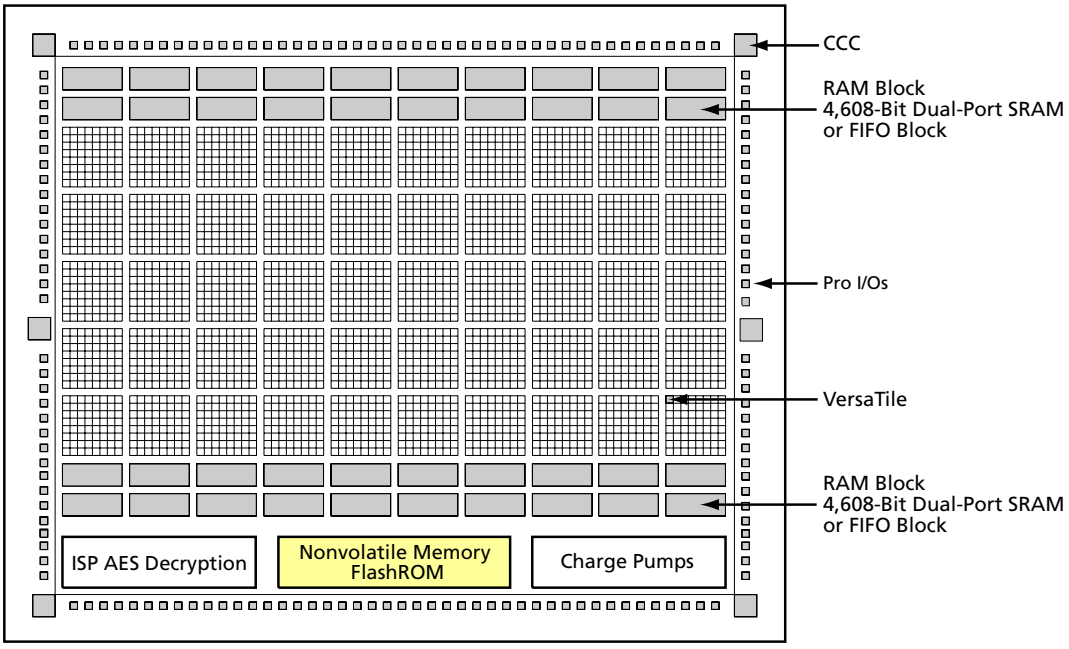


Figure 5-3 • **ProASIC3 and IGLOO Device Architecture**

FlashROM Applications

The SmartGen core generator is used to configure FlashROM content. You can configure each page independently. SmartGen enables you to create and modify regions within a page; these regions can be 1 to 16 bytes long (Figure 5-4).

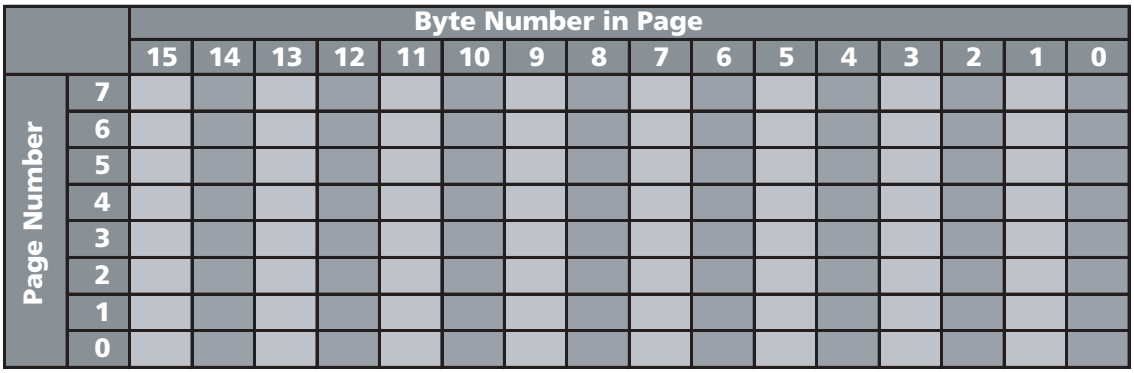


Figure 5-4 • **FlashROM Configuration**

The FlashROM content can be changed independently of the FPGA core content. It can be easily accessed and programmed via JTAG, depending on the security settings of the device. The SmartGen core generator enables each region to be independently updated (described in the "Programming and Accessing FlashROM" section on page 5-6). This enables you to change the FlashROM content on a per-part basis while keeping some regions "constant" for all parts. These features allow the FlashROM to be used in diverse system applications. Consider the following possible uses of FlashROM:

- Internet protocol (IP) addressing (wireless or fixed)
- System calibration settings
- Restoring configuration after unpredictable system power-down
- Device serialization and/or inventory control
- Subscription-based business models (e.g., set-top boxes)
- Secure key storage
- Asset management tracking
- Date stamping
- Version management

FlashROM Security

Low-power flash devices have an on-chip Advanced Encryption Standard (AES) decryption core, combined with an enhanced version of the Actel flash-based lock technology (FlashLock®). Together, they provide unmatched levels of security in a programmable logic device. This security applies to both the FPGA core and FlashROM content. These devices use the 128-bit AES (Rijndael) algorithm to encrypt programming files for secure transmission to the on-chip AES decryption core. The same algorithm is then used to decrypt the programming file. This key size provides approximately 3.4×10^{38} possible 128-bit keys. A computing system that could find a DES key in a second would take approximately 149 trillion years to crack a 128-bit AES key. The 128-bit FlashLock feature in low-power flash devices works via a FlashLock security Pass Key mechanism, where the user locks or unlocks the device with a user-defined key. Refer to *Security in Low-Power Flash Devices*.

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.

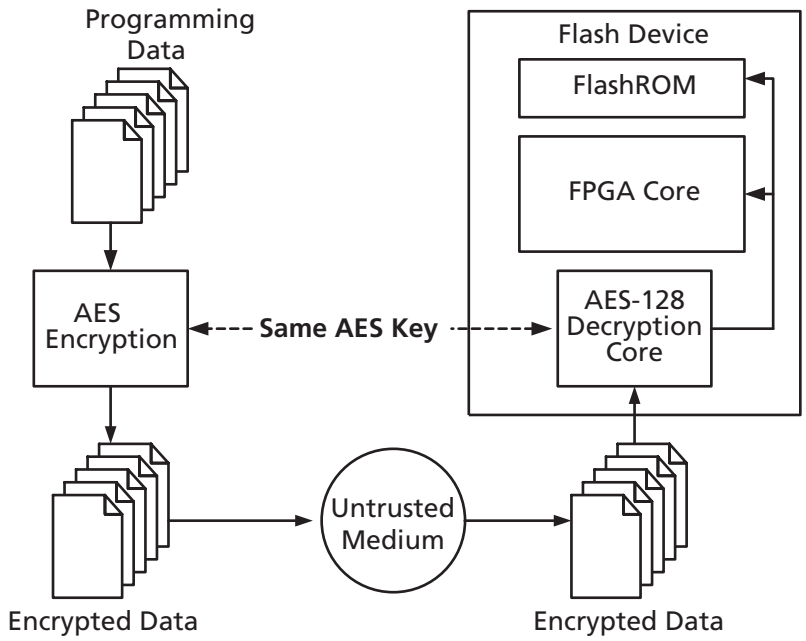


Figure 5-5 • **Programming FlashROM Using AES**

Programming and Accessing FlashROM

The FlashROM content can only be programmed via JTAG, but it can be read back selectively through the JTAG programming interface, the UJTAG interface, or via direct FPGA core addressing. The pages of the FlashROM can be made secure to prevent read-back via JTAG. In that case, read-back on these secured pages is only possible by the FPGA core fabric or via UJTAG.

A 7-bit address from the FPGA core defines which of the eight pages (three MSBs) is being read, and which of the 16 bytes within the selected page (four LSBs) are being read. The FlashROM content can be read on a random basis; the access time is 10 ns for a device supporting commercial specifications. The FPGA core will be powered down during writing of the FlashROM content. FPGA power-down during FlashROM programming is managed on-chip, and FPGA core functionality is not available during programming of the FlashROM. Table 5-2 summarizes various FlashROM access scenarios.

Table 5-2 • **FlashROM Read/Write Capabilities by Access Mode**

Access Mode	FlashROM Read	FlashROM Write
JTAG	Yes	Yes
UJTAG	Yes	No
FPGA core	Yes	No

Figure 5-6 shows the accessing of the FlashROM using the UJTAG macro. This is similar to FPGA core access, where the 7-bit address defines which of the eight pages (three MSBs) is being read and which of the 16 bytes within the selected page (four LSBs) are being read. Refer to *UJTAG Applications in Actel's Low-Power Flash Devices* for details on using the UJTAG macro to read the FlashROM.

Figure 5-7 on page 5-7 and Figure 5-8 on page 5-7 show the FlashROM access from the JTAG port. The FlashROM content can be read on a random basis. The three-bit address defines which page is being read or updated.

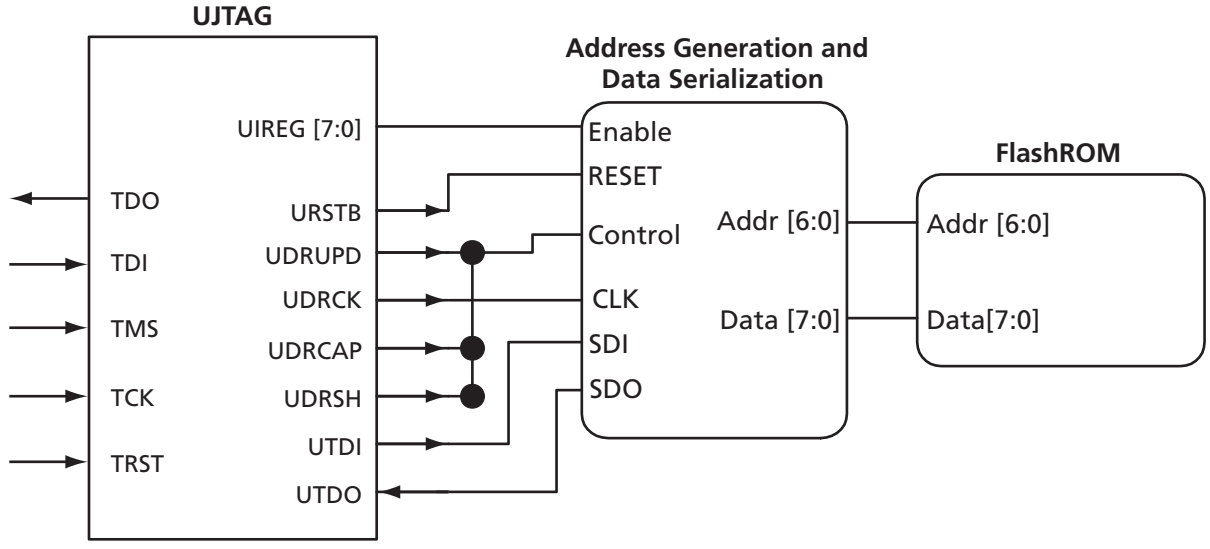


Figure 5-6 • **Block Diagram of Using UJTAG to Read FlashROM Contents**

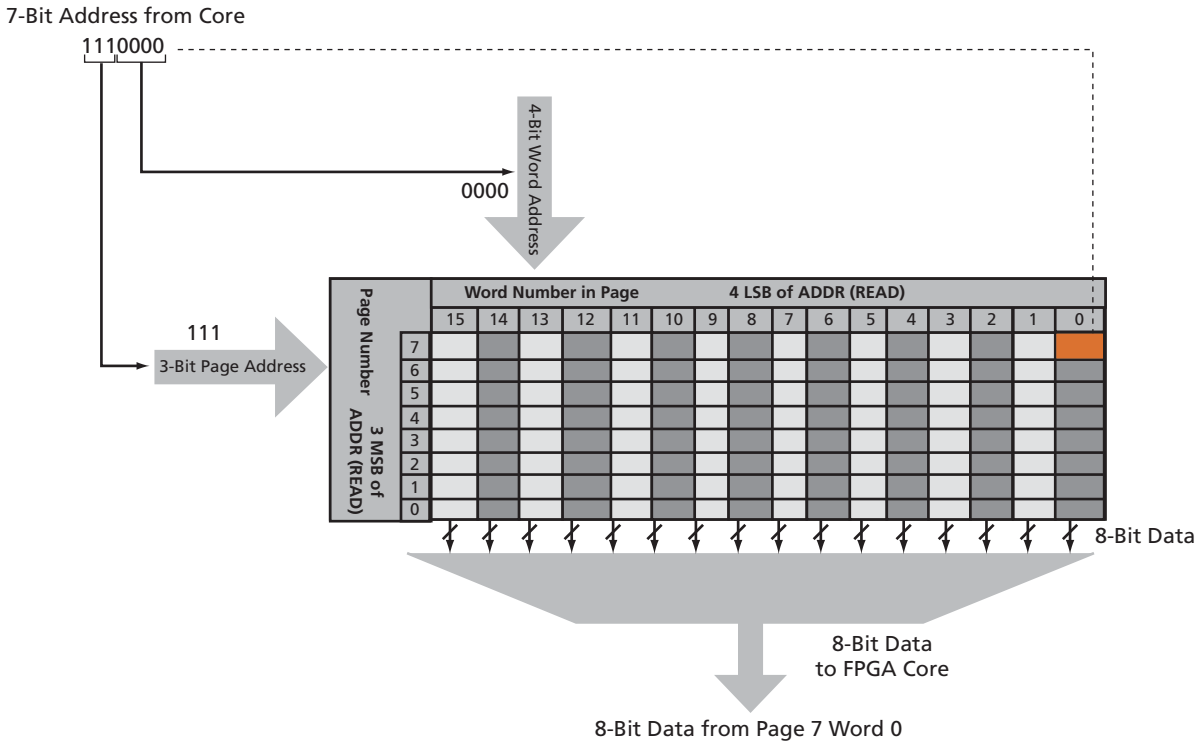


Figure 5-7 • **Accessing FlashROM Using FPGA Core**

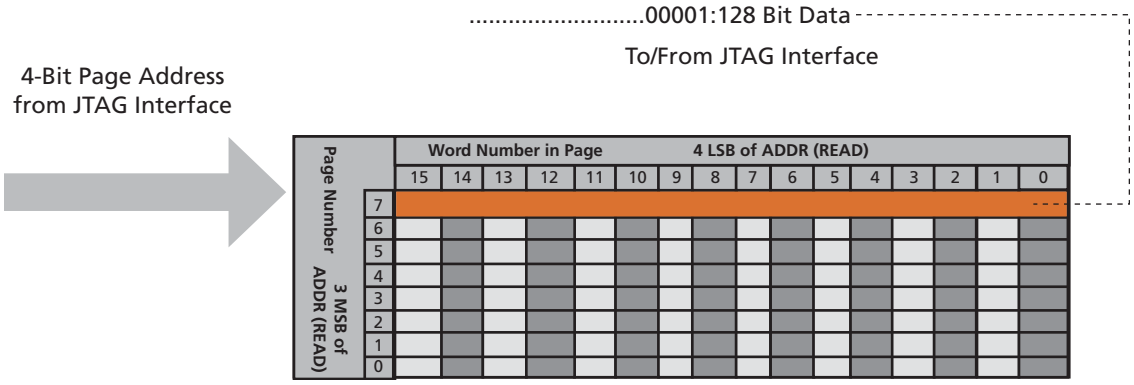


Figure 5-8 • **Accessing FlashROM Using JTAG Port**

FlashROM Design Flow

The Actel Libero® Integrated Design Environment (IDE) software has extensive FlashROM support, including FlashROM generation, instantiation, simulation, and programming. Figure 5-9 shows the user flow diagram. In the design flow, there are three main steps:

1. FlashROM generation and instantiation in the design
2. Simulation of FlashROM design
3. Programming file generation for FlashROM design

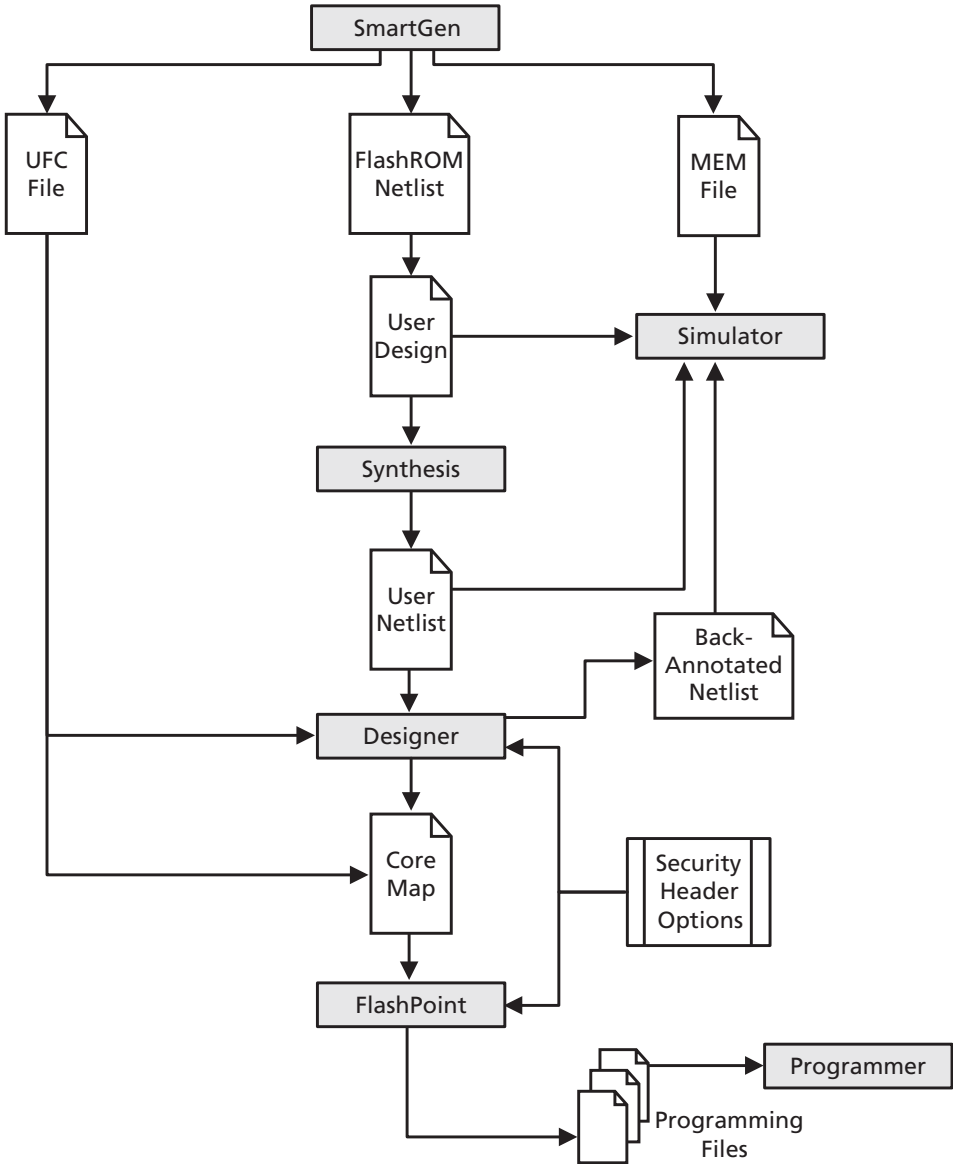


Figure 5-9 • **FlashROM Design Flow**

FlashROM Generation and Instantiation in the Design

The SmartGen core generator, available in Libero IDE and Designer, is the only tool that can be used to generate the FlashROM content. SmartGen has several user-friendly features to help generate the FlashROM contents. Instead of selecting each byte and assigning values, you can create a region within a page, modify the region, and assign properties to that region. The FlashROM user interface, shown in Figure 5-10 on page 5-9, includes the configuration grid, existing regions list, and properties field. The properties field specifies the region-specific information and defines the data used for that region. You can assign values to the following properties:

1. Static Fixed Data—Enables you to fix the data so it cannot be changed during programming time. This option is useful when you have fixed data stored in this region, which is required for the operation of the design in the FPGA. Key storage is one example.
2. Static Modifiable Data—Select this option when the data in a particular region is expected to be static data (such as a version number, which remains the same for a long duration but could conceivably change in the future). This option enables you to avoid changing the value every time you enter new data.
3. Read from File—This provides the full flexibility of FlashROM usage to the customer. If you have a customized algorithm for generating the FlashROM data, you can specify this setting. You can then generate a text file with data for as many devices as you wish to program, and load that into the FlashPoint programming file generation software to get programming files that include all the data. SmartGen will optionally pass the location of the file where the data is stored if the file is specified in SmartGen. Each text file has only one type of data format (binary, decimal, hex, or ASCII text). The length of each data file must be shorter than or equal to the selected region length. If the data is shorter than the selected region length, the most significant bits will be padded with 0s. For multiple text files for multiple regions, the first lines are for the first device. In SmartGen, **Load Sim. Value From File** allows you to load the first device data in the MEM file for simulation.
4. Auto Increment/Decrement—This scenario is useful when you specify the contents of FlashROM for a large number of devices in a series. You can specify the step value for the serial number and a maximum value for inventory control. During programming file generation, the actual number of devices to be programmed is specified and a start value is fed to the software.

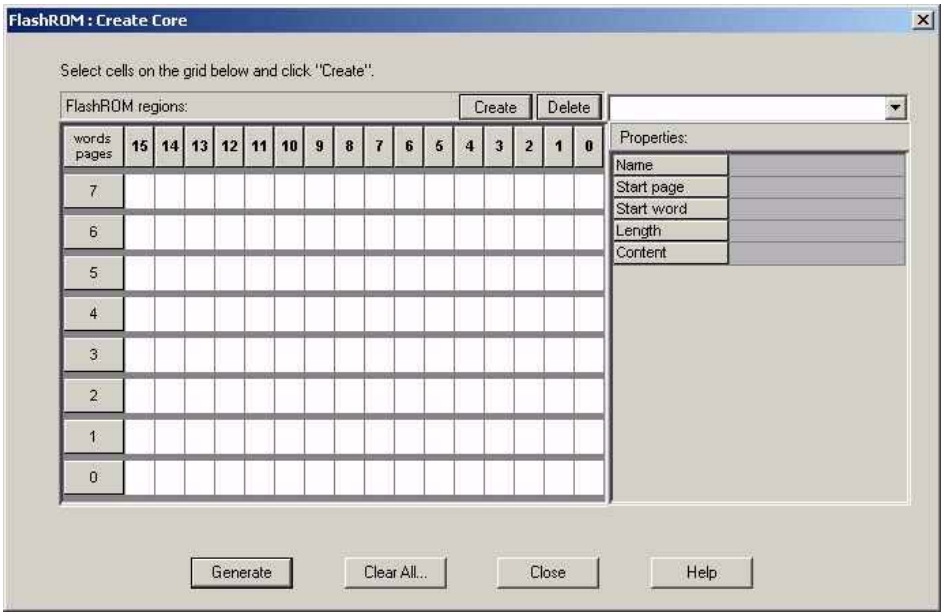


Figure 5-10 • **SmartGen GUI of the FlashROM**

SmartGen allows you to generate the FlashROM netlist in VHDL, Verilog, or EDIF format. After the FlashROM netlist is generated, the core can be instantiated in the main design like other SmartGen cores. Note that the macro library name for FlashROM is UFROM. The following is a sample FlashROM VHDL netlist that can be instantiated in the main design:

```
library ieee;
use ieee.std_logic_1164.all;
library fusion;

entity FROM_a is
  port( ADDR : in std_logic_vector(6 downto 0); DOUT : out std_logic_vector(7 downto 0));
end FROM_a;

architecture DEF_ARCH of  FROM_a is

  component UFROM
    generic (MEMORYFILE:string);
    port(DO0, DO1, DO2, DO3, DO4, DO5, DO6, DO7 : out std_logic;
      ADDR0, ADDR1, ADDR2, ADDR3, ADDR4, ADDR5, ADDR6 : in std_logic := 'U') ;
  end component;

  component GND
    port( Y : out std_logic);
  end component;

signal U_7_PIN2 : std_logic ;

begin

  GND_1_net : GND port map(Y => U_7_PIN2);
  UFROM0 : UFROM
  generic map(MEMORYFILE => "FROM_a.mem")
  port map(DO0 => DOUT(0), DO1 => DOUT(1), DO2 => DOUT(2), DO3 => DOUT(3), DO4 => DOUT(4),
    DO5 => DOUT(5), DO6 => DOUT(6), DO7 => DOUT(7), ADDR0 => ADDR(0), ADDR1 => ADDR(1),
    ADDR2 => ADDR(2), ADDR3 => ADDR(3), ADDR4 => ADDR(4), ADDR5 => ADDR(5),
    ADDR6 => ADDR(6));

end DEF_ARCH;
```

SmartGen generates the following files along with the netlist. These are located in the SmartGen folder for the Libero IDE project.

1. MEM (Memory Initialization) file
2. UFC (User Flash Configuration) file
3. Log file

The MEM file is used for simulation, as explained in the "Simulation of FlashROM Design" section. The UFC file, generated by SmartGen, has the FlashROM configuration for single or multiple devices and is used during STAPL generation. It contains the region properties and simulation values. Note that any changes in the MEM file will not be reflected in the UFC file. Do not modify the UFC to change FlashROM content. Instead, use the SmartGen GUI to modify the FlashROM content. See the "Programming File Generation for FlashROM Design" section on page 5-11 for a description of how the UFC file is used during the programming file generation. The log file has information regarding the file type and file location.

Simulation of FlashROM Design

The MEM file has 128 rows of 8 bits, each representing the contents of the FlashROM used for simulation. For example, the first row represents page 0, byte 0; the next row is page 0, byte 1; and so the pattern continues. Note that the three MSBs of the address define the page number, and the four LSBs define the byte number. So, if you send address 0000100 to FlashROM, this corresponds to the page 0 and byte 4 location, which is the fifth row in the MEM file. SmartGen defaults to 0s for any unspecified location of the FlashROM. Besides using the MEM file generated by SmartGen, you can create a binary file with 128 rows of 8 bits each and use this as a MEM file. Actel recommends that you use different file names if you plan to generate multiple MEM files. During simulation, Libero IDE passes the MEM file used as the generic file in the netlist, along with the design files and testbench. If you want to use different MEM files during simulation, you need to modify the generic file reference in the netlist.

```
.....
UFROM0: UFROM
--generic map(MEMORYFILE => "F:\Appsnotes\FROM\test_designs\testa\smartgen\FROM_a.mem")
--generic map(MEMORYFILE => "F:\Appsnotes\FROM\test_designs\testa\smartgen\FROM_b.mem")
.....
```

The VITAL and Verilog simulation models accept the generics passed by the netlist, read the MEM file, and perform simulation with the data in the file.

Programming File Generation for FlashROM Design

FlashPoint is the programming software used to generate the programming files for flash devices. Depending on the applications, you can use the FlashPoint software to generate a STAPL file with different FlashROM contents. In each case, optional AES decryption is available. To generate a STAPL file that contains the same FPGA core content and different FlashROM contents, the FlashPoint software needs an Array Map file for the core and UFC file(s) for the FlashROM. This final STAPL file represents the combination of the logic of the FPGA core and FlashROM content.

FlashPoint generates the STAPL files you can use to program the desired FlashROM page and/or FPGA core of the FPGA device contents. FlashPoint supports the encryption of the FlashROM content and/or FPGA Array configuration data. In the case of using the FlashROM for device serialization, a sequence of unique FlashROM contents will be generated. When generating a programming file with multiple unique FlashROM contents, you can specify in FlashPoint whether to include all FlashROM content in a single STAPL file or generate a different STAPL file for each FlashROM (Figure 5-11). The programming software (FlashPro) handles the single STAPL file that contains the FlashROM content from multiple devices. It enables you to program the FlashROM content into a series of devices sequentially (Figure 5-11). See the *FlashPro User's Guide* for information on serial programming.

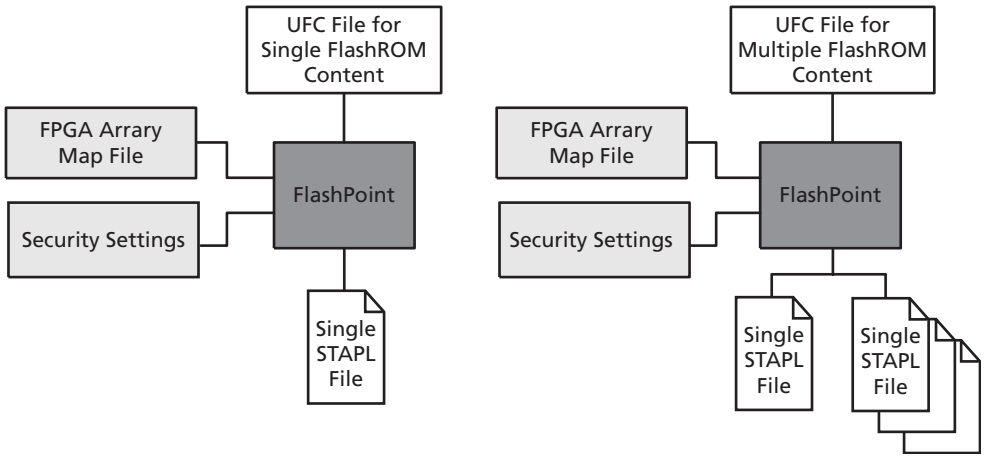


Figure 5-11 • **Single or Multiple Programming File Generation**

Figure 5-12 on page 5-12 shows the programming file generator, which enables different STAPL file generation methods. When you select **Program FlashROM** and choose the UFC file, the FlashROM Settings window appears, as shown in Figure 5-13 on page 5-12. In this window, you can select the FlashROM page you want to program and the data value for the configured regions. This enables you to use a different page for different programming files.

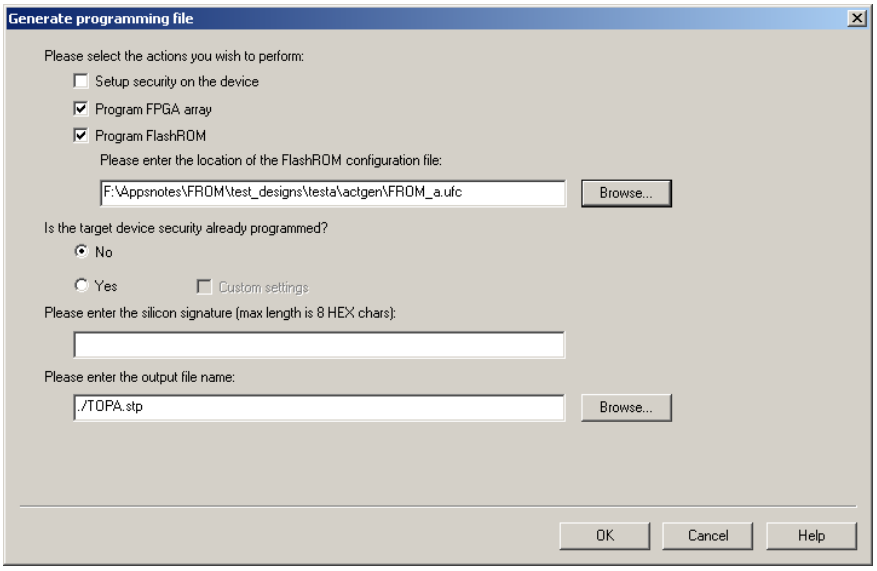


Figure 5-12 • **Programming File Generator**

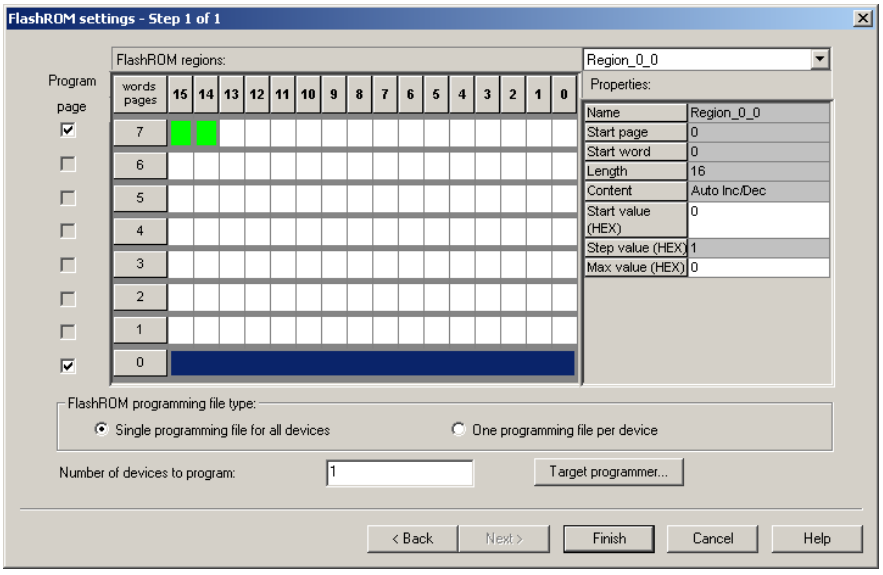


Figure 5-13 • **Setting FlashROM during Programming File Generation**

The programming hardware and software can load the FlashROM with the appropriate STAPL file. Programming software handles the single STAPL file that contains multiple FlashROM contents for multiple devices, and programs the FlashROM in sequential order (e.g., for device serialization).

This feature is supported in the programming software. After programming with the STAPL file, you can run **DEVICE_INFO** to check the FlashROM content.

DEVICE_INFO displays the FlashROM content, serial number, Design Name, and checksum as shown below:

```
EXPORT IDCODE[32] = 123261CF
EXPORT SILSIG[32] = 00000000
User information :
CHECKSUM: 61A0
Design Name:        TOP
Programming Method: STAPL
Algorithm Version: 1
Programmer: UNKNOWN
=======================================
FlashROM Information :
EXPORT Region_7_0[128] = FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
=======================================
Security Setting :
Encrypted FlashROM Programming Enabled.
Encrypted FPGA Array Programming Enabled.
=======================================
```

The Libero IDE file manager recognizes the UFC and MEM files and displays them in the appropriate view. Libero IDE also recognizes the multiple programming files, if you choose the option to generate multiple files for multiple FlashROM content in Designer. These features enable a user-friendly flow for the FlashROM generation and programming in Libero IDE.

Custom Serialization Using FlashROM

You can use FlashROM for device serialization or inventory control by using the Auto Inc region or Read From File region. FlashPoint will automatically generate the serial number sequence for the Auto Inc region with the **Start Value**, **Max Value**, and **Step Value** provided. If you have a unique serial number generation scheme that you prefer, the Read From File region allows you to import the file with your serial number scheme programmed into the region. See the *FlashPro User's Guide* for custom serialization file format information.

The following steps describe how to perform device serialization or inventory control using FlashROM:

1. Generate FlashROM using SmartGen. From the **Properties** section in the **FlashROM Settings** dialog box, select **Auto Inc** or **Read From File** region. For **Auto Inc** region, specify the desired step value. You will not be able to modify this value in the FlashPoint software.
2. Go through the regular design flow and finish place-and-route.
3. Select **Programming File** in Designer and open **Generate Programming File** (Figure 5-12 on page 5-12).
4. Click **Program FlashROM**, browse to the UFC file, and click **Next**. The FlashROM Settings window appears, as shown in Figure 5-13 on page 5-12.
5. Select the FlashROM page you want to program and the data value for the configured regions. The STAPL file generated will contain only the data that targets the selected FlashROM page.
6. Modify properties for the serialization.
   - For Auto Inc region, specify the **Start** and **Max** values.
   - For Read From File region, select the file name of the custom serialization file.
7. Select the FlashROM programming file type you want to generate from the two options below:
   - Single programming file for all devices: generates one programming file with all FlashROM values.
   - One programming file per device: generates a separate programming file for each FlashROM value.

8. Enter the number of devices you want to program and generate the required programming file.
9. Open the programming software and load the programming file. The programming software, FlashPro3 and Silicon Sculptor II, supports the device serialization feature. If, for some reason, the device fails to program a part during serialization, the software allows you to reuse the serial data or skip the serial data. Refer to the *FlashPro User's Guide* for details.

Conclusion

The Fusion, IGLOO, and ProASIC3 families are the only FPGAs that offer on-chip FlashROM support. This document presents information on the FlashROM architecture, possible applications, programming, access through the JTAG and UJTAG interface, and integration into your design. In addition, the Libero IDE tool set enables easy creation and modification of the FlashROM content.

The nonvolatile FlashROM block in the FPGA can be customized, enabling multiple applications.

Additionally, the security offered by the low-power flash devices keeps both the contents of FlashROM and the FPGA design safe from system over-builders, system cloners, and IP thieves.

Related Documents

Handbook Documents

Security in Low-Power Flash Devices

www.actel.com/documents/LPD_Security_HBs.pdf

UJTAG Applications in Actel's Low-Power Flash Devices

http://www.actel.com/documents/LPD_UJTAG_HBs.pdf

User's Guides

FlashPro User's Guide

http://www.actel.com/documents/FlashPro_UG.pdf

Part Number and Revision Date

This document contains content extracted from the Device Architecture section of the datasheet, combined with content previously published as an application note describing features and functions of the device. To improve usability for customers, the device architecture information has now been combined with usage information, to reduce duplication and possible inconsistencies in published information. No technical changes were made to the datasheet content unless explicitly listed. Changes to the application note content were made only to be consistent with existing datasheet information.

Part Number 51700094-007-3

Revised October 2008

List of Changes

The following table lists critical changes that were made in the current version of the chapter.

Previous Version	Changes in Current Version (v1.3)	Page
v1.2 (June 2008)	The "FlashROM Support in Low-Power Devices" section was revised to include new families and make the information more concise.	5-2
	Figure 5-2 · Fusion Device Architecture Overview (AFS600) was replaced. Figure 5-5 · Programming FlashROM Using AES was revised to change "Fusion" to "Flash Device."	5-3, 5-5
	The *FlashPoint User's Guide* was removed from the "User's Guides" section, as its content is now part of the *FlashPro User's Guide*.	
v1.1 (March 2008)	The following changes were made to the family descriptions in Table 5-1 · Low-Power Flash Families: • ProASIC3L was updated to include 1.5 V. • The number of PLLs for ProASIC3E was changed from five to six.	5-2
v1.0 (January 2008)	The chapter was updated to include the IGLOO PLUS family and information regarding 15 k gate devices. The "IGLOO Terminology" section and "ProASIC3 Terminology" section are new.	N/A

6 – SRAM and FIFO Memories in Actel's Low-Power Flash Devices

Introduction

As design complexity grows, greater demands are placed upon an FPGA's embedded memory. Actel Fusion,® IGLOO,® and ProASIC®3 devices provide the flexibility of true dual-port and two-port SRAM blocks. The embedded memory, along with built-in, dedicated FIFO control logic, can be used to create cascading RAM blocks and FIFOs without using additional logic gates.

IGLOO, IGLOO PLUS, and ProASIC3L FPGAs contain an additional feature that allows the device to be put in a low-power mode called Flash*Freeze. In this mode, the core draws minimal power (on the order of 4 to 127 µW) and still retains values on the embedded SRAM/FIFO and registers. Flash*Freeze technology allows the user to switch to Active mode on demand, thus simplifying power management and the use of SRAM/FIFOs.

Device Architecture

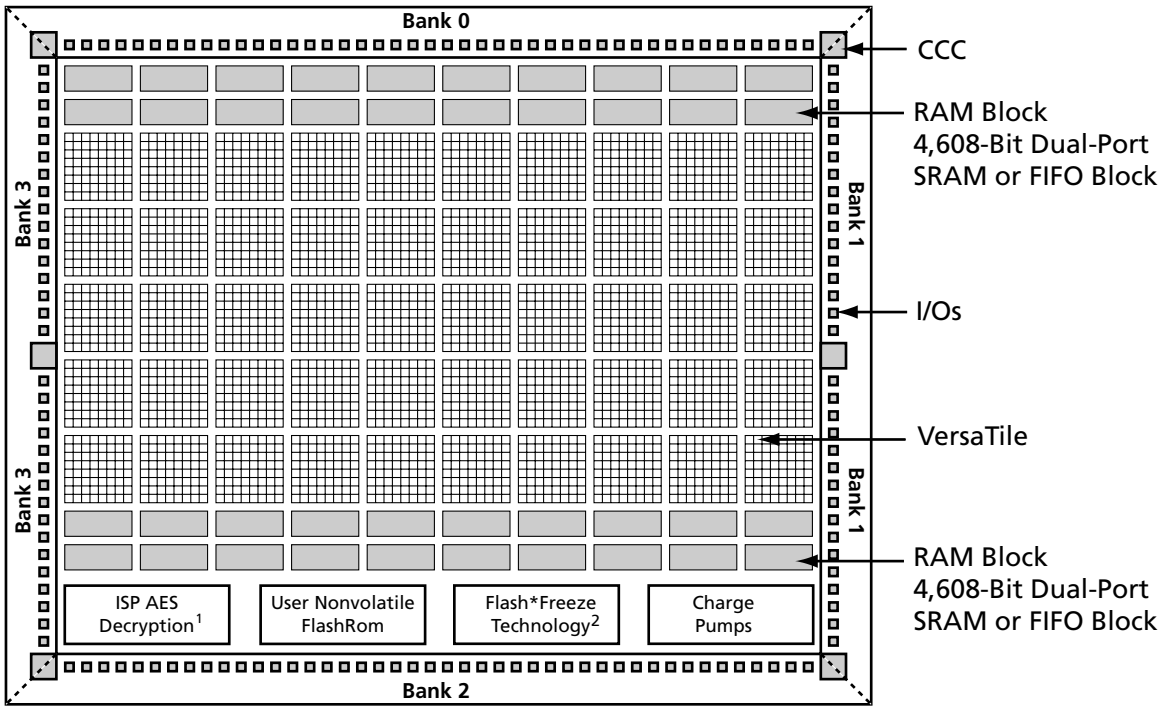
The low-power flash devices feature up to 504 kbits of RAM in 4,608-bit blocks (Figure 6-1 on page 6-2 and Figure 6-2 on page 6-3). The total embedded SRAM for each device can be found in the datasheets. These memory blocks are arranged along the top and bottom of the device to allow better access from the core and I/O (in some devices, they are only available on the north side of the device). Every RAM block has a flexible, hardwired, embedded FIFO controller, enabling the user to implement efficient FIFOs without sacrificing user gates.

In the IGLOO and ProASIC3 families of devices, the following memories are supported:

- 15 k and 30 k gate devices do not support SRAM and FIFO.
- 60 k and 125 k gate devices support memories on the north side of the device only.
- 250 k devices and larger support memories on the north and south sides of the device.

In Fusion devices, the following memories are supported:

- AFS090 and AFS250 support memories on the north side of the device only.
- AFS600 and AFS1500 support memories on the north and south sides of the device.



Notes:

1. *AES decryption not supported in 15 k and 30 k gate devices*
2. *Flash*Freeze is supported only in IGLOO, IGLOO PLUS, and IGLOOe and ProASIC3L devices.*

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

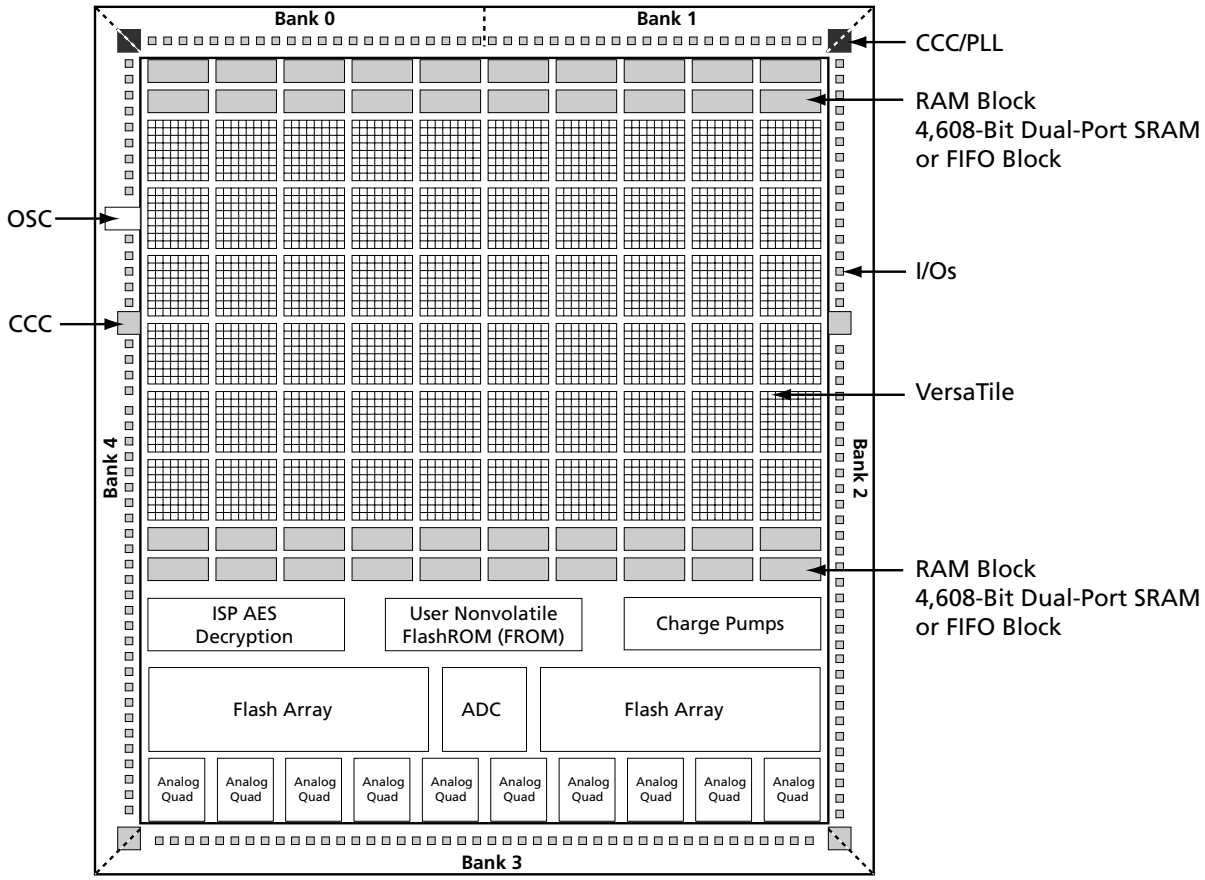


Figure 6-2 • **Fusion Device Architecture Overview (AFS600)**

SRAM/FIFO Support in Low-Power Devices

The low-power flash families listed in Table 6-1 support SRAM and FIFO blocks and the functions described in this document.

Table 6-1 • **Low-Power Flash Families**

Product Line	Family*	Description
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
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 PLUS	IGLOO FPGAs with enhanced I/O capabilities
ProASIC3	ProASIC3	Low-power, high-performance 1.5 V FPGAs
	ProASIC3E	Higher density ProASIC3 FPGAs with six PLLs and additional I/O standards
	ProASIC3L	ProASIC3 FPGAs supporting 1.2 V to 1.5 V with Flash*Freeze technology
	Automotive ProASIC3	ProASIC3 FPGAs qualified for automotive applications
	Military ProASIC3/EL	Military temperature A3PE600L, A3P1000, and A3PE3000L
	RT ProASIC3	Radiation-tolerant RT3PE600L and RT3PE3000L

*Note: *The family names link to the appropriate datasheet, including product brief, DC and switching characteristics, and packaging information.*

IGLOO Terminology

In documentation, the terms IGLOO families and IGLOO devices refer to all of the IGLOO products as listed in Table 6-1. Where the information applies to only one family or limited devices, these exclusions will be explicitly stated.

ProASIC3 Terminology

In documentation, the terms ProASIC3 families and ProASIC3 devices refer to all of the ProASIC3 families as listed in Table 6-1. Where the information applies to only one family or limited devices, these exclusions will be explicitly stated.

To further understand the differences between the IGLOO and ProASIC3 families, refer to the *Industry's Lowest Power FPGAs Portfolio*.

SRAM and FIFO Architecture

To meet the needs of high-performance designs, the memory blocks operate strictly in synchronous mode for both read and write operations. The read and write clocks are completely independent, and each can operate at any desired frequency up to 250 MHz.

- 4k×1, 2k×2, 1k×4, 512×9 (dual-port RAM—2 read / 2 write or 1 read / 1 write)
- 512×9, 256×18 (2-port RAM—1 read / 1 write)
- Sync write, sync pipelined / nonpipelined read

Automotive ProASIC3 devices support single-port SRAM capabilities or dual-port SRAM only under specific conditions. Dual-port mode is supported if the clocks to the two SRAM ports are the same and 180° out of phase (i.e., the port A clock is the inverse of the port B clock). The Actel Libero® Integrated Design Environment (IDE) software macro libraries support a dual-port macro only. For use of this macro as a single-port SRAM, the inputs and clock of one port should be tied off (grounded) to prevent errors during design compile. For use in dual-port mode, the same clock with an inversion between the two clock pins of the macro should be used in the design to prevent errors during compile.

The memory block includes dedicated FIFO control logic to generate internal addresses and external flag logic (FULL, EMPTY, AFULL, AEMPTY).

Simultaneous dual-port read/write and write/write operations at the same address are allowed when certain timing requirements are met.

During RAM operation, addresses are sourced by the user logic, and the FIFO controller is ignored. In FIFO mode, the internal addresses are generated by the FIFO controller and routed to the RAM array by internal MUXes.

The low-power flash device architecture enables the read and write sizes of RAMs to be organized independently, allowing for bus conversion. For example, the write size can be set to 256×18 and the read size to 512×9.

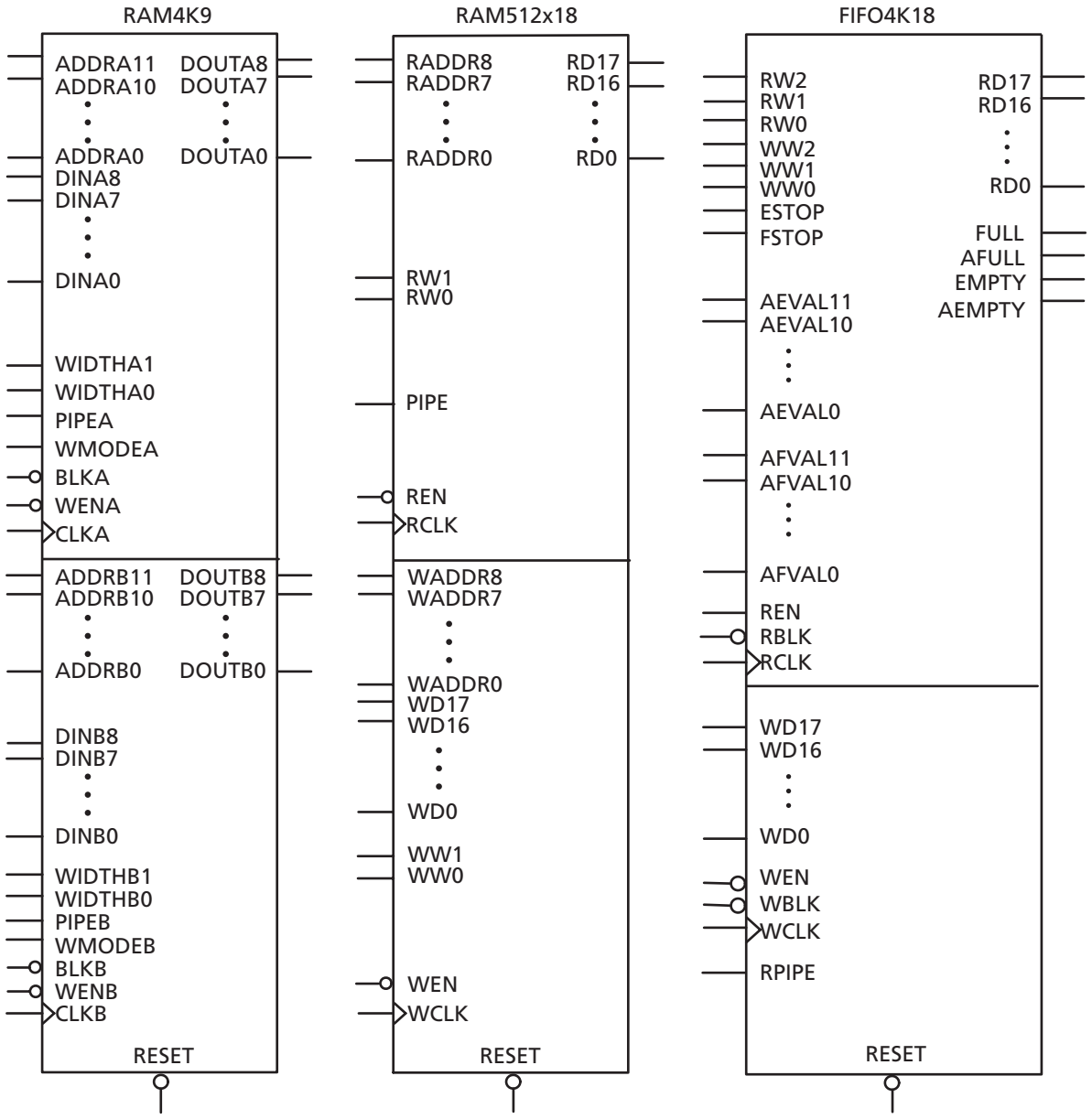
Both the write width and read width for the RAM blocks can be specified independently with the WW (write width) and RW (read width) pins. The different D×W configurations are 256×18, 512×9, 1k×4, 2k×2, and 4k×1. When widths of one, two, or four are selected, the ninth bit is unused. For example, when writing nine-bit values and reading four-bit values, only the first four bits and the second four bits of each nine-bit value are addressable for read operations. The ninth bit is not accessible.

Conversely, when writing four-bit values and reading nine-bit values, the ninth bit of a read operation will be undefined. The RAM blocks employ little-endian byte order for read and write operations.

Memory Blocks and Macros

Memory blocks can be configured with many different aspect ratios, but are generically supported in the macro libraries as one of two memory elements: RAM4K9 or RAM512X18. The RAM4K9 is configured as a true dual-port memory block, and the RAM512X18 is configured as a two-port memory block. Dual-port memory allows the RAM to both read from and write to either port independently. Two-port memory allows the RAM to read from one port and write to the other using a common clock or independent read and write clocks. If needed, the RAM4K9 blocks can be configured as two-port memory blocks. The memory block can be configured as a FIFO by combining the basic memory block with dedicated FIFO controller logic. The FIFO macro is named FIFO4KX18 (Figure 6-3 on page 6-6).

Clocks for the RAM blocks can be driven by the VersaNet (global resources) or by regular nets. When using local clock segments, the clock segment region that encompasses the RAM blocks can drive the RAMs. In the dual-port configuration (RAM4K9), each memory block port can be driven by either rising-edge or falling-edge clocks. Each port can be driven by clocks with different edges. Though only a rising-edge clock can drive the physical block itself, the Actel Designer software will automatically bubble-push the inversion to properly implement the falling-edge trigger for the RAM block.



Note: Automotive ProASIC3 devices restrict RAM4K9 to a single port or to dual ports with the same clock 180° out of phase (inverted) between clock pins. In single-port mode, inputs to port B should be tied to ground to prevent errors during compile. For FIFO4K18, the same clock 180° out of phase (inverted) between clock pins should be used.

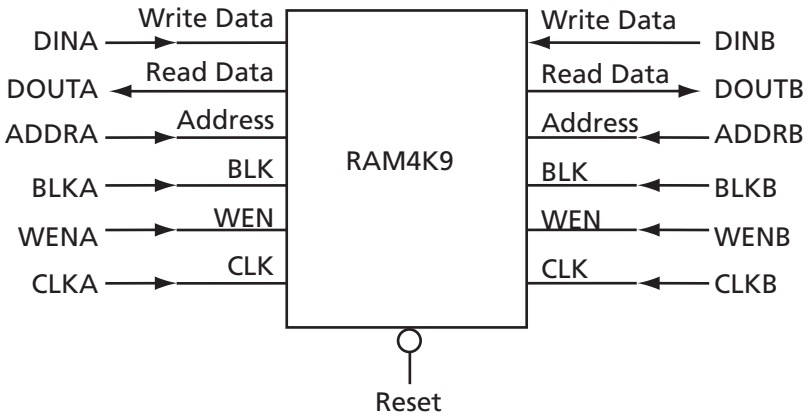
Figure 6-3 • **Supported Basic RAM Macros**

SRAM Features

RAM4K9 Macro

RAM4K9 is the dual-port configuration of the RAM block (Figure 6-4). The RAM4K9 nomenclature refers to both the deepest possible configuration and the widest possible configuration the dual-port RAM block can assume, and does not denote a possible memory aspect ratio. The RAM block can be configured to the following aspect ratios: 4,096x1, 2,048x2, 1,024x4, and 512x9. RAM4K9 is fully synchronous and has the following features:

- Two ports that allow fully independent reads and writes at different frequencies
- Selectable pipelined or nonpipelined read
- Active-low block enables for each port
- Toggle control between read and write mode for each port
- Active-low asynchronous reset
- Pass-through write data or hold existing data on output. In pass-through mode, the data written to the write port will immediately appear on the read port.
- Designer software will automatically facilitate falling-edge clocks by bubble-pushing the inversion to previous stages.



Note: For timing diagrams of the RAM signals, refer to the appropriate family datasheet.

Figure 6-4 • **RAM4K9 Simplified Configuration**

Signal Descriptions for RAM4K9

Note: Automotive ProASIC3 devices support single-port SRAM capabilities, or dual-port SRAM only under specific conditions. Dual-port mode is supported if the clocks to the two SRAM ports are the same and 180° out of phase (i.e., the port A clock is the inverse of the port B clock). Since Actel Libero IDE macro libraries support a dual-port macro only, certain modifications must be made. These are detailed below.

The following signals are used to configure the RAM4K9 memory element:

WIDTHA and WIDTHB

These signals enable the RAM to be configured in one of four allowable aspect ratios (Table 6-2 on page 6-8).

Note: When using the SRAM in single-port mode for Automotive ProASIC3 devices, WIDTHB should be tied to ground.

Table 6-2 • **Allowable Aspect Ratio Settings for WIDTHA[1:0]**

WIDTHA[1:0]	WIDTHB[1:0]	D×W
00	00	4k×1
01	01	2k×2
10	10	1k×4
11	11	512×9

Note: The aspect ratio settings are constant and cannot be changed on the fly.

BLKA and BLKB

These signals are active-low and will enable the respective ports when asserted. When a BLKx signal is deasserted, that port's outputs hold the previous value.

Note: When using the SRAM in single-port mode for Automotive ProASIC3 devices, BLKB should be tied to ground.

WENA and WENB

These signals switch the RAM between read and write modes for the respective ports. A LOW on these signals indicates a write operation, and a HIGH indicates a read.

Note: When using the SRAM in single-port mode for Automotive ProASIC3 devices, WENB should be tied to ground.

CLKA and CLKB

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

Note: For Automotive ProASIC3 devices, dual-port mode is supported if the clocks to the two SRAM ports are the same and 180° out of phase (i.e., the port A clock is the inverse of the port B clock). For use of this macro as a single-port SRAM, the inputs and clock of one port should be tied off (grounded) to prevent errors during design compile.

PIPEA and PIPEB

These signals are used to specify pipelined read on the output. A LOW on PIPEA or PIPEB indicates a nonpipelined read, and the data appears on the corresponding output in the same clock cycle. A HIGH indicates a pipelined read, and data appears on the corresponding output in the next clock cycle.

Note: When using the SRAM in single-port mode for Automotive ProASIC3 devices, PIPEB should be tied to ground. For use in dual-port mode, the same clock with an inversion between the two clock pins of the macro should be used in the design to prevent errors during compile.

WMODEA and WMODEB

These signals are used to configure the behavior of the output when the RAM is in write mode. A LOW on these signals makes the output retain data from the previous read. A HIGH indicates pass-through behavior, wherein the data being written will appear immediately on the output. This signal is overridden when the RAM is being read.

Note: When using the SRAM in single-port mode for Automotive ProASIC3 devices, WMODEB should be tied to ground.

RESET

This active-low signal resets the control logic, forces the output hold state registers to zero, disables reads and writes from the SRAM block, and clears the data hold registers when asserted. It does not reset the contents of the memory array.

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.

ADDRA and ADDRB

These are used as read or write addresses, and they are 12 bits wide. When a depth of less than 4 k is specified, the unused high-order bits must be grounded (Table 6-3 on page 6-9).

Note: When using the SRAM in single-port mode for Automotive ProASIC3 devices, ADDRB should be tied to ground.

Table 6-3 • **Address Pins Unused/Used for Various Supported Bus Widths**

D×W	ADDRx	
	Unused	Used
4k×1	None	[11:0]
2k×2	[11]	[10:0]
1k×4	[11:10]	[9:0]
512×9	[11:9]	[8:0]

Note: The "x" in ADDRx implies A or B.

DINA and DINB

These are the input data signals, and they are nine bits wide. Not all nine bits are valid in all configurations. When a data width less than nine is specified, unused high-order signals must be grounded (Table 6-4).

Note: When using the SRAM in single-port mode for Automotive ProASIC3 devices, DINB should be tied to ground.

DOUTA and DOUTB

These are the nine-bit output data signals. Not all nine bits are valid in all configurations. As with DINA and DINB, high-order bits may not be used (Table 6-4). The output data on unused pins is undefined.

Table 6-4 • **Unused/Used Input and Output Data Pins for Various Supported Bus Widths**

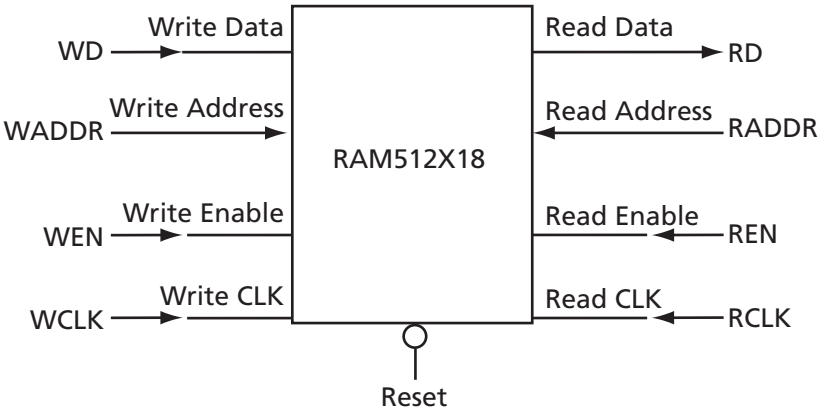
D×W	DINx/DOUTx	
	Unused	Used
4k×1	[8:1]	[0]
2k×2	[8:2]	[1:0]
1k×4	[8:4]	[3:0]
512×9	None	[8:0]

Note: The "x" in DINx or DOUTx implies A or B.

RAM512X18 Macro

RAM512X18 is the two-port configuration of the same RAM block (Figure 6-5 on page 6-10). Like the RAM4K9 nomenclature, the RAM512X18 nomenclature refers to both the deepest possible configuration and the widest possible configuration the two-port RAM block can assume. In two-port mode, the RAM block can be configured to either the 512x9 aspect ratio or the 256x18 aspect ratio. RAM512X18 is also fully synchronous and has the following features:

- Dedicated read and write ports
- Active-low read and write enables
- Selectable pipelined or nonpipelined read
- Active-low asynchronous reset
- Designer software will automatically facilitate falling-edge clocks by bubble-pushing the inversion to previous stages.



Note: For timing diagrams of the RAM signals, refer to the appropriate family datasheet.

Figure 6-5 • **512X18 Two-Port RAM Block Diagram**

Signal Descriptions for RAM512X18

RAM512X18 has slightly different behavior from RAM4K9, as it has dedicated read and write ports.

WW and RW

These signals enable the RAM to be configured in one of the two allowable aspect ratios (Table 6-5).

Table 6-5 • **Aspect Ratio Settings for WW[1:0]**

WW[1:0]	RW[1:0]	D×W
01	01	512×9
10	10	256×18
00, 11	00, 11	Reserved

WD and RD

These are the input and output data signals, and they are 18 bits wide. When a 512×9 aspect ratio is used for write, WD[17:9] are unused and must be grounded. If this aspect ratio is used for read, RD[17:9] are undefined.

WADDR and RADDR

These are read and write addresses, and they are nine bits wide. When the 256×18 aspect ratio is used for write or read, WADDR[8] and RADDR[8] are unused and must be grounded.

WCLK and RCLK

These signals are the write and read clocks, respectively. They can be clocked on the rising or falling edge of WCLK and RCLK.

WEN and REN

These signals are the write and read enables, respectively. They are both active-low by default. These signals can be configured as active-high.

RESET

This active-low signal resets the control logic, forces the output hold state registers to zero, disables reads and writes from the SRAM block, and clears the data hold registers when asserted. It does not reset the contents of the memory array.

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.

PIPE

This signal is used to specify pipelined read on the output. A LOW on PIPE 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.

SRAM Usage

The following descriptions refer to the usage of both RAM4K9 and RAM512X18.

Clocking

The dual-port SRAM blocks are only clocked on the rising edge. SmartGen allows falling-edge-triggered clocks by adding inverters to the netlist, hence achieving dual-port SRAM blocks that are clocked on either edge (rising or falling). For dual-port SRAM, each port can be clocked on either edge and by separate clocks by port. Note that for Automotive ProASIC3, the same clock, with an inversion between the two clock pins of the macro, should be used in design to prevent errors during compile.

Low-power flash devices support inversion (bubble-pushing) throughout the FPGA architecture, including the clock input to the SRAM modules. Inversions added to the SRAM clock pin on the design schematic or in the HDL code will be automatically accounted for during design compile without incurring additional delay in the clock path.

The two-port SRAM can be clocked on the rising or falling edge of WCLK and RCLK.

If negative-edge RAM and FIFO clocking is selected for memory macros, clock edge inversion management (bubble-pushing) is automatically used within the development tools, without performance penalty.

Modes of Operation

There are two read modes and one write mode:

- Read Nonpipelined (synchronous—1 clock edge): In the standard read mode, new data is driven onto the RD bus in the same clock cycle following RA and REN valid. The read address is registered on the read port clock active edge, and data appears at RD after the RAM access time. Setting PIPE to OFF enables this mode.
- Read Pipelined (synchronous—2 clock edges): The pipelined mode incurs an additional clock delay from address to data but enables operation at a much higher frequency. The read address is registered on the read port active clock edge, and the read data is registered and appears at RD after the second read clock edge. Setting PIPE to ON enables this mode.
- Write (synchronous—1 clock edge): On the write clock active edge, the write data is written into the SRAM at the write address when WEN is HIGH. The setup times of the write address, write enables, and write data are minimal with respect to the write clock.

RAM Initialization

Each SRAM block can be individually initialized on power-up by means of the JTAG port using the UJTAG mechanism. The shift register for a target block can be selected and loaded with the proper bit configuration to enable serial loading. The 4,608 bits of data can be loaded in a single operation.

FIFO Features

The FIFO4KX18 macro is created by merging the RAM block with dedicated FIFO logic (Figure 6-6 on page 6-12). Since the FIFO logic can only be used in conjunction with the memory block, there is no separate FIFO controller macro. As with the RAM blocks, the FIFO4KX18 nomenclature does not refer to a possible aspect ratio, but rather to the deepest possible data depth and the widest possible data width. FIFO4KX18 can be configured into the following aspect ratios: 4,096x1, 2,048x2, 1,024x4, 512x9, and 256x18. In addition to being fully synchronous, the FIFO4KX18 also has the following features:

- Four FIFO flags: Empty, Full, Almost-Empty, and Almost-Full
- EMPTY flag is synchronized to the read clock
- FULL flag is synchronized to the write clock
- Both Almost-Empty and Almost-Full flags have programmable thresholds

- Active-low asynchronous reset
- Active-low block enable
- Active-low write enable
- Active-high read enable
- Ability to configure the FIFO to either stop counting after the empty or full states are reached or to allow the FIFO counters to continue
- Designer software will automatically facilitate falling-edge clocks by bubble-pushing the inversion to previous stages.

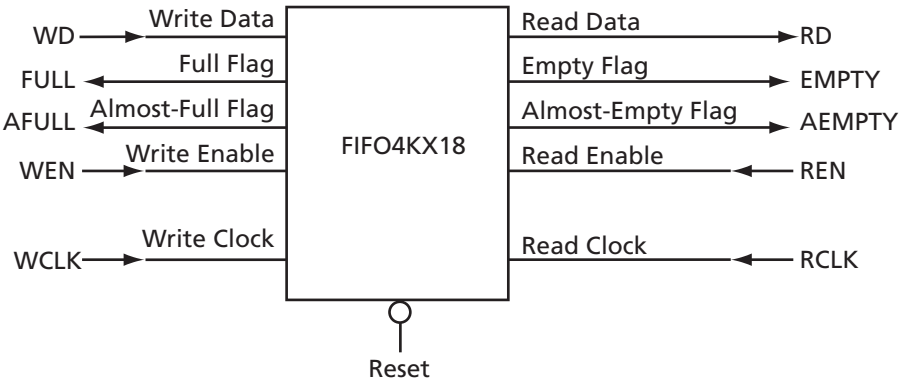


Figure 6-6 • **FIFO4KX18 Block Diagram**

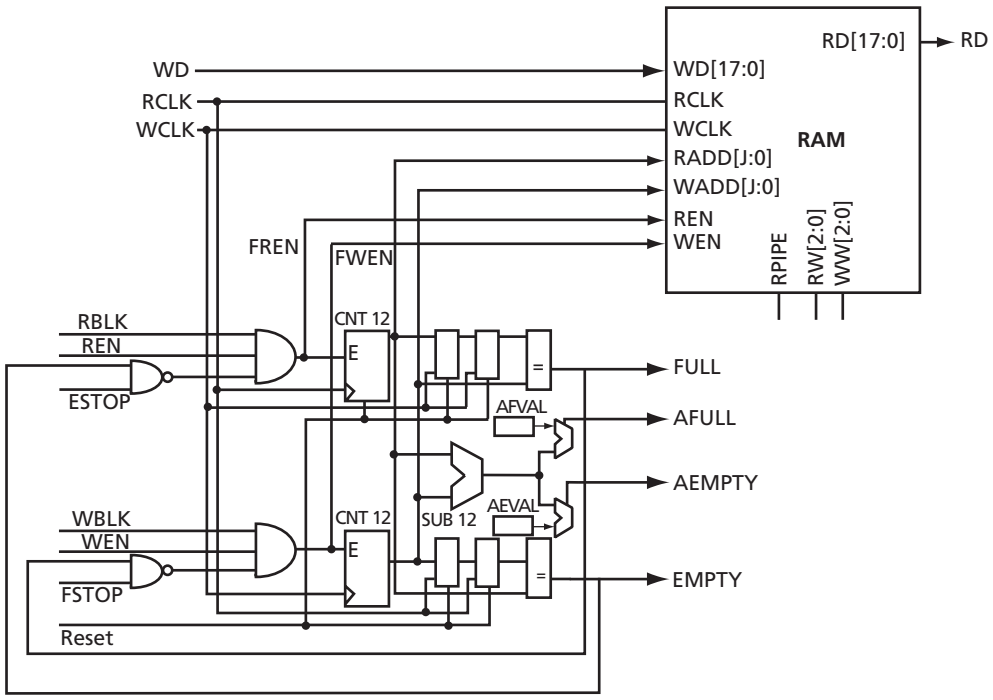


Figure 6-7 • **RAM Block with Embedded FIFO Controller**

The FIFOs maintain a separate read and write address. Whenever the difference between the write address and the read address is greater than or equal to the almost-full value (AFVAL), the Almost-Full flag is asserted. Similarly, the Almost-Empty flag is asserted whenever the difference between the write address and read address is less than or equal to the almost-empty value (AEVAL).

Due to synchronization between the read and write clocks, the Empty flag will deassert after the second read clock edge from the point that the write enable asserts. However, since the Empty flag is synchronized to the read clock, it will assert after the read clock reads the last data in the FIFO. Also, since the Full flag is dependent on the actual hardware configuration, it will assert when the actual physical implementation of the FIFO is full.

For example, when a user configures a 128×18 FIFO, the actual physical implementation will be a 256×18 FIFO element. Since the actual implementation is 256×18, the Full flag will not trigger until the 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 6-14).

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).

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 on page 6-15.

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 6-15.

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 6-15.

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.

FIFO Flag Usage Considerations

The AEVAL and AFVAL pins are used to specify the 12-bit AEMPTY and AFULL threshold values. The FIFO contains separate 12-bit write address (WADDR) and read address (RADDR) counters. WADDR is incremented every time a write operation is performed, and RADDR is incremented every time a read operation is performed. Whenever the difference between WADDR and RADDR is greater than or equal to AFVAL, the AFULL output is asserted. Likewise, whenever the difference between WADDR and RADDR is less than or equal to AEVAL, the AEMPTY output is asserted. To handle different read and write aspect ratios, AFVAL and AEVAL are expressed in terms of total data bits instead of total data words. When users specify AFVAL and AEVAL in terms of read or write words, the SmartGen tool translates them into bit addresses and configures these signals automatically. SmartGen configures the AFULL flag to assert when the write address exceeds the read address by at least a predefined value. In a 2k×8 FIFO, for example, a value of 1,500 for AFVAL means that the AFULL flag will be asserted after a write when the difference between the write address and the read address reaches 1,500 (there have been at least 1,500 more writes than reads). It will stay asserted until the difference between the write and read addresses drops below 1,500.

The AEMPTY flag is asserted when the difference between the write address and the read address is less than a predefined value. In the example above, a value of 200 for AEVAL means that the AEMPTY flag will be asserted when a read causes the difference between the write address and the read address to drop to 200. It will stay asserted until that difference rises above 200. Note that the FIFO can be configured with different read and write widths; in this case, the AFVAL setting is based on the number of write data entries, and the AEVAL setting is based on the number of read data entries. For aspect ratios of 512×9 and 256×18, only 4,096 bits can be addressed by the 12 bits of AFVAL and AEVAL. The number of words must be multiplied by 8 and 16 instead of 9 and 18. The SmartGen tool automatically uses the proper values. To avoid halfwords being written or read, which could happen if different read and write aspect ratios were specified, the FIFO will assert FULL or EMPTY as soon as at least one word cannot be written or read. For example, if a two-bit word is written and a four-bit word is being read, the FIFO will remain in the empty state when the first word is written. This occurs even if the FIFO is not completely empty, because in this case, a complete word cannot be read. The same is applicable in the full state. If a four-bit word is written and a two-bit word is read, the FIFO is full and one word is read. The FULL flag will remain asserted because a complete word cannot be written at this point.

Variable Aspect Ratio and Cascading

Variable aspect ratio and cascading allow users to configure the memory in the width and depth required. The memory block can be configured as a FIFO by combining the basic memory block with dedicated FIFO controller logic. The FIFO macro is named FIFO4KX18. Low-power flash device RAM can be configured as 1, 2, 4, 9, or 18 bits wide. By cascading the memory blocks, any multiple of those widths can be created. The RAM blocks can be from 256 to 4,096 bits deep, depending on the aspect ratio, and the blocks can also be cascaded to create deeper areas. Refer to the aspect ratios available for each macro cell in the "SRAM Features" section on page 6-7. The largest continuous configurable memory area is equal to half the total memory available on the device, because the RAM is separated into two groups, one on each side of the device.

The Actel SmartGen core generator will automatically configure and cascade both RAM and FIFO blocks. Cascading is accomplished using dedicated memory logic and does not consume user gates for depths up to 4,096 bits deep and widths up to 18, depending on the configuration. Deeper memory will utilize some user gates to multiplex the outputs.

Generated RAM and FIFO macros can be created as either structural VHDL or Verilog for easy instantiation into the design. Users of Actel Libero IDE can create a symbol for the macro and incorporate it into a design schematic.

Table 6-10 on page 6-17 shows the number of memory blocks required for each of the supported depth and width memory configurations, and for each depth and width combination. For example, a 256-bit deep by 32-bit wide two-port RAM would consist of two 256×18 RAM blocks. The first 18 bits would be stored in the first RAM block, and the remaining 14 bits would be implemented in the other 256×18 RAM block. This second RAM block would have four bits of unused storage. Similarly, a dual-port memory block that is 8,192 bits deep and 8 bits wide would be implemented using 16 memory blocks. The dual-port memory would be configured in a 4,096×1 aspect ratio. These blocks would then be cascaded two deep to achieve 8,192 bits of depth, and eight wide to achieve the eight bits of width.

Table 6-8 and Table 6-9 show the maximum potential width and depth configuration for each device. Note that 15 k and 30 k gate devices do not support RAM or FIFO.

Table 6-8 • **Memory Availability per IGLOO and ProASIC3 Devices**

Device		RAM Blocks	Maximum Potential Width[1]		Maximum Potential Depth[2]	
IGLOO/ IGLOO PLUS	ProASIC3/ ProASIC3L		Depth	Width	Depth	Width
AGL060 / AGLP060	A3P060	4	256	72 (4×18)	16,384 (4,096×4)	1
AGL125 AGLP125	A3P125	8	256	144 (8×18)	32,768 (4,094×8)	1
AGL250	A3P250/L	8	256	144 (8×18)	32,768 (4,096×8)	1
	A3P400	12	256	216 (12×18)	49,152 (4,096×12)	1
AGL600	A3P600/L	24	256	432 (24×18)	98,304 (4,096×24)	1
AGL1000	A3P1000/L	32	256	576 (32×18)	131,072 (4,096×32)	1
AGLE600	A3PE600	24	256	432 (24×18)	98,304 (4,096×24)	1
	A3PE1500	60	256	1,080 (60×18)	245,760 (4,096×60)	1
AGLE3000	A3PE3000/L	112	256	2,016 (112×18)	458,752 (4,096×112)	1

Notes:

1. *Maximum potential width uses the two-port configuration.*
2. *Maximum potential depth uses the dual-port configuration.*

Table 6-9 • **Memory Availability per Fusion Device**

Device	RAM Blocks	Maximum Potential Width[1]		Maximum Potential Depth[2]	
		Depth	Width	Depth	Width
AFS090	6	256	108 (6×18)	24,576 (4,094×6)	1
AFS250	8	256	144 (8×18)	32,768 (4,094×8)	1
AFS600	24	256	432 (24×18)	98,304 (4,096×24)	1
AFS1500	60	256	1,080 (60×18)	245,760 (4,096×60)	1

Notes:

1. *Maximum potential width uses the two-port configuration.*
2. *Maximum potential depth uses the dual-port configuration.*

Table 6-10 • **RAM and FIFO Memory Block Consumption**

			Depth									
			256		512	1,024	2,048	4,096	8,192	16,384	32,768	65,536
			Two-Port	Dual-Port	Dual-Port	Dual-Port	Dual-Port	Dual-Port	Dual-Port	Dual-Port	Dual-Port	Dual-Port
Width	1	Number Block	1	1	1	1	1	1	2	4	8	16 × 1
		Configuration	Any	Any	Any	1,024 × 4	2,048 × 2	4,096 × 1	2 × (4,096 × 1) Cascade Deep	4 × (4,096 × 1) Cascade Deep	8 × (4,096 × 1) Cascade Deep	16 × (4,096 × 1) Cascade Deep
	2	Number Block	1	1	1	1	1	2	4	8	16	32
		Configuration	Any	Any	Any	1,024×4	2,048 × 2	2 × (4,096 × 1) Cascaded Wide	4 × (4,096 × 1) Cascaded 2 Deep and 2 Wide	8 × (4,096 × 1) Cascaded 4 Deep and 2 Wide	16 × (4,096 × 1) Cascaded 8 Deep and 2 Wide	32 × (4,096 × 1) Cascaded 16 Deep and 2 Wide
	4	Number Block	1	1	1	1	2	4	8	16	32	64
		Configuration	Any	Any	Any	1,024 × 4	2 × (2,048 × 2) Cascaded Wide	4 × (4,096 × 1) Cascaded Wide	4 × (4,096 × 1) Cascaded 2 Deep and 4 Wide	16 × (4,096 × 1) Cascaded 4 Deep and 4 Wide	32 × (4,096 × 1) Cascaded 8 Deep and 4 Wide	64 × (4,096 × 1) Cascaded 16 Deep and 4 Wide
	8	Number Block	1	1	1	2	4	8	16	32	64	
		Configuration	Any	Any	Any	2 × (1,024 × 4) Cascaded Wide	4 × (2,048 × 2) Cascaded Wide	8 × (4,096 × 1) Cascaded Wide	16 × (4,096 × 1) Cascaded 2 Deep and 8 Wide	32 × (4,096 × 1) Cascaded 4 Deep and 8 Wide	64 × (4,096 × 1) Cascaded 8 Deep and 8 Wide	
	9	Number Block	1	1	1	2	4	8	16	32		
		Configuration	Any	Any	Any	2 × (512 × 9) Cascaded Deep	4 × (512 × 9) Cascaded Deep	8 × (512 × 9) Cascaded Deep	16 × (512 × 9) Cascaded Deep	32 × (512 × 9) Cascaded Deep		
	16	Number Block	1	1	1	4	8	16	32	64		
		Configuration	256 × 18	256 × 18	256 × 18	4 × (1,024 × 4) Cascaded Wide	8 × (2,048 × 2) Cascaded Wide	16 × (4,096 × 1) Cascaded Wide	32 × (4,096 × 1) Cascaded 2 Deep and 16 Wide	32 × (4,096 × 1) Cascaded 4 Deep and 16 Wide		
	18	Number Block	1	2	2	4	8	18	32			
		Configuration	256 × 8	2 × (512 × 9) Cascaded Wide	2 × (512 × 9) Cascaded Wide	4 × (512 × 9) Cascaded 2 Deep and 2 Wide	8 × (512 × 9) Cascaded 4 Deep and 2 Wide	16 × (512 × 9) Cascaded 8 Deep and 2 Wide	16 × (512 × 9) Cascaded 16 Deep and 2 Wide			
	32	Number Block	2	4	4	8	16	32	64			
		Configuration	2 × (256 × 18) Cascaded Wide	4 × (512 × 9) Cascaded Wide	4 × (512 × 9) Cascaded Wide	8 × (1,024 × 4) Cascaded Wide	16 × (2,048 × 2) Cascaded Wide	32 × (4,096 × 1) Cascaded Wide	64 × (4,096 × 1) Cascaded 2 Deep and 32 Wide			
	36	Number Block	2	4	4	8	16	32				
		Configuration	2 × (256 × 18) Cascaded Wide	4 × (512 × 9) Cascaded Wide	4 × (512 × 9) Cascaded Wide	4 × (512 × 9) Cascaded 2 Deep and 4 Wide	16 × (512 × 9) Cascaded 4 Deep and 4 Wide	16 × (512 × 9) Cascaded 8 Deep and 4 Wide				
	64	Number Block	4	8	8	16	32	64				
		Configuration	4 × (256 × 18) Cascaded Wide	8 × (512 × 9) Cascaded Wide	8 × (512 × 9) Cascaded Wide	16 × (1,024 × 4) Cascaded Wide	32 × (2,048 × 2) Cascaded Wide	64 × (4,096 × 1) Cascaded Wide				
	72	Number Block	4	8	8	16	32					
		Configuration	4 × (256 × 18) Cascaded Wide	8 × (512 × 9) Cascaded Wide	8 × (512 × 9) Cascaded Wide	16 × (512 × 9) Cascaded Wide	16 × (512 × 9) Cascaded 4 Deep and 8 Wide					

Note: Memory configurations represented by grayed cells are not supported.

Initializing the RAM/FIFO

The SRAM blocks can be initialized with data to use as a lookup table (LUT). Data initialization can be accomplished either by loading the data through the design logic or through the UJTAG interface. The UJTAG macro is used to allow access from the JTAG port to the internal logic in the device. By sending the appropriate initialization string to the JTAG Test Access Port (TAP) Controller, the designer can put the JTAG circuitry into a mode that allows the user to shift data into the array logic through the JTAG port using the UJTAG macro. For a more detailed explanation of the UJTAG macro, refer to *UJTAG Applications in Actel's Low-Power Flash Devices*.

A user interface is required to receive the user command, initialization data, and clock from the UJTAG macro. The interface must synchronize and load the data into the correct RAM block of the design. The main outputs of the user interface block are the following:

- Memory block chip select: Selects a memory block for initialization. The chip selects signals for each memory block that can be generated from different user-defined pockets or simple logic, such as a ring counter (see below).
- Memory block write address: Identifies the address of the memory cell that needs to be initialized.
- Memory block write data: The interface block receives the data serially from the UTDI port of the UJTAG macro and loads it in parallel into the write data ports of the memory blocks.
- Memory block write clock: Drives the WCLK of the memory block and synchronizes the write data, write address, and chip select signals.

Figure 6-8 shows the user interface between UJTAG and the memory blocks.

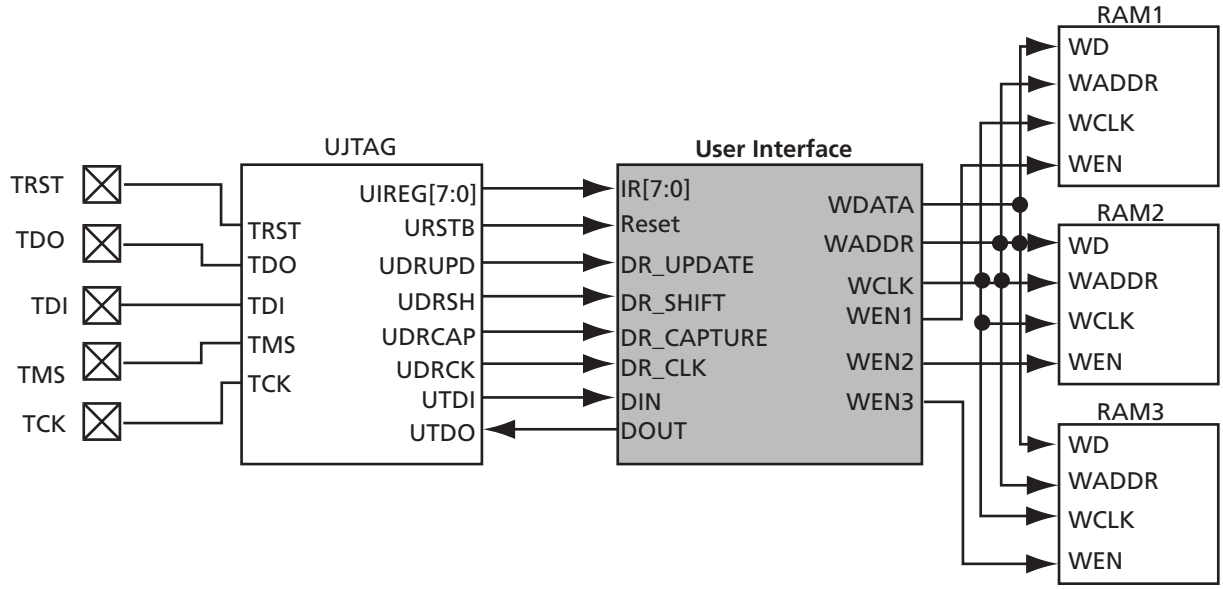


Figure 6-8 • **Interfacing TAP Ports and SRAM Blocks**

An important component of the interface between the UJTAG macro and the RAM blocks is a serial-in/parallel-out shift register. The width of the shift register should equal the data width of the RAM blocks. The RAM data arrives serially from the UTDI output of the UJTAG macro. The data must be shifted into a shift register clocked by the JTAG clock (provided at the UDRCK output of the UJTAG macro).

Then, after the shift register is fully loaded, the data must be transferred to the write data port of the RAM block. To synchronize the loading of the write data with the write address and write clock, the output of the shift register can be pipelined before driving the RAM block.

The write address can be generated in different ways. It can be imported through the TAP using a different instruction opcode and another shift register, or generated internally using a simple

counter. Using a counter to generate the address bits and sweep through the address range of the RAM blocks is recommended, since it reduces the complexity of the user interface block and the board-level JTAG driver.

Moreover, using an internal counter for address generation speeds up the initialization procedure, since the user only needs to import the data through the JTAG port.

The designer may use different methods to select among the multiple RAM blocks. Using counters along with demultiplexers is one approach to set the write enable signals. Basically, the number of RAM blocks needing initialization determines the most efficient approach. For example, if all the blocks are initialized with the same data, one enable signal is enough to activate the write procedure for all of them at the same time. Another alternative is to use different opcodes to initialize each memory block. For a small number of RAM blocks, using counters is an optimal choice. For example, a ring counter can be used to select from multiple RAM blocks. The clock driver of this counter needs to be controlled by the address generation process.

Once the addressing of one block is finished, a clock pulse is sent to the (ring) counter to select the next memory block.

Figure 6-9 illustrates a simple block diagram of an interface block between UJTAG and RAM blocks.

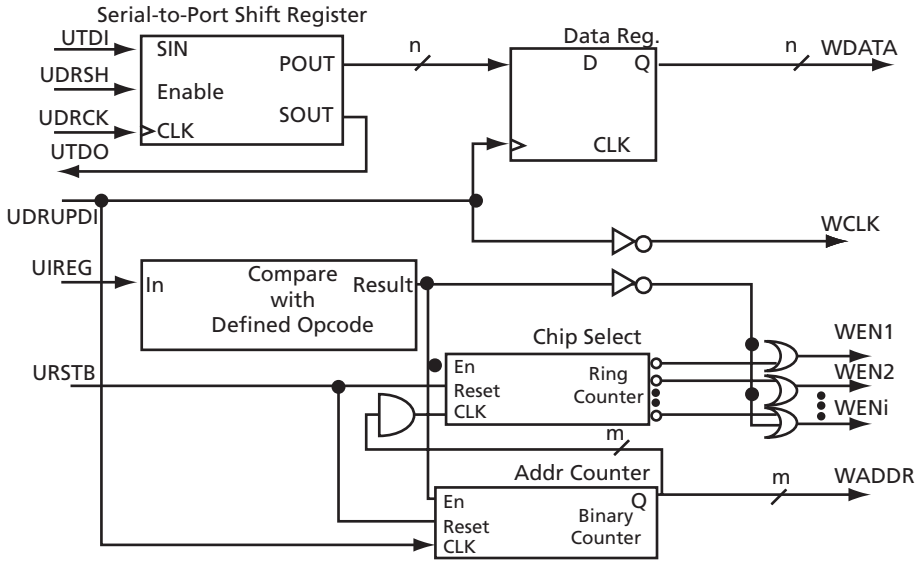


Figure 6-9 • **Block Diagram of a Sample User Interface**

In the circuit shown in Figure 6-9, the shift register is enabled by the UDRSH output of the UJTAG macro. The counters and chip select outputs are controlled by the value of the TAP Instruction Register. The comparison block compares the UIREG value with the "start initialization" opcode value (defined by the user). If the result is true, the counters start to generate addresses and activate the WEN inputs of appropriate RAM blocks.

The UDRUPD output of the UJTAG macro, also shown in Figure 6-9, is used for generating the write clock (WCLK) and synchronizing the data register and address counter with WCLK. UDRUPD is HIGH when the TAP Controller is in the Data Register Update state, which is an indication of completing the loading of one data word. Once the TAP Controller goes into the Data Register Update state, the UDRUPD output of the UJTAG macro goes HIGH. Therefore, the pipeline register and the address counter place the proper data and address on the outputs of the interface block. Meanwhile, WCLK is defined as the inverted UDRUPD. This will provide enough time (equal to the UDRUPD HIGH time) for the data and address to be placed at the proper ports of the RAM block before the rising edge of WCLK. The inverter is not required if the RAM blocks are clocked at the falling edge of the write clock. An example of this is described in the "Example of RAM Initialization" section on page 6-20.

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 6-21.

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.

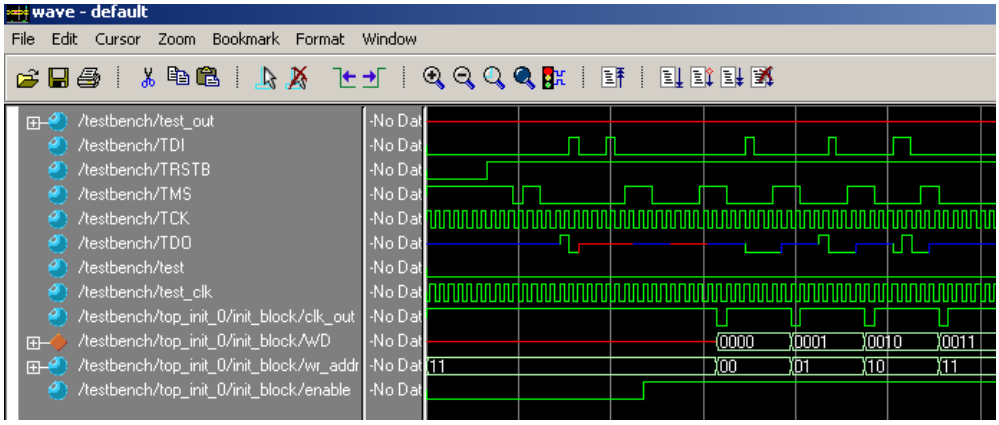


***Figure 6-10* • Simulation of Initialization Step**

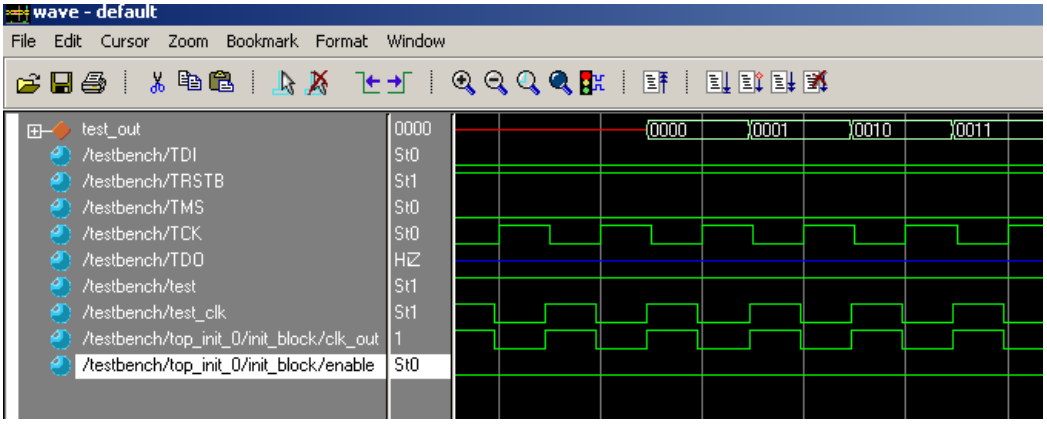


***Figure 6-11* • Simulation of the Test Procedure of the Example**

The ROM emulation application is based on RAM block initialization. If the user's main design has access only to the read ports of the RAM block (RADDR, RD, RCLK, and REN), and the contents of the RAM are already initialized through the TAP, then the memory blocks will emulate ROM functionality for the core design. In this case, the write ports of the RAM blocks are accessed only by the user interface block, and the interface is activated only by the TAP Instruction Register contents.

Users should note that the contents of the RAM blocks are lost in the absence of applied power. However, the 1 kbit of flash memory, FlashROM, in low-power flash devices can be used to retain data after power is removed from the device. Refer to *FlashROM in Actel's Low-Power Flash Devices* for more information.

Sample Verilog Code

Interface Block

```
`define Initialize_start 8'h22 //INITIALIZATION START COMMAND VALUE
`define Initialize_stop 8'h23 //INITIALIZATION START COMMAND VALUE

module interface(IR, rst_n, data_shift, clk_in, data_update, din_ser, dout_ser, test,
  test_out,test_clk,clk_out,wr_en,rd_en,write_word,read_word,rd_addr, wr_addr);

input [7:0] IR;
input [3:0] read_word; //RAM DATA READ BACK
input rst_n, data_shift, clk_in, data_update, din_ser; //INITIALIZATION SIGNALS
input test, test_clk; //TEST PROCEDURE CLOCK AND COMMAND INPUT
output [3:0] test_out; //READ DATA
output [3:0] write_word; //WRITE DATA
output [1:0] rd_addr; //READ ADDRESS
output [1:0] wr_addr; //WRITE ADDRESS
output dout_ser; //TDO DRIVER
output clk_out, wr_en, rd_en;

wire [3:0] write_word;
wire [1:0] rd_addr;
wire [1:0] wr_addr;
wire [3:0] Q_out;
wire enable, test_active;

reg clk_out;

//SELECT CLOCK FOR INITIALIZATION OR READBACK TEST
always @(enable or test_clk or data_update)
begin
  case ({test_active})
    1 : clk_out = test_clk ;
    0 : clk_out = !data_update;
    default : clk_out = 1'b1;
  endcase
end

assign test_active = test && (IR == 8'h23);
assign enable = (IR == 8'h22);
assign wr_en = !enable;
assign rd_en = !test_active;
assign test_out = read_word;
assign dout_ser = Q_out[3];

//4-bit SIN/POUT SHIFT REGISTER
shift_reg data_shift_reg (.Shiften(data_shift), .Shiftin(din_ser), .Clock(clk_in),
  .Q(Q_out));

//4-bit PIPELINE REGISTER
D_pipeline pipeline_reg (.Data(Q_out), .Clock(data_update), .Q(write_word));
```

```
//
addr_counter counter_1 (.Clock(data_update), .Q(wr_addr), .Aset(rst_n),
  .Enable(enable));
addr_counter counter_2 (.Clock(test_clk), .Q(rd_addr), .Aset(rst_n),
  .Enable( test_active));

endmodule
```

Interface Block / UJTAG Wrapper

This example is a sample wrapper, which connects the interface block to the UJTAG and the memory blocks.

```
// WRAPPER
module top_init (TDI, TRSTB, TMS, TCK, TDO, test, test_clk, test_ out);

input TDI, TRSTB, TMS, TCK;
output TDO;
input test, test_clk;
output [3:0] test_out;

wire [7:0] IR;
wire reset, DR_shift, DR_cap, init_clk, DR_update, data_in, data_out;
wire clk_out, wen, ren;
wire [3:0] word_in, word_out;
wire [1:0] write_addr, read_addr;

UJTAG UJTAG_U1 (.UIREG0(IR[0]), .UIREG1(IR[1]), .UIREG2(IR[2]), .UIREG3(IR[3]),
  .UIREG4(IR[4]), .UIREG5(IR[5]), .UIREG6(IR[6]), .UIREG7(IR[7]), .URSTB(reset),
  .UDRSH(DR_shift), .UDRCAP(DR_cap), .UDRCK(init_clk), .UDRUPD(DR_update),
  .UT-DI(data_in), .TDI(TDI), .TMS(TMS), .TCK(TCK), .TRSTB(TRSTB), .TDO(TDO),
  .UT-DO(data_out));
mem_block RAM_block (.DO(word_out), .RCLOCK(clk_out), .WCLOCK(clk_out), .DI(word_in),
  .WRB(wen), .RDB(ren), .WAD-DR(write_addr), .RADDR(read_addr));
interface init_block (.IR(IR), .rst_n(reset), .data_shift(DR_shift), .clk_in(init_clk),
  .data_update(DR_update), .din_ser(data_in), .dout_ser(data_out), .test(test),
  .test_out(test_out), .test_clk(test_clk), .clk_out(clk_out), .wr_en(wen),
  .rd_en(ren), .write_word(word_in), .read_word(word_out), .rd_addr(read_addr),
  .wr_addr(write_addr));

endmodule
```

Address Counter

```
module addr_counter (Clock, Q, Aset, Enable);

input Clock;
output [1:0] Q;
input Aset;
input Enable;

reg [1:0] Qaux;

always @(posedge Clock or negedge Aset)
begin
  if (!Aset) Qaux <= 2'b11;
  else if (Enable) Qaux <= Qaux + 1;
end

assign Q = Qaux;

endmodule
```

Pipeline Register

```
module D_pipeline (Data, Clock, Q);

input [3:0] Data;
input Clock;
output [3:0] Q;

reg [3:0] Q;

always @ (posedge Clock) Q <= Data;

endmodule
```

4x4 RAM Block (created by SmartGen Core Generator)

```
module mem_block(DI,DO,WADDR,RADDR,WRB,RDB,WCLOCK,RCLOCK);

input [3:0] DI;
output [3:0] DO;
input [1:0] WADDR, RADDR;
input WRB, RDB, WCLOCK, RCLOCK;

wire WEBP, WEAP, VCC, GND;

VCC VCC_1_net(.Y(VCC));
GND GND_1_net(.Y(GND));
INV WEBUBBLEB(.A(WRB), .Y(WEBP));
RAM4K9 RAMBLOCK0(.ADDRA11(GND), .ADDRA10(GND), .ADDRA9(GND), .ADDRA8(GND),
  .ADDRA7(GND), .ADDRA6(GND), .ADDRA5(GND), .ADDRA4(GND), .ADDRA3(GND), .ADDRA2(GND),
  .ADDRA1(RADDR[1]), .ADDRA0(RADDR[0]), .ADDRB11(GND), .ADDRB10(GND), .ADDRB9(GND),
  .ADDRB8(GND), .ADDRB7(GND), .ADDRB6(GND), .ADDRB5(GND), .ADDRB4(GND), .ADDRB3(GND),
  .ADDRB2(GND), .ADDRB1(WADDR[1]), .ADDRB0(WADDR[0]), .DINA8(GND), .DINA7(GND),
  .DINA6(GND), .DINA5(GND), .DINA4(GND), .DINA3(GND), .DINA2(GND), .DINA1(GND),
  .DINA0(GND), .DINB8(GND), .DINB7(GND), .DINB6(GND), .DINB5(GND), .DINB4(GND),
  .DINB3(DI[3]), .DINB2(DI[2]), .DINB1(DI[1]), .DINB0(DI[0]), .WIDTHA0(GND),
  .WIDTHA1(VCC), .WIDTHB0(GND), .WIDTHB1(VCC), .PIPEA(GND), .PIPEB(GND),
  .WMODEA(GND), .WMODEB(GND), .BLKA(WEAP), .BLKB(WEBP), .WENA(VCC), .WENB(GND),
  .CLKA(RCLOCK), .CLKB(WCLOCK), .RESET(VCC), .DOUTA8(), .DOUTA7(), .DOUTA6(),
  .DOUTA5(), .DOUTA4(), .DOUTA3(DO[3]), .DOUTA2(DO[2]), .DOUTA1(DO[1]),
  .DOUTA0(DO[0]), .DOUTB8(), .DOUTB7(), .DOUTB6(), .DOUTB5(), .DOUTB4(), .DOUTB3(),
  .DOUTB2(), .DOUTB1(), .DOUTB0());
INV WEBUBBLEA(.A(RDB), .Y(WEAP));

endmodule
```

Software Support

The SmartGen core generator is the easiest way to select and configure the memory blocks (Figure 6-12). SmartGen automatically selects the proper memory block type and aspect ratio, and cascades the memory blocks based on the user's selection. SmartGen also configures any additional signals that may require tie-off.

SmartGen will attempt to use the minimum number of blocks required to implement the desired memory. When cascading, SmartGen will configure the memory for width before configuring for depth. For example, if the user requests a 256×8 FIFO, SmartGen will use a 512×9 FIFO configuration, not 256×18.

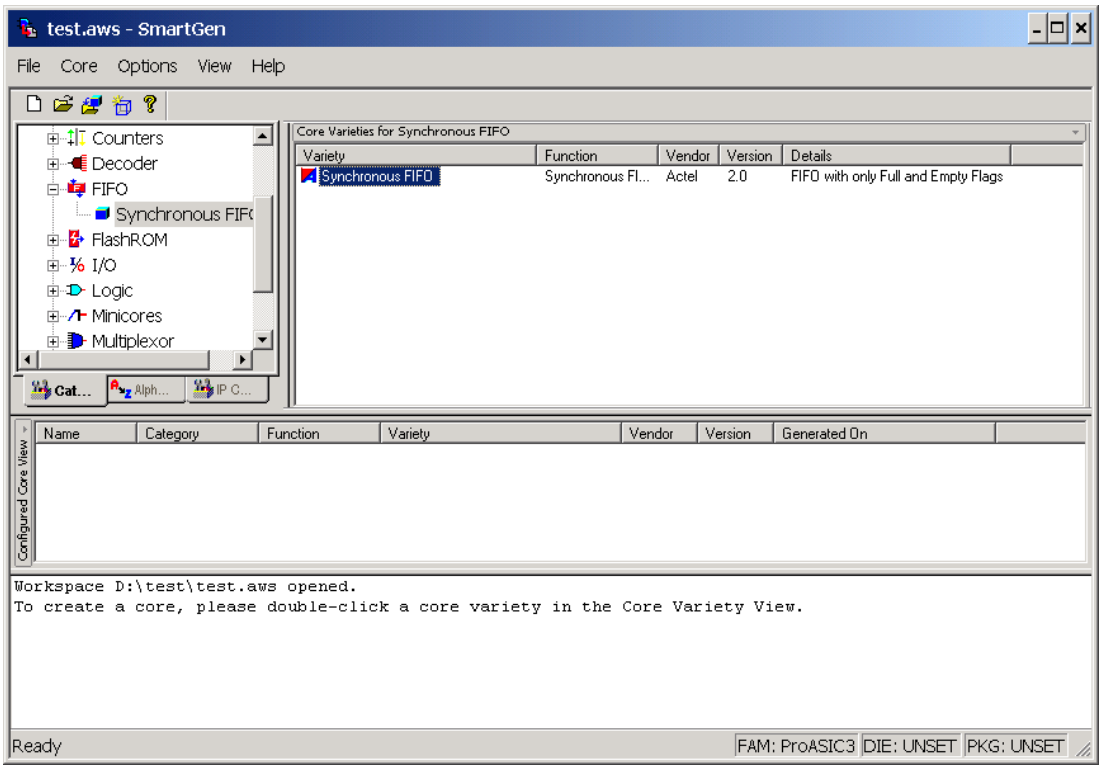


Figure 6-12 • **SmartGen Core Generator Interface**

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).

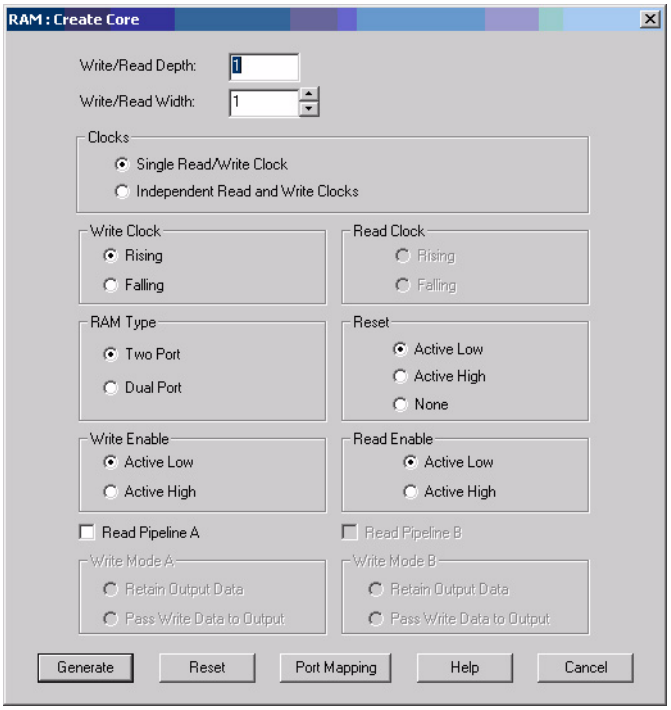


***Figure 6-13* • SmartGen Memory Configuration Interface**

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).

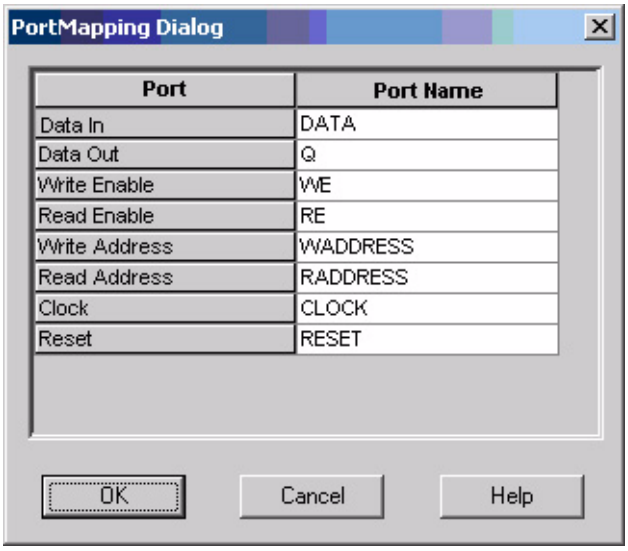


***Figure 6-14* • Port Mapping Interface for SmartGen-Generated Memory**

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 without reprogramming the device. Dynamic flag settings are determined by register values and can be altered without reprogramming the device by reloading the register values either from the design or through the UJTAG interface described in the "Initializing the RAM/FIFO" section on page 6-18.

SmartGen can also configure the FIFO to continue counting after the FIFO is full. In this configuration, the FIFO write counter will wrap after the counter is full and continue to write data. With the FIFO configured to continue to read after the FIFO is empty, the read counter will also wrap and re-read data that was previously read. This mode can be used to continually read back repeating data patterns stored in the FIFO (Figure 6-15).

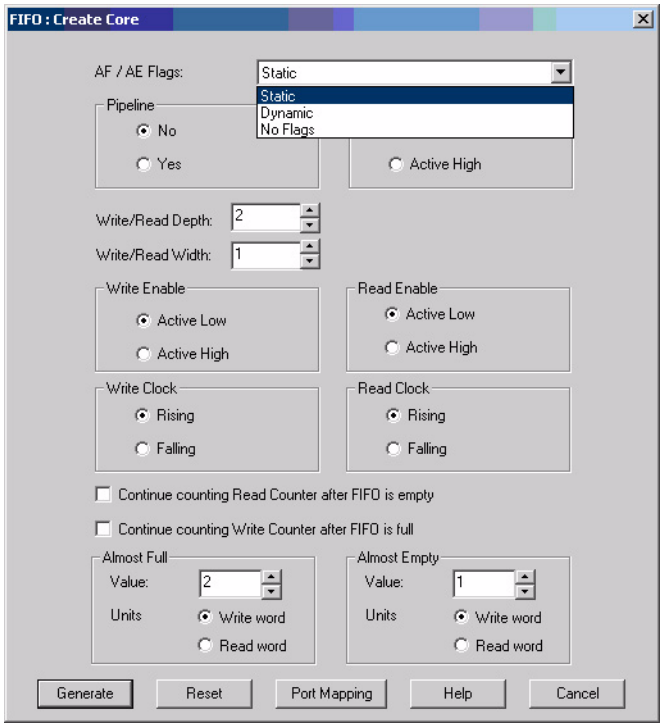


Figure 6-15 • **SmartGen FIFO Configuration Interface**

FIFOs configured using SmartGen can also make use of the port mapping feature to configure the names of the ports.

Limitations

Users should be aware of the following limitations when configuring SRAM blocks for low-power flash devices:

- SmartGen does not track the target device in a family, so it cannot determine if a configured memory block will fit in the target device.
- Dual-port RAMs with different read and write aspect ratios are not supported.
- Cascaded memory blocks can only use a maximum of 64 blocks of RAM.
- The Full flag of the FIFO is sensitive to the maximum depth of the actual physical FIFO block, not the depth requested in the SmartGen interface.

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 Actel SmartGen core generator provides a fast and easy way to configure these memory elements for use in designs.

Related Documents

Handbook Documents

UJTAG Applications in Actel's Low-Power Flash Devices

www.actel.com/documents/LPD_UJTAG_HBs.pdf

FlashROM in Actel's Low-Power Flash Devices

http://www.actel.com/documents/LPD_FlashROM_HBs.pdf

Part Number and Revision Date

This document contains content extracted from the Device Architecture section of the datasheet, combined with content previously published as an application note describing features and functions of the device. To improve usability for customers, the device architecture information has now been combined with usage information, to reduce duplication and possible inconsistencies in published information. No technical changes were made to the datasheet content unless explicitly listed. Changes to the application note content were made only to be consistent with existing datasheet information.

Part Number 51700094-008-4

Revised October 2008

List of Changes

The following table lists critical changes that were made in the current version of the chapter.

Previous Version	Changes in the Current Version (v1.4)	Page
v1.3 (August 2008)	The "SRAM/FIFO Support in Low-Power Devices" section was revised to include new families and make the information more concise.	6-4
	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.	6-5
	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.	6-11
	The "ESTOP and FSTOP Usage" section was updated to refer to FIFO counters in devices in general rather than only IGLOO and ProASIC3E devices.	6-15
v1.2 (June 2008)	The note was removed from Figure 6-7 · RAM Block with Embedded FIFO Controller and placed in the WCLK and RCLK description.	6-12
	The "WCLK and RCLK" description was revised.	6-13
v1.1 (March 2008)	The following changes were made to the family descriptions in Table 6-1 · Low-Power Flash Families: • ProASIC3L was updated to include 1.5 V. • The number of PLLs for ProASIC3E was changed from five to six.	6-4
v1.0 (January 2008)	The "Introduction" section was updated to include the IGLOO PLUS family.	6-1
	The "Device Architecture" section was updated to state that 15 k gate devices do not support SRAM and FIFO.	6-1
	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.	6-3
	The Table 6-1 · Low-Power Flash Families table and associated text were updated to include the IGLOO PLUS family. The "IGLOO Terminology" section and "ProASIC3 Terminology" section are new.	6-4
	The text introducing Table 6-8 · Memory Availability per IGLOO and ProASIC3 Devices was updated to replace "A3P030 and AGL030" with "15 k and 30 k gate devices. Table 6-8 · Memory Availability per IGLOO and ProASIC3 Devices was updated to remove AGL400 and AGLE1500 and include IGLOO PLUS and ProASIC3L devices.	6-16

I/O Descriptions and Usage

Actel
POWER MATTERS

7 – I/O Structures in IGLOO PLUS Devices

Introduction

Low-power flash devices feature a flexible I/O structure, supporting a range of mixed voltages (1.2 V, 1.5 V, 1.8 V, 2.5 V, and 3.3 V) through bank-selectable voltages. The IGLOO® PLUS family supports IGLOO PLUS I/Os.

Users designing I/O solutions are faced with a number of implementation decisions and configuration choices that can directly impact the efficiency and effectiveness of their final design. The flexible I/O structure, supporting a wide variety of voltages and I/O standards, enables users to meet the growing challenges of their many diverse applications. The Actel Libero® Integrated Design Environment (IDE) provides an easy way to implement I/O that will result in robust I/O design.

This document describes IGLOO PLUS I/O types in terms of the supported standards. It then explains the individual features and how to implement them in Actel's Libero IDE.

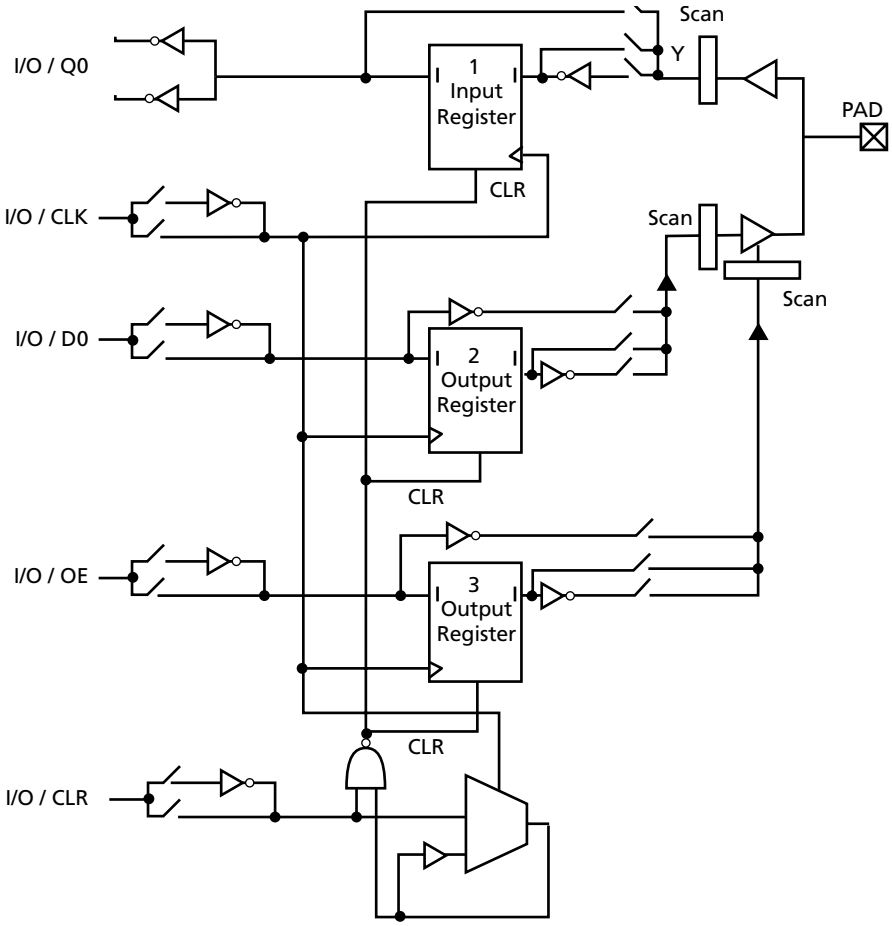


Figure 7-1 • **IGLOO PLUS I/O Block Logical Representation**

Low-Power Flash Device I/O Support

The low-power flash families listed in Table 7-1 support I/Os and the functions described in this document.

Table 7-1 • **Low-Power Flash Families**

Product Line	Family*	Description
IGLOO	IGLOO PLUS	IGLOO FPGAs with enhanced I/O capabilities

*Note: *The family name links to the appropriate datasheet, including product brief, DC and switching characteristics, and packaging information.*

IGLOO Terminology

In documentation, the terms IGLOO families and IGLOO devices refer to all of the IGLOO products as listed in Table 7-1. Where the information applies to only one family or limited devices, these exclusions will be explicitly stated.

IGLOO PLUS I/Os

Table 7-2 and Table 7-3 show the voltages and compatible I/O standards for IGLOO PLUS family.

I/Os provide programmable slew rates, drive strengths, and weak pull-up and pull-down circuits. Selectable Schmitt trigger and 5 V tolerant receivers are offered. See the "5 V Input Tolerance" section on page 7-11 for possible implementations of 5 V tolerance.

All I/Os are in a known state during power-up, and any power-up sequence is allowed without current impact. Refer to the "I/O Power-Up and Supply Voltage Thresholds for Power-On Reset (Commercial and Industrial)" section in the datasheet for more information. During power-up, before reaching activation levels, the I/O input and output buffers are disabled while the weak pull-up is enabled. Activation levels are described in the datasheet.

Table 7-2 • **Supported I/O Standards**

IGLOO PLUS	AGLP030	AGLP060	AGLP125
Single-Ended			
LVTTL/LVCMOS 3.3 V, LVCMOS 2.5 V / 1.8 V / 1.5 V / 1.2 V	✓	✓	✓

I/O Banks and I/O Standards Compatibility

I/Os are grouped into I/O voltage banks. All IGLOO PLUS devices have four I/O banks.

Each I/O voltage bank has dedicated I/O supply and ground voltages. This isolation is necessary to minimize simultaneous switching noise from the input and output (SSI and SSO). The switching noise (ground bounce and power bounce) is generated by the output buffers and transferred into input buffer circuits, and vice versa. Because of these dedicated supplies, only I/Os with compatible standards can be assigned to the same I/O voltage bank. Table 7-3 shows the required voltage compatibility values for each of these voltages.

I/O standards are compatible if their V_{CCI} values are identical. For more information about I/O and global assignments to I/O banks in a device, refer to the specific pin table for the device in the packaging section of the datasheet and the "User I/O Naming Convention" section on page 7-18.

Table 7-3 • **V_{CCI} Voltages and Compatible IGLOO PLUS Standards**

V_{CCI} (typical)	Compatible Standards
3.3 V	LVTTL/LVCMOS 3.3
2.5 V	LVCMOS 2.5
1.8 V	LVCMOS 1.8
1.5 V	LVCMOS 1.5
1.2 V	LVCMOS 1.2

Features Supported on Every I/O

Table 7-4 lists all features supported by transmitter/receiver for single-ended I/Os. Table 7-5 lists the performance of each IO technology.

Table 7-4 • **I/O Features**

Feature	Description
All I/O	• High performance (Table 7-5) • Electrostatic discharge (ESD) protection • I/O register combining option
Single-Ended Transmitter Features	• Hot-swap • I/Os can be configured to behave in Flash*Freeze mode as tristate, HIGH, LOW, or to hold the previous state. • Programmable output slew rate: high and low • Optional weak pull-up and pull-down resistors • Output drive: 3 drive strengths (except for LVCMOS 1.2 V) • LVTTL/LVCMOS 3.3 V outputs compatible with 5 V TTL inputs
Single-Ended Receiver Features	• Selectable Schmitt trigger • 5 V–input–tolerant receiver (Table 7-11 on page 7-11) • Separate ground plane for GNDQ pin and power plane for V_{CCI} pin are used for input buffer to reduce output-induced noise.

Table 7-5 • **Maximum I/O Frequency**

Specification	Maximum Performance	
	IGLOO PLUS V2 or V5 Devices, 1.5 V DC Core Supply Voltage	IGLOO PLUS V2, 1.2 V DC Core Supply Voltage
LVTTL/LVCMOS 3.3 V	180 MHz	TBD
LVCMOS 2.5 V	230 MHz	TBD
LVCMOS 1.8 V	180 MHz	TBD
LVCMOS 1.5 V	120 MHz	TBD
LVCMOS 1.2 V	N/A	TBD

I/O Architecture

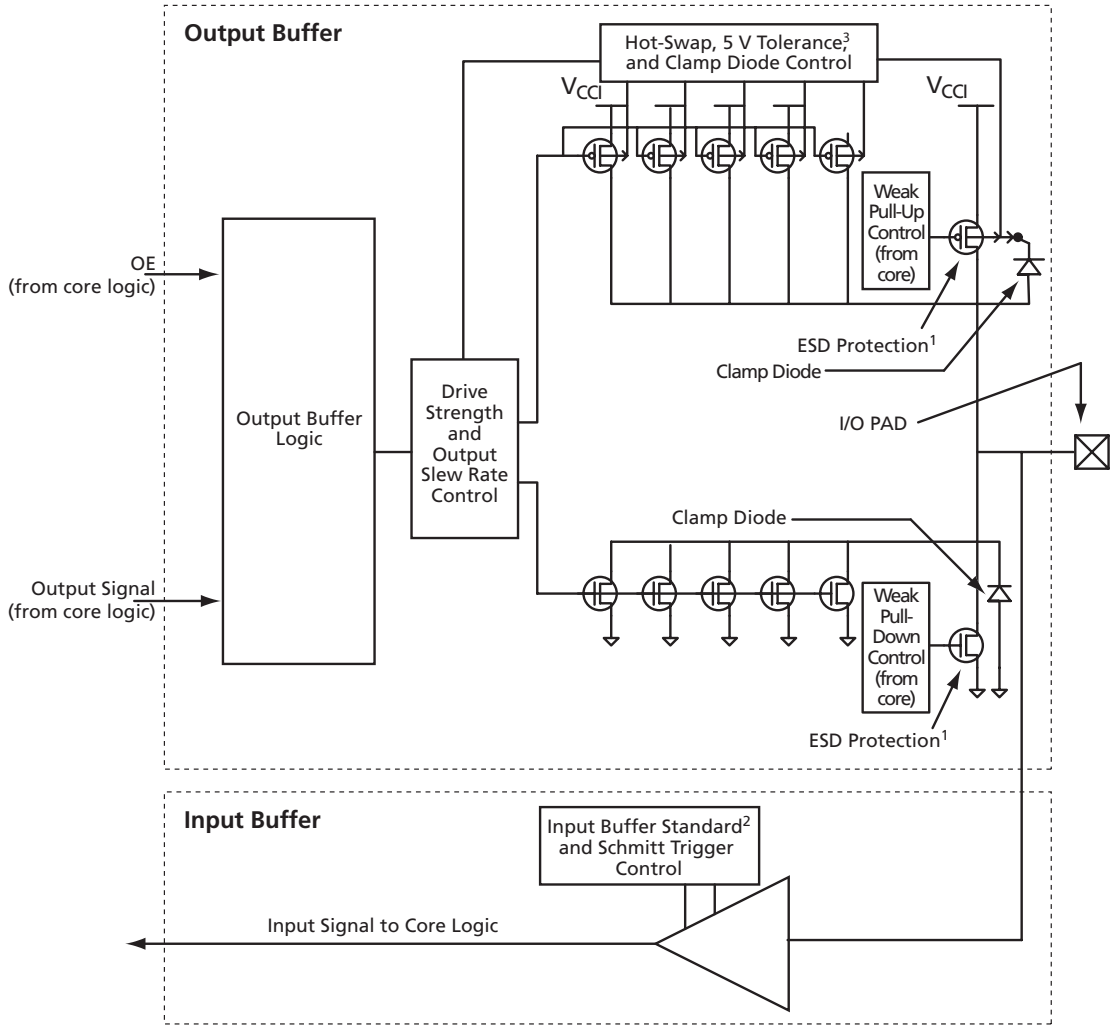
I/O Tile

The I/O tile provides a flexible, programmable structure for implementing a large number of I/O standards. In addition, the registers available in the I/O tile can be used to support high-performance register inputs and outputs, with register enable if desired (Figure 7-1 on page 7-1).

As depicted in Figure 7-1 on page 7-1, all I/O registers share one CLR port.

I/O Bank Structure

Low-power flash device I/Os are divided into multiple technology banks. The IGLOO PLUS devices have four banks. Each bank has its own V_{CCI} power supply pin. Refer to Figure 7-2 for more information.



Notes:

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

Figure 7-2 • **Simplified I/O Buffer Circuitry**

I/O Registers

Each I/O module contains several input and output registers. Refer to Figure 7-2 on page 7-5 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 7-1 on page 7-1) between registers to implement single-ended 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. The I/O register combining requires that no combinatorial logic be present between the register and the I/O.

I/O Standards

Single-Ended Standards

These I/O standards use a push-pull CMOS output stage with a voltage referenced to system ground to designate logical states. The input buffer configuration, output drive, and I/O supply voltage (V_{CCI}) vary among the I/O standards (Figure 7-3).

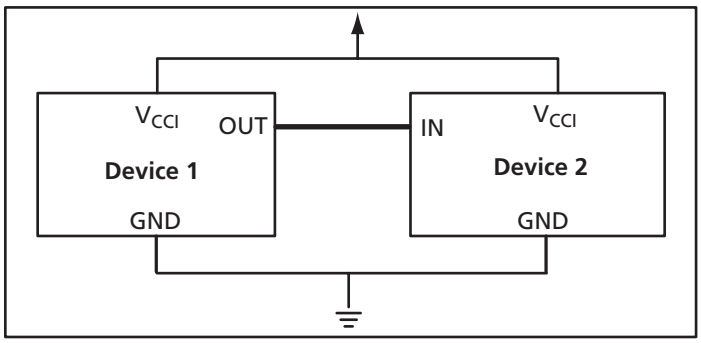


Figure 7-3 • **Single-Ended I/O Standard Topology**

The advantage of these standards is that a common ground can be used for multiple I/Os. This simplifies board layout and reduces system cost. Their low-edge-rate (*dv/dt*) data transmission causes less electromagnetic interference (EMI) on the board. However, they are not suitable for high-frequency (>200 MHz) switching due to noise impact and higher power consumption.

LVTTL (Low-Voltage TTL)

This is a general-purpose standard (EIA/JESD8-B) for 3.3 V applications. It uses an LVTTL input buffer and a push-pull output buffer. The LVTTL output buffer can have up to six different programmable drive strengths. Refer to "I/O Programmable Features" on page 7-7 for details. Refer to Table 7-13 on page 7-16 for details.

LVCMOS (Low-Voltage CMOS)

The low-power flash devices provide five voltage levels for LVCMOS: LVCMOS 3.3 V, LVCMOS 2.5 V, LVCMOS 1.8 V, LVCMOS 1.5 V, and LVCMOS 1.2 V. LVCMOS 3.3 V is an extension of the LVCMOS standard (JESD8-B–compliant) used for general-purpose 3.3 V applications. LVCMOS 2.5 V is an extension of the LVCMOS standard (JESD8-5–compliant) used for general-purpose 2.5 V applications.

LVCMOS 1.8 V is an extension of the LVCMOS standard (JESD8-7–compliant) used for general-purpose 1.8 V applications. LVCMOS 1.5 V is an extension of the LVCMOS standard (JESD8-11–compliant) used for general-purpose 1.5 V applications. LVCMOS 1.2V is an extension of the LVCMOS standard (JESD8-12A-compliant) used for general-purpose 1.2 V applications. The V_{CCI} values for these standards are 3.3 V, 2.5 V, 1.8 V, 1.5 V, and 1.2 V, respectively. All these versions use a 3.3 V–tolerant CMOS input buffer and a push-pull output buffer. Like LVTTL, the output buffer has up to six different programmable drive strengths (2, 4, 6, 8, 12, and 16 mA). Refer to "IGLOO PLUS Output Drive and Slew" on page 7-16 for details.

I/O Features

IGLOO PLUS devices support multiple I/O features that make board design easier. For example, an I/O feature like Schmitt Trigger in the 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 IGLOO PLUS 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 7-6 lists programmable I/O features and their ranges.

Table 7-6 • **Programmable I/O Features (user control via I/O Attribute Editor)**

Feature	Description	Range
Slew Control	Output slew rate	HIGH, LOW
Output Drive (mA)	Output drive strength	Depends on I/O type
Resistor Pull	Weak resistor pull circuit	Up, Down, None
Schmitt Trigger	Schmitt trigger for input only	ON, OFF

Hot-Swap Support

All devices in the IGLOO PLUS family are hot-swappable.

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 *Power-Up/-Down Behavior of Low-Power Flash Devices* for details on hot-swapping.

Hot-swapping 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 7-7 on page 7-8 to Table 7-10 on page 7-9. The I/Os also need to be configured in hot-insertion mode if hot-plugging compliance is required. IGLOO PLUS 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 7-7 • **Hot-Swap Level 1**

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

Table 7-8 • **Hot-Swap Level 2**

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

Table 7-9 • **Hot-Swap Level 3**

Description	Hot-swap while bus idle
Power Applied to Device	Yes
Bus State	Held idle (no ongoing I/O processes during insertion/removal)
Card Ground Connection	Reset must be maintained for 1 ms before, during, and after insertion/removal.
Device Circuitry Connected to Bus Pins	Must remain glitch-free during power-up or power-down
Example Application	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.
Compliance of IGLOO PLUS Devices	Compliant

Table 7-10 • **Hot-Swap Level 4**

Description	Hot-swap on an active bus
Power Applied to Device	Yes
Bus State	Bus may have active I/O processes ongoing, but device being inserted or removed must be idle.
Card Ground Connection	Reset must be maintained for 1 ms before, during, and after insertion/removal.
Device Circuitry Connected to Bus Pins	Must remain glitch-free during power-up or power-down
Example Application	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.
Compliance of IGLOO PLUS Devices	Compliant

For Level 3 and Level 4 compliance with the IGLOO PLUS devices, cards with two levels of staging should have the following sequence:

- Grounds
- Powers, I/Os, and other pins

Cold-Sparing Support

Cold-sparing refers to the ability of a device to leave system data undisturbed when the system is powered up, while the component itself is powered down, or when power supplies are floating.

Cold-sparing is supported on IGLOO PLUS devices only when the user provides resistors from each power supply to ground. The resistor value is calculated based on the decoupling capacitance on a given power supply. The RC constant should be greater than 3 µs.

To remove resistor current during operation, it is suggested that the resistor be disconnected (e.g., with an NMOS switch) from the power supply after the supply has reached its final value. Refer to the *Power-Up/-Down Behavior of Low-Power Flash Devices* chapter of the *ProASIC3* and *ProASIC3E* handbooks for details on cold-sparing.

Cold-sparing means that a subsystem with no power applied (usually a circuit board) is electrically connected to the system that is in operation. This means that all input buffers of the subsystem must present very high input impedance with no power applied so as not to disturb the operating portion of the system.

When targeting low-power applications, I/O cold-sparing may add additional current if a pin is configured with either a pull-up or pull-down resistor and driven in the opposite direction. A small static current is induced on each I/O pin when the pin is driven to a voltage opposite to the weak pull resistor. The current is equal to the voltage drop across the input pin divided by the pull resistor. Refer to the "Detailed I/O DC Characteristics" section of the appropriate family datasheet for the specific pull resistor value for the corresponding I/O standard.

For example, assuming an LVTTL 3.3 V input pin is configured with a weak pull-up resistor, a current will flow through the pull-up resistor if the input pin is driven LOW. For LVTTL 3.3 V, the pull-up resistor is ~45 kΩ, and the resulting current is equal to 3.3 V / 45 kΩ = 73 µA when the IO pin is driven LOW. This is true also when a weak pull-down is chosen and the input pin is driven HIGH. This current can be avoided by driving the input LOW when a weak pull-down resistor is used and driving it HIGH when a weak pull-up resistor is used.

This current draw can occur in the following cases:

- 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 PLUS devices are qualified to the Human Body Model (HBM) and the Charged Device Model (CDM).

Table 7-11 • **I/O Hot-Swap and 5 V Input Tolerance Capabilities in IGLOO PLUS Devices**

I/O Assignment	Clamp Diode	Hot Insertion	5 V Input Tolerance	Input Buffer	Output Buffer
3.3 V LVTTL/LVCMOS	No	Yes	Yes*	Enabled/Disabled	
LVCMOS 2.5 V	No	Yes	No	Enabled/Disabled	
LVCMOS 1.8 V	No	Yes	No	Enabled/Disabled	
LVCMOS 1.5 V	No	Yes	No	Enabled/Disabled	
LVCMOS 1.2 V	No	Yes	No	Enabled/Disabled	

* *Can be implemented with an external IDT bus switch, resistor divider, or Zener with resistor.*

5 V Input and Output Tolerance

IGLOO PLUS devices can be made 5 V–input–tolerant for certain I/O standards by using external level shifting techniques. 5 V output compliance can be achieved using certain I/O standards.

Table 7-4 on page 7-4 shows the I/O standards that support 5 V input tolerance. Only 3.3 V LVTTL/LVCMOS standards support 5 V output tolerance.

5 V Input Tolerance

I/Os can support 5 V input tolerance when LVTTL 3.3 V, LVCMOS 3.3 V, LVCMOS 2.5 V, LVCMOS 1.8 V, LVCMOS 1.5 V, and LVCMOS 1.2 V configurations are used (see Table 7-11). There are three recommended solutions for achieving 5 V receiver tolerance (see Figure 7-4 on page 7-12 to Figure 7-6 on page 7-13 for details of board and macro setups). All the solutions meet a common requirement of limiting the voltage at the input to 3.6 V or less. In fact, the I/O absolute maximum voltage rating is 3.6 V, and any voltage above 3.6 V may cause long-term gate oxide failures.

Solution 1

The board-level design must ensure that the reflected waveform at the pad does not exceed the limits provided in the recommended operating conditions in the datasheet. This is a requirement to ensure long-term reliability.

This solution requires two board resistors, as demonstrated in Figure 7-4 on page 7-12. Here are some examples of possible resistor values (based on a simplified simulation model with no line effects and 10 Ω transmitter output resistance, where Rtx_out_high = (V_{CCI} – V_{OH}) / I_{OH} and Rtx_out_low = V_{OL} / I_{OL}).

Example 1 (high speed, high current):

Rtx_out_high = Rtx_out_low = 10 Ω

R1 = 36 Ω (±5%), P(r1)min = 0.069 Ω

R2 = 82 Ω (±5%), P(r2)min = 0.158 Ω

Imax_tx = 5.5 V / (82 × 0.95 + 36 × 0.95 + 10) = 45.04 mA

t_{RISE} = t_{FALL} = 0.85 ns at C_pad_load = 10 pF (includes up to 25% safety margin)

t_{RISE} = t_{FALL} = 4 ns at C_pad_load = 50 pF (includes up to 25% safety margin)

Example 2 (low–medium speed, medium current):

Rtx_out_high = Rtx_out_low = 10 Ω

R1 = 220 Ω (±5%), P(r1)min = 0.018 Ω

R2 = 390 Ω (±5%), P(r2)min = 0.032 Ω

Imax_tx = 5.5 V / (220 × 0.95 + 390 × 0.95 + 10) = 9.17 mA

t_{RISE} = t_{FALL} = 4 ns at C_pad_load = 10 pF (includes up to 25% safety margin)

t_{RISE} = t_{FALL} = 20 ns at C_pad_load = 50 pF (includes up to 25% safety margin)

Other values of resistors are also allowed as long as the resistors are sized appropriately to limit the voltage at the receiving end to 2.5 V < Vin (rx) < 3.6 V when the transmitter sends a logic 1. This range of Vin_dc(rx) must be assured for any combination of transmitter supply (5 V ± 0.5 V), transmitter output resistance, and board resistor tolerances.

Temporary overshoots are allowed according to the overshoot and undershoot table in the datasheet.

Solution 1

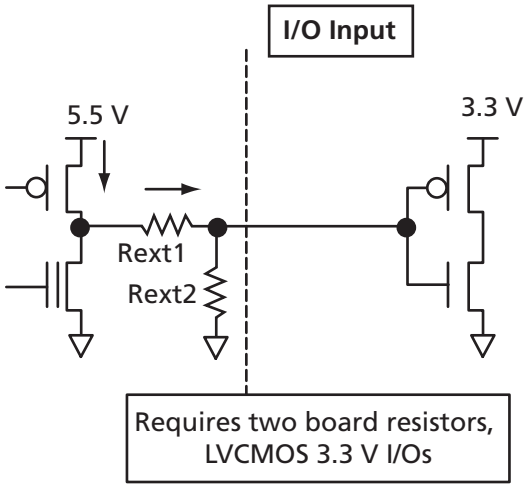


Figure 7-4 • **Solution 1**

Solution 2

This solution requires one board resistor and one Zener 3.3 V diode, as demonstrated in Figure 7-5.

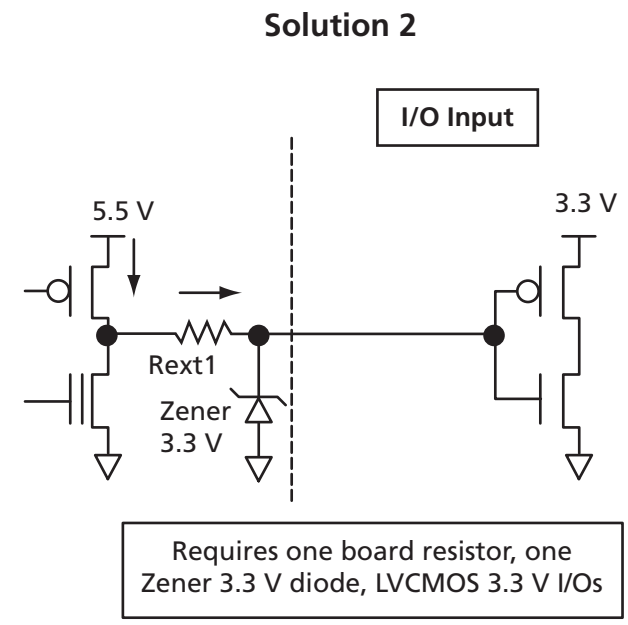


Figure 7-5 • **Solution 2**

Solution 3

This solution requires a bus switch on the board, as demonstrated in Figure 7-6.

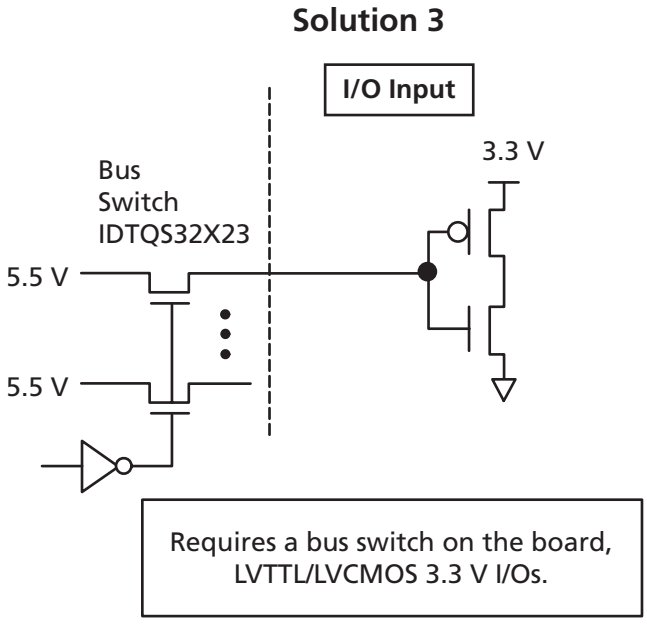


Figure 7-6 • **Solution 3**

Table 7-12 • **Comparison Table for 5 V–Compliant Receiver Solutions**

Solution	Board Components	Speed	Current Limitations
1	Two resistors	Low to High[1]	Limited by transmitter's drive strength
2	Resistor and Zener 3.3 V	Medium	Limited by transmitter's drive strength
3	Bus switch	High	N/A

Notes:

1. *Speed and current consumption increase as the board resistance values decrease.*
2. *Resistor values ensure I/O diode long-term reliability.*
3. *At 70°C, customers could still use 420 Ω on every I/O.*
4. *At 85°C, a 5 V solution on every other I/O is permitted, since the resistance is lower (150 Ω) and the current is higher. Also, the designer can still use 420 Ω and use the solution on every I/O.*
5. *At 100°C, the 5 V solution on every I/O is permitted, since 420 Ω are used to limit the current to 5.9 mA.*

5 V Output Tolerance

IGLOO PLUS I/Os must be set to 3.3 V LVTTL or 3.3 V LVCMOS mode to reliably drive 5 V TTL receivers. It is also critical that there be NO external I/O pull-up resistor to 5 V, since this resistor would pull the I/O pad voltage beyond the 3.6 V absolute maximum value and consequently cause damage to the I/O.

When set to 3.3 V LVTTL or 3.3 V LVCMOS mode, the I/Os can directly drive signals into 5 V TTL receivers. In fact, V_{OL} = 0.4 V and V_{OH} = 2.4 V in both 3.3 V LVTTL and 3.3 V LVCMOS modes exceeds the V_{IL} = 0.8 V and V_{IH} = 2 V level requirements of 5 V TTL receivers. Therefore, level 1 and level 0 will be recognized correctly by 5 V TTL receivers.

Schmitt Trigger

A Schmitt trigger is a buffer used to convert a slow or noisy input signal into a clean one before passing it to the FPGA. Using Schmitt trigger buffers guarantees a fast, noise-free input signal to the FPGA.

IGLOO PLUS devices have Schmitt triggers built into their I/O circuitry. Schmitt Trigger is available on all I/O configurations.

This feature can be implemented by using a Physical Design Constraints (PDC) command (Table 7-4 on page 7-4) or by selecting a check box in the I/O Attribute Editor in Designer. The check box is cleared by default.

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 in the *I/O Software Control in Low-Power Flash Devices* section of the handbook).

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 function:
   - Both the Output and Output Enable registers must not have an E pin (clock enable).

Weak Pull-Up and Weak Pull-Down Resistors

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

For low-power applications, Configuration of the pull-up or pull-down of the I/O can be used to set the I/O to a known state while the device is in Flash*Freeze mode. Refer to *Flash*Freeze Technology and Low-Power Modes in IGLOO and ProASIC3L Devices* for more information.

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

Output Slew Rate Control

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

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

The slew rate can be implemented by using a PDC command (Table 7-4 on page 7-4), setting "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."

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

Output Drive

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

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

Refer to Table 7-9 on page 7-9 for more information about the slew rate and drive strength specification for LVTTL/LVCMOS 3.3 V, LVCMOS 2.5 V, LVCMOS 1.8 V, LVCMOS 1.5 V, and LVCMOS 1.2 V output buffers.

Table 7-13 • **IGLOO PLUS Output Drive and Slew**

I/O Standards	2 mA	4 mA	6 mA	8 mA	12 mA	16 mA	Slew	
LVTTL / LVCMOS 3.3 V	✓	✓	✓	✓	✓	✓	High	Low
LVCMOS 2.5 V	✓	✓	✓	✓	✓	–	High	Low
LVCMOS 1.8 V	✓	✓	✓	✓	–	–	High	Low
LVCMOS 1.5 V	✓	✓	–	–	–	–	High	Low
LVCMOS 1.2 V	✓	–	–	–	–	–	High	Low

Simultaneously Switching Outputs (SSOs) and Printed Circuit Board Layout

Each I/O voltage bank has a separate ground and power plane for input and output circuits. This isolation is necessary to minimize simultaneous switching noise from the input and output (SSI and SSO). The switching noise (ground bounce and power bounce) is generated by the output buffers and transferred into input buffer circuits, and vice versa.

SSOs can cause signal integrity problems on adjacent signals that are not part of the SSO bus. Both inductive and capacitive coupling parasitics of bond wires inside packages and of traces on PCBs will transfer noise from SSO busses onto signals adjacent to those busses. Additionally, SSOs can produce ground bounce noise and V_{CCI} dip noise. These two noise types are caused by rapidly changing currents through GND and V_{CCI} package pin inductances during switching activities (EQ 7-1 and EQ 7-2).

$$\text{Ground bounce noise voltage} = L(GND) \times di/dt \quad \textit{EQ 7-1}$$

$$V_{CCI} \text{ dip noise voltage} = L(V_{CCI}) \times di/dt \quad \textit{EQ 7-2}$$

Any group of four or more input pins switching on the same clock edge is considered an SSO bus. The shielding should be done both on the board and inside the package unless otherwise described.

In-package shielding can be achieved in several ways; the required shielding will vary depending on whether pins next to the SSO bus are LVTTL/LVCMOS inputs, LVTTL/LVCMOS outputs, or GTL/SSTL/HSTL/LVDS/LVPECL inputs and outputs. Board traces in the vicinity of the SSO bus have to be adequately shielded from mutual coupling and inductive noise that can be generated by the SSO bus. Also, noise generated by the SSO bus needs to be reduced inside the package.

PCBs perform an important function in feeding stable supply voltages to the IC and, at the same time, maintaining signal integrity between devices.

Key issues that need to be considered are as follows:

- Power and ground plane design and decoupling network design
- Transmission line reflections and terminations

For extensive data per package on the SSO and PCB issues, refer to *ProASIC3/E SSO and Pin Placement and Guidelines* chapter of the handbook.

I/O Software Support

In Actel's Libero IDE software, default settings have been defined for the various I/O standards supported. Changes can be made to the default settings via the use of attributes; however, not all I/O attributes are applicable for all I/O standards.

Table 7-14 • **IGLOO PLUS I/O Attributes vs. I/O Standard Applications**

I/O Standard	SLEW (output only)	OUT_DRIVE (output only)	RES_PULL	OUT_LOAD (output only)	Schmitt Trigger	Hold State	Combine Register
LVTTL/ LVCMOS3.3	✓	✓ (12)	✓	✓	✓	✓	✓
LVCMOS2.5	✓	✓ (12)	✓	✓	✓	✓	✓
LVCMOS1.8	✓	✓ (8)	✓	✓	✓	✓	✓
LVCMOS1.5	✓	✓ (4)	✓	✓	✓	✓	✓
LVCMOS1.2	✓	✓ (2)	✓	✓	✓	✓	✓
Software Defaults	HIGH	Refer to the numbers in parentheses in the above cells.	None	5 pF	Off	Off	No

User I/O Naming Convention

Due to the comprehensive and flexible nature of IGLOO PLUS I/Os, a naming scheme is used to show the details of each I/O (Figure 7-7). The name identifies to which I/O bank it belongs.

I/O Nomenclature = FF/Gmn/IOuxwBy

Gmn is only used for I/Os that also have CCC access—i.e., global pins.

FF = Indicates the I/O dedicated for the Flash*Freeze mode activation pin

G = Global

m = Global pin location associated with each CCC on the device: A (northwest corner), B (northeast corner), C (east middle), D (southeast corner), E (southwest corner), and F (west middle)

n = Global input MUX and pin number of the associated Global location m—either A0, A1, A2, B0, B1, B2, C0, C1, or C2. Refer to *Global Resources in Actel Low-Power Flash Devices* for information about the three input pins per clock source MUX at CCC location m.

u = I/O pair number in the bank, starting at 00 from the northwest I/O bank and proceeding in a clockwise direction

x = R (Regular—single-ended) for the I/Os that support single-ended standards.

w = S (Single-Ended)

B = Bank

y = Bank number (0–3). The Bank number starts at 0 from the northwest I/O bank and proceeds in a clockwise direction.

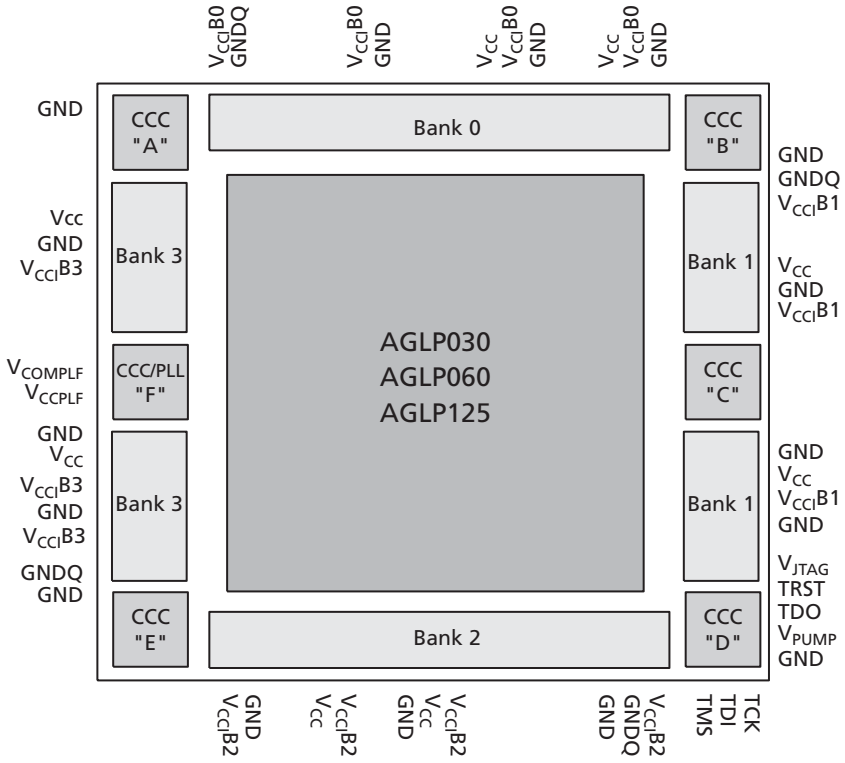


Figure 7-7 • **Naming Conventions of IGLOO PLUS Devices – Top View**

Board-Level Considerations

Low-power flash devices have robust I/O features that can help in reducing board-level components. The devices offer single-chip solutions, which makes the board layout simpler and more immune to signal integrity issues. Although, in many cases, these devices resolve board-level issues, special attention should always be given to overall signal integrity. This section covers important board-level considerations to facilitate optimum device performance.

Termination

Proper termination of all signals is essential for good signal quality. Nonterminated signals, especially clock signals, can cause malfunctioning of the device.

For general termination guidelines, refer to the *Board-Level Considerations* application note for Actel FPGAs. Also refer to *Pin Descriptions* for termination requirements for specific pins.

Low-power flash I/Os are equipped with on-chip pull-up/-down resistors. The user can enable these resistors by instantiating them either in the top level of the design (refer to the *IGLOO, Fusion, and ProASIC3 Macro Library Guide* for the available I/O macros with pull-up/-down) or in the I/O Attribute Editor in Designer if generic input or output buffers are instantiated in the top level. Unused I/O pins are configured as inputs with pull-up resistors.

As mentioned earlier, low-power flash devices have multiple programmable drive strengths, and the user can eliminate unwanted overshoot and undershoot by adjusting the drive strengths.

Power-Up Behavior

Low-power flash devices are power-up/-down friendly; i.e., no particular sequencing is required for power-up and power-down. This eliminates extra board components for power-up sequencing, such as a power-up sequencer.

During power-up, all I/Os are tristated, irrespective of I/O macro type (input buffers, output buffers, I/O buffers with weak pull-ups or weak pull-downs, etc.). Once I/Os become activated, they are set to the user-selected I/O macros. Refer to the *Power-Up/-Down Behavior of Low-Power Flash Devices* chapter of the *ProASIC3* and *ProASIC3E* handbooks for details.

Drive Strength

Low-power flash devices have up to seven programmable output drive strengths. The user can select the drive strength of a particular output in the I/O Attribute Editor or can instantiate a specialized I/O macro, such as OUTBUF_S_12 (slew = low, out_drive = 12 mA).

The maximum available drive strength is 16 mA per I/O. Though no I/O should be forced to source or sink more than 16 mA indefinitely, I/Os may handle a higher amount of current (refer to the device IBIS model for maximum source/sink current) during signal transition (AC current). Every device package has its own power dissipation limit; hence, power calculation must be performed accurately to determine how much current can be tolerated per I/O within that limit.

I/O Interfacing

Low-power flash devices are 5 V–input– and 5 V–output–tolerant without adding any extra circuitry. Along with other low-voltage I/O macros, this 5 V tolerance makes these devices suitable for many types of board component interfacing.

Table 7-15 shows some high-level interfacing examples using low-power flash devices.

Table 7-15 • **IGLOO PLUS High-Level Interface**

Interface	Clock		I/O			
	Type	Frequency	Type	Signals In	Signals Out	Data I/O
GM	Src Sync	125 MHz	LVTTL	8	8	125 Mbps
TBI	Src Sync	125 MHz	LVTTL	10	10	125 Mbps

Conclusion

IGLOO PLUS support for multiple I/O standards minimizes board-level components and makes possible a wide variety of applications. The Actel Designer software, integrated with Actel Libero IDE, 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 IGLOO PLUS 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

Handbook Documents

Board-Level Considerations
http://www.actel.com/documents/BoardLevelCons_AN.pdf
DDR for Actel's Low-Power Flash Devices
http://www.actel.com/documents/LPD_DDR_HBs.pdf
*Flash*Freeze Technology and Low-Power Modes in IGLOO and ProASIC3L Devices*
http://www.actel.com/documents/LPD_FlashFreeze_HBs.pdf
Global Resources in Actel Low-Power Flash Devices
http://www.actel.com/documents/LPD_Global_HBs.pdf
Pin Descriptions
http://www.actel.com/documents/LPD_PinDescriptions_HBs.pdf
Power-Up/-Down Behavior of Low-Power Flash Devices
http://www.actel.com/documents/LPD_PowerUp_HBs.pdf
ProASIC3/E SSO and Pin Placement and Guidelines
http://www.actel.com/documents/PA3_E_SSO_HBs.pdf

User's Guides

Actel Libero IDE User's Guide
http://www.actel.com/documents/libero_ug.pdf
IGLOO, Fusion, and ProASIC3 Macro Library Guide
http://www.actel.com/documents/pa3_libguide_ug.pdf
SmartGen Core Reference Guide
http://www.actel.com/documents/genguide_ug.pdf

Part Number and Revision Date

This document contains content extracted from the Device Architecture section of the datasheet, combined with content previously published as an application note describing features and functions of the device. To improve usability for customers, the device architecture information has now been combined with usage information, to reduce duplication and possible inconsistencies in published information. No technical changes were made to the datasheet content unless explicitly listed. Changes to the application note content were made only to be consistent with existing datasheet information.

Part Number 51700094-025-2

Revised October 2008

List of Changes

The following table lists critical changes that were made in the current version of the chapter.

Previous Version	Changes in Current Version (v1.2)	Page
v1.1 (June 2008)	The "Low-Power Flash Device I/O Support" section was revised.	7-2
v1.0 (March 2008)	The following changes were made to the family descriptions in Table 7-1 · Low-Power Flash Families: • ProASIC3L was updated to include 1.5 V. • The number of PLLs for ProASIC3E was changed from five to six.	7-2

8 – I/O Software Control in Low-Power Flash Devices

Actel Fusion,® IGLOO,® and ProASIC®3 I/Os provide more design flexibility, allowing the user to control specific features by enabling certain I/O standards. Some features are selectable only for certain I/O standards, whereas others are available for all I/O standards. For example, slew control is not supported by differential I/O standards. Conversely, I/O register combining is supported by all I/O standards. For detailed information about which I/O standards and features are available on each device and each I/O type, refer to the I/O Structures section of the handbook for the device you are using.

Figure 8-1 shows the various points in the software design flow where a user can provide input or control of the I/O selection and parameters. A detailed description is provided throughout this document.

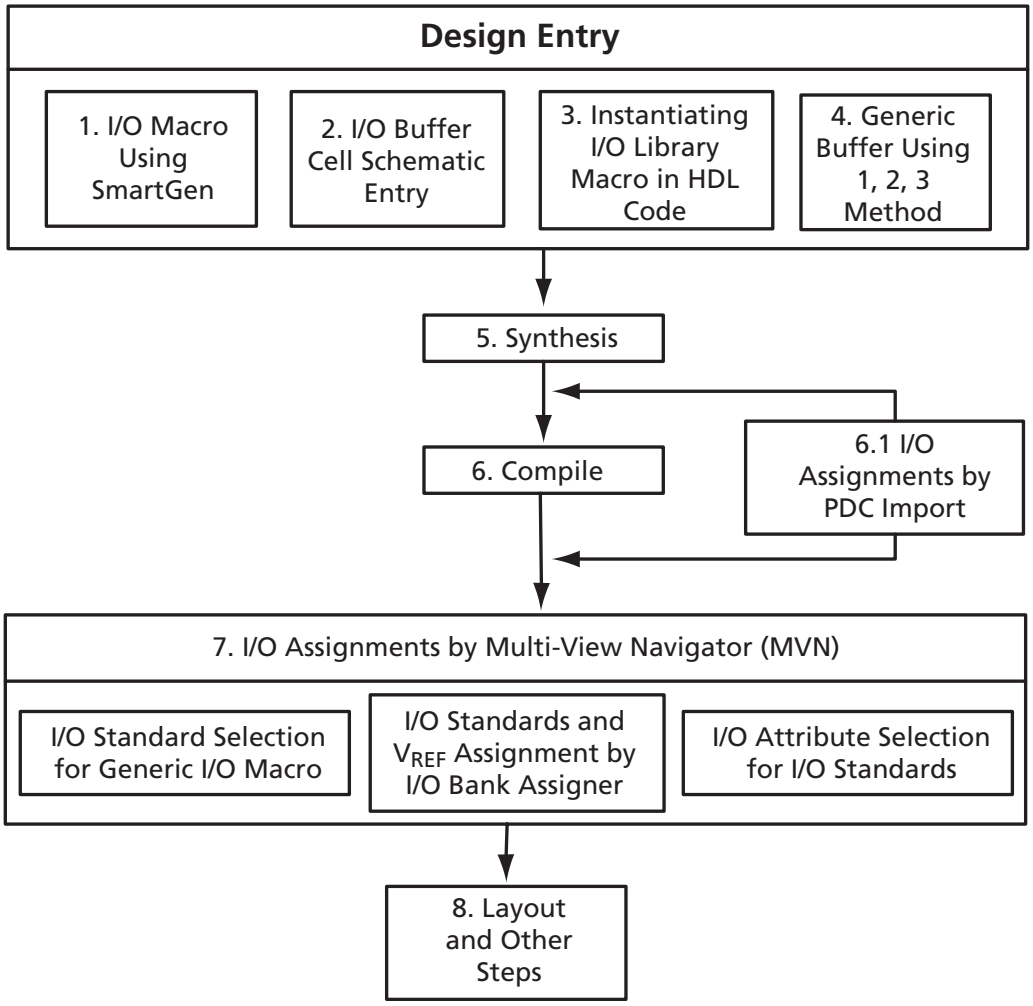


Figure 8-1 • **User I/O Assignment Flow Chart**

Low-Power Flash Families I/O Support

The low-power flash families listed in Table 8-1 support I/Os and the functions described in this document.

Table 8-1 • **Low-Power Flash Families**

Product Line	Family*	Description
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
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 PLUS	IGLOO FPGAs with enhanced I/O capabilities
ProASIC3	ProASIC3	Low-power, high-performance 1.5 V FPGAs
	ProASIC3E	Higher density ProASIC3 FPGAs with six PLLs and additional I/O standards
	ProASIC3L	ProASIC3 FPGAs supporting 1.2 V to 1.5 V with Flash*Freeze technology
	Automotive ProASIC3	ProASIC3 FPGAs qualified for automotive applications
	Military ProASIC3/EL	Military temperature A3PE600L, A3P1000, and A3PE3000L
	RT ProASIC3	Radiation-tolerant RT3PE600L and RT3PE3000L

*Note: *The family names link to the appropriate datasheet, including product brief, DC and switching characteristics, and packaging information.*

IGLOO Terminology

In documentation, the terms IGLOO families and IGLOO devices refer to all of the IGLOO products as listed in Table 8-1. Where the information applies to only one family or limited devices, these exclusions will be explicitly stated.

ProASIC3 Terminology

In documentation, the terms ProASIC3 families and ProASIC3 devices refer to all of the ProASIC3 families as listed in Table 8-1. Where the information applies to only one family or limited devices, these exclusions will be explicitly stated.

To further understand the differences between the IGLOO and ProASIC3 families, refer to the *Industry's Lowest Power FPGAs Portfolio*.

Software-Controlled I/O Attributes

Users may modify these programmable I/O attributes using the I/O Attribute Editor. Modifying an I/O attribute may result in a change of state in Designer. Table 8-2 details which steps have to be re-run as a function of modified I/O attribute.

Table 8-2 • **Designer State (resulting from I/O attribute modification)**

	Designer States				
I/O Attribute	**Compile**	**Layout**	**Fuse**	**Timing**	**Power**
Slew Control	No	No	Yes	Yes	Yes
Output Drive (mA)	No	No	Yes	Yes	Yes
Skew Control	No	No	Yes	Yes	Yes
Resistor Pull	No	No	Yes	Yes	Yes
Input Delay	No	No	Yes	Yes	Yes
Schmitt Trigger	No	No	Yes	Yes	Yes
OUT_LOAD	No	No	No	Yes	Yes
COMBINE_REGISTER	Yes	Yes	N/A	N/A	N/A

Notes:

1. *No = Remains the same, Yes = Re-run the step, N/A = Not applicable*
2. *Skew control and input delay do not apply to IGLOO PLUS.*

Implementing I/Os in Actel Software

Actel Libero® Integrated Design Environment (IDE) is integrated with design entry tools such as the SmartGen macro builder, the ViewDraw schematic entry tool, and an HDL editor. It is also integrated with the synthesis and Designer tools. In this section, all necessary steps to implement the I/Os are discussed.

Design Entry

There are three ways to implement I/Os in a design:

1. Use the SmartGen macro builder to configure I/Os by generating specific I/O library macros and then instantiating them in top-level code. This is especially useful when creating I/O bus structures.
2. Use an I/O buffer cell in a schematic design.
3. Manually instantiate specific I/O macros in the top-level code.

If technology-specific macros, such as INBUF_LVCMOS33 and OUTBUF_PCI, are used in the HDL code or schematic, the user will not be able to change the I/O standard later on in Designer. If generic I/O macros are used, such as INBUF, OUTBUF, TRIBUF, CLKBUF, and BIBUF, the user can change the I/O standard using the Designer I/O Attribute Editor tool.

Using SmartGen for I/O Configuration

The SmartGen tool in Libero IDE provides a GUI-based method of configuring the I/O attributes. The user can select certain I/O attributes while configuring the I/O macro in SmartGen. The steps to configure an I/O macro with specific I/O attributes are as follows:

1. Open Libero IDE.
2. On the left hand side of the Catalog View, select **I/O**, as shown in Figure 8-2.

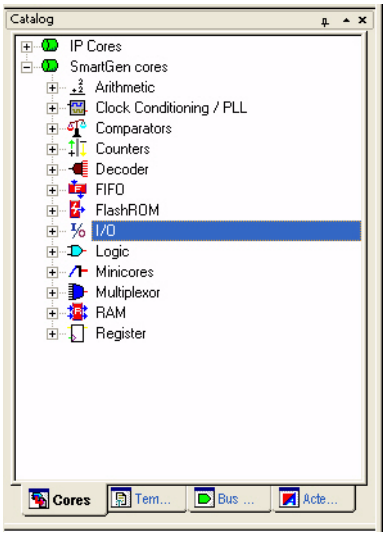


Figure 8-2 • **SmartGen Catalog**

3. Expand the I/O section and double-click one of the options (Figure 8-3).

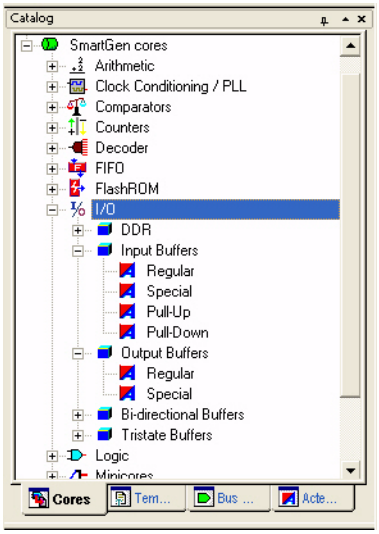


Figure 8-3 • **Expanded I/O Section**

4. Double-click any of the varieties. The I/O Create Core window opens (Figure 8-4).

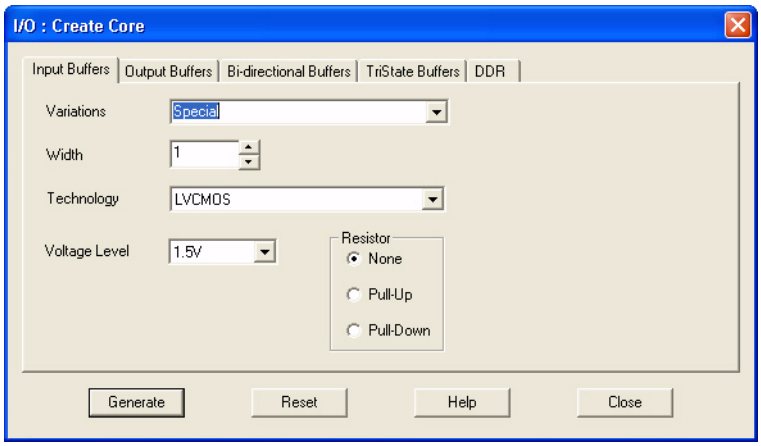


Figure 8-4 • **I/O Create Core Window**

As seen in Figure 8-4, there are five tabs to configure the I/O macro: Input Buffers, Output Buffers, Bidirectional Buffers, Tristate Buffers, and DDR.

Input 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, Voltage Level, and Resistor Pull-Up/-Down options (see Figure 8-4). All the I/O standards and supply voltages (V_{CCI}) supported for the device family are available for selection.

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.

5. Once the desired configuration is selected, click the **Generate** button. The Generate Core window opens (Figure 8-5).
6. Enter a name for the macro. Click **OK**. The core will be generated and saved to the appropriate location within the project files (Figure 8-6 on page 8-7).

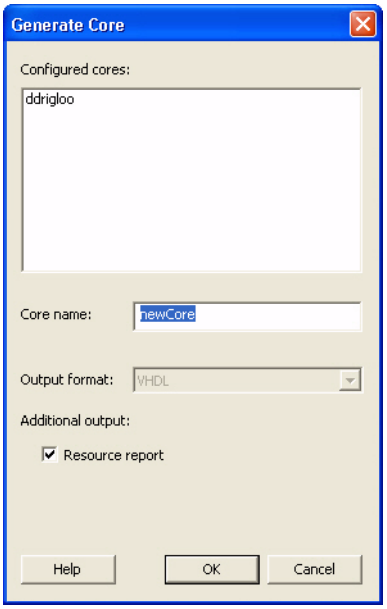


Figure 8-5 • **Generate Core Window**

7. 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:

Rules for the DDR I/O Function

- The fanout between an I/O pin (D or Y) and a DDR (DDR_REG or DDR_OUT) macro must be equal to one for the combining to happen on that pin.
- If a DDR_REG macro and a DDR_OUT macro are combined on the same bidirectional I/O, they must share the same clear signal.
- Registers will not be combined in an I/O in the presence of DDR combining on the same I/O.

Using the I/O Buffer Schematic Cell

Libero IDE includes the ViewDraw schematic entry tool. Using ViewDraw, the user can insert any supported I/O buffer cell in the top-level schematic. Figure 8-6 shows a top-level schematic with different I/O buffer cells. When synthesized, the netlist will contain the same I/O macro.

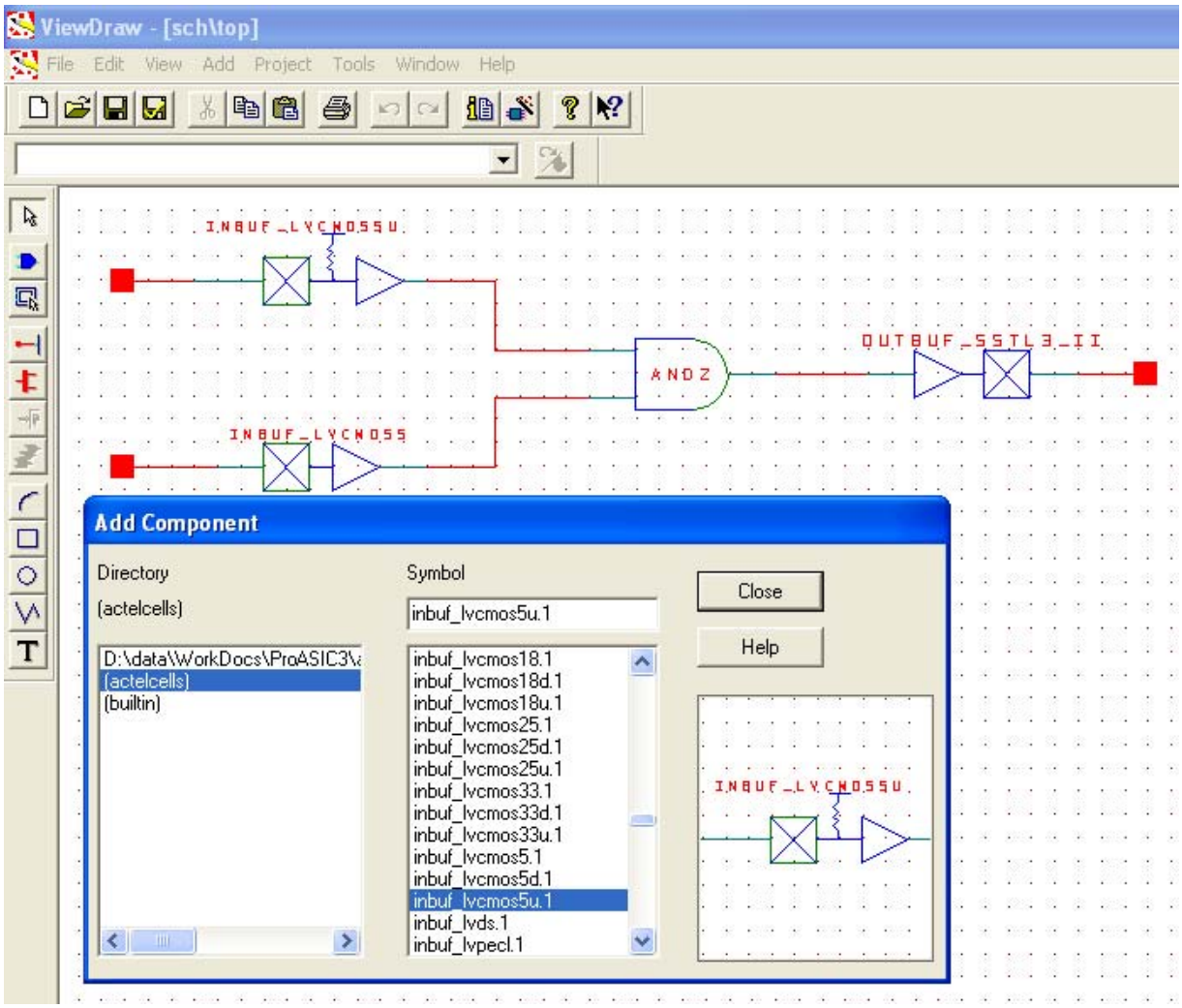


Figure 8-6 • **I/O Buffer Schematic Cell Usage**

Instantiating in HDL code

All the supported I/O macros can be instantiated in the top-level HDL code (refer to the *IGLOO, Fusion, and ProASIC3 Macro Library Guide* for a detailed list of all I/O macros). The following is an example:

```
library ieee;
use ieee.std_logic_1164.all;
library proasic3e;

entity TOP is
  port(IN2, IN1 : in std_logic; OUT1 : out std_logic);
end TOP;

architecture DEF_ARCH of TOP is

  component INBUF_LVCMOS5U
    port(PAD : in std_logic := 'U'; Y : out std_logic);
  end component;

  component INBUF_LVCMOS5
    port(PAD : in std_logic := 'U'; Y : out std_logic);
  end component;

  component OUTBUF_SSTL3_II
    port(D : in std_logic := 'U'; PAD : out std_logic);
  end component;

  Other component …..

signal x, y, z…….other signals : std_logic;

begin

  I1 : INBUF_LVCMOS5U
    port map(PAD => IN1, Y =>x);
  I2 : INBUF_LVCMOS5
    port map(PAD => IN2, Y => y);
  I3 : OUTBUF_SSTL3_II
    port map(D => z, PAD => OUT1);

  other port mapping…

end DEF_ARCH;
```

Synthesizing the Design

Libero IDE integrates with the Synplify® synthesis tool. Other synthesis tools can also be used with Libero IDE. Refer to the *Actel Libero IDE User's Guide* or Libero IDE online help for details on how to set up the Libero IDE tool profile with synthesis tools from other vendors.

During synthesis, the following rules apply:

- Generic macros:
  - Users can instantiate generic INBUF, OUTBUF, TRIBUF, and BIBUF macros.
  - Synthesis will automatically infer generic I/O macros.
  - The default I/O technology for these macros is LVTTL.
  - Users will need to use the I/O Attribute Editor in Designer to change the default I/O standard if needed (see Figure 8-7 on page 8-9).
- Technology-specific I/O macros:
  - Technology-specific I/O macros, such as INBUF_LVCMO25 and OUTBUF_GTL25, can be instantiated in the design. Synthesis will infer these I/O macros in the netlist.

– The I/O standard of technology-specific I/O macros cannot be changed in the I/O Attribute Editor (see Figure 8-7).
– The user MUST instantiate differential I/O macros (LVDS/LVPECL) in the design. This is the only way to use these standards in the design.
– To implement the DDR I/O function, the user must instantiate a DDR_REG or DDR_OUT macro. This is the only way to use a DDR macro in the design.

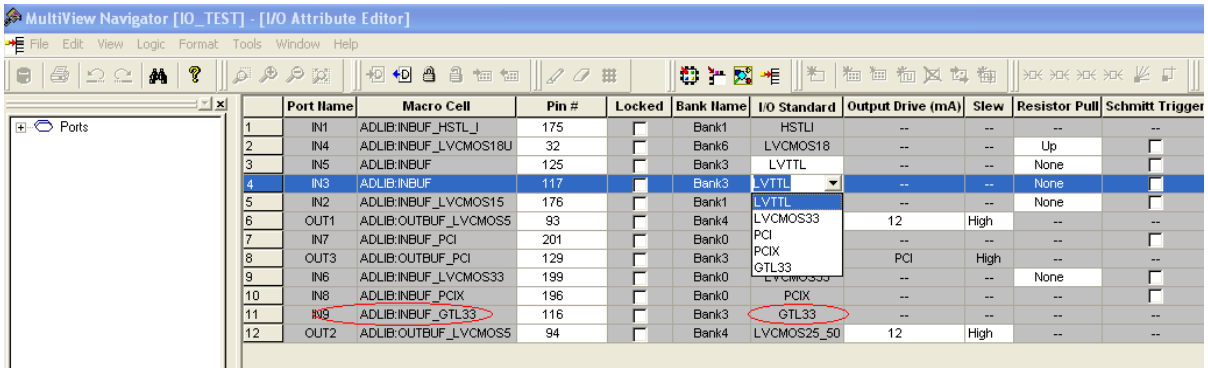


Figure 8-7 • **Assigning a Different I/O Standard to the Generic I/O Macro**

Performing Place-and-Route on the Design

The netlist created by the synthesis tool should now be imported into Designer and compiled. During Compile, the user can specify the I/O placement and attributes by importing the PDC file. The user can also specify the I/O placement and attributes using ChipPlanner and the I/O Attribute Editor, under MVN.

Defining I/O Assignments in the PDC File

A PDC file is a Tcl script file specifying physical constraints. This file can be imported to and exported from Designer.

Table 8-3 shows I/O assignment constraints supported in the PDC file.

Table 8-3 • **PDC I/O Constraints**

Command	Action	Example	Comment
I/O Banks Setting Constraints			

set_iobank	Sets the I/O supply voltage, V_{CCI}, and the input reference voltage, V_{REF}, for the specified I/O bank.	`set_iobank bankname [-vcci vcci_voltage] [-vref vref_voltage]`
		
		`set_iobank Bank7 -vcci 1.50 -vref 0.75`
			Must use in case of mixed I/O voltage (V_{CCI}) design

set_vref	Assigns a V_{REF} pin to a bank.	`set_vref -bank [bankname] [pinnum]`
		
		`set_vref -bank Bank0 685 704 723 742 761`
			Must use if voltage-referenced I/Os are used

Note: Refer to the Actel Libero IDE User's Guide for detailed rules on PDC naming and syntax conventions.

Table 8-3 • **PDC I/O Constraints (continued)**

Command	Action	Example	Comment

set_vref_defaults	Sets the default V_{REF} pins for the specified bank. This command is ignored if the bank does not need a V_{REF} pin.	`set_vref_defaults bankname`
		
		`set_vref_defaults bank2`	

I/O Attribute Constraint			

set_io	Sets the attributes of an I/O	`set_io portname`
		`[-pinname value]`
		`[-fixed value]`
		`[-iostd value]`
		`[-out_drive value]`
		`[-slew value]`
		`[-res_pull value]`
		`[-schmitt_trigger value]`
		`[-in_delay value]`
		`[-skew value]`
		`[-out_load value]`
		`[-register value]`
		
		`set_io IN2 -pinname 28`
		`-fixed yes -iostd LVCMOS15`
		`-out_drive 12 -slew high`
		`-RES_PULL None`
		`-SCHMITT_TRIGGER Off`
		`-IN_DELAY Off –skew off`
		`-REGISTER No`
			If the I/O macro is generic (e.g., INBUF) or technology-specific (INBUF_LVCMOS25), then all I/O attributes can be assigned using this constraint.
			If netlist has an I/O macro that specifies one of its attributes, that attribute cannot be changed using this constraint, though other attributes can be changed.
			Example: OUTBUF_S_24 (low slew, output drive 24 mA)
			Slew and output drive cannot be changed.

I/O Region Placement Constraints			

define_region	Defines either a rectangular region or a rectilinear region	`define_region`
		`-name [region_name]`
		`-type [region_type] x1 y1 x2 y2`
		
		`define_region -name test`
		`-type inclusive 0 15 2 29`
			If any number of I/Os must be assigned to a particular I/O region, such a region can be created with this constraint.

assign_region	Assigns a set of macros to a specified region	`assign_region [region name]`
		`[macro_name...]`
		
		`assign_region test U12`
			This constraint assigns I/O macros to the I/O regions. When assigning an I/O macro, PDC naming conventions must be followed if the macro name contains special characters; e.g., if the macro name is \\$1I19\\, the correct use of escape characters is \\\\\$1I19\\\\.

Note: Refer to the Actel Libero IDE User's Guide *for detailed rules on PDC naming and syntax conventions.*

Compiling the Design

During Compile, a PDC I/O constraint file can be imported along with the netlist file. If only the netlist file is compiled, certain I/O assignments need to be completed before proceeding to Layout. All constraints that can be entered in PDC can also be entered using ChipPlanner, I/O Attribute Editor, and PinEditor.

There are certain rules that must be followed in implementing I/O register combining and the I/O DDR macro (refer to the I/O Registers section of the handbook for the device that you are using and the "DDR" section on page 8-6 for details). Provided these rules are met, the user can enable or disable I/O register combining by using the PDC command `set_io portname –register yes|no` in the I/O Attribute Editor or selecting a check box in the Compile Options dialog box (see Figure 8-8). The Compile Options dialog box appears when the design is compiled for the first time. It can also be accessed by choosing **Options** > **Compile** during successive runs. I/O register combining is off by default. The PDC command overrides the setting in the Compile Options dialog box.

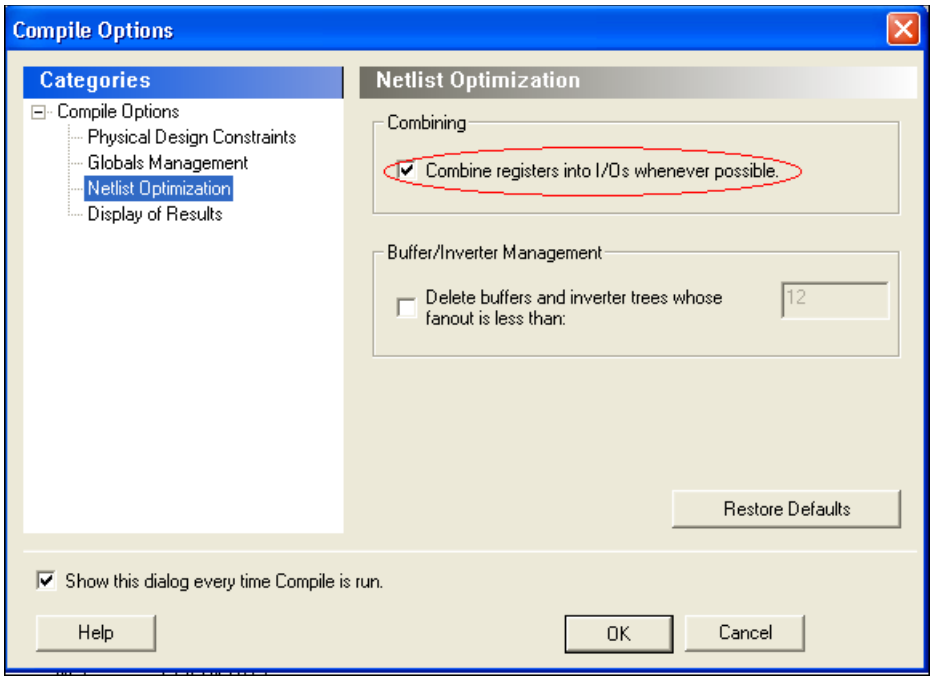


Figure 8-8 • **Setting Register Combining During Compile**

Understanding the Compile Report

The I/O bank report is generated during Compile and displayed in the log window. This report lists the I/O assignments necessary before Layout can proceed.

When Designer is started, the I/O Bank Assigner tool is run automatically if the Layout command is executed. The I/O Bank Assigner takes care of the necessary I/O assignments. However, these assignments can also be made manually with MVN or by importing the PDC file. Refer to the "Assigning Technologies and V_{REF} to I/O Banks" section on page 8-14 for further description.

The I/O bank report can also be extracted from Designer by choosing **Tools** > **Report** and setting the Report Type to **IOBank**.

This report has the following tables: I/O Function, I/O Technology, I/O Bank Resource Usage, and I/O Voltage Usage. This report is useful if the user wants to do I/O assignments manually.

I/O Function

Figure 8-9 shows an example of the I/O Function table included in the I/O bank report:

I/O Function:

Type	w/o register	w/ register	w/ DDR register
Input I/O	7	0	1
Output I/O	1	1	0
Bidirectional I/O	0	0	0
Differential Input I/O Pairs	0	0	0
Differential Output I/O Pairs	0	0	1

Figure 8-9 • **I/O Function Table**

This table lists the number of input I/Os, output I/Os, bidirectional I/Os, and differential input and output I/O pairs that use I/O and DDR registers.

Certain rules must be met to implement registered and DDR I/O functions (refer to the I/O Structures section of the handbook for the device you are using and the "DDR" section on page 8-6).

I/O Technology

The I/O Technology table (shown in Figure 8-10) gives the values of V_{CCI} and V_{REF} (reference voltage) for all the I/O standards used in the design. The user should assign these voltages appropriately.

I/O Technology:

	Voltages		I/Os		
I/O Standard(s)	Vcci	Vref	Input	Output	Bidirectional
LVTTL	3.30v	N/A	1	1	0
LVCMOS33	3.30v	N/A	1	0	0
LVCMOS25_50	2.50v	N/A	1	1	0
LVCMOS18	1.80v	N/A	1	0	0
LVCMOS15	1.50v	N/A	1	0	0
PCIX	3.30v	N/A	1	0	0
LVDS	2.50v	N/A	0	2	0
SSTL3I (Input/Bidirectional)	3.30v	1.50v	1	0	0
GTLP33 (Input/Bidirectional)	3.30v	1.00v	1	0	0

Figure 8-10 • **I/O Technology Table**

I/O Bank Resource Usage

This is an important portion of the report. The user must meet the requirements stated in this table. Figure 8-11 shows the I/O Bank Resource Usage table included in the I/O bank report:

```
I/O Bank Resource Usage:

          |   Voltages    | Single I/Os  | Diff I/O Pairs |        Vref I/Os
          |-------|-------|------|-------|-------|--------|------|-------|----------
          | Vcci  | Vref  | Used | Total | Used  | Total  | Used | Total | Vref Pins
    ------|-------|-------|------|-------|-------|--------|------|-------|----------
    Bank0 | N/A   | N/A   | 0    | 25    | 0     | 12     | N/A  | N/A   | N/A
    Bank1 | N/A   | N/A   | 0    | 15    | 0     | 7      | N/A  | N/A   | N/A
    Bank2 | N/A   | N/A   | 0    | 17    | 0     | 6      | N/A  | N/A   | N/A
    Bank3 | N/A   | N/A   | 0    | 16    | 0     | 7      | N/A  | N/A   | N/A
    Bank4 | N/A   | N/A   | 0    | 15    | 0     | 7      | N/A  | N/A   | N/A
    Bank5 | N/A   | N/A   | 0    | 22    | 0     | 10     | N/A  | N/A   | N/A
    Bank6 | N/A   | N/A   | 0    | 19    | 0     | 9      | N/A  | N/A   | N/A
    Bank7 | N/A   | N/A   | 0    | 18    | 0     | 7      | N/A  | N/A   | N/A

Warning: IOPRL1: 8 I/O Bank(s) have not been assigned any voltages.
         The I/O modules located in these banks cannot be assigned any I/O macro.
```

Figure 8-11 • **I/O Bank Resource Usage Table**

The example in Figure 8-11 shows that none of the I/O macros is assigned to the bank because more than one V_{CCI} is detected.

I/O Voltage Usage

The I/O Voltage Usage table provides the number of V_{REF} (E devices only) and V_{CCI} assignments required in the design. If the user decides to make I/O assignments manually (PDC or MVN), the issues listed in this table must be resolved before proceeding to Layout. As stated earlier, V_{REF} assignments must be made if there are any voltage-referenced I/Os.

Figure 8-12 shows the I/O Voltage Usage table included in the I/O bank report.

```
I/O Voltage Usage:

      Voltages    |     I/Os
    -------|-------|------|-------
     Vcci  | Vref  | Used | Total
    -------|-------|------|-------
     1.50v | N/A   | 1*   | 0
     1.80v | N/A   | 1*   | 0
     2.50v | N/A   | 4*   | 0
     3.30v | N/A   | 6*   | 0
     3.30v | 1.00v | 1*   | 0
     3.30v | 1.50v | 1*   | 0

Warning: IOPRL3: This design has infeasible I/O voltage requirement(s),
         which are indicated with a '*' in the I/O Voltage Usage table.
         Please consider importing a Physical Design Constraint (PDC) file or
         use the MultiView Navigator (MVN) to resolve the design's voltage requirements
         before running Layout.
```

Figure 8-12 • **I/O Voltage Usage Table**

The table in Figure 8-12 indicates that there are two voltage-referenced I/Os used in the design. Even though both of the voltage-referenced I/O technologies have the same V_{CCI} voltage, their V_{REF} voltages are different. As a result, two I/O banks are needed to assign the V_{CCI} and V_{REF} voltages.

In addition, there are six single-ended I/Os used that have the same V_{CCI} voltage. Since two banks are already assigned with the same V_{CCI} voltage and there are enough unused bonded I/Os in those banks, the user does not need to assign the same V_{CCI} voltage to another bank. The user needs to assign the other three V_{CCI} voltages to three more banks.

Assigning Technologies and V_{REF} to I/O Banks

Low-power flash devices offer a wide variety of I/O standards, including voltage-referenced standards. Before proceeding to Layout, each bank must have the required V_{CCI} voltage assigned for the corresponding I/O technologies used for that bank. The voltage-referenced standards require the use of a reference voltage (V_{REF}). This assignment can be done manually or automatically. The following sections describe this in detail.

Manually Assigning Technologies to I/O Banks

The user can import the PDC at this point and resolve this requirement. The PDC command is

```
set_iobank [bank name] –vcci [vcci value]
```

Another method is to use the I/O Bank Settings dialog box (**MVN** > **Edit** > **I/O Bank Settings**) to set up the V_{CCI} voltage for the bank (Figure 8-13).

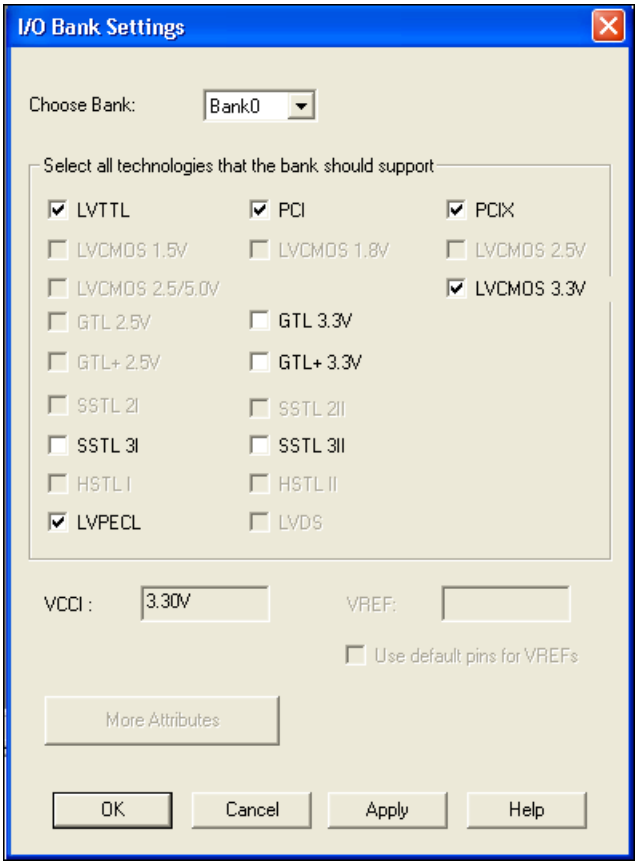


Figure 8-13 • **Setting V_{CCI} for a Bank**

The procedure is as follows:

1. Select the bank to which you want V_{CCI} to be assigned from the **Choose Bank** list.
2. Select the I/O standards for that bank. If you select any standard, the tool will automatically show all compatible standards that have a common V_{CCI} voltage requirement.
3. Click **Apply**.
4. Repeat steps 1–3 to assign V_{CCI} voltages to other banks. Refer to Figure 8-12 on page 8-13 to find out how many I/O banks are needed for V_{CCI} bank assignment.

Manually Assigning V_{REF} Pins

Voltage-referenced inputs require an input reference voltage (V_{REF}). The user must assign V_{REF} pins before running Layout. Before assigning a V_{REF} pin, the user must set a V_{REF} technology for the bank to which the pin belongs.

V_{REF} Rules for the Implementation of Voltage-Referenced I/O Standards

The V_{REF} rules are as follows:

1. Any I/O (except JTAG I/Os) can be used as a V_{REF} pin.
2. One V_{REF} pin can support up to 15 I/Os. It is recommended, but not required, that eight of them be on one side and seven on the other side (in other words, all 15 can still be on one side of V_{REF}).
3. SSTL3 (I) and (II): Up to 40 I/Os per north or south bank in any position
4. LVPECL / GTL+ 3.3 V / GTL 3.3 V: Up to 48 I/Os per north or south bank in any position
5. SSTL2 (I) and (II) / GTL+ 2.5 V / GTL 2.5 V: Up to 72 I/Os per north or south bank in any position.
6. V_{REF} minibanks partition rule: Each I/O bank is physically partitioned into V_{REF} minibanks. The V_{REF} pins within a V_{REF} minibank are interconnected internally, and consequently, only one V_{REF} voltage can be used within each V_{REF} minibank. If a bank does not require a V_{REF} signal, the V_{REF} pins of that bank are available as user I/Os.
7. The first V_{REF} minibank includes all I/Os starting from one end of the bank to the first power triple and eight more I/Os after the power triple. Therefore, the first V_{REF} minibank may contain (0 + 8), (2 + 8), (4 + 8), (6 + 8), or (8 + 8) I/Os.

   The second V_{REF} minibank is adjacent to the first V_{REF} minibank and contains eight I/Os, a power triple, and eight more I/Os after the triple. An analogous rule applies to all other V_{REF} minibanks but the last.

   The last V_{REF} minibank is adjacent to the previous one but contains eight I/Os, a power triple, and all I/Os left at the end of the bank. This bank may also contain (8 + 0), (8 + 2), (8 + 4), (8 + 6), or (8 + 8) available I/Os.

Example:

4 I/Os → Triple → 8 I/Os, 8 I/Os → Triple → 8 I/Os, 8 I/Os → Triple → 2 I/Os

i.e., minibank A = (4 + 8) I/Os, minibank B = (8 + 8) I/Os, minibank C = (8 + 2) I/Os

Assigning the V_{REF} Voltage to a Bank

When importing the PDC file, the V_{REF} voltage can be assigned to the I/O bank. The PDC command is as follows:

```
set_iobank –vref [value]
```

Another method for assigning V_{REF} is by using **MVN** > **Edit** > **I/O Bank Settings** (Figure 8-14 on page 8-16).

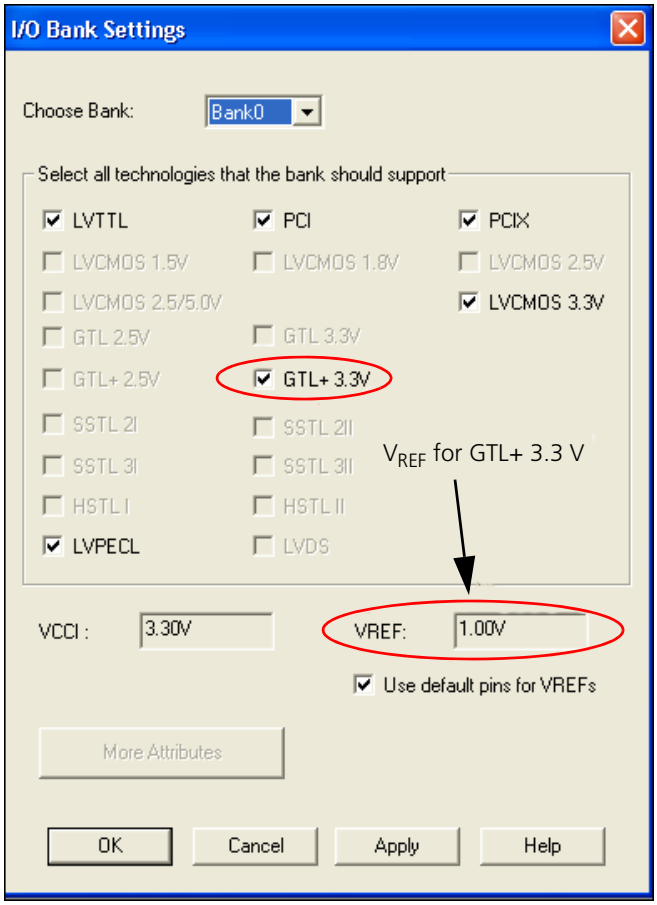


Figure 8-14 • **Selecting V_{REF} Voltage for the I/O Bank**

Assigning V_{REF} Pins for a Bank

The user can use default pins for V_{REF}. In this case, select the **Use default pins for V_{REF}s** check box (Figure 8-14). This option guarantees full V_{REF} coverage of the bank. The equivalent PDC command is as follows:

```
set_vref_default [bank name]
```

To be able to choose V_{REF} pins, adequate V_{REF} pins must be created to allow legal placement of the compatible voltage-referenced I/Os.

To assign V_{REF} pins manually, the PDC command is as follows:

```
set_vref –bank [bank name] [package pin numbers]
```

For ChipPlanner/PinEditor to show the range of a V_{REF} pin, perform the following steps:

1. Assign V_{CCI} to a bank using **MVN** > **Edit** > **I/O Bank Settings**.
2. Open **ChipPlanner**. Zoom in on an I/O package pin in that bank.
3. Highlight the pin and then right-click. Choose **Use Pin for V_{REF}**.

4. Right-click and then choose **Show V_{REF} range**. All the pins covered by that V_{REF} pin will be highlighted (Figure 8-15).

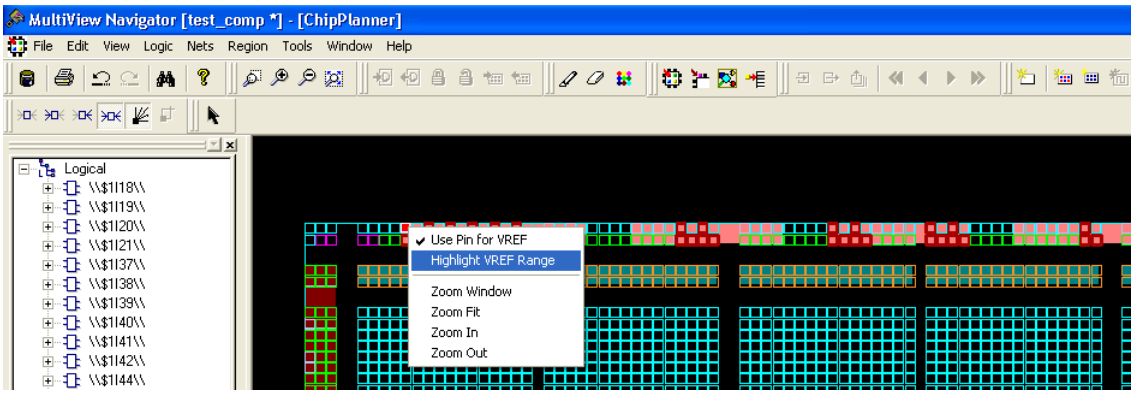


Figure 8-15 • V_{REF} Range

Using PinEditor or ChipPlanner, V_{REF} pins can also be assigned (Figure 8-16).

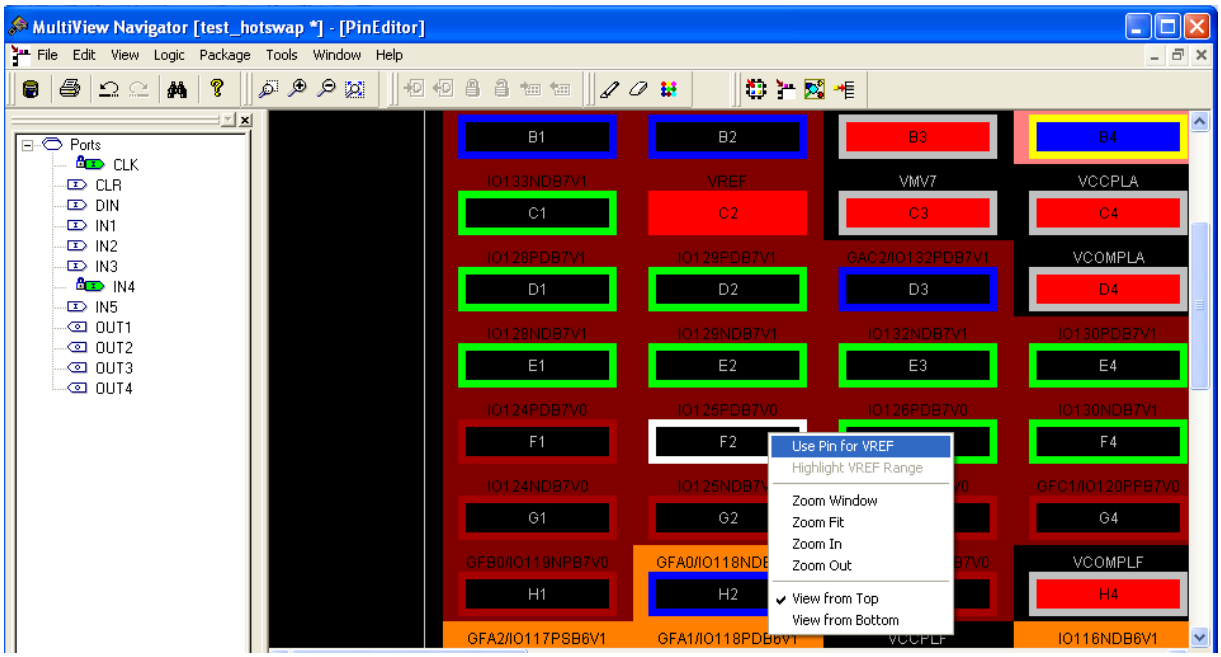


Figure 8-16 • Assigning V_{REF} from PinEditor

To unassign a V_{REF} pin:

1. Select the pin to unassign.
2. Right-click and choose **Use Pin for V_{REF}**. The check mark next to the command disappears. The V_{REF} pin is now a regular pin.

Resetting the pin may result in unassigning I/O cores, even if they are locked. In this case, a warning message appears so you can cancel the operation.

After you assign the V_{REF} pins, right-click a V_{REF} pin and choose **Highlight VREF Range** to see how many I/Os are covered by this pin. To unhighlight the range, choose **Unhighlight All** from the **Edit** menu.

Automatically Assigning Technologies to I/O Banks

The I/O Bank Assigner (IOBA) tool runs automatically when you run Layout. You can also use this tool from within the MultiView Navigator (Figure 8-18). The IOBA tool automatically assigns technologies and V_{REF} pins (if required) to every I/O bank that does not currently have any technologies assigned to it. This tool is available when at least one I/O bank is unassigned.

To automatically assign technologies to I/O banks, choose **I/O Bank Assigner** from the **Tools** menu (or click the I/O Bank Assigner's toolbar button, shown in Figure 8-17).

***Figure 8-17* • I/O Bank Assigner's Toolbar Button**

Messages will appear in the Output window informing you when the automatic I/O bank assignment begins and ends. If the assignment is successful, the message "I/O Bank Assigner completed successfully" appears in the Output window, as shown in Figure 8-18.

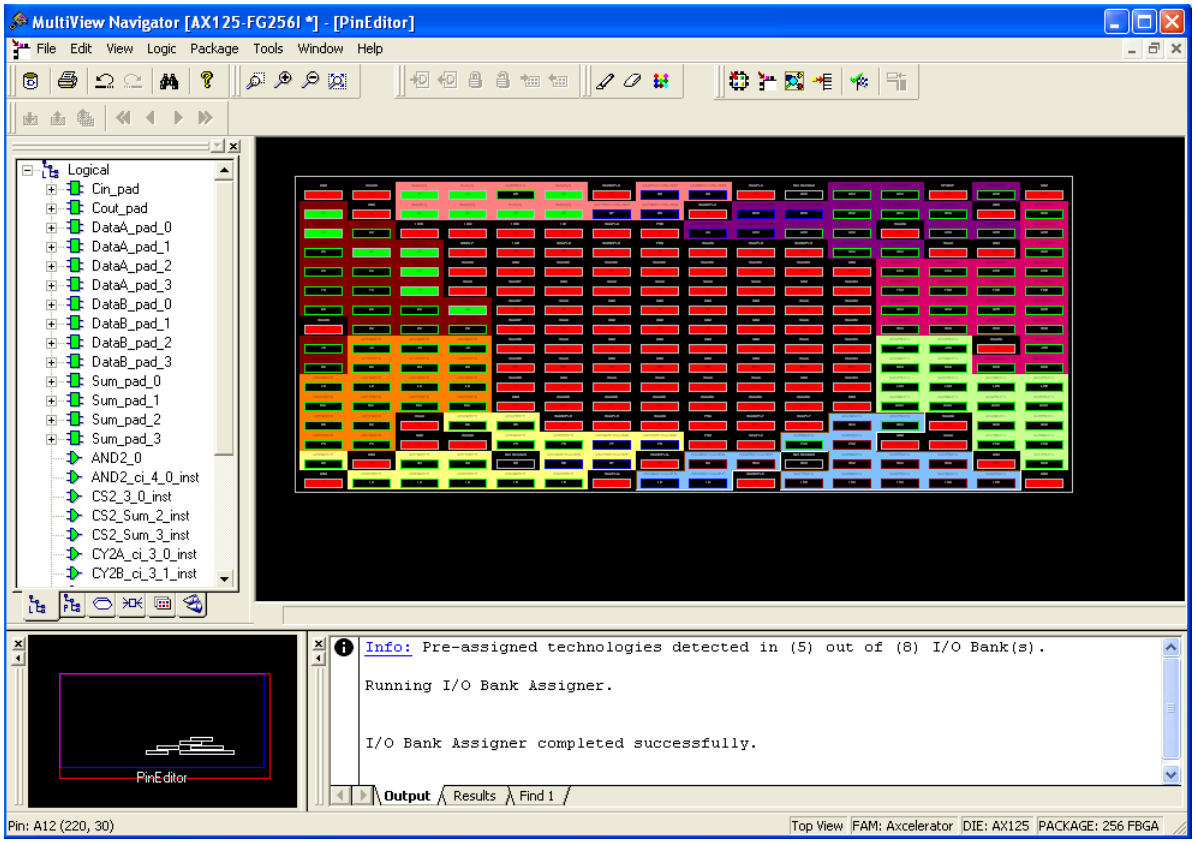


***Figure 8-18* • I/O Bank Assigner Displays Messages in Output Window**

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

Actel Fusion, IGLOO, and ProASIC3 support for multiple I/O standards minimizes board-level components and makes possible a wide variety of applications. The Actel Designer software, integrated with Actel Libero IDE, 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

Handbook Documents

DDR for Actel's Low-Power Flash Devices
http://www.actel.com/documents/LPD_DDR_HBs.pdf
*Flash*Freeze Technology and Low-Power Modes in IGLOO and ProASIC3L Devices*
http://www.actel.com/documents/LPD_FlashFreeze_HBs.pdf
Global Resources in Actel Low-Power Flash Devices
http://www.actel.com/documents/LPD_Global_HBs.pdf
I/O Structures in IGLOO and ProASIC3 Devices
http://www.actel.com/documents/IGLOO_PA3_IO_HBs.pdf
I/O Structures in IGLOO PLUS Devices
http://www.actel.com/documents/IGLOOPLUS_IO_HBs.pdf
I/O Structures in IGLOOe and ProASIC3E Devices
http://www.actel.com/documents/IGLOOe_PA3E_IO_HBs.pdf
Pin Descriptions
http://www.actel.com/documents/LPD_PinDescriptions_HBs.pdf
Power-Up/-Down Behavior of Low-Power Flash Devices
http://www.actel.com/documents/LPD_PowerUp_HBs.pdf
ProASIC3/E SSO and Pin Placement and Guidelines
http://www.actel.com/documents/PA3_E_SSO_HBs.pdf

User's Guides

Actel Libero IDE User's Guide
http://www.actel.com/documents/libero_ug.pdf
IGLOO, Fusion, and ProASIC3 Macro Library Guide
http://www.actel.com/documents/pa3_libguide_ug.pdf
SmartGen Core Reference Guide
http://www.actel.com/documents/genguide_ug.pdf

Part Number and Revision Date

This document contains content extracted from the Device Architecture section of the datasheet, combined with content previously published as an application note describing features and functions of the device. To improve usability for customers, the device architecture information has now been combined with usage information, to reduce duplication and possible inconsistencies in published information. No technical changes were made to the datasheet content unless explicitly listed. Changes to the application note content were made only to be consistent with existing datasheet information.

Part Number 51700094-026-2

Revised October 2008

List of Changes

The following table lists critical changes that were made in the current version of the document.

Previous Version	Changes in Current Version (v1.3)	Page
v1.2 (June 2008)	The "Low-Power Flash Families I/O Support" section was revised to include new families and make the information more concise.	8-2
v1.1 (March 2008)	The following changes were made to the family descriptions in Table 8-1 · Low-Power Flash Families: • ProASIC3L was updated to include 1.5 V. • The number of PLLs for ProASIC3E was changed from five to six.	8-2
v1.0 (January 2008)	This document was previously part of the I/O Structures in IGLOO and ProASIC3 Devices document. The content was separated and made into a new document.	N/A
	Table 8-2 · Designer State (resulting from I/O attribute modification) was updated to include note 2 for IGLOO PLUS.	8-3

Packaging and Pin Descriptions

9 – Pin Descriptions

Supply Pins

GND **Ground**

Ground supply voltage to the core, I/O outputs, and I/O logic.

GNDQ **Ground (quiet)**

Quiet ground supply voltage to input buffers of I/O banks. Within the package, the GNDQ plane is decoupled from the simultaneous switching noise originated from the output buffer ground domain. This minimizes the noise transfer within the package and improves input signal integrity. GNDQ must always be connected to GND on the board.

V_{CC} **Core Supply Voltage**

Supply voltage to the FPGA core, nominally 1.5 V for ProASIC®3/E devices, 1.5 V for IGLOO®/e V5 devices, and 1.2 V or 1.5 V for IGLOO/e V2 and ProASIC3L devices. V_{CC} is required for powering the JTAG state machine in addition to V_{JTAG}. Even when a device is in bypass mode in a JTAG chain of interconnected devices, both V_{CC} and V_{JTAG} must remain powered to allow JTAG signals to pass through the device.

For IGLOO/e V2 and ProASIC3L devices, V_{CC} can be switched dynamically from 1.2 V to 1.5 V or vice versa. This allows in-system programming (ISP) when V_{CC} is at 1.5 V and the benefit of low-power operation when V_{CC} is at 1.2 V.

V_{CCI}Bx **I/O Supply Voltage**

Supply voltage to the bank's I/O output buffers and I/O logic. Bx is the I/O bank number. There are up to eight I/O banks on low-power flash devices plus a dedicated V_{JTAG} bank. Each bank can have a separate V_{CCI} connection. All I/Os in a bank will run off the same V_{CCI}Bx supply. V_{CCI} can be 1.2 V (not supported on ProASIC3/E devices), 1.5 V, 1.8 V, 2.5 V, or 3.3 V, nominal voltage. Unused I/O banks should have their corresponding V_{CCI} pins tied to GND.

VMVx **I/O Supply Voltage (quiet)**

Quiet supply voltage to the input buffers of each I/O bank. *x* is the bank number. Within the package, the VMV plane is decoupled from the simultaneous switching noise originated from the output buffer V_{CCI} domain. This minimizes the noise transfer within the package and improves input signal integrity. Each bank must have at least one VMV connection, and no VMV should be left unconnected. All I/Os in a bank run off the same VMVx supply. VMV is used to provide a quiet supply voltage to the input buffers of each I/O bank. VMVx can be 1.2 V (ProASIC3L and IGLOO/e devices only), 1.5 V, 1.8 V, 2.5 V, or 3.3 V, nominal voltage. Unused I/O banks should have their corresponding VMV pins tied to GND. VMV and V_{CCI} should be at the same voltage within a given I/O bank. Used VMV pins must be connected to the corresponding V_{CCI} pins of the same bank (i.e., VMV0 to V_{CCI}B0, VMV1 to V_{CCI}B1, etc.).

$V_{CCPLA/B/C/D/E/F}$ **PLL Supply Voltage**

Supply voltage to analog PLL, nominally 1.5 V or 1.2 V, depending on the device family.

- 1.5 V for IGLOO V5, IGLOOe V5, ProASIC3, and ProASIC3E devices
- 1.2 V or 1.5 V for IGLOO V2, IGLOOe V2, ProASIC3L, and ProASIC3EL devices

When the PLLs are not used, the Actel Designer place-and-route tool automatically disables the unused PLLs to lower power consumption. The user should tie unused V_{CCPLx} and V_{COMPLx} pins to ground. Actel recommends tying V_{CCPLx} to V_{CC} and using proper filtering circuits to decouple V_{CC} noise from PLL. Refer to the PLL Power Supply Decoupling section of *Clock Conditioning Circuits in IGLOO and ProASIC3 Devices* for a complete board solution for the PLL analog power supply and ground.

- There is one V_{CCPLF} pin on IGLOO, IGLOO PLUS, ProASIC3L, and ProASIC3 devices.
- There are six V_{CCPLX} pins on IGLOOe, ProASIC3EL, and ProASIC3E devices.

$V_{COMPLA/B/C/D/E/F}$ PLL Ground

Ground to analog PLL power supplies. When the PLLs are not used, the Actel Designer place-and-route tool automatically disables the unused PLLs to lower power consumption. The user should tie unused V_{CCPLx} and V_{COMPLx} pins to ground.

- There is one V_{COMPLF} pin on IGLOO, ProASIC3L, and ProASIC3 devices.
- There are six V_{COMPL} pins (PLL ground) on IGLOOe, ProASIC3EL, and ProASIC3E devices.

V_{JTAG} JTAG Supply Voltage

Low-power flash devices have a separate bank for the dedicated JTAG pins. The JTAG pins can be run at any voltage from 1.5 V to 3.3 V (nominal). Isolating the JTAG power supply in a separate I/O bank gives greater flexibility in supply selection and simplifies power supply and PCB design. If the JTAG interface is neither used nor planned for use, the V_{JTAG} pin together with the TRST pin could be tied to GND. It should be noted that V_{CC} is required to be powered for JTAG operation; V_{JTAG} alone is insufficient. If a device is in a JTAG chain of interconnected boards, the board containing the device can be powered down, provided both V_{JTAG} and V_{CC} to the part remain powered; otherwise, JTAG signals will not be able to transition the device, even in bypass mode.

Actel recommends that V_{PUMP} and V_{JTAG} power supplies be kept separate with independent filtering capacitors rather than supplying them from a common rail.

V_{PUMP} Programming Supply Voltage

IGLOO, ProASIC3L, and ProASIC3 devices support single-voltage ISP of the configuration flash and FlashROM. For programming, V_{PUMP} should be 3.3 V nominal. During normal device operation, V_{PUMP} can be left floating or can be tied (pulled up) to any voltage between 0 V and the V_{PUMP} maximum. Programming power supply voltage (V_{PUMP}) range is listed in the datasheet.

When the V_{PUMP} pin is tied to ground, it will shut off the charge pump circuitry, resulting in no sources of oscillation from the charge pump circuitry.

For proper programming, 0.01 µF and 0.33 µF capacitors (both rated at 16 V) are to be connected in parallel across V_{PUMP} and GND, and positioned as close to the FPGA pins as possible.

Actel recommends that V_{PUMP} and V_{JTAG} power supplies be kept separate with independent filtering capacitors rather than supplying them from a common rail.

User-Defined Supply Pins

V_{REF} I/O Voltage Reference

Reference voltage for I/O minibanks in IGLOOe, ProASIC3EL, and ProASIC3E devices. V_{REF} pins are configured by the user from regular I/Os, and any I/O in a bank, except JTAG I/Os, can be designated the voltage reference I/O. Only certain I/O standards require a voltage reference—HSTL (I) and (II), SSTL2 (I) and (II), SSTL3 (I) and (II), and GTL/GTL+. One V_{REF} pin can support the number of I/Os available in its minibank.

User Pins

I/O User Input/Output

The I/O pin functions as an input, output, tristate, or bidirectional buffer. Input and output signal levels are compatible with the I/O standard selected.

During programming, I/Os become tristated and weakly pulled up to V_{CCI}. With V_{CCI}, VMV, and V_{CC} supplies continuously powered up, when the device transitions from programming to operating mode, the I/Os are instantly configured to the desired user configuration.

Unused I/Os are configured as follows:

- Output buffer is disabled (with tristate value of high impedance)
- Input buffer is disabled (with tristate value of high impedance)
- Weak pull-up is programmed

GL Globals

GL I/Os have access to certain clock conditioning circuitry (and the PLL) and/or have direct access to the global network (spines). Additionally, the global I/Os can be used as regular I/Os, since they have identical capabilities. Unused GL pins are configured as inputs with pull-up resistors.

See more detailed descriptions of global I/O connectivity in *Clock Conditioning Circuits in IGLOO and ProASIC3 Devices*. All inputs labeled GC/GF are direct inputs into the quadrant clocks. For example, if GAA0 is used for an input, GAA1 and GAA2 are no longer available for input to the quadrant globals. All inputs labeled GC/GF are direct inputs into the chip-level globals, and the rest are connected to the quadrant globals. The inputs to the global network are multiplexed, and only one input can be used as a global input.

Refer to the I/O Structure section of the handbook for the device you are using for an explanation of the naming of global pins.

FF Flash*Freeze Mode Activation Pin

Flash*Freeze is available on IGLOO, ProASIC3L, and RT ProASIC3 devices. It is not supported on ProASIC3/E devices. The FF pin is a dedicated input pin used to enter and exit Flash*Freeze mode. The FF pin is active-low, has the same characteristics as a single-ended I/O, and must meet the maximum rise and fall times. When Flash*Freeze mode is not used in the design, the FF pin is available as a regular I/O. For IGLOOe, ProASIC3EL and RT ProASIC3 only, the FF pin can be configured as a Schmitt trigger input.

When Flash*Freeze mode is used, the FF pin must not be left floating to avoid accidentally entering Flash*Freeze mode. While in Flash*Freeze mode, the Flash*Freeze pin should be constantly asserted.

The Flash*Freeze pin can be used with any single-ended I/O standard supported by the I/O bank in which the pin is located, and input signal levels compatible with the I/O standard selected. The FF pin should be treated as a sensitive asynchronous signal. When defining pin placement and board layout, simultaneously switching outputs (SSOs) and their effects on sensitive asynchronous pins must be considered.

Unused FF or I/O pins are tristated with weak pull-up. This default configuration applies to both Flash*Freeze mode and normal operation mode. No user intervention is required.

Table 9-1 shows the Flash*Freeze pin location on the available packages for IGLOO and ProASIC3L devices. The Flash*Freeze pin location is independent of device (except for a PQ208 package), allowing migration to larger or smaller IGLOO devices while maintaining the same pin location on the board. Refer to *Flash*Freeze Technology and Low-Power Modes in IGLOO and ProASIC3L Devices* for more information on I/O states during Flash*Freeze mode.

Table 9-1 • **Flash*Freeze Pin Location in IGLOO and ProASIC3L Family Packages (device-independent)**

IGLOO and ProASIC3L Packages	Flash*Freeze Pin
CS81/UC81	H2
CS121	J5
CS196	P3
QN68	18
QN132	B12
CS281	W2
CS201 (package available only for IGLOO PLUS devices)	R4
CS289 (package available only for IGLOO PLUS devices)	TBD
VQ100	27
FG144	L3
FG256	T3
FG484	W6
CG/LG484 (package available only for RT ProASIC3 devices)	W6
FG896	AH4
CG/LG896 (package available only for RT ProASIC3 devices)	TBD
PQ208 (package available only for ProASIC3L devices)	
PQ208-A3P250	56
PQ208-A3P600L	55
PQ2097-A3P1000L	55
PQ208-A3PE3000L	58

JTAG Pins

Low-power flash devices have a separate bank for the dedicated JTAG pins. The JTAG pins can be run at any voltage from 1.5 V to 3.3 V (nominal). V_{CC} must also be powered for the JTAG state machine to operate, even if the device is in bypass mode; V_{JTAG} alone is insufficient. Both V_{JTAG} and V_{CC} to the part must be supplied to allow JTAG signals to transition the device. Isolating the JTAG power supply in a separate I/O bank gives greater flexibility in supply selection and simplifies power supply and PCB design. If the JTAG interface is neither used nor planned for use, the V_{JTAG} pin together with the TRST pin could be tied to GND.

TCK Test Clock

Test clock input for JTAG boundary scan, ISP, and UJTAG. The TCK pin does not have an internal pull-up/-down resistor. If JTAG is not used, Actel recommends tying off TCK to GND through a resistor placed close to the FPGA pin. This prevents JTAG operation in case TMS enters an undesired state.

Note that to operate at all V_{JTAG} voltages, 500 Ω to 1 kΩ will satisfy the requirements. Refer to Table 9-2 for more information.

Table 9-2 • **Recommended Tie-Off Values for the TCK and TRST Pins**

V_{JTAG}	Tie-Off Resistance
V_{JTAG} at 3.3 V	200 Ω to 1 kΩ
V_{JTAG} at 2.5 V	200 Ω to 1 kΩ
V_{JTAG} at 1.8 V	500 Ω to 1 kΩ
V_{JTAG} at 1.5 V	500 Ω to 1 kΩ

Notes:

1. *Equivalent parallel resistance if more than one device is on the JTAG chain*
2. *The TCK pin can be pulled up/down.*
3. *The TRST pin is pulled down.*

TDI Test Data Input

Serial input for JTAG boundary scan, ISP, and UJTAG usage. There is an internal weak pull-up resistor on the TDI pin.

TDO Test Data Output

Serial output for JTAG boundary scan, ISP, and UJTAG usage.

TMS Test Mode Select

The TMS pin controls the use of the IEEE 1532 boundary scan pins (TCK, TDI, TDO, TRST). There is an internal weak pull-up resistor on the TMS pin.

TRST Boundary Scan Reset Pin

The TRST pin functions as an active-low input to asynchronously initialize (or reset) the boundary scan circuitry. There is an internal weak pull-up resistor on the TRST pin. If JTAG is not used, an external pull-down resistor could be included to ensure the test access port (TAP) is held in reset mode. The resistor values must be chosen from Table 9-2 and must satisfy the parallel resistance value requirement. The values in Table 9-2 correspond to the resistor recommended when a single device is used, and the equivalent parallel resistor when multiple devices are connected via a JTAG chain.

In critical applications, an upset in the JTAG circuit could allow entrance to an undesired JTAG state. In such cases, Actel recommends tying off TRST to GND through a resistor placed close to the FPGA pin.

Note that to operate at all V_{JTAG} voltages, 500 Ω to 1 kΩ will satisfy the requirements.

Special Function Pins

NC No Connect

This pin is not connected to circuitry within the device. These pins can be driven to any voltage or can be left floating with no effect on the operation of the device.

DC Do Not Connect

This pin should not be connected to any signals on the PCB. These pins should be left unconnected.

Related Documents

Handbook Documents

Clock Conditioning Circuits in IGLOO and ProASIC3 Devices
http://www.actel.com/documents/LPD_CCC_HBs.pdf
I/O Structures in IGLOO PLUS Devices
http://www.actel.com/documents/IGLOOPLUS_IO_HBs.pdf
I/O Structures in IGLOO and ProASIC3 Devices
http://www.actel.com/documents/IGLOO_PA3_IO_HBs.pdf
I/O Structures in IGLOOe and ProASIC3E Devices
http://www.actel.com/documents/IGLOOe_PA3E_IO_HBs.pdf
*Flash*Freeze Technology and Low-Power Modes in IGLOO and ProASIC3L Devices*
http://www.actel.com/documents/LPD_FlashFreeze_HBs.pdf

Part Number and Revision Date

This document contains content extracted from the Device Architecture section of the datasheet. To improve usability for customers, the device architecture information has now been split into handbook sections, which also include usage information. No technical changes were made to the content unless explicitly listed.

Part Number 51700094-011-2

Revised October 2008

List of Changes

The following table lists critical changes that were made in the current version of the chapter.

Previous Version	Changes in Current Version (v1.2)	Page
v1.1 (March 2008)	The "FF Flash*Freeze Mode Activation Pin" section was updated to include RT ProASIC3 devices.	9-3
	CG/LG484 and CG/LG896 were added to Table 9-1 · Flash*Freeze Pin Location in IGLOO and ProASIC3L Family Packages (device-independent).	9-4
v1.0 (January 2008)	The "VCCIBx I/O Supply Voltage" section was revised to note that 1.2 V is not supported for ProASIC3/E devices. The "VMVx I/O Supply Voltage (quiet)" section was updated to state that VMVx can also be 1.2 V nominal voltage on ProASIC3L and IGLOO/e devices.	9-1
	The "Handbook Documents" section was revised to include the three different I/O Structures chapters for IGLOO and ProASIC3 device families.	9-6
	The "$V_{CCPLA/B/C/D/E/F}$ PLL Supply Voltage" section and "$V_{COMPLA/B/C/D/E/F}$ PLL Ground" section were revised. The "V_{CCPLF} PLL Supply Voltage" section and "V_{COMPLF} PLL Ground" section were removed.	9-1 to 9-2
	The following packages were added to Table 9-1 · Flash*Freeze Pin Location in IGLOO and ProASIC3L Family Packages (device-independent): UC81, QN68, CS201, and CS289. Flash*Freeze pin W2 was specified for the CS281 package. The PG208 package was changed to the correct designation of PQ208.	9-4

10 – Packaging

Semiconductor technology is constantly shrinking in size while growing in capability and functional integration. To enable next-generation silicon technologies, semiconductor packages have also evolved to provide improved performance and flexibility.

Actel consistently delivers packages that provide the necessary mechanical and environmental protection to ensure consistent reliability and performance. Actel IC packaging technology efficiently supports high-density FPGAs with large-pin-count Ball Grid Arrays (BGAs), but is also flexible enough to accommodate stringent form factor requirements for Chip Scale Packaging (CSP). In addition, Actel offers a variety of packages designed to meet your most demanding application and economic requirements for today's embedded and mobile systems.

The following documents provide packaging information and device selection for low-power flash devices.

Package Selector Guide

http://www.actel.com/documents/selguide.pdf

Lists devices currently recommended for new designs and the packages available for each member of the family. Use this document or the datasheet tables to determine the best package for your design, and which package drawing to use.

Package Mechanical Drawings

http://www.actel.com/documents/PckgMechDrwngs.pdf

This document contains the package mechanical drawings for all packages currently or previously supplied by Actel. Use the bookmarks to navigate to the package mechanical drawings.

Related Documents

Additional packaging materials are available at http://www.actel.com/products/solutions/package/docs.aspx.

Part Number and Revision Date

This document contains content extracted from the Device Architecture section of the datasheet, combined with content previously published as an application note describing features and functions of the device. To improve usability for customers, the device architecture information has now been combined with usage information, to reduce duplication and possible inconsistencies in published information. No technical changes were made to the datasheet content unless explicitly listed. Changes to the application note content were made only to be consistent with existing datasheet information.

Part Number 5170009-012-0

Revised January 2008

Programming and Security

Actel
POWER MATTERS

11 – Programming Flash Devices

Introduction

This document provides an overview of the various programming options available for the Actel flash families. The electronic version of this document includes active links to all programming resources, which are available at http://www.actel.com/products/hardware/default.aspx. For Actel antifuse devices, refer to the *Programming Antifuse Devices* document.

Summary of Programming Support

FlashPro3 is a high-performance in-system programming (ISP) tool targeted at the latest generation of low-power flash devices offered by Actel: Fusion, IGLOO,® and ProASIC®3, including ARM®-enabled devices. FlashPro3 offers extremely high performance through the use of USB 2.0, is high-speed compliant for full use of the 480 Mbps bandwidth, and can program ProASIC3 devices in under 30 seconds. Powered exclusively via USB, FlashPro3 provides a V_{PUMP} voltage of 3.3 V for programming these devices.

Silicon Sculptor 3 is an easy-to-use, single-site programming tool for Actel FPGAs that delivers high data throughput and promotes ease of use while lowering the overall cost of ownership. Silicon Sculptor 3 includes a high-speed USB 2.0 interface that allows a customer to connect as many as 12 programmers to a single PC. Furthermore, Silicon Sculptor 3 is compatible with adapter modules from Silicon Sculptor II, thereby preserving a customer's investment and enabling a seamless upgrade to this latest generation of the tool.

For details of programmer support for each device, refer to Table 11-6 on page 11-10.

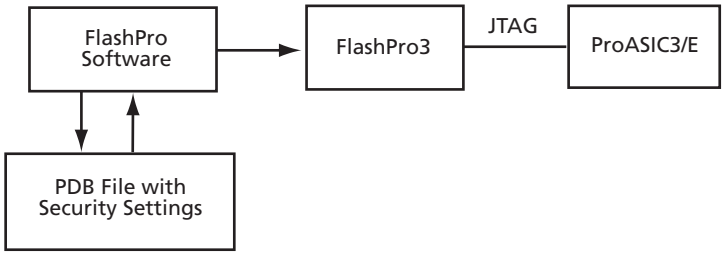


***Figure 11-1* • FlashPro Programming Setup**

Programming Support in Flash Devices

The flash families listed in Table 11-1 support flash in-system programming and the functions described in this document.

Table 11-1 • **Low-Power Flash Families**

Product Line	Family*	Description
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
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 PLUS	IGLOO FPGAs with enhanced I/O capabilities
ProASIC3	ProASIC3	Low-power, high-performance 1.5 V FPGAs
	ProASIC3E	Higher density ProASIC3 FPGAs with six PLLs and additional I/O standards
	ProASIC3L	ProASIC3 FPGAs supporting 1.2 V to 1.5 V with Flash*Freeze technology
	Automotive ProASIC3	ProASIC3 FPGAs qualified for automotive applications
	Military ProASIC3/EL	Military temperature A3PE600L, A3P1000, and A3PE3000L
	RT ProASIC3	Radiation-tolerant RT3PE600L and RT3PE3000L
ProASIC	ProASIC	First generation ProASIC devices
	ProASIC PLUS	Second generation ProASIC devices

*Note: *The family names link to the appropriate datasheet, including product brief, DC and switching characteristics, and packaging information.*

IGLOO Terminology

In documentation, the terms IGLOO families and IGLOO devices refer to all of the IGLOO products as listed in Table 11-1. Where the information applies to only one family or limited devices, these exclusions will be explicitly stated.

ProASIC3 Terminology

In documentation, the terms ProASIC3 families and ProASIC3 devices refer to all of the ProASIC3 families as listed in Table 11-1. Where the information applies to only one family or limited devices, these exclusions will be explicitly stated.

To further understand the differences between the IGLOO and ProASIC3 families, refer to the *Industry's Lowest Power FPGAs Portfolio*.

General Flash Programming Information

Programming Basics

When choosing a programming solution, there are a number of options available. This section provides a brief overview of those options. The next sections provide more detail on those options as they apply to Actel FPGAs.

Reprogrammable or One-Time-Programmable (OTP)

Depending on the technology chosen, devices may be reprogrammable or one-time-programmable. As the name implies, a reprogrammable device can be programmed many times. Generally, the contents of such a device will be completely overwritten when it is reprogrammed. All Actel flash devices are reprogrammable.

An OTP device is programmable one time only. Once programmed, no more changes can be made to the contents. Actel flash devices provide the option of disabling the reprogrammability for security purposes. This combines the convenience of reprogrammability during design verification with the security of an OTP technology for highly sensitive designs.

Device Programmer or In-System Programming

There are two fundamental ways to program an FPGA: using a device programmer or, if the technology permits, using in-system programming. A device programmer is a piece of equipment in a lab or on the production floor that is used for programming devices. The devices are placed into a socket mounted in a programming adapter module, and the appropriate electrical interface is applied. The device can then be placed on the board. A typical programmer, used during development, programs a single device at a time and is referred to as a single-site engineering programmer.

With ISP, the device is already mounted onto the board when programming occurs, most typically via the JTAG pins. The JTAG pins can be controlled either by an on-board resource, such as a microprocessor, or by an off-board programmer through a header connection. Once mounted, it can be programmed repeatedly. If the application requires it, the system can be designed to reprogram itself using a microprocessor, without the use of any external programmer.

For production, high-volume multi-site production programmers handle designs that require device programmers. In addition, Actel can preprogram devices for production, negating the need for further programming. This service is referred to as in-house programming (IHP).

Live at Power-Up (LAPU) or Boot PROM

Utilizing the technology of the FPGA significantly impacts board-level power-up considerations. Some technologies are nonvolatile and are considered functional, or "live," as soon as power reaches the operational level. All Actel FPGA technologies are live at power-up. By contrast, SRAM technology is volatile, and devices built using SRAM cells lose their contents when power cycling occurs. These devices must be reprogrammed every time power is applied. Such a design must include nonvolatile storage for the contents as well as the means to reprogram. There is a delay before SRAM devices are functional; other parts of the board must come alive first to reprogram these types of FPGAs. Therefore, such devices can never be part of critical boot circuits.

Design Security

Design security is a growing concern for systems designers. The choice of programming methodology and technology affects system security. Use of Actel programming technology is the most secure option available, providing much better protection than SRAM-based devices and ASICs. Actel provides a number of ways to ensure designs are protected. General information on design security can be found on the Actel website:

http://www.actel.com/products/solutions/security/default.aspx

Programming Features for Actel Devices

Actel provides two types of FPGAs: flash and antifuse (Table 11-2). Some programming methods are common to both and some are exclusive to flash. This document describes only the programming solutions supported for flash devices.

Table 11-2 • **Programming Features for Actel Devices**

Feature	Flash	Antifuse
Reprogrammable	Yes	No
In-system programmable	Yes	No
One-time programmable	Yes (option)	Yes
Live at power-up	Yes	Yes
Secure	Yes	Yes
Single-site programmer support	Yes	Yes
Multi-site programmer support	Yes	Yes
In-house programming support	Yes	Yes

Flash Devices

The flash devices supplied by Actel are reprogrammable by either a generic device programmer or ISP. Actel supports ISP using JTAG, which is supported by the FlashPro3, FlashPro, FlashPro Lite, and Sculptor programmers.

Levels of ISP support vary depending on the device chosen:

- All Fusion, IGLOO, and ProASIC3 devices support ISP.
- ProASIC3L devices operate using a 1.2 V core voltage and support ISP at 1.5 V only. Voltage switching is required in-system to switch from a 1.2 V core to 1.5 V core for programming.
- IGLOO, IGLOOe, and IGLOO PLUS V5 devices can be programmed in-system when the device is using a 1.5 V supply voltage to the FPGA core.
- IGLOO, IGLOOe, and IGLOO PLUS V2 devices can operate using either 1.2 V core voltage or 1.5 V core voltage. Although the device can operate at 1.2 V core voltage, the device can only be reprogrammed when the core voltage is 1.5 V. Voltage switching is required in-system to switch from a 1.2 V supply (V_{CC}, V_{CCI}, and V_{JTAG}) to 1.5 V for programming.

Since flash devices are nonvolatile, they are live at power-up. This is different from an SRAM-based device, which loads its programming information when it is powered up. SRAM devices require a time on the order of hundreds of milliseconds before the system is active.

There are multiple levels of security available in flash devices. Use of a security key will lock the device. The device can then only be reprogrammed by first unlocking the device with the appropriate security key. It can also be locked permanently, which means there is no key that can access the device. The command to secure the device is embedded within the programming file, optionally enabled by the programming software. This is also referred to as the OTP version of flash, allowing for only a single programming instance. This is discussed in more detail in the *Implementation of Security in Actel's ProASIC and ProASIC^{PLUS} Flash-Based FPGAs* application note, and in the *Security in Low-Power Flash Devices* handbook section.

Flash devices can also be programmed using single-site or multi-site programmers as well as volume-programming services from Actel or other vendors.

Types of Programming for Flash Devices

The number of devices to be programmed will influence the optimal programming methodology. Those available are listed below:

- In-system programming
  - Using a programmer
  - Using a microprocessor or microcontroller
- Device programmers
  - Single-site programmers
  - Multi-site programmers, batch programmers, or gang programmers
  - Automated production (robotic) programmers
- Volume programming services
  - Actel in-house programming
  - Programming centers

In-System Programming

Device Type Supported: Flash

ISP refers to programming the FPGA after it has been mounted on the system board. The FPGA may be preprogrammed and later reprogrammed using ISP.

The advantage of using ISP is the ability to update the FPGA design many times without any changes to the board. This eliminates the requirement of using a socket for the FPGA, saving cost and improving reliability. It also reduces programming hardware expenses, as the ISP methodology is die-/package-independent.

There are two methods of in-system programming: external and internal.

- Programmer ISP—Refer to *In-System Programming (ISP) of Actel's Low-Power Flash Devices Using FlashPro3* for more information.

  Using an external programmer and a cable, the device can be programmed through a header on the system board. In Actel documentation, this is referred to as external ISP. Actel provides FlashPro3, FlashPro Lite, FlashPro, or Silicon Sculptor 3 to perform external ISP. Note that Silicon Sculptor II and Silicon Sculptor 3 can only provide ISP for ProASIC and ProASIC^{PLUS}® families, not for Fusion, IGLOO, or ProASIC3.

  - Advantages: Allows local control of programming and data files for maximum security. The programming algorithms and hardware are available from Actel. The only hardware required on the board is a programming header.
  - Limitations: A negligible board space requirement for the programming header and JTAG signal routing
- Microprocessor ISP—Refer to *MicroProcessor Programming of Actel's Low-Power Flash Devices* for more information.

  Using a microprocessor and an external or internal memory, you can store the program in memory and use the microprocessor to perform the programming. In Actel documentation, this is referred to as internal ISP. Both the code for the programming algorithm and the FPGA programming file must be stored in memory on the board. Programming voltages must also be generated on the board.

  - Advantages: The programming code is stored in the system memory. An external programmer is not required during programming.
  - Limitations: This is the approach that requires the most design work, since some way of getting and/or storing the data is needed; a system interface to the device must be designed; and the low-level API to the programming firmware must be written and linked into the code provided by Actel. While there are benefits to this methodology, serious thought and planning should go into the decision.

Device Programmers

Device Type Supported: Flash and Antifuse

Device programmers are used to program a device before it is mounted on the system board.

The advantage of using device programmers is that no programming hardware is required on the system board. Therefore, no additional components or board space are required.

If devices are to be reprogrammed multiple times, or if the quantity of devices to be programmed is relatively low, a single-site device programmer is the simplest solution. For applications in which design security is paramount (often the case in military or space designs), the use of on-site programing maintains design security at all times.

Adapter modules are purchased with the programmers to support the FPGA packages used. The FPGA is placed in the adapter module and the programming software is run from a PC. Actel supplies the programming software for all of the Actel programmers. The software allows for the selection of the correct die/package and programming files. It will then program and verify the device.

- Single-site programmers

  A single-site programmer programs one device at a time. Actel offers Silicon Sculptor 3 as a single-site programmer.

  – Advantages: Lower cost than multi-site programmers. No additional overhead for programming on the system board. Allows local control of programming and data files for maximum security. Allows on-demand programming on-site.

  – Limitations: Only programs one device at a time.

- Multi-site programmers

  Often referred to as batch or gang programmers, multi-site programmers can program multiple devices at the same time using the same programming file. This is often used for large volume programming and by programming houses. The sites often have independent processors and memory enabling the sites to operate concurrently, meaning each site may start programming the same file independently. This enables the operator to change one device while the other sites continue programming, which increases throughput. Multiple adapter modules for the same package are required when using a multi-site programmer. Silicon Sculptor I, II, and 3 programmers can be cascaded to program multiple devices in a chain. Multi-site programmers can also be purchased from BP Microsystems.

  – Advantages: Provides the capability of programming multiple devices at the same time. No additional overhead for programming on the system board. Allows local control of programming and data files for maximum security.

  – Limitations: More expensive than a single-site programmer

- Automated production (robotic) programmers

  Automated production programmers are based on multi-site programmers. They consist of a large input tray holding multiple parts and a robotic arm to select and place parts into appropriate programming sockets automatically. When the programming of the parts is complete, the parts are removed and placed in a finished tray. The automated programmers are often used in volume programming houses to program parts for which the programming time is small.

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 11-3 describes the volume programming services.

Table 11-3 • **Volume Programming Services**

Programmer	Vendor	Availability
In-House Programming	Actel	Contact Actel 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.

- Actel In-House Programming

  When purchasing Actel 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 Actel. Sample parts with the design programmed, First Articles, will be returned for customer approval. Once approval of First Articles has been received, Actel will proceed with programming the remainder of the order. To request Actel IHP, contact your local Actel 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.

- Independent Programming Centers

  There are many programming centers that specialize only in programming but are not directly affiliated with Actel or our distributors. These programming centers must follow the guidelines for programming Actel devices and use certified programmers to program Actel devices. Actel does not have recommendations for external programming centers.

Programming Solutions

Details for the available programmers can be found in the programmer user's guides listed in the "Related Documents" section on page 11-14. Refer to Table 11-6 on page 11-10 for more information concerning programming solutions.

All of the programmers except the FlashPro3, FlashPro Lite, and FlashPro require adapter modules, which are designed to support device packages. The modules are all listed on the Actel website at http://www.actel.com/products/hardware/program_debug/ss/modules.aspx. They are not listed in this document, since this list is updated frequently with new package options and any upgrades required to improve programming yield or support new families.

Table 11-4 • **Programming Solutions**

Programmer	Vendor	ISP	Single Device	Multi-Device	Availability
FlashPro3	Actel	Only	Yes	Yes[1]	Available
FlashPro Lite	Actel	Only	Yes	Yes[1]	Available
FlashPro	Actel	Only	Yes	Yes[1]	Available
Silicon Sculptor 3	Actel	Yes[2]	Yes	Cascade option (up to two)	Available
Silicon Sculptor II	Actel	Yes[2]	Yes	Cascade option (up to two)	Available
Silicon Sculptor	Actel	Yes	Yes	Cascade option (up to four)	Discontinued
Sculptor 6X	Actel	No	Yes	Yes	Discontinued
BP MicroProgrammers	BP Microsystems	No	Yes	Yes	Contact BP Microsystems at www.bpmicro.com

Notes:

1. *Multiple devices can be connected in the same JTAG chain for programming.*
2. *Silicon Sculptor II and Silicon Sculptor 3 can only provide ISP for ProASIC and ProASIC^{PLUS} families, not for Fusion, IGLOO, or ProASIC3 devices.*

Programmer Ordering Codes

The products shown in Table 11-5 can be ordered through Actel sales and will be shipped directly from Actel. Products can also be ordered from Actel distributors, but will still be shipped directly from Actel. Table 11-5 includes ordering codes for the full kit, as well as codes for replacement items and any related hardware. Some additional products can be purchased from external suppliers for use with the programmers. Ordering codes for adapter modules used with Silicon Sculptor are available on the Actel website at http://www.actel.com/products/hardware/program_debug/ss/modules.aspx.

Table 11-5 • **Programming Ordering Codes**

Description	Vendor	Ordering Code	Comment
FlashPro3 ISP programmer	Actel	FLASHPRO 3	Uses a 2x5, RA male header connector
FlashPro Lite ISP programmer	Actel	FLASHPRO LITE	Supports small programming header or large header through header converter (not included)
FlashPro ISP programmer	Actel	FLASH PRO	Supports small programming header or large header through header converter (not included)
Silicon Sculptor 3	Actel	SILICON-SCULPTOR 3	USB 2.0 high-speed production programmer
Silicon Sculptor II	Actel	SILICON-SCULPTOR II	Requires add-on adapter modules to support devices
Silicon Sculptor ISP module	Actel	SMPA-ISP-ACTEL-3-KIT	Ships with both large and small header support

** A maximum of two Silicon Sculptor II programmers can be chained together using a standard IEEE 1284 parallel port cable.*

Table 11-5 • **Programming Ordering Codes (continued)**

Description	Vendor	Ordering Code	Comment
Concurrent programming cable	Actel	SS-EXPANDER	Used to cascade Silicon Sculptor I programmers together*
Software for Silicon Sculptor	Actel	SCULPTOR-SOFTWARE-CD	http://www.actel.com/download/program_debug/ss/
ISP cable for small header	Actel	ISP-CABLE-S	Supplied with SMPA-ISP-ACTEL-3-KIT
ISP cable for large header	Actel	PA-ISP-CABLE	Supplied with SMPA-ISP-ACTEL-3-KIT
Header converter	Actel	Header-Converter	Converts from small to large header
Small programming header	Samtec	FTSH-113-01-L-D-K	Supported by FlashPro, FlashPro Lite, and Silicon Sculptor In migrating to ProASIC3/E devices, an FP3-26PIN-ADAPTER is required.
10-pin 0.1" pitch cable header (right-angle PCB mount angle)	AMP	103310-1	Supported by FlashPro3
10-pin 0.1" pitch cable header (straight PCB mount angle)	3M	2510-6002UB	Supported by FlashPro3
Compact programming header (10-pin 0.05" pitch, 2 rows of 5 pins)	Samtec	FTSH-105-01-L-D-K	Supported by FlashPro3, FP3-26PIN-ADAPTER required. Used for boards where space is at a premium.
Migration and compact header adapter	Actel	FP3-26PIN-ADAPTER	Required with the use of FTSH-105-01-L-D-K
Large programming header 0.062" board thickness	3M	3429-6502	Supported by Silicon Sculptor by default, FlashPro, and FlashPro Lite, with header converter
Large programming header 0.094" to 0.125" board thickness	3M	3429-6503	Supported by Silicon Sculptor by default, FlashPro, and FlashPro Lite, with header converter
Plug-in header small	Actel	SMPA-ISP-HEADER-S	Required for small header for ProASIC only; not used for ProASIC^{PLUS}
Plug-in header	Actel	SMPA-ISP-HEADER	Required for large header for ProASIC only; not used for ProASIC^{PLUS}
Vacuum pens for PQ, TQ, VQ; <208 pins	Actel	PENVAC	
Vacuum pens for PQ, TQ, VQ; ≥208 pins	Actel	PENVAC-HD	Heavy-duty, provides stronger vacuum

* *A maximum of two Silicon Sculptor II programmers can be chained together using a standard IEEE 1284 parallel port cable.*

Programmer Device Support

Refer to Table 11-6 to determine which general-purpose flash devices have programmer device support. To learn more about the different Actel families, refer to the Actel website: http://www.actel.com/products/devices.aspx.

Data in Table 11-6 also applies to ARM-enabled M7 device versions of Fusion, IGLOO, and ProASIC3 devices. Refer to the appropriate family datasheets for information on die/package combinations available as ARM-enabled versions.

Table 11-6 • **Programmer Device Support**

Actel Family	Device	ARM-Enabled	Silicon Sculptor	Silicon Sculptor 6X	Silicon Sculptor II	Silicon Sculptor 3	FlashPro	FlashPro Lite	FlashPro3
Fusion	AFS090		No	No	Yes. No ISP support.	Yes. No ISP support.	No	No	Yes. ISP support
	AFS250								
	AFS600	✓							
	AFS1500	✓							
IGLOO	AGL015		No	No	Yes. No ISP support.	Yes. No ISP support.	No	No	Yes. ISP support.
	AGL030								
	AGL060								
	AGL125								
	AGL250	✓							
	AGL600	✓							
	AGL1000	✓							
IGLOOe	AGLE600	✓	No	No	Yes. No ISP support	Yes. No ISP support.	No	No	Yes. ISP support
	AGLE3000	✓							
IGLOO PLUS	AGLP030		No	No	Yes. No ISP support	Yes. No ISP support.	No	No	Yes. ISP support
	AGLP060								
	AGLP125								
ProASIC3L	A3P250L	✓	No	No	Yes. No ISP support.	Yes. No ISP support.	No	No	Yes. ISP support.
	A3P600L	✓							
	A3P1000L	✓							
	A3PE3000L	✓							
ProASIC3	A3P015		No	No	Yes. No ISP support.	Yes. No ISP support.	No	No	Yes. ISP support.
	A3P030								
	A3P060								
	A3P125								
	A3P250	✓							
	A3P400	✓							
	A3P600	✓							
	A3P1000	✓							
ProASIC3E	A3PE600	✓	No	No	Yes. No ISP support.	Yes. No ISP support.	No	No	Yes. ISP support.
	A3PE1500	✓							
	A3PE3000	✓							

** Refer to the "Certified Programming Solutions" section on page 11-11 for more information on programmer support.*

Table 11-6 • **Programmer Device Support (continued)**

Actel Family	Device	ARM-Enabled	Silicon Sculptor	Silicon Sculptor 6X	Silicon Sculptor II	Silicon Sculptor 3	FlashPro	FlashPro Lite	FlashPro3
ProASIC^{PLUS}	APA075		Yes. ISP support.	Yes. ISP support.	Yes. ISP support.	Yes. ISP support.	Yes. ISP support.	Yes. ISP support.	No
	APA150								
	APA300								
	APA450								
	APA600								
	APA750								
	APA1000								
ProASIC	A500K50		Yes	Yes	Yes	Yes	Yes	No	No
	A500K130								
	A500K180								
	A500K270								

* *Refer to the "Certified Programming Solutions" section on page 11-11 for more information on programmer support.*

Certified Programming Solutions

The Actel-certified programmers for flash devices are FlashPro3, FlashPro Lite, FlashPro, Silicon Sculptor I and II, and any programmer that is built by BP Microsystems. All other programmers are considered noncertified programmers.

- **FlashPro3, FlashPro Lite, FlashPro**

  The Actel family of FlashPro device programmers provides in-system programming in an easy-to-use, compact system that supports all flash families. Whether programming a board containing a single device or multiple devices connected in a chain, the Actel line of FlashPro programmers enables fast programming and reprogramming. Programming with the FlashPro series of programmers saves board space and money as it eliminates the need for sockets on the board. There are no built-in algorithms, so there is no delay between product release and programming support.

- **Silicon Sculptor II**

  Silicon Sculptor II is a robust, compact, single-device programmer with standalone software for the PC. It is designed to enable concurrent programming of multiple units from the same PC with speeds equivalent to or faster than previous Actel programmers. It replaces Silicon Sculptor I as the Actel programmer of choice.

- **Silicon Sculptor I and Silicon Sculptor 6X**

  Actel no longer offers Silicon Sculptor I or Silicon Sculptor 6X for sale. Both items have been discontinued. Actel does support Silicon Sculptor I and Silicon Sculptor 6X by continuing to release new software that enables improved programming of previously covered Actel devices; new Actel devices are only supported on Silicon Sculptor II. All software support for Silicon Sculptor I and Silicon Sculptor 6X programmers will be disconnected by the end of 2005; no support for these older programmers will be offered in 2006. Actel recommends that all customers upgrade to Silicon Sculptor II or a BP multi-site programmer.

- **Noncertified Programmers**

  Actel does not test programming solutions from other vendors, and CANNOT guarantee programming yield. Also, Actel will not perform any failure analysis on devices programmed by hardware from other vendors.

- **Programming Centers**

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

Flash Programming Guidelines

Preprogramming Setup

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

Use Proper Handling and Electrostatic Discharge (ESD) Precautions

Actel 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 *Actel 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 Actel flash devices (*.bit, *.stp) contain important information about the switches that will be programmed in the FPGA. Find the latest version and corresponding release notes at http://www.actel.com/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.actel.com/download/program_debug/flashpro/
- Silicon Sculptor: http://www.actel.com/download/program_debug/ss/

Use the Most Recent Adapter Module with Silicon Sculptor

Occasionally, Actel 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.actel.com/products/hardware/program_debug/ss/modules.aspx.

Perform Routine Hardware Self-Diagnostic Test

- FlashPro

  The self-test is only applicable when programming with FlashPro and FlashPro3 programmers. It is not supported with FlashPro Lite. To run the self-diagnostic test, follow the instructions given in the "Performing a Self-Test" section of http://www.actel.com/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 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**.

Programming Flash FPGAs

Programming a flash device is a one-step process, whether programming is conducted with a socket adapter module or via ISP. The Execute function will automatically erase the device, program the flash cells, and verify that it is programmed correctly. Actel recommends confirming the security status is correct before programming.

The following steps are required to program Actel flash FPGAs.

Programming with FlashPro

Setup

Properly connect the FlashPro ribbon cable with the programming header and turn on the switch. Actel recommends running the self-test before programming any devices; see the "Perform Routine Hardware Self-Diagnostic Test" section on page 11-12.

In the programming software, from the File menu, choose **Connect**. In the FlashPro Connect to Programmer dialog box that appears, select the port to which the FlashPro programmer is connected, and select the device family. Disable voltages from the programmer if they are available on the board.

Click **Connect**. A successful connect or any errors appear in the Log window.

Analyze Chain and Device Selection

From the File menu, choose **Analyze Chain**. Chain details appear in the Log window. If any failures appear, refer to the error and troubleshooting section of the *FlashPro User's Guide*. Select the device to be programmed from the **Device** list. If only one device is present in the chain, performing Analyze Chain selects that device automatically from the Device list.

Loading the STAPL file

FlashPro3, FlashPro Lite, and FlashPro programmers use a STAPL (*.stp) file to program the device. To load the STAPL file, from the File menu, choose **Open STAPL file**, or click the **Open File** button in the toolbar.

Selecting an Action

After loading the STAPL file, select an action from the Action list. See the "Programming File Actions" section in the *FlashPro User's Guide* for a definition of each action.

Programming the Device

To program the device, in the Action list, select **Program**. Make the required selections and click **Execute** to start programming. The progress of the programming action displays in the Log window. The message "Exit 0" indicates that the device has successfully been programmed.

Note: Do not interrupt the programming sequence; it may damage the device or programmer.

Verify Correct Programming

To verify the device is programmed with the correct STAPL file, load the STAPL file and in the Action list and click **Verify**. Click **Execute** to start the verification process. A successful verification results in "Exit 0."

Note: Verification is also performed in the previous "Programming the Device" step; clicking **Verify** is an additional standalone option.

Programming Failure Allowances

Flash FPGAs are reprogrammable, so Actel tests the programmability for 100% of the devices shipped.

Return Material Authorization (RMA) Policies

Actel consistently strives to exceed customer expectations by continuing to improve the quality of our products and our quality management system. Actel has RMA procedures in place to address programming fallout. Customers should be mindful of the following RMA policies.

All devices submitted for an RMA, must be within the Actel warranty period of one year from date of shipment. Actel will reject RMAs for devices that are no longer under warranty.

RMAs will only be authorized for current Actel devices. Devices that have been discontinued will not receive RMAs. All functional failure analysis requests must be initiated by opening a case with Actel Technical Support. Devices returned for failure analysis against an RMA should be in their original packaging and must have an RMA number issued by Actel.

Contacting the Customer Support Group

Highly skilled engineers staff the Customer Applications Center from 7:00 A.M. to 6:00 P.M., Pacific time, Monday through Friday. You can contact the center by one of the following methods:

Electronic Mail

You can communicate your technical questions to our email address and receive answers back by email, fax, or phone. Also, if you have design problems, you can email your design files to receive assistance. Actel monitors the email account throughout the day. When sending your request to us, please be sure to include your full name, company name, and contact information for efficient processing of your request. The technical support email address is tech@actel.com.

Telephone

Our Technical Support Hotline answers all calls. The center retrieves information, such as your name, company name, telephone number, and question. Once this is done, a case number is assigned. Then the center forwards the information to a queue where the first available applications engineer receives the data and returns your call. The phone hours are from 7:00 A.M. to 6:00 P.M., Pacific time, Monday through Friday.

The Customer Applications Center number is (800) 262-1060.

European customers can call +44 (0) 1256 305 600.

Related Documents

Below is a list of related documents, their location on the Actel website, and a brief summary of each document.

Application Notes

Programming Antifuse Devices

http://www.actel.com/documents/AntifuseProgram_AN.pdf

Implementation of Security in Actel's ProASIC and ProASIC^{PLUS} Flash-Based FPGAs

http://www.actel.com/documents/Flash_Security_AN.pdf

Handbook Documents

Security in Low-Power Flash Devices

http://www.actel.com/documents/LPD_Security_HBs.pdf

In-System Programming (ISP) of Actel's Low-Power Flash Devices Using FlashPro3

http://www.actel.com/documents/LPD_ISP_HBs.pdf

MicroProcessor Programming of Actel's Low-Power Flash Devices

http://www.actel.com/documents/LPD_Microprocessor_HBs.pdf

User's Guides

FlashPro Programmers

FlashPro3, FlashPro Lite, and FlashPro

http://www.actel.com/products/hardware/program_debug/flashpro/default.aspx

FlashPro User's Guide

http://www.actel.com/documents/FlashPro_UG.pdf

The FlashPro User's Guide includes hardware and software setup, self-test instructions, use instructions, and a troubleshooting / error message guide.

Silicon Sculptor 3 and Silicon Sculptor II

http://www.actel.com/products/hardware/program_debug/ss/default.aspx

Other Documents

http://www.actel.com/products/solutions/security/default.aspx#flashlock

The security resource center describes security in Actel Flash FPGAs.

Actel Quality and Reliability Guide

http://www.actel.com/documents/RelGuide.pdf

Part Number and Revision Date

This document was previously published as an application note describing features and functions of the device, and as such has now been incorporated into the device handbook format. No technical changes have been made to the content.

Part Number 51700094-013-2

Revised October 2008

List of Changes

The following table lists critical changes that were made in the current version of the chapter.

Previous Version	Changes in Current Version (v1.2)	Page
v1.1 (March 2008)	The "Programming Support in Flash Devices" section was revised to include new families and make the information more concise.	11-2
	Figure 11-1 · FlashPro Programming Setup and the "Programming Support in Flash Devices" section are new.	11-1, 11-2
	Table 11-6 · Programmer Device Support was updated to include A3PE600L with the other Pro ASIC3L devices, and the RT ProASIC3 family was added.	11-10
v1.0 (January 2008)	The "Flash Devices" section was updated to include the IGLOO PLUS family. The text, "Voltage switching is required in-system to switch from a 1.2 V core to 1.5 V core for programming" was revised to state, "Although the device can operate at 1.2 V core voltage, the device can only be reprogrammed when the core voltage is 1.5 V. Voltage switching is required in-system to switch from a 1.2 V supply (V_{CC}, V_{CCI}, and V_{JTAG}) to 1.5 V for programming."	11-4
	The ProASIC3L family was added to Table 11-6 · Programmer Device Support as a separate set of rows rather than combined with ProASIC3 and ProASIC3E devices. The IGLOO PLUS family was included, and AGL015 and A3P015 were added.	11-10

12 – Security in Low-Power Flash Devices

Security in Programmable Logic

The need for security on FPGA programmable logic devices (PLDs) has never been greater than today. If the contents of the FPGA can be read by an external source, the intellectual property (IP) of the system is vulnerable to unauthorized copying. Actel Fusion,® IGLOO,® and ProASIC®3 devices contain state-of-the-art circuitry to make the flash-based devices secure during and after programming. Low-power flash devices have a built-in 128-bit Advanced Encryption Standard (AES) decryption core (except for 15 k and 30 k gate devices). The decryption core facilitates secure in-system programming (ISP) of the FPGA core array fabric, the FlashROM, and the Flash Memory Blocks (FBs) in Fusion devices. The FlashROM, Flash Blocks, and FPGA core fabric can be programmed independently of each other, allowing the FlashROM or Flash Blocks to be updated without the need for change to the FPGA core fabric.

Actel has incorporated the AES decryption core into the low-power flash devices and has also included the Actel flash-based lock technology, FlashLock.® Together, they provide leading-edge security in a programmable logic device. Configuration data loaded into a device can be decrypted prior to being written to the FPGA core using the AES 128-bit block cipher standard. The AES encryption key is stored in on-chip, nonvolatile flash memory.

This document outlines the security features offered in low-power flash devices, some applications and uses, as well as the different software settings for each application.

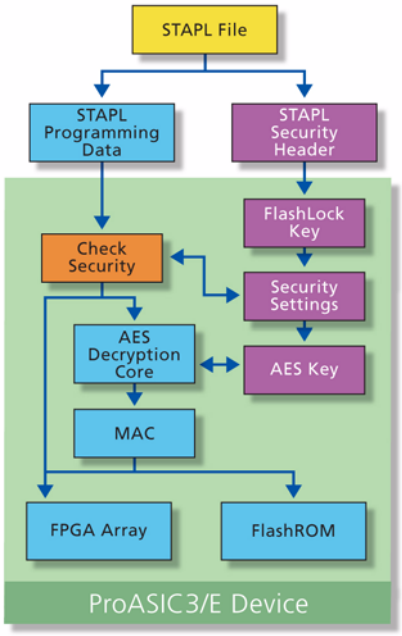


Figure 12-1 • **Overview on Security**

Security Support in Low-Power Devices

The low-power flash families listed in Table 12-1 support the security feature and the functions described in this document.

Table 12-1 • **Low-Power Flash Families**

Product Line	Family*	Description
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
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 PLUS	IGLOO FPGAs with enhanced I/O capabilities
ProASIC3	ProASIC3	Low-power, high-performance 1.5 V FPGAs
	ProASIC3E	Higher density ProASIC3 FPGAs with six PLLs and additional I/O standards
	ProASIC3L	ProASIC3 FPGAs supporting 1.2 V to 1.5 V with Flash*Freeze technology
	Automotive ProASIC3	ProASIC3 FPGAs qualified for automotive applications
	Military ProASIC3/EL	Military temperature A3PE600L, A3P1000, and A3PE3000L
	RT ProASIC3	Radiation-tolerant RT3PE600L and RT3PE3000L

*Note: *The family names link to the appropriate datasheet, including product brief, DC and switching characteristics, and packaging information.*

IGLOO Terminology

In documentation, the terms IGLOO families and IGLOO devices refer to all of the IGLOO products as listed in Table 12-1. Where the information applies to only one family or limited devices, these exclusions will be explicitly stated.

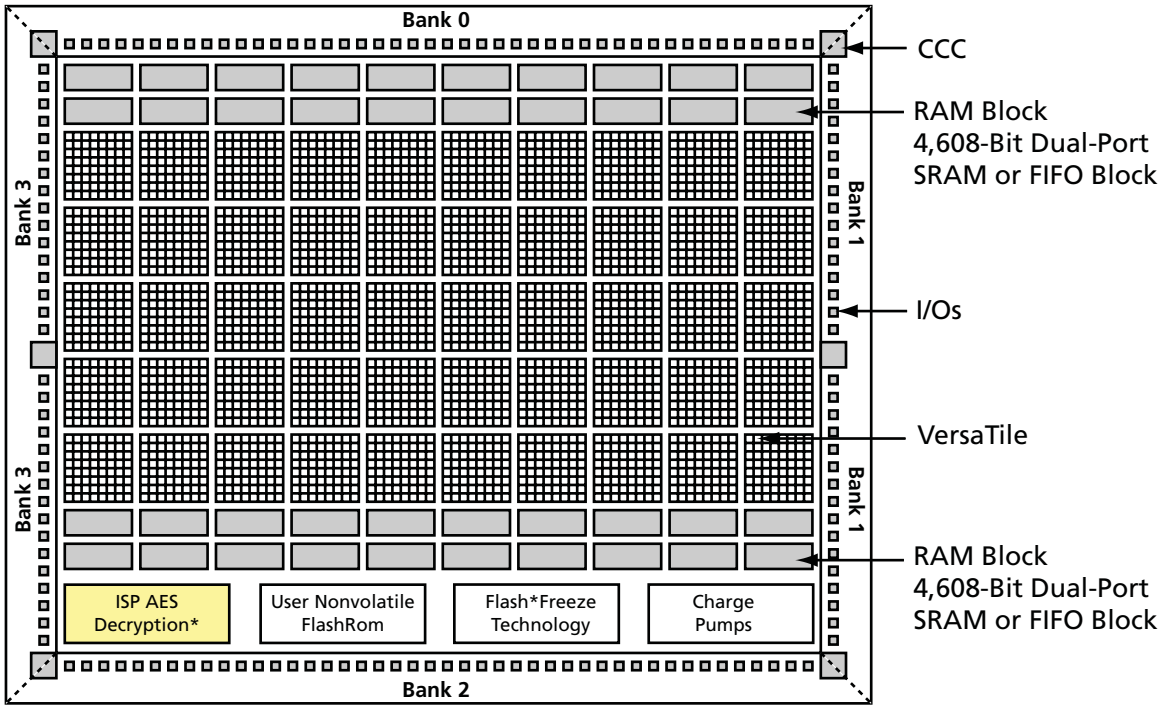
ProASIC3 Terminology

In documentation, the terms ProASIC3 families and ProASIC3 devices refer to all of the ProASIC3 families as listed in Table 12-1. Where the information applies to only one family or limited devices, these exclusions will be explicitly stated.

To further understand the differences between the IGLOO and ProASIC3 families, refer to the *Industry's Lowest Power FPGAs Portfolio*.

Security Architecture

Fusion, IGLOO, and ProASIC3 devices have been designed with the most comprehensive programming logic design security in the industry. In the architecture of these devices, security has been designed into the very fabric. The flash cells are located beneath seven metal layers, and the use of many device design and layout techniques makes invasive attacks difficult. Since device layers cannot be removed without disturbing the charge on the programmed (or erased) flash gates, devices cannot be easily deconstructed to decode the design. Low-power flash devices are unique in being reprogrammable and having inherent resistance to both invasive and noninvasive attacks on valuable IP. Secure, remote ISP is now possible with AES encryption capability for the programming file during electronic transfer. Figure 12-2 shows a view of the AES decryption core inside an IGLOO device; Figure 12-3 on page 12-4 shows the AES decryption core inside a Fusion device. The AES core is used to decrypt the encrypted programming file when programming.



Note: ISP AES Decryption is not supported by 15 k and 30 k gate devices. For details of other architecture features by device, refer to the appropriate family datasheet.

***Figure 12-2* • Block Representation of the AES Decryption Core in IGLOO and ProASIC3 Devices**

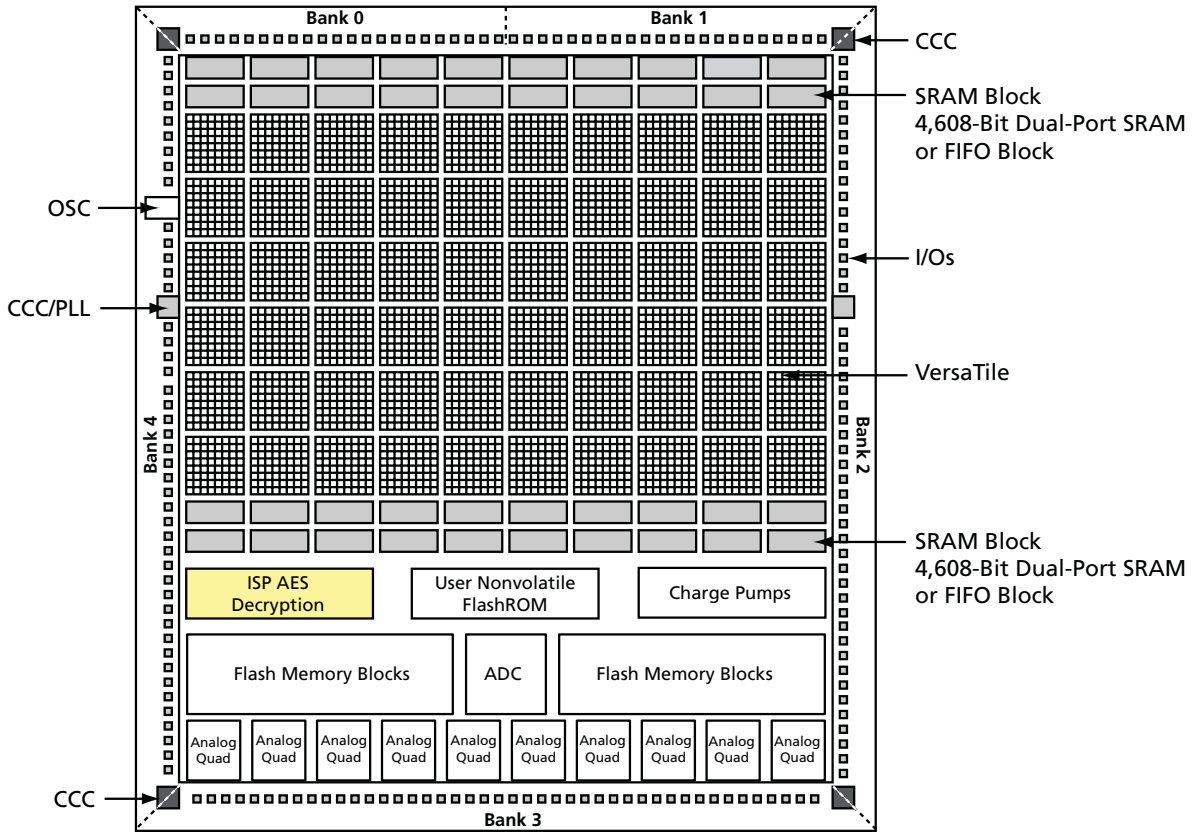


Figure 12-3 • **Block Representation of the AES Decryption Core in a Fusion AFS600 FPGA**

Security Features

IGLOO and ProASIC3 devices have two entities inside: FlashROM and the FPGA core fabric. Fusion devices contain three entities: FlashROM, FBs, and the FPGA core fabric. The parts can be programmed or updated independently with a STAPL programming file. The programming files can be AES-encrypted or plaintext. This allows maximum flexibility in providing security to the entire device. Refer to *FlashROM in Actel's Low-Power Flash Devices* for information on the FlashROM structure.

Unlike SRAM-based FPGA devices, which require a separate boot PROM to store programming data, low-power flash devices are nonvolatile, and the secured configuration data is stored in on-chip flash cells that are part of the FPGA fabric. Once programmed, this data is an inherent part of the FPGA array and does not need to be loaded at system power-up. SRAM-based FPGAs load the configuration bitstream upon power-up; therefore, the configuration is exposed and can be read easily.

The built-in FPGA core, FB, and FlashROM support programming files encrypted with the 128-bit AES (FIPS-192) block ciphers. The AES key is stored in dedicated, on-chip flash memory and can be programmed before the device is shipped to other parties (allowing secure remote field updates).

Security in ARM-Enabled Low-Power Flash Devices

There are slight differences between the regular flash devices and the ARM®-enabled flash devices, which have the M1 and M7 prefix.

The AES key is used by Actel and preprogrammed into the device to protect the ARM IP. As a result, the design is encrypted along with the ARM IP, according to the details below.

CoreMP7 Device Security

ARM7 (M7-enabled) devices are shipped with the following security features:

- FPGA array enabled for AES-encrypted programming and verification
- FlashROM enabled for plaintext Read and Write

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 Actel 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 Actel 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 3.4×10^{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.

Actel has implemented the 128-bit AES (Rijndael) algorithm in low-power flash devices. With this key size, there are approximately 3.4×10^{38} possible 128-bit keys. DES has a 56-bit key size, which provides approximately 7.2×10^{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.[1]

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 Actel 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 12-4 and Figure 12-4 on page 12-7 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

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.*

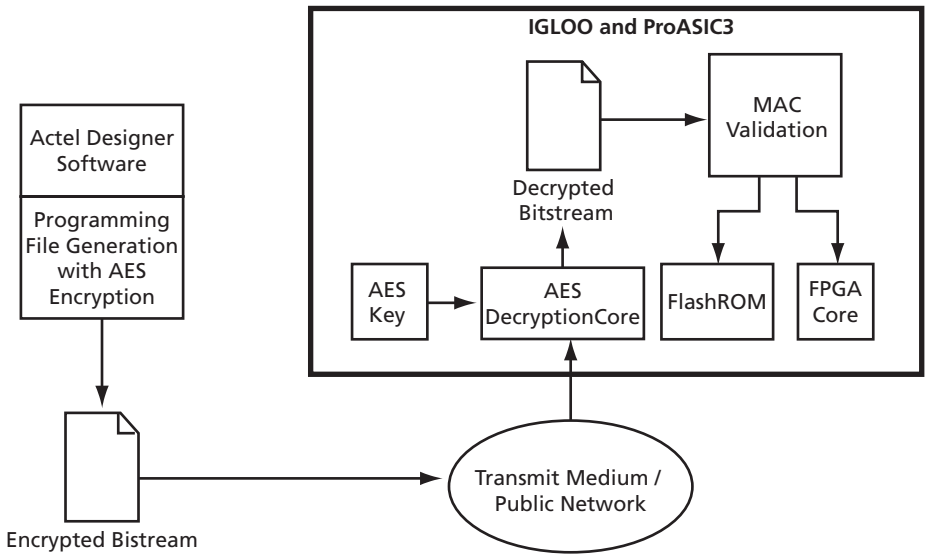


***Figure 12-4* • Example Application Scenario Using AES in IGLOO and ProASIC3 Devices**

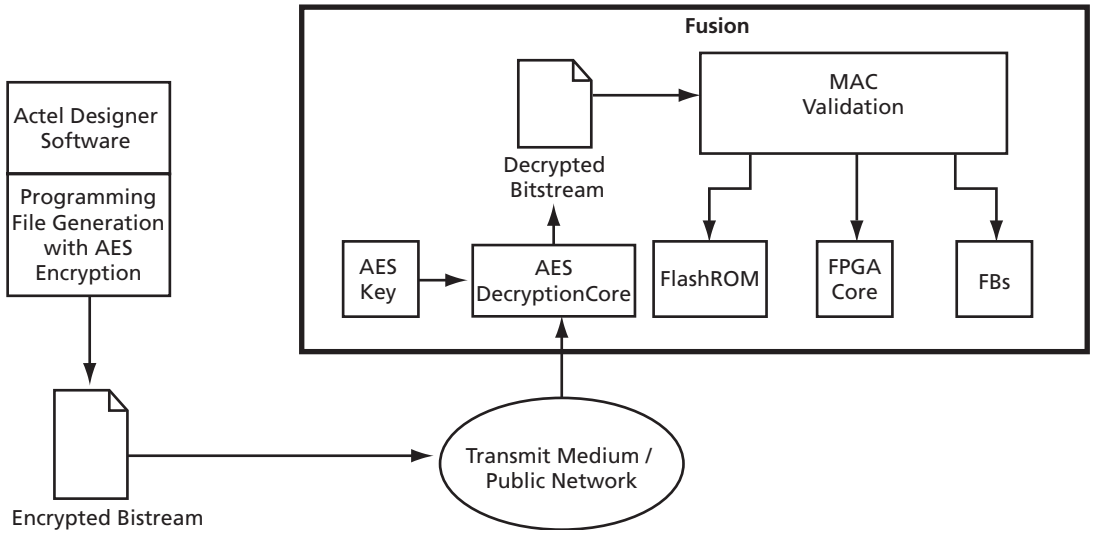


***Figure 12-5* • Example Application Scenario Using AES in Fusion Devices**

FlashLock

Additional Options for IGLOO and ProASIC3 Devices

The user also has the option of prohibiting Write operations to the FPGA array but allowing Verify operations on the FPGA array and/or Read operations on the FlashROM without the use of the FlashLock Pass Key. This option provides the user the freedom of verifying the FPGA array and/or reading the FlashROM contents after the device is programmed, without having to provide the FlashLock Pass Key. The user can incorporate AES encryption on the programming files to better enhance the level of security used.

Permanent Security Setting Options

In applications where a permanent lock is not desired, yet the security settings should not be modifiable, IGLOO and ProASIC3 devices can accommodate this requirement.

This application is particularly useful in cases where a device is located at a remote location and must be reprogrammed with a design or data update. Refer to the "Application 3: Nontrusted Environment—Field Updates/Upgrades" section on page 12-10 for further discussion and examples of how this can be achieved.

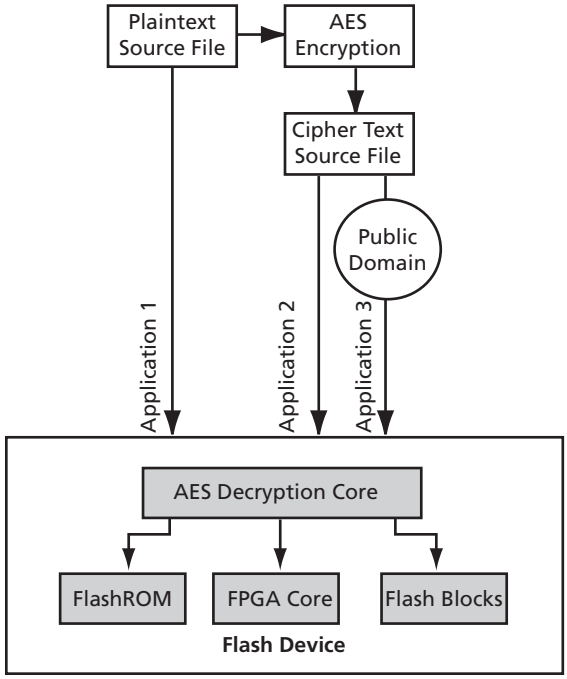
The user must be careful when considering the Permanent FlashLock or Permanent Security Settings option. Once the design is programmed with the permanent settings, it is not possible to reconfigure the security settings already employed on the device. Therefore, exercise careful consideration before programming permanent settings.

Permanent FlashLock

The purpose of the permanent lock feature is to provide the benefits of the highest level of security to IGLOO and ProASIC3 devices. If selected, the permanent FlashLock feature will create a permanent barrier, preventing any access to the contents of the device. This is achieved by permanently disabling Write and Verify access to the array, and Write and Read access to the FlashROM. After permanently locking the device, it has been effectively rendered one-time-programmable. This feature is useful if the intended applications do not require design or system updates to the device.

Security in Action

This section illustrates some applications of the security advantages of Actel's devices (Figure 12-6).



Note: Flash blocks are only used in Fusion devices.

Figure 12-6 • **Security Options**

Application 1: Trusted Environment

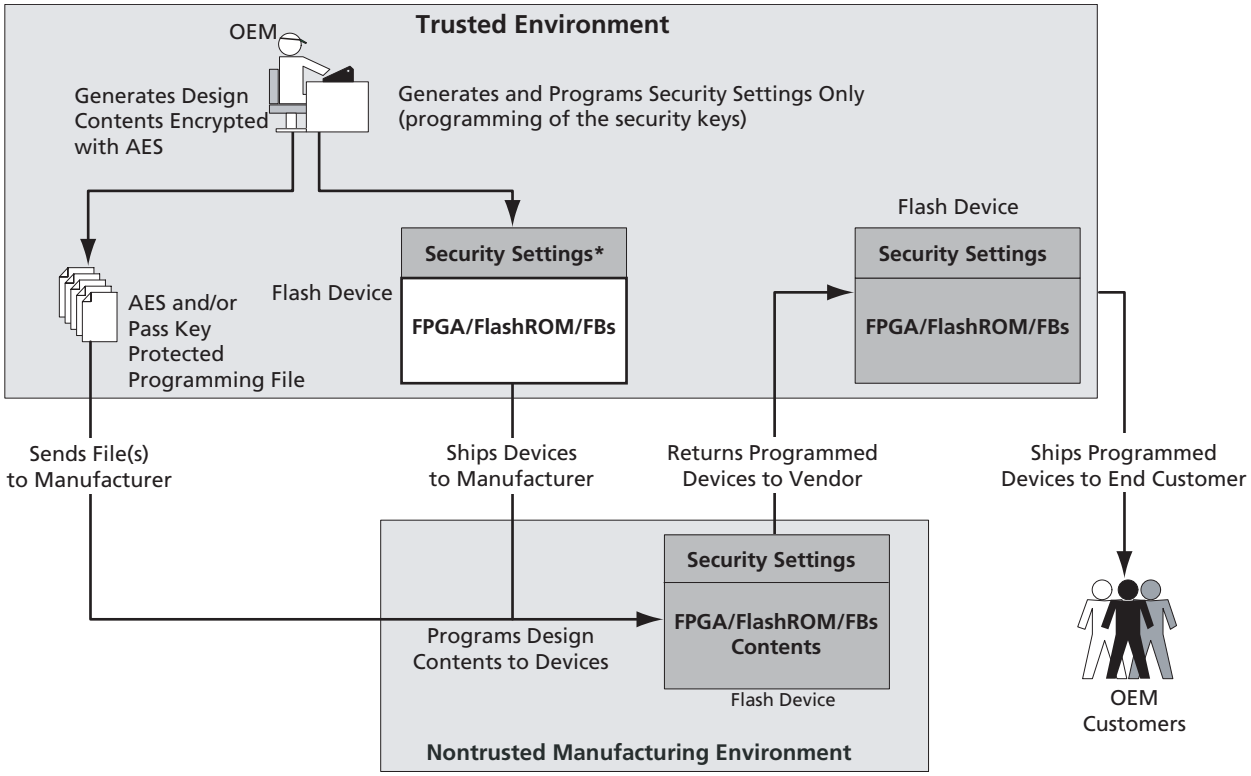
As illustrated in Figure 12-7 on page 12-10, this application allows the programming of devices at design locations where research and development take place. Therefore, encryption is not necessary and is optional to the user. This is often a secure way to protect the design, since the design program files are not sent elsewhere. In situations where production programming is not available at the design location, programming centers (such as Actel In-House Programming) provide a way of programming designs at an alternative, secure, and trusted location. In this scenario, the user generates a STAPL programming file from the Designer software in plaintext format, containing information on the entire design or the portion of the design to be programmed. The user can choose to employ the FlashLock Pass Key feature with the design. Once the design is programmed to unprogrammed devices, the design is protected by this FlashLock Pass Key. If no future programming is needed, the user can consider permanently securing the IGLOO and ProASIC3 device, as discussed in the "Permanent FlashLock" section on page 12-8.

Application 2: Nontrusted Environment—Unsecured Location

Often, programming of devices is not performed in the same location as actual design implementation, to reduce manufacturing cost. Overseas programming centers and contract manufacturers are examples of this scenario.

To achieve security in this case, the AES key and the FlashLock Pass Key can be initially programmed in-house (trusted environment). This is done by generating a programming file with only the security settings and no design contents. The design FPGA core, FlashROM, and (for Fusion) FB contents are generated in a separate programming file. This programming file must be set with the same AES key that was used to program to the device previously so the device will correctly decrypt this encrypted programming file. As a result, the encrypted design content programming file can

be safely sent off-site to nontrusted programming locations for design programming. Figure 12-7 shows a more detailed flow for this application.



Notes:

1. *Programmed portion indicated with dark gray.*
2. *Programming of FBs applies to Fusion only*

Figure 12-7 • **Application 2: Device Programming in a Nontrusted Environment**

Application 3: Nontrusted Environment—Field Updates/Upgrades

Programming or reprogramming of devices may occur at remote locations. Reconfiguration of devices in consumer products/equipment through public networks is one example. Typically, the remote system is already programmed with particular design contents. When design update (FPGA array contents update) and/or data upgrade (FlashROM and/or FB contents upgrade) is necessary, an updated programming file with AES encryption can be generated, sent across public networks, and transmitted to the remote system. Reprogramming can then be done using this AES-encrypted programming file, providing easy and secure field upgrades. Low-power flash devices support this secure ISP using AES. The detailed flow for this application is shown in Figure 12-8 on page 12-11. Refer to *Microprocessor Programming of Actel's Low-Power Flash Devices* for more information.

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

For example, if the designer wants to use an AES key for the FPGA array and the FlashROM, **Permanent** needs to be chosen for this setting. At first, the user would do this by choosing the options to use an AES key for the FPGA array and the FlashROM, and then choosing **Permanently lock the security settings**. A unique AES key would be chosen. Once this programming file is

generated and programmed to the devices, the AES key is permanently stored in the on-chip memory, where it is secured safely. The devices would be sent to distant locations for the intended application. When an update is needed, a new programming file must be generated. The programming file must use the same AES key for encryption; otherwise, the authentication will fail and the file will not get programmed in the device.

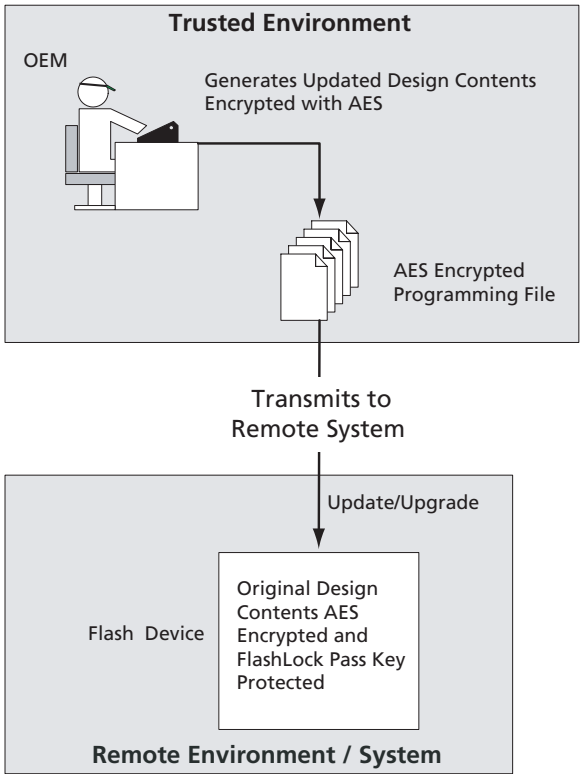


Figure 12-8 • **Application 3: Nontrusted Environment—Field Updates/Upgrades**

FlashROM Security Use Models

Each of the subsequent sections describes in detail the available selections in Actel Designer as an aid to understanding security applications and generating appropriate programming files for those applications. Before proceeding, it is helpful to review Figure 12-7 on page 12-10, which gives a general overview of the programming file generation flow within the Designer software as well as what occurs during the device programming stage. Specific settings are discussed in the following sections.

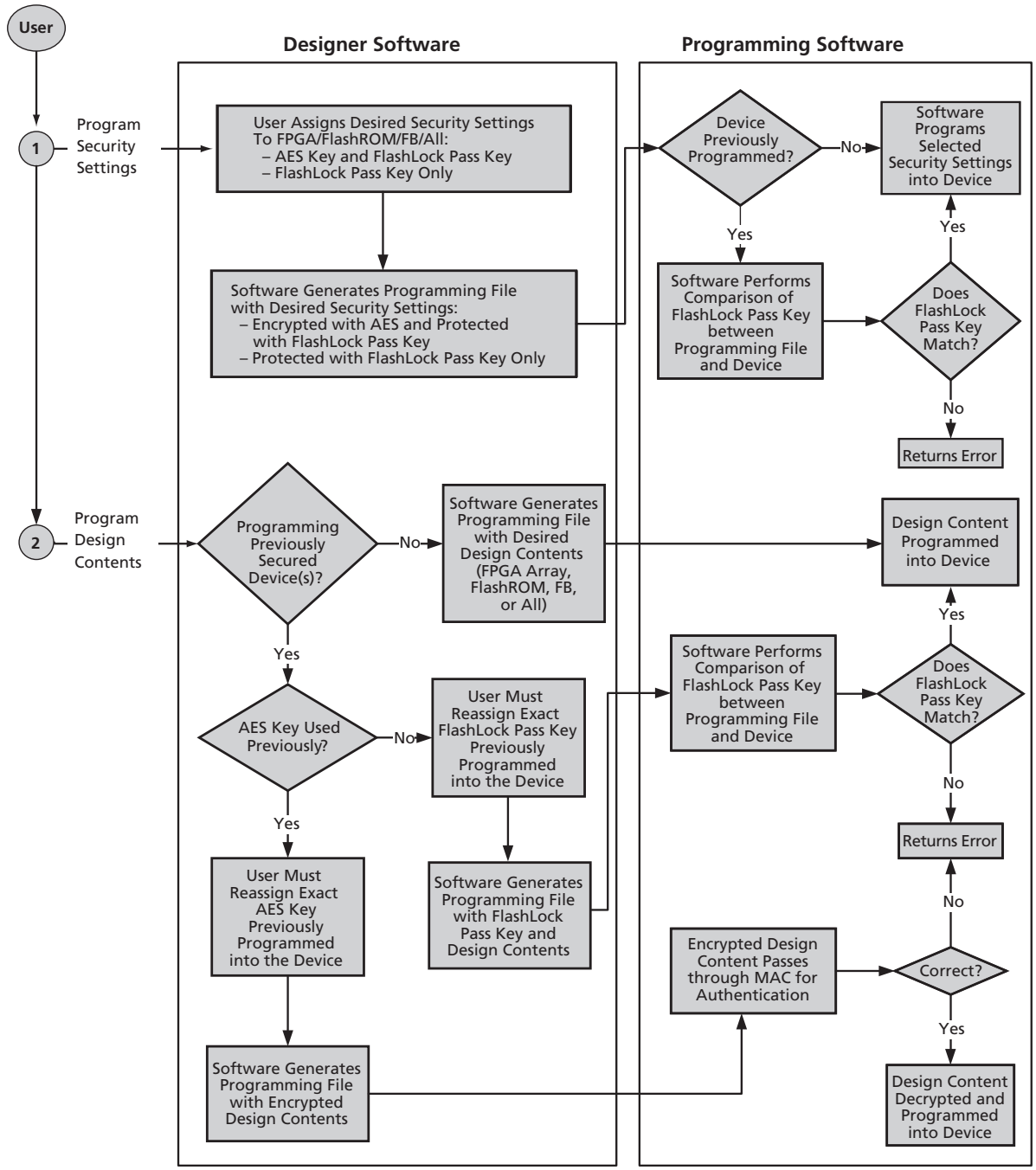
In Figure 12-7 on page 12-10, the flow consists of two sub-flows. Sub-flow 1 describes programming security settings to the device only, and sub-flow 2 describes programming the design contents only.

In Application 1, described in the "Application 1: Trusted Environment" section on page 12-9, the user does not need to generate separate files but can generate one programming file containing both security settings and design contents. Then programming of the security settings and design contents is done in one step. Both sub-flow 1 and sub-flow 2 are used.

In Application 2, described in the "Application 2: Nontrusted Environment—Unsecured Location" section on page 12-9, the trusted site should follow sub-flows 1 and 2 separately to generate two separate programming files. The programming file from sub-flow 1 will be used at the trusted site to program the device(s) first. The programming file from sub-flow 2 will be sent off-site for production programming.

In Application 3, described in the "Application 3: Nontrusted Environment—Field Updates/Upgrades" section on page 12-10, typically only sub-flow 2 will be used because only updates to the design content portion are needed and no security settings need to be changed.

In the event that update of the security settings is necessary, see the "Reprogramming Devices" section on page 12-21 for details. For more information on programming low-power flash devices, refer to *In-System Programming (ISP) of Actel's Low-Power Flash Devices Using FlashPro3*.



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

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

Generating Programming Files

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

As discussed in the "Application 1: Trusted Environment" section on page 12-9, in a trusted environment, the user can choose to program the device with plaintext bitstream content. It is possible to use plaintext for programming even when the FlashLock Pass Key option has been selected. In this application, it is not necessary to employ AES encryption protection. For AES encryption settings, refer to the next sections.

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

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

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

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

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

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

For this scenario, generate the programming file as follows:

1. Select the **Silicon features to be programmed** (Security Settings, FPGA Array, FlashROM, Flash Memory Block), as shown in Figure 12-10 on page 12-14 and Figure 12-11 on page 12-14. Click **Next**.

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

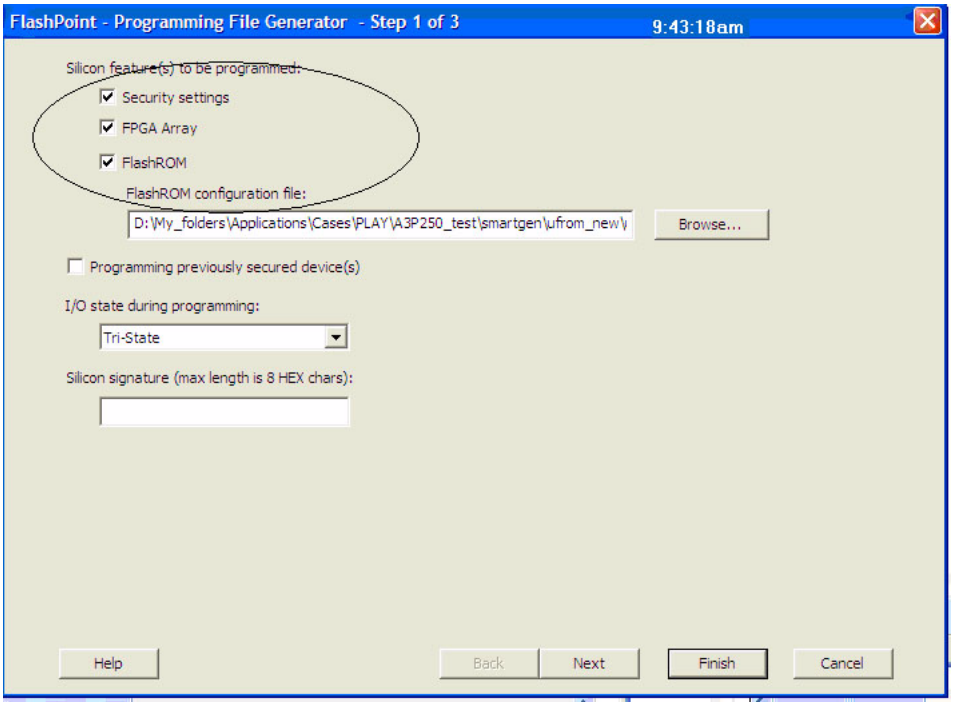


***Figure 12-10* • All Silicon Features Checked for IGLOO and ProASIC3 Devices**

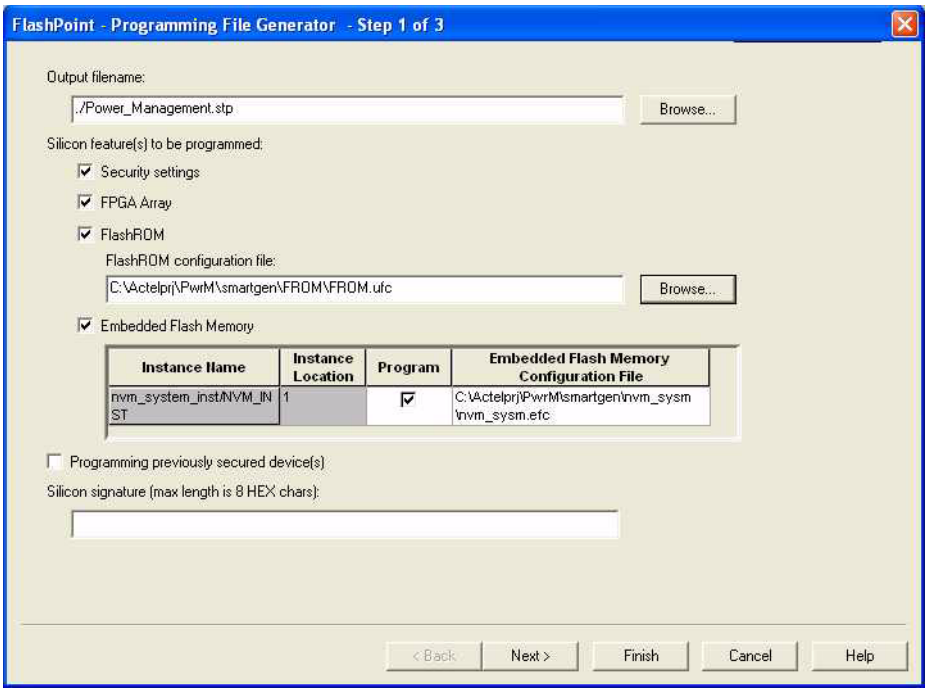


***Figure 12-11* • All Silicon Features Checked for Fusion**

2. Choose the appropriate security level setting and enter a FlashLock Pass Key. The default is the **Medium** security level (Figure 12-12). Click **Next**.

   If you want to select different options for the FPGA and/or FlashROM, this can be set by clicking **Custom Level**. Refer to the "Advanced Options" section on page 12-22 for different custom security level options and descriptions of each.

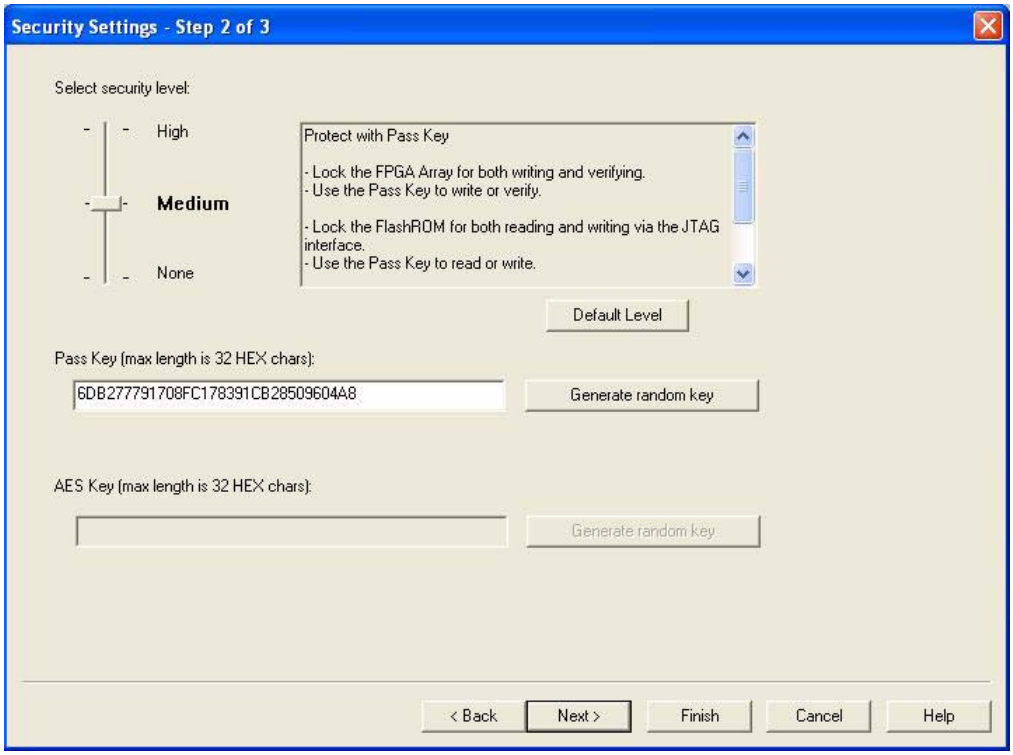


Figure 12-12 • **Medium Security Level Selected for Low-Power Flash Devices**

3. Choose the desired settings for the FlashROM configurations to be programmed (Figure 12-13). Click **Finish** to generate the STAPL programming file for the design.

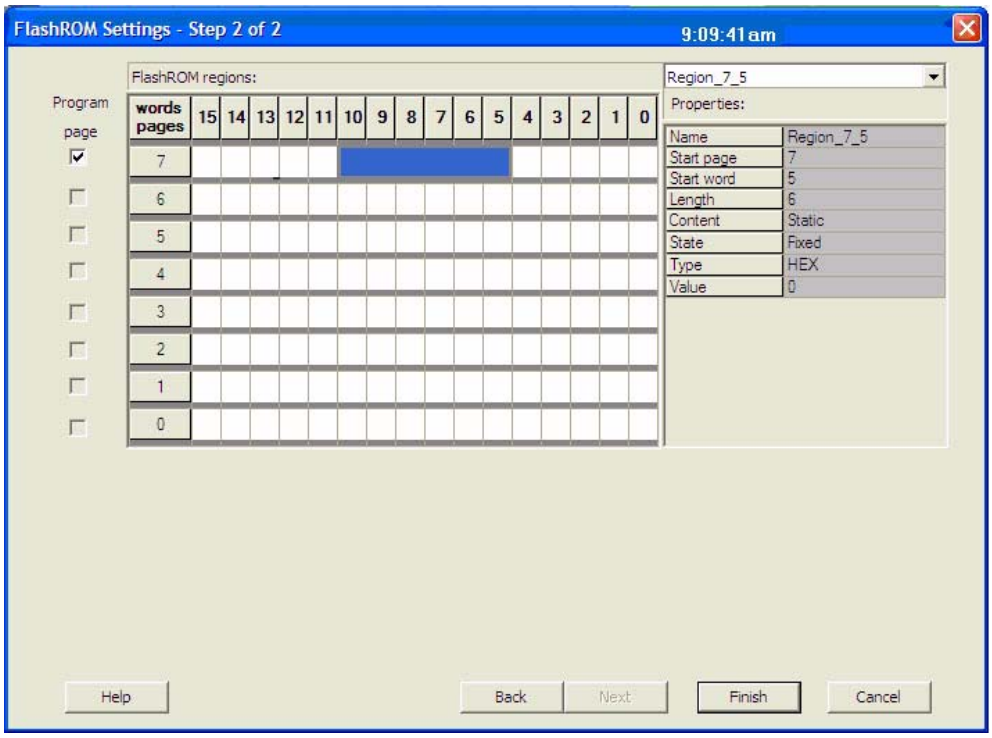


Figure 12-13 • **FlashROM Configuration Settings for Low-Power Flash Devices**

Generation of Security Header Programming File Only—Application 2

As mentioned in the "Application 2: Nontrusted Environment—Unsecured Location" section on page 12-9, the designer may employ FlashLock Pass Key protection or FlashLock Pass Key with AES encryption on the device before sending it to a nontrusted or unsecured location for device programming. To achieve this, the user needs to generate a programming file containing only the security settings desired (Security Header programming file).

Note: If AES encryption is configured, FlashLock Pass Key protection must also be configured.

The available security options are indicated in Table 12-4 and Table 12-5 on page 12-17.

Table 12-4 • **FlashLock Security Options for IGLOO and ProASIC3**

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

Table 12-5 • **FlashLock Security Options for Fusion**

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

For this scenario, generate the programming file as follows:

1. Select only the **Security settings** option, as indicated in Figure 12-14 and Figure 12-15 on page 12-18. Click **Next**.

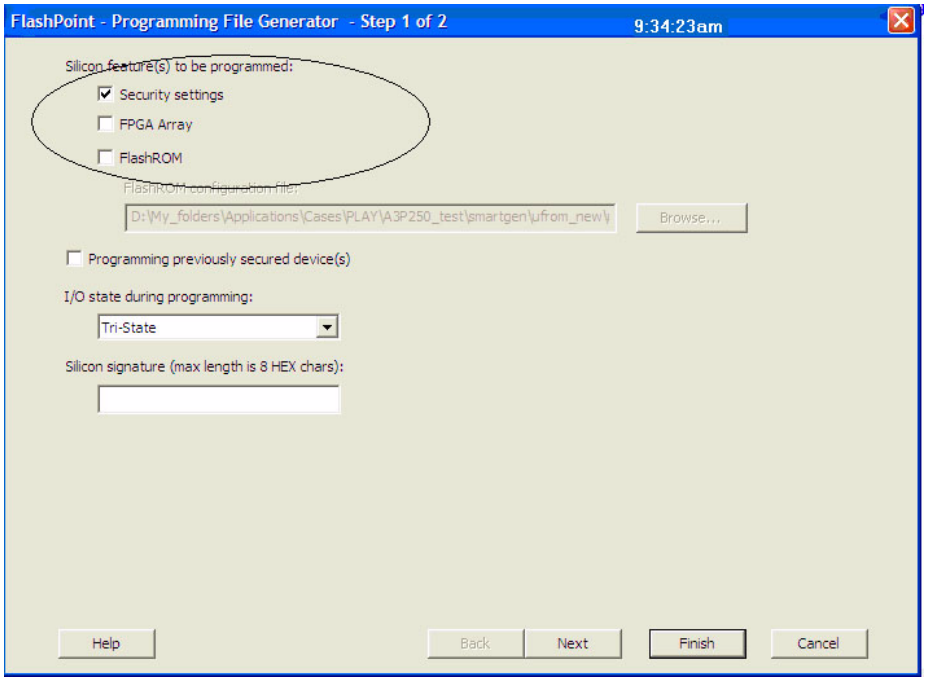


Figure 12-14 • **Programming IGLOO and ProASIC3 Security Settings Only**

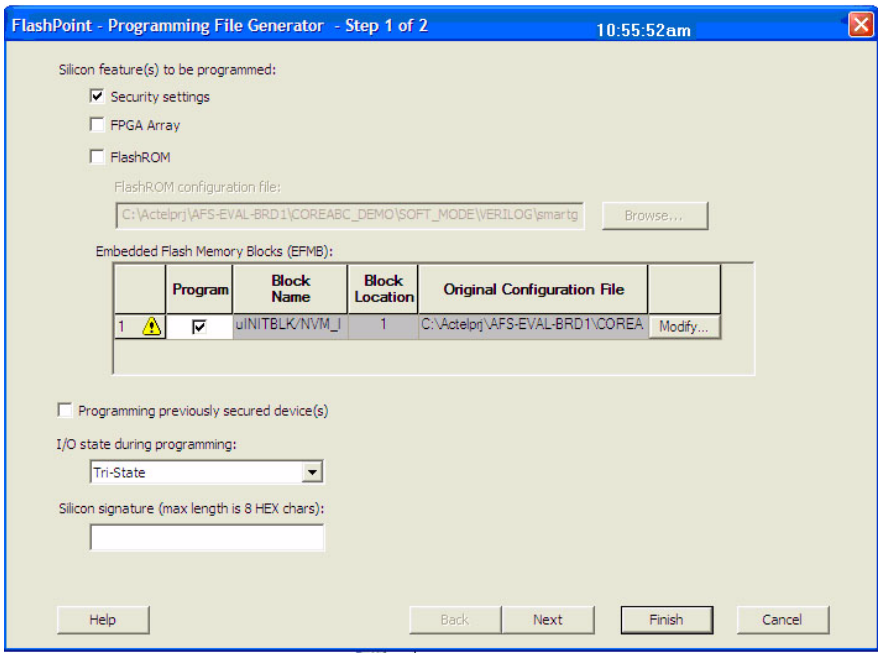


Figure 12-15 • **Programming Fusion Security Settings Only**

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.

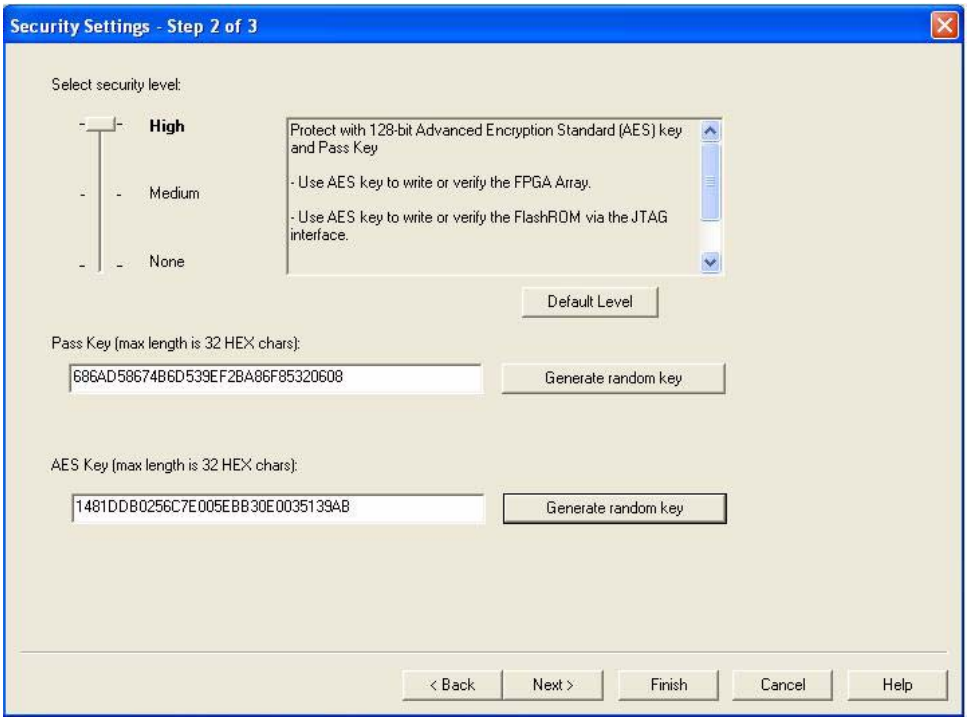


Figure 12-16 • **High Security Level to Implement FlashLock Pass Key and AES Key Protection**

Table 12-6 and Table 12-7 show all available options. If you want to implement custom levels, refer to the "Advanced Options" section on page 12-22 for information on each option and how to set it.

3. When done, click **Finish** to generate the Security Header programming file.

Table 12-6 • **All IGLOO and ProASIC3 Header File Security Options**

Security Option	FlashROM Only	FPGA Core Only	Both FlashROM and FPGA
No AES / No FlashLock	✓	✓	✓
FlashLock only	✓	✓	✓
AES and FlashLock	✓	✓	✓

Note: ✓ = options that may be used

Table 12-7 • **All Fusion Header File Security Options**

Security Option	FlashROM Only	FPGA Core Only	FB Core Only	All
No AES / No FlashLock	✓	✓	✓	✓
FlashLock	✓	✓	✓	✓
AES and FlashLock	✓	✓	✓	✓

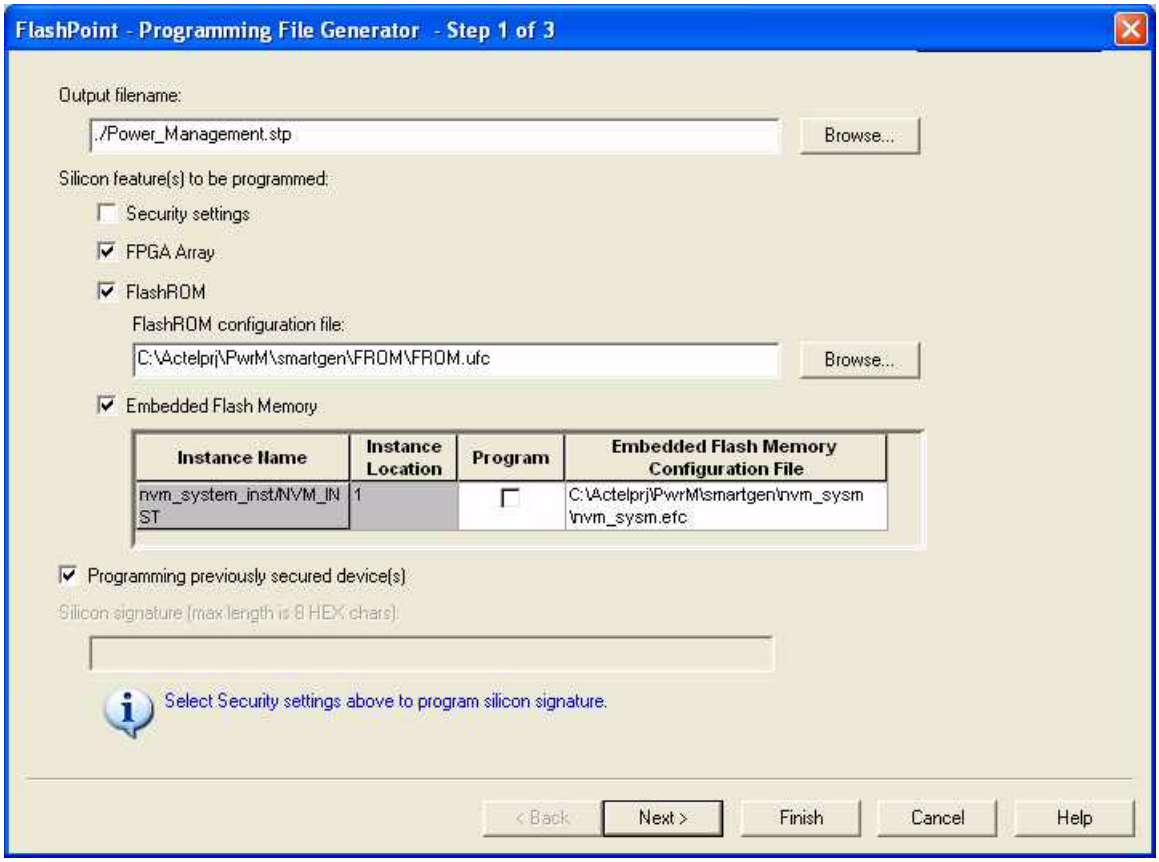
Generation of Programming Files with AES Encryption—Application 3

This section discusses how to generate design content programming files needed specifically at unsecured or remote locations to program devices with a security header (FlashLock Pass Key and AES key) already programmed ("Application 2: Nontrusted Environment—Unsecured Location" section on page 12-9 and "Application 3: Nontrusted Environment—Field Updates/Upgrades" section on page 12-10). In this case, the encrypted programming file must correspond to the AES key already programmed into the device. If AES encryption was previously selected to encrypt the FlashROM, FB, and FPGA array, AES encryption must be set when generating the programming file for them. AES encryption can be applied to the FlashROM only, the FB only, the FPGA array only, or all. The user must ensure both the FlashLock Pass Key and the AES key match those already programmed to the device(s), and all security settings must match what was previously programmed. Otherwise, the encryption and/or device unlocking will not be recognized when attempting to program the device with the programming file.

The generated programming file will be AES-encrypted.

In this scenario, generate the programming file as follows:

1. Deselect the **Security settings** and select the portion of the device to be programmed (Figure 12-17 on page 12-20). Select **Programming previously secured device(s)**. Click **Next**.



Note: The settings in this figure are used to show the generation of an AES-encrypted programming file for the FPGA array, FlashROM, and FB contents. One or all locations may be selected for encryption.

Figure 12-17 • **Settings to Program a Device Secured with FlashLock and using AES Encryption**

Choose the **High** security level to reprogram devices using both the FlashLock Pass Key and AES key protection (Figure 12-18 on page 12-21). Enter the AES key and click **Next**.

A device that has already been secured with FlashLock and has an AES key loaded must recognize the AES key to program the device and generate a valid bitstream in authentication. The FlashLock Key is only required to unlock the device and change the security settings.

This is what makes it possible to program in an untrusted environment. The AES key is protected inside the device by the FlashLock Key, so you can only program if you have the correct AES key. In fact, the AES key is not in the programming file either. It is the key used to encrypt the data in the file. The same key previously programmed with the FlashLock Key matches to decrypt the file.

If you had an AES-encrypted file programmed to a device without FlashLock, this would not be secure, since without FlashLock to protect the AES key, you could simply reprogram the AES key first, then program with any AES key you wanted or no AES key at all. This option is therefore not available in the software.

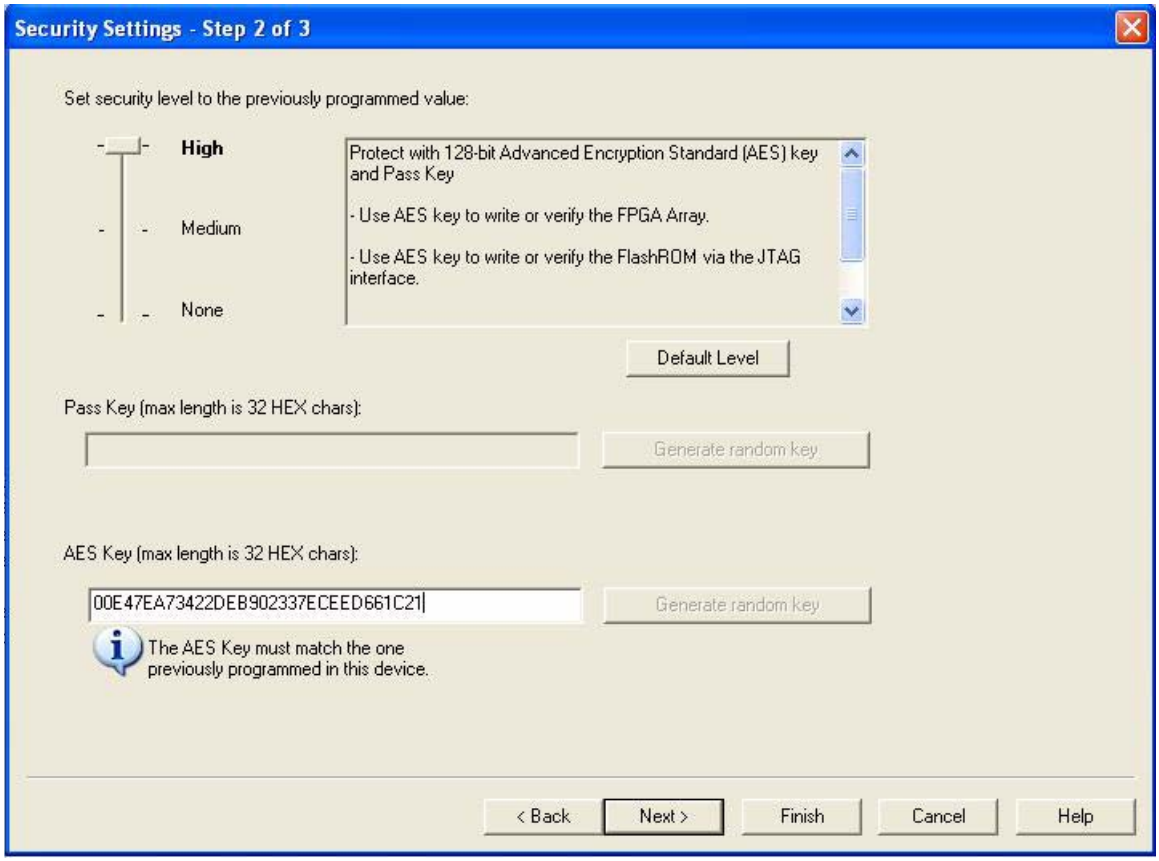


Figure 12-18 • **Security Level Set High to Reprogram Device with AES Key**

Programming with this file is intended for an unsecured environment. The AES key encrypts the programming file with the same AES key already used in the device and utilizes it to program the device.

Reprogramming Devices

Previously programmed devices can be reprogrammed using the steps in the "Generation of the Programming File in a Trusted Environment—Application 1" section on page 12-13 and "Generation of Security Header Programming File Only—Application 2" section on page 12-16. In the case where a FlashLock Pass Key has been programmed previously, the user must generate the new programming file with a FlashLock Pass Key that matches the one previously programmed into the device. The software will check the FlashLock Pass Key in the programming file against the FlashLock Pass Key in the device. The keys must match before the device can be unlocked to perform further programming with the new programming file.

Figure 12-10 on page 12-14 and Figure 12-11 on page 12-14 show the option **Programming previously secured device(s)**, which the user should select before proceeding. Upon going to the next step, the user will be notified that the same FlashLock Pass Key needs to be entered, as shown in Figure 12-19 on page 12-22.

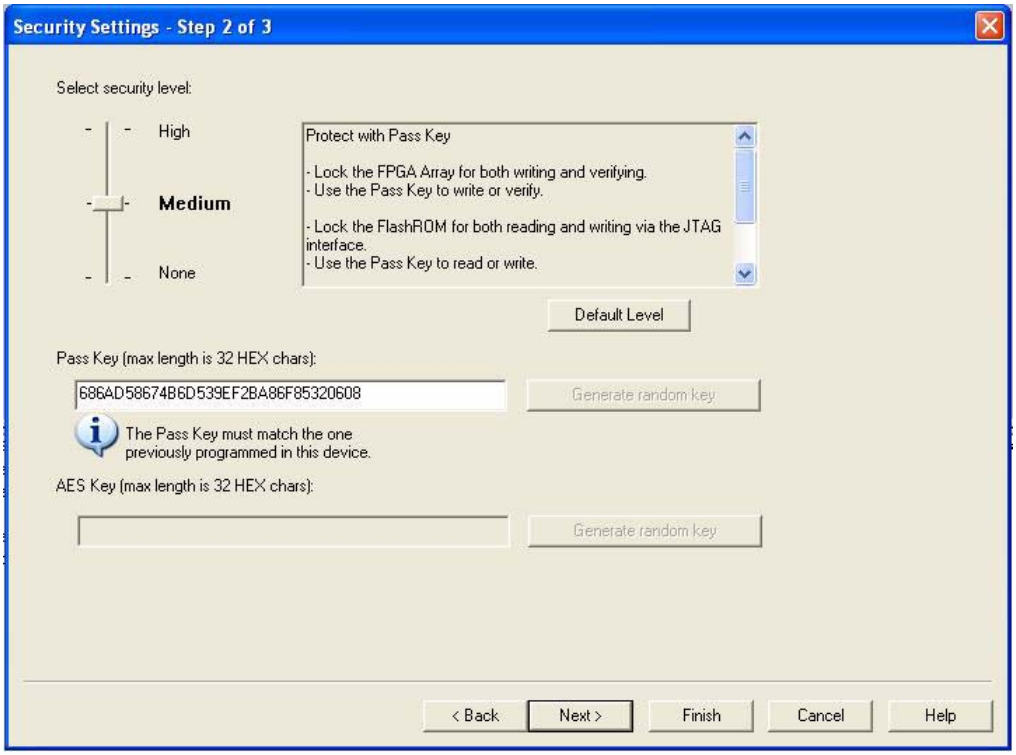


Figure 12-19 • **FlashLock Pass Key, Previously Programmed Devices**

It is important to note that when the security settings need to be updated, the user also needs to select the **Security settings** check box in Step 1, as shown in Figure 12-10 on page 12-14 and Figure 12-11 on page 12-14, to modify the security settings. The user must consider the following:

- If only a new AES key is necessary, the user must re-enter the same Pass Key previously programmed into the device in Designer and then generate a programming file with the same Pass Key and a different AES key. This ensures the programming file can be used to access and program the device and the new AES key.
- If a new Pass Key is necessary, the user can generate a new programming file with a new Pass Key (with the same or a new AES key if desired). However, for programming, the user must first load the original programming file with the Pass Key that was previously used to unlock the device. Then the new programming file can be used to program the new security settings.

Advanced Options

As mentioned, there may be applications where more complicated security settings are required. The "Custom Security Levels" section in the *FlashPro User's Guide* describes different advanced options available to aid the user in obtaining the best available security settings.

Programming File Header Definition

In each STAPL programming file generated, there will be information about how the AES key and FlashLock Pass Key are configured. Table 12-8 shows the header definitions in STAPL programming files for different security levels.

Table 12-8 • **STAPL Programming File Header Definitions by Security Level**

Security Level	STAPL File Header Definition
No security (no FlashLock Pass Key or AES key)	`NOTE "SECURITY" "Disable";`
FlashLock Pass Key with no AES key	`NOTE "SECURITY" "KEYED ";`
FlashLock Pass Key with AES key	`NOTE "SECURITY" "KEYED ENCRYPT ";`
Permanent Security Settings option enabled	`NOTE "SECURITY" "PERMLOCK ENCRYPT ";`
AES-encrypted FPGA array (for programming updates)	`NOTE "SECURITY" "ENCRYPT CORE ";`
AES-encrypted FlashROM (for programming updates)	`NOTE "SECURITY" "ENCRYPT FROM ";`
AES-encrypted FPGA array and FlashROM (for programming updates)	`NOTE "SECURITY" "ENCRYPT FROM CORE ";`

Example File Headers

STAPL Files Generated with FlashLock Key and AES Key Contain Key Information

- FlashLock Key / AES key indicated in STAPL file header definition
- Intended ONLY for secured/trusted environment programming applications

```
================================================
NOTE "CREATOR" "Designer Version: 6.1.1.108";
NOTE "DEVICE" "A3PE600";
NOTE "PACKAGE" "208 PQFP";
NOTE "DATE" "2005/04/08";
NOTE "STAPL_VERSION" "JESD71";
NOTE "IDCODE" "$123261CF";
NOTE "DESIGN" "counter32";
NOTE "CHECKSUM" "$EDB9";
NOTE "SAVE_DATA" "FRomStream";
NOTE "SECURITY" "KEYED ENCRYPT ";
NOTE "ALG_VERSION" "1";
NOTE "MAX_FREQ" "20000000";
NOTE "SILSIG" "$00000000";
NOTE "PASS_KEY" "$00123456789012345678901234567890";
NOTE "AES_KEY" "$ABCDEFABCDEFABCDEFABCDEFABCDEFAB";
================================================
```

STAPL File with AES Encryption

- Does not contain AES key / FlashLock Key information
- Intended for transmission through web or service to unsecured locations for programming

```
============================================
NOTE "CREATOR" "Designer Version: 6.1.1.108";
NOTE "DEVICE" "A3PE600";
NOTE "PACKAGE" "208 PQFP";
NOTE "DATE" "2005/04/08";
NOTE "STAPL_VERSION" "JESD71";
NOTE "IDCODE" "$123261CF";
NOTE "DESIGN" "counter32";
NOTE "CHECKSUM" "$EF57";
NOTE "SAVE_DATA" "FRomStream";
NOTE "SECURITY" "ENCRYPT FROM CORE ";
NOTE "ALG_VERSION" "1";
NOTE "MAX_FREQ" "20000000";
NOTE "SILSIG" "$00000000";
```

Conclusion

The new and enhanced security features offered in Actel Fusion, IGLOO, and ProASIC3 devices provide state-of-the-art security to designs programmed into these flash-based devices. Actel low-power flash devices employ the encryption standard used by NIST and the U.S. government—AES using the 128-bit Rijndael algorithm.

The combination of an on-chip AES decryption engine and Actel FlashLock technology provides the highest level of security against invasive attacks and design theft, implementing the most robust and secure ISP solution. These security features protect IP within the FPGA and protect the system from cloning, wholesale “black box” copying of a design, invasive attacks, and explicit IP or data theft.

Glossary

Term	Explanation
Security Header programming file	Programming file used to program the FlashLock Pass Key and/or AES key into the device to secure the FPGA, FlashROM, and/or FBs.
AES (encryption) key	128-bit key defined by the user when the AES encryption option is set in the Actel Designer software when generating the programming file.
FlashLock Pass Key	128-bit key defined by the user when the FlashLock option is set in the Actel Designer software when generating the programming file. The FlashLock Key protects the security settings programmed to the device. Once a device is programmed with FlashLock, whatever settings were chosen at that time are secure.
FlashLock	The combined security features that protect the device content from attacks. These features are the following: • Flash technology that does not require an external bitstream to program the device • FlashLock Pass Key that secures device content by locking the security settings and preventing access to the device as defined by the user • AES key that allows secure, encrypted device reprogrammability

References

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.

Related Documents

Handbook Documents

FlashROM in Actel's Low-Power Flash Devices

http://www.actel.com/documents/LPD_FlashROM_HBs.pdf

Programming ProASIC3/E Using a Microprocessor

http://www.actel.com/documents/LPD_Microprocessor_HBs.pdf

In-System Programming (ISP) of Actel's Low-Power Flash Devices Using FlashPro3

http://www.actel.com/documents/LPD_ISP_HBs.pdf

User's Guides

FlashPro User's Guide

http://www.actel.com/documents/flashpro_ug.pdf

Part Number and Revision Date

This document contains content extracted from the Device Architecture section of the datasheet, combined with content previously published as an application note describing features and functions of the device. To improve usability for customers, the device architecture information has now been combined with usage information, to reduce duplication and possible inconsistencies in published information. No technical changes were made to the datasheet content unless explicitly listed. Changes to the application note content were made only to be consistent with existing datasheet information.

Part Number 51700094-014-3

Revised October 2008

List of Changes

The following table lists critical changes that were made in the current version of the chapter.

Previous Version	Changes in Current Version (v1.3)	Page
v1.2 (June 2008)	The "Security Support in Low-Power Devices" section was revised to include new families and make the information more concise.	12-2
v1.1 (March 2008)	The following changes were made to the family descriptions in Table 12-1 · Low-Power Flash Families: • ProASIC3L was updated to include 1.5 V. • The number of PLLs for ProASIC3E was changed from five to six.	12-2
v1.0 (January 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.	12-2

13 – In-System Programming (ISP) of Actel's Low-Power Flash Devices Using FlashPro3

Introduction

Actel's low-power flash devices are all in-system programmable. This document describes the general requirements for programming a device and specific requirements for the FlashPro3 programmer.

Fusion, IGLOO,® and ProASIC®3 devices offer a low-power, single-chip, live-at-power-up solution with the ASIC advantages of security and low unit cost through nonvolatile flash technology. Each device contains 1 kbit of on-chip, user-accessible, nonvolatile FlashROM. The FlashROM can be used in diverse system applications such as Internet Protocol (IP) addressing, user system preference storage, device serialization, or subscription-based business models. Fusion, IGLOO, and ProASIC3 devices offer the best in-system programming (ISP) solution, FlashLock® security features, and AES-decryption-based ISP.

ISP Architecture

Low-power flash devices support ISP via JTAG and require a single V_{PUMP} voltage of 3.3 V during programming. In addition, programming via a microcontroller in a target system is also supported.

Refer to *Microprocessor Programming of Actel's Low-Power Flash Devices*.

Family-specific support:

- Fusion, ProASIC3, and ProASIC3E devices support ISP.
- ProASIC3L devices operate using a 1.2 V core voltage and support ISP at 1.5 V only. Voltage switching is required in-system to switch from a 1.2 V core to 1.5 V core for programming.
- IGLOO and IGLOOe V5 devices can be programmed in-system when the device is using a 1.5 V supply voltage to the FPGA core.
- IGLOO, IGLOO PLUS, and IGLOOe V2 devices can operate using either a 1.2 V core voltage or a 1.5 V core voltage. Although the device can operate at 1.2 V core voltage, the device can only be reprogrammed when all supplies (V_{CC}, V_{CCI}, and V_{JTAG}) are at 1.5 V. Voltage switching is required in-system to switch from a 1.2 V core to 1.5 V core for programming.

IGLOO devices cannot be programmed in-system when the device is in Flash*Freeze mode. The device should exit Flash*Freeze mode and be in normal operation for programming to start. Programming operations in IGLOO devices can be achieved when the device is in normal operating mode and a 1.5 V core voltage is used.

JTAG 1532

Fusion, IGLOO, and ProASIC3 devices support the JTAG-based IEEE 1532 standard for ISP. To start JTAG operations, the IGLOO device should exit Flash*Freeze™ mode and be in normal operation before starting to send JTAG commands to the device. As part of this support, when a device is in an unprogrammed state, all user I/O pins are disabled. This is achieved by keeping the global IO_EN signal deactivated, which also has the effect of disabling the input buffers. The SAMPLE/PRELOAD instruction captures the status of pads in parallel and shifts them out as new data is shifted in for loading into the Boundary Scan Register (BSR). When the device is in an unprogrammed state, the SAMPLE/PRELOAD instruction has no effect on I/O status; however, it will continue to shift in new data to be loaded into the BSR. Therefore, when SAMPLE/PRELOAD is used on an unprogrammed device, the BSR will be loaded with undefined data. For JTAG timing information on setup, hold, and fall times, refer to the *FlashPro User's Guide*.

ISP Support in Low-Power Devices

The low-power flash families listed in Table 13-1 support the ISP feature and the functions described in this document.

Table 13-1 • **Low-Power Flash Families**

Product Line	Family*	Description
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
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 PLUS	IGLOO FPGAs with enhanced I/O capabilities
ProASIC3	ProASIC3	Low-power, high-performance 1.5 V FPGAs
	ProASIC3E	Higher density ProASIC3 FPGAs with six PLLs and additional I/O standards
	ProASIC3L	ProASIC3 FPGAs supporting 1.2 V to 1.5 V with Flash*Freeze technology
	Automotive ProASIC3	ProASIC3 FPGAs qualified for automotive applications
	Military ProASIC3/EL	Military temperature A3PE600L, A3P1000, and A3PE3000L
	RT ProASIC3	Radiation-tolerant RT3PE600L and RT3PE3000L

*Note: *The family names link to the appropriate datasheet, including product brief, DC and switching characteristics, and packaging information.*

IGLOO Terminology

In documentation, the terms IGLOO families and IGLOO devices refer to all of the IGLOO products as listed in Table 13-1. Where the information applies to only one family or limited devices, these exclusions will be explicitly stated.

ProASIC3 Terminology

In documentation, the terms ProASIC3 families and ProASIC3 devices refer to all of the ProASIC3 families as listed in Table 13-1. Where the information applies to only one family or limited devices, these exclusions will be explicitly stated.

To further understand the differences between the IGLOO and ProASIC3 families, refer to the *Industry's Lowest Power FPGAs Portfolio*.

Programming Voltage (V_{PUMP}) and V_{JTAG}

Low-power flash devices support on-chip charge pumps, and therefore require only a single 3.3 V programming voltage for the V_{PUMP} pin during programming. When the device is not being programmed, the V_{PUMP} pin can be left floating or can be tied (pulled up) to any voltage between 0 V and 3.6 V. During programming, the target board or the FlashPro3 programmer can provide V_{PUMP}. FlashPro3 is capable of supplying V_{PUMP} to a single device. If more than one device is to be programmed using FlashPro3 on a given board, FlashPro3 should not be relied on to supply the V_{PUMP} voltage.

Low-power flash device I/Os support a bank-based, voltage-supply architecture that simultaneously supports multiple I/O voltage standards (Table 13-2 on page 13-3). By isolating the JTAG power supply in a separate bank from the user I/Os, low-power flash devices provide greater flexibility with supply selection and simplify power supply and printed circuit board (PCB) design. The JTAG pins can be run at any voltage from 1.5 V to 3.3 V (nominal). Actel recommends that TCK be tied to GND or V_{JTAG} when not used. This prevents a possible totempole current on the input buffer stage. For TDI, TMS, and TRST pins, the devices provide an internal nominal 10 kΩ pull-up resistor. During programming, all I/O pins, except for JTAG interface pins, are tristated and weakly pulled up to V_{CCI}. This isolates the part and prevents the signals from floating. The JTAG interface pins are driven by the FlashPro3 during programming, including the TRST pin, which is driven HIGH.

Table 13-2 • **Power Supplies**

Power Supply	Programming Mode	Current during Programming
V_{CC}	1.5 V	< 70 mA
V_{CCI}	1.5 V / 1.8 V / 2.5 V / 3.3 V (bank-selectable)	I/Os are weakly pulled up.
V_{JTAG}	1.5 V / 1.8 V / 2.5 V / 3.3 V	< 20 mA
V_{PUMP}	3.0 V to 3.6 V	< 80 mA

Note: All supply voltages should be at 1.5 V or higher, regardless of the setting during normal operation.

IEEE 1532 (JTAG) Interface

The supported industry-standard IEEE 1532 programming interface builds on the IEEE 1149.1 (JTAG) standard. IEEE 1532 defines the standardized process and methodology for ISP. Both silicon and software issues are addressed in IEEE 1532 to create a simplified ISP environment. Any IEEE 1532–compliant programmer can be used to program low-power flash devices. However, only limited security and FlashROM features are supported when using the IEEE 1532 standard. The Actel FlashPro3 programmer was developed exclusively for these devices and will support all the security and device serialization features. Refer to the standard for detailed information about IEEE 1532.

Security

Unlike SRAM-based FPGAs that require loading at power-up from an external source such as a microcontroller or boot PROM, Actel nonvolatile devices are live at power-up, and there is no bitstream required to load the device when power is applied. The unique flash-based architecture prevents reverse engineering of the programmed code on the device, because the programmed data is stored in nonvolatile memory cells. Each nonvolatile memory cell is made up of small capacitors and any physical deconstruction of the device will disrupt stored electrical charges.

Each low-power flash device has a built-in 128-bit Advanced Encryption Standard (AES) decryption core, except for the 15 k and 30 k gate devices. Any FPGA core or FlashROM content loaded into the device can optionally be sent as encrypted bitstream and decrypted as it is loaded. This is

particularly suitable for applications where device updates must be transmitted over an unsecured network such as the Internet. The embedded AES decryption core can prevent sensitive data from being intercepted (Figure 13-1 on page 13-4). A single 128-bit AES Key (32 hex characters) is used to encrypt FPGA core programming data and/or FlashROM programming data in the Actel tools. The low-power flash devices also decrypt with a single 128-bit AES Key. In addition, low-power flash devices support a Message Authentication Code (MAC) for authentication of the encrypted bitstream on-chip. This allows the encrypted bitstream to be authenticated and prevents erroneous data from being programmed into the device. The FPGA core, FlashROM, and Flash Memory Blocks (FBs), in Fusion only, can be updated independently using a programming file that is AES-encrypted (cipher text) or uses plain text.

Security in ARM-Enabled Low-Power Flash Devices

There are slight differences between the regular flash device and the ARM®-enabled flash devices, which have the M1 and M7 prefix.

The AES key is used by Actel and pre-programmed into the device to protect the ARM IP. As a result, the design will be encrypted along with the ARM IP, according to the details below.

CoreMP7 Device Security

ARM7™ (M7-enabled) devices are shipped with the following security features:

- FPGA Array enabled for AES encrypted programming and verification
- FlashROM enabled for plaintext read and write

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.

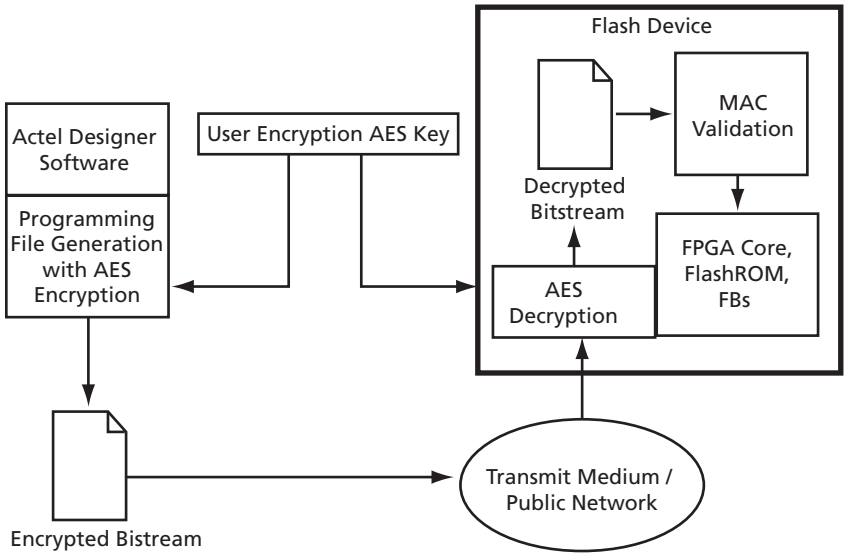


***Figure 13-1* • AES-128 Security Features**

Figure 13-2 on page 13-5 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.

Actel low-power flash devices also provide the unique Actel 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. Low-power flash devices do 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 Actel Designer software. Refer to *Security in Low-Power Flash Devices* for more information.

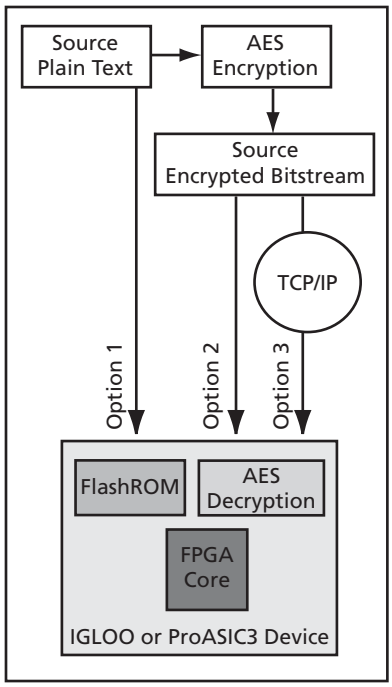


***Figure 13-2* • Different ISP Use Models**

FlashROM and Programming Files

Each low-power flash device has 1 kbit of on-chip, nonvolatile flash memory that can be accessed from the FPGA core. This nonvolatile FlashROM is arranged in eight pages of 128 bits (Figure 13-3). Each page can be programmed independently, with or without the 128-bit AES encryption. The FlashROM can only be programmed via the IEEE 1532 JTAG port and cannot be programmed from the FPGA core. In addition, during programming of the FlashROM, the FPGA core is powered down automatically by the on-chip programming control logic.

Using FlashROM combined with AES, many subscription-based applications or device serialization applications are possible. SmartGen supports easy management of the FlashROM contents even over large numbers of devices. SmartGen can support FlashROM contents that contain the following:

- Static values

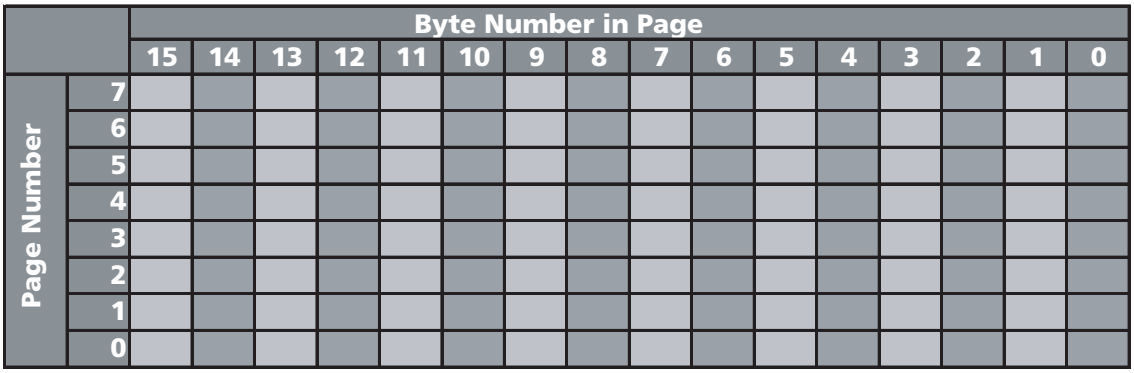


Figure 13-3 • **FlashROM Architecture**

- Random numbers
- Values read from a file
- Independent updates of each page

In addition, auto-incrementing of fields is possible. In applications where the FlashROM content is different for each device, you have the option to generate a single STAPL file for all the devices or individual serialization files for each device. For more information on how to generate the FlashROM content for device serialization, refer to *FlashROM in Actel's Low-Power Flash Devices*.

Actel Libero® Integrated Designed Environment (IDE) includes a unique tool to support the generation and management of FlashROM and FPGA programming files. This tool is called FlashPoint.

Depending on the applications, designers can use the FlashPoint software to generate a STAPL file with different contents. In each case, optional AES encryption and/or different security settings can be set.

In Designer, when you click the Programming File icon, FlashPoint launches, and you can generate STAPL file(s) with four different cases (Figure 13-4 on page 13-7). When the serialization feature is used during the configuration of FlashROM in SmartGen, you can generate a single STAPL file that will program all the devices or an individual STAPL file for each device.

The following cases present the FPGA core and FlashROM programming file combinations that can be used for different applications. In each case, you can set the optional security settings (FlashLock Pass Key and/or AES Key) depending on the application.

1. A single STAPL file or multiple STAPL files with multiple FlashROM contents and the FPGA core content. A single STAPL file will be generated if the device serialization feature is not used. You can program the whole FlashROM or selectively program individual pages.
2. A single STAPL file for the FPGA core content
3. A single STAPL file or multiple STAPL files with multiple FlashROM contents. A single STAPL file will be generated if the device serialization feature is not used. You can program the whole FlashROM or selectively program individual pages.
4. A single STAPL file to configure the security settings for the device, such as the AES Key and/or Pass Key.

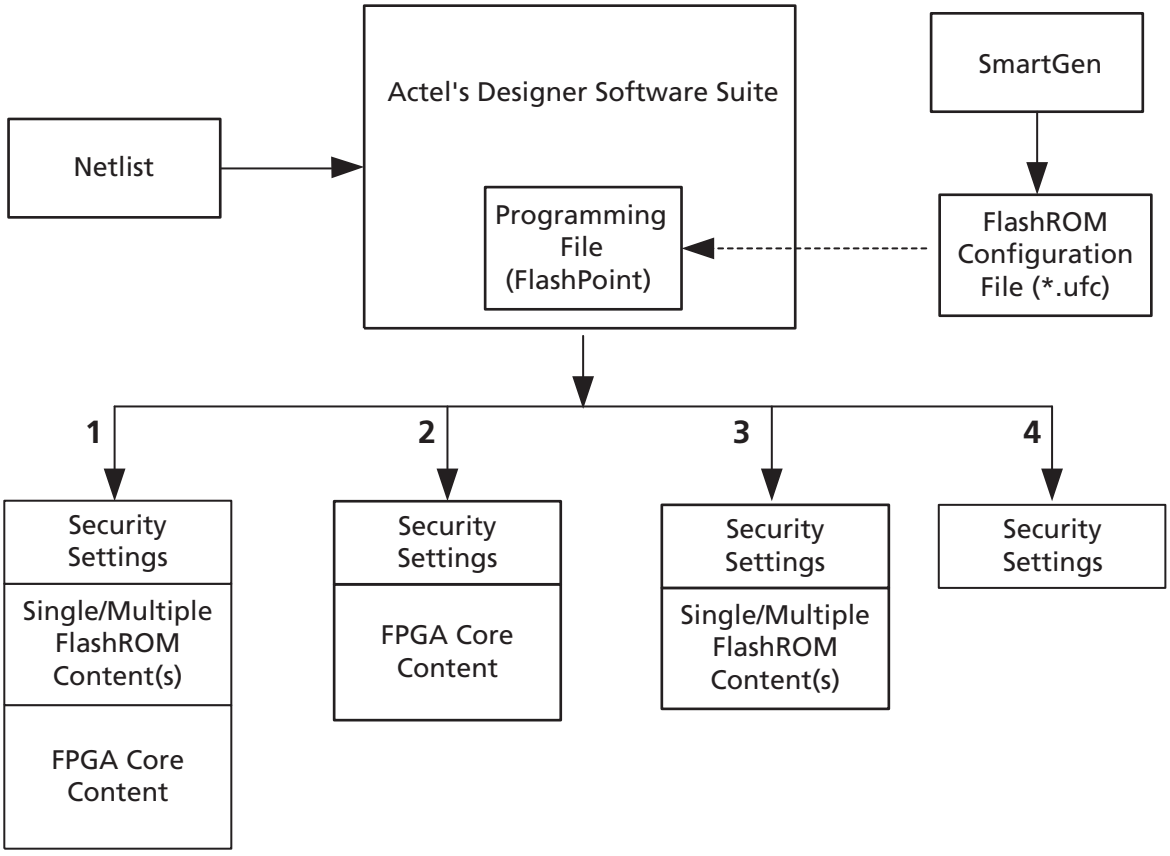


***Figure 13-4* • Flexible Programming File Generation for Different Applications**

Programming Solution

For device programming, any IEEE 1532–compliant programmer can be used; however, the FlashPro3 programmer must be used to control the low-power flash device's rich security features and FlashROM programming options. The FlashPro3 programmer is a low-cost portable programmer for the Actel flash families. It can also be used with a powered USB hub for parallel programming. General specifications for the FlashPro3 programmer are as follows:

- Programming clock – TCK is used with a maximum frequency of 20 MHz, and the default frequency is 4 MHz.
- Programming file – STAPL
- Daisy chain – Supported. You can use the ChainBuilder software to build the programming file for the chain.
- Parallel programming – Supported. Multiple FlashPro3 programmers can be connected together using a powered USB hub or through the multiple USB ports on the PC.
- Power supply – The target board must provide V_{CC}, V_{CCI}, V_{PUMP}, and V_{JTAG} during programming. However, if there is only one device on the target board, the FlashPro3 programmer can generate the required V_{PUMP} voltage from the USB port.

ISP Programming Header Information

The FlashPro3 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 Actel (FP3-26PIN-ADAPTER).

Existing ProASIC^{PLUS} family customers who are using the Samtec Small Programming Header (FTSH-113-01-L-D-K) and are planning to migrate to IGLOO or ProASIC3 devices can order a separate adapter kit from Actel (FP3-10PIN-ADAPTER-KIT), which contains a compact 10-pin adapter kit as well as 26-pin migration capability.

Table 13-3 • **Programming Header Ordering Code**

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.
Actel	FP3-10PIN-ADAPTER-KIT	Compact header and migration kit

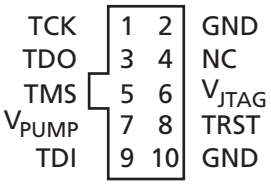


Figure 13-5 • **Programming Header (top view)**

Table 13-4 • **Programming Header Pin Numbers and Description**

Pin	Signal	Source	Description
1	TCK	Programmer	JTAG Clock
2	GND[1]	–	Signal Reference
3	TDO	Target Board	Test Data Output
4	NC	–	No Connect
5	TMS	Programmer	Test Mode Select
6	V_{JTAG}	Target Board	JTAG Supply Voltage
7	V_{PUMP}[2]	Programmer/Target Board	Programming Supply Voltage
8	nTRST	Programmer	JTAG Test Reset (Hi-Z with 10 kΩ pull-down, HIGH, LOW, or toggling)
9	TDI	Programmer	Test Data Input
10	GND[1]	–	Signal Reference

Notes:

1. *Both GND pins must be connected.*
2. *FlashPro3 can provide V_{PUMP} if there is only one device on the target board.*

Board-Level Considerations

A bypass capacitor is required from V_{PUMP} to GND for all low-power flash devices during programming. This bypass capacitor protects the devices from voltage spikes that may occur on the V_{PUMP} supplies during the erase and programming cycles. Refer to *Pin Descriptions* for specific recommendations. For proper programming, 0.01 µF and 0.33 µF capacitors (both rated at 16 V) are to be connected in parallel across V_{PUMP} and GND, and positioned as close to the FPGA pins as possible. The bypass capacitor must be placed within 2.5 cm of the device pins.

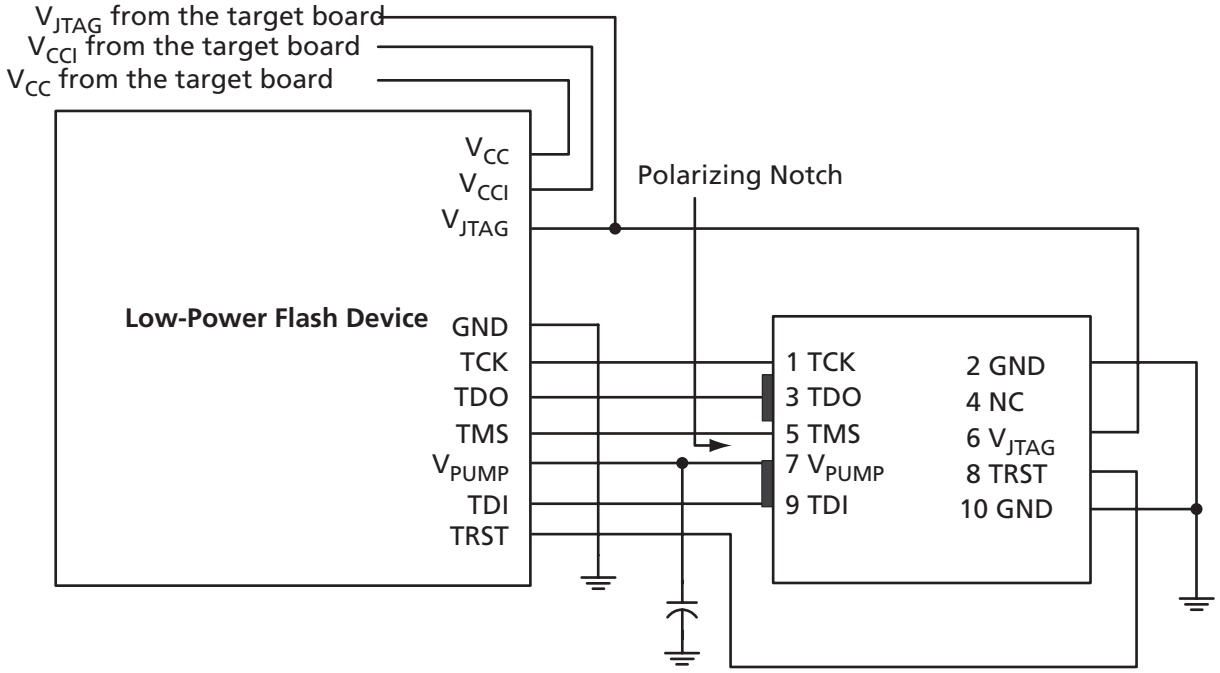


***Figure 13-6* • Board Layout and Programming Header Top View**

Troubleshooting Signal Integrity

Symptoms of a Signal Integrity Problem

A signal integrity problem can manifest itself in many ways. The problem may show up as extra or dropped bits during serial communication, changing the meaning of the communication. There is a normal variation of threshold voltage and frequency response between parts even from the same lot. Because of this, the effects of signal integrity may not always affect different devices on the same board in the same way. Sometimes, replacing a device appears to make signal integrity problems go away, but this is just masking the problem. Different parts on identical boards will exhibit the same problem sooner or later. It is important to fix signal integrity problems early. Unless the signal integrity problems are severe enough to completely block all communication between the device and the programmer, they may show up as subtle problems. Some of the FlashPro3 exit codes that are caused by signal integrity problems are listed below. Signal integrity problems are not the only possible cause of these errors, but this list is intended to show where problems can occur. FlashPro3 allows TCK to be lowered from 24 MHz down to 1 MHz to allow you to address some signal integrity problems that may occur with impedance mismatching at higher frequencies.

Chain Integrity Test Error or Analyze Chain Failure

Normally, the FlashPro3 Analyze Chain command expects to see 0x2 on the TDO pin. If the command reports reading 0x0 or 0x3, it is seeing the TDO pin stuck at 0 or 1. The only time the TDO pin comes out of tristate is when the JTAG TAP state machine is in the Shift-IR or Shift-DR state. If

noise or reflections on the TCK or TMS lines have disrupted the correct state transitions, the device's TAP state controller might not be in one of these two states when the programmer tries to read the device. When this happens, the output is floating when it is read and does not match the expected data value. This can also be caused by a broken TDO net. Only a small amount of data is read from the device during the Analyze Chain command, so marginal problems may not always show up during this command.

Exit 11

This error occurs during the verify stage of programming a device. After programming the design into the device, the device is verified to ensure it is programmed correctly. The verification is done by shifting the programming data into the device. An internal comparison is performed within the device to verify that all switches are programmed correctly. Noise induced by poor signal integrity can disrupt the writes and reads or the verification process and produce a verification error. While technically a verification error, the root cause is often related to signal integrity.

Refer to the *FlashPro User's Guide* for other error messages and solutions. For the most up-to-date known issues and solutions, refer to http://www.actel.com/support.

Conclusion

Fusion, IGLOO, and ProASIC3 devices offer a low-cost, single-chip solution that is live at power-up through nonvolatile flash technology. The FlashLock Pass Key and 128-bit AES Key security features enable secure ISP in an untrusted environment. On-chip FlashROM enables a host of new applications, including device serialization, subscription-based applications, and IP addressing. Additionally, as the FlashROM is nonvolatile, all of these services can be provided without battery backup.

Related Documents

Handbook Documents

Microprocessor Programming of Actel's Low-Power Flash Devices

http://www.actel.com/documents/LPD_Microprocessor_HBs.pdf

Security in Low-Power Flash Devices

http://www.actel.com/LPD_Security_HBs.pdf

FlashROM in Actel's Low-Power Flash Devices

http://www.actel.com/documents/LPD_FlashROM_HBs.pdf

Pin Descriptions

http://www.actel.com/documents/LPD_PinDescriptions_HBs.pdf

User's Guides

FlashPro User's Guide

http://www.actel.com/documents/flashpro_ug.pdf

Part Number and Revision Date

This document was previously published as an application note describing features and functions of the device, and as such has now been incorporated into the device handbook format. No technical changes have been made to the content.

Part Number 51700094-015-3

Revised October 2008

List of Changes

The following table lists critical changes that were made in the current version of the chapter.

Previous Version	Changes in Current Version (v1.3)	Page
v1.2 (June 2008)	The "ISP Support in Low-Power Devices" section was revised to include new families and make the information more concise.	13-2
v1.1 (March 2008)	The following changes were made to the family descriptions in Table 13-1 · Low-Power Flash Families: • ProASIC3L was updated to include 1.5 V. • The number of PLLs for ProASIC3E was changed from five to six.	13-2
v1.0 (January 2008)	The "ISP Architecture" section was updated to included the IGLOO PLUS family in the discussion of family-specific support. The text, "When 1.2 V is used, the device can be reprogrammed in-system at 1.5 V only" was revised to state, "Although the device can operate at 1.2 V core voltage, the device can only be reprogrammed when all supplies (V_{CC}, V_{CCI}, and V_{JTAG}) are at 1.5 V."	13-1
	The "ISP Support in Low-Power Devices" section and Table 13-1 · Low-Power Flash Families were updated to include the IGLOO PLUS family. The "IGLOO Terminology" section and "ProASIC3 Terminology" section are new.	13-2
	The "Security" section was updated to mention that 15 k gate devices do not have a built-in 128-bit decryption core.	13-3
	Table 13-2 · Power Supplies was revised to remove the Normal Operation column and add a table note stating, "All supply voltages should be at 1.5 V or higher, regardless of the setting during normal operation."	13-3
	The "ISP Programming Header Information" section was revised to change FP3-26PIN-ADAPTER to FP3-10PIN-ADAPTER-KIT. Table 13-3 · Programming Header Ordering Code was updated with the same change, as well as adding the part number FFSD-05-D-06.00-01-N, a 10-pin cable with 50-mil-pitch sockets.	13-8
	The "Board-Level Considerations" section was updated to describe connecting two capacitors in parallel across V_{PUMP} and GND for proper programming.	13-9
51900055-2/7.06	Information was added to the "Programming Voltage (VPUMP) and VJTAG" section about the JTAG interface pin.	13-3
51900055-1/1.05	ACTgen was changed to SmartGen.	N/A
	In Figure 13-6 · Board Layout and Programming Header Top View, the order of the text was changed to: V_{JTAG} from the target board V_{CCI} from the target board V_{CC} from the target board	13-9

14 – Core Voltage Switching Circuit for IGLOO and ProASIC3L In-System Programming

Introduction

The IGLOO® and ProASIC®3L families offer devices that can be powered by either 1.5 V or 1.2 V core voltage.

Since IGLOO and ProASIC3L devices are flash-based, they can be programmed and reprogrammed multiple times in-system using Actel FlashPro3. Actel FlashPro3 uses the JTAG standard interface (IEEE 1532) and STAPL file (IEEE 1149) to program a device. Programming can also be executed by other methods, such as an embedded microcontroller that follows the same standards above.

All IGLOO and ProASIC3L devices must be programmed with the V_{CC} core voltage at 1.5 V. Therefore, applications using IGLOO or ProASIC3L devices powered by a 1.2 V supply must switch the core supply to 1.5 V for in-system programming.

The purpose of this document is to describe an easy-to-use and cost-effective solution for switching the core supply voltage from 1.2 V to 1.5 V during in-system programming for IGLOO and ProASIC3L devices.

Actel's Flash Families Support Voltage Switching Circuit

The low-power flash families listed in Table 14-1 support the voltage switching circuit feature and the functions described in this document.

Table 14-1 • **Low-Power Flash Families**

Product Line	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 PLUS	IGLOO FPGAs with enhanced I/O capabilities
ProASIC3	ProASIC3L	ProASIC3 FPGAs supporting 1.2 V to 1.5 V with Flash*Freeze technology
	Military ProASIC3/EL	Military temperature A3PE600L, A3P1000, and A3PE3000L
	RT ProASIC3	Radiation-tolerant RT3PE600L and RT3PE3000L

*Note: *The family names link to the appropriate datasheet, including product brief, DC and switching characteristics, and packaging information.*

IGLOO Terminology

In documentation, the terms IGLOO families and IGLOO devices refer to all of the IGLOO products as listed in Table 14-1. Where the information applies to only one family or limited devices, these exclusions will be explicitly stated.

ProASIC3 Terminology

In documentation, the terms ProASIC3 families and ProASIC3 devices refer to all of the ProASIC3 families as listed in Table 14-1. Where the information applies to only one family or limited devices, these exclusions will be explicitly stated.

To further understand the differences between the IGLOO and ProASIC3 families, refer to the *Industry's Lowest Power FPGAs Portfolio*.

Circuit Description

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 to 1.5 V during in-system programming (ISP). There are several possible techniques to meet this requirement. Actel recommends utilizing a linear voltage regulator, a resistor voltage divider, and an N-Channel Digital FET to set the appropriate V_{CC} voltage, as shown in Figure 14-1.

The main component of Actel'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 kΩ and an R37 value of 80.6 kΩ, 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 V_{CC} 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.

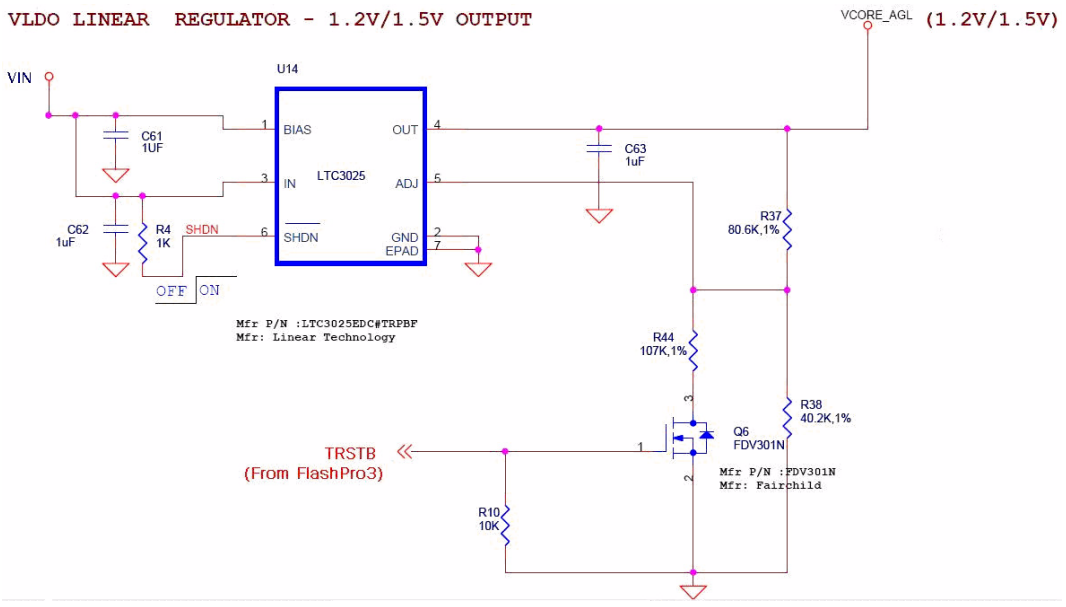


Figure 14-1 • **Circuit Diagram**

Circuit Verification

The power switching circuit recommended above is implemented on Actel's Icicle board (Figure 14-2). On the Icicle board, VJTAGENB is used to control the N-Channel Digital FET; however, this circuit was modified to use TRST instead of VJTAGENB in this application. There are three important aspects of this circuit that were verified:

1. The rise on V_{CC} from 1.2 V to 1.5 V when TRST is HIGH
2. V_{CC} rises to 1.5 V before programming begins.
3. V_{CC} switches from 1.5 V to 1.2 V when TRST is LOW.

Verification Steps

1. The rise on V_{CC} from 1.2 V to 1.5 V when TRST is HIGH.

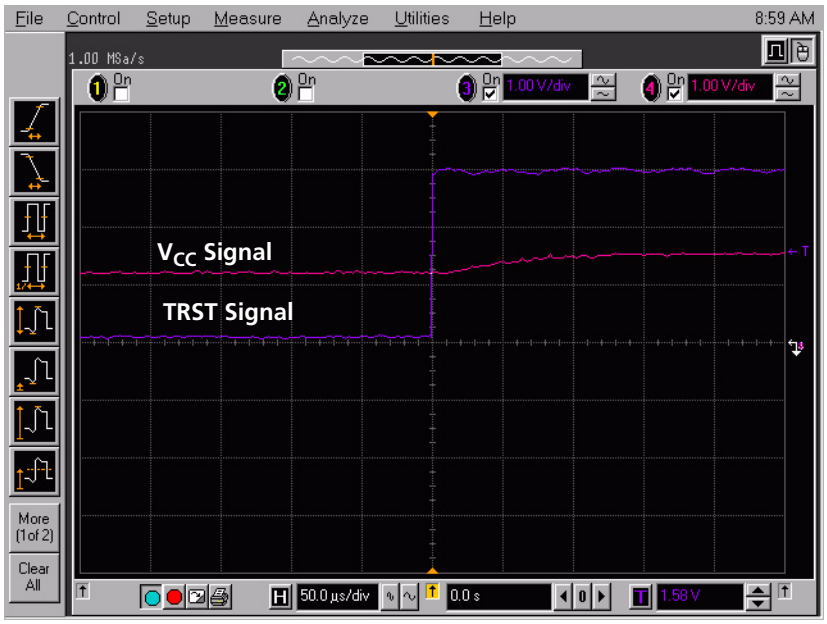


***Figure 14-2* • Core Voltage on the IGLOO AGL125-QNG132 Device**

In the oscilloscope plots (Figure 14-2), the TRST from FlashPro3 and the V_{CC} core voltage of the IGLOO device are labeled. This plot shows the rise characteristic of the TRST signal from FlashPro3. Once the TRST signal is asserted HIGH, the LTC3025 shown in Figure 14-1 on page 14-3 senses the increase in voltage and changes the output from 1.2 V to 1.5 V. It takes the circuit approximately 100 µs to respond to TRST and change the voltage to 1.5 V on the V_{CC} core.

2. V_{CC} rises to 1.5 V before programming begins.

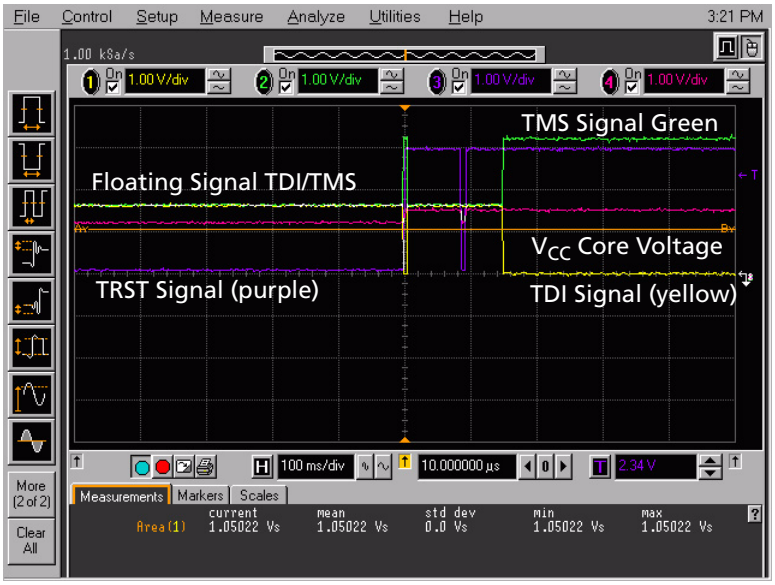


Figure 14-3 • **Programming Algorithm**

The oscilloscope plot in Figure 14-3 shows a wider time interval for the programming algorithm and includes the TDI and TMS signals from the FlashPro3. These signals carry the programming information that is programmed into the device and should only start toggling after the V_{CC} core voltage reaches 1.5 V. Again, the TRST from FlashPro3 and the V_{CC} voltage core of the IGLOO device are labeled. As shown in Figure 14-3, TDI and TMS are floating initially, and the core voltage is 1.2 V. When a programming command on the FlashPro3 is executed, the TRST is driven HIGH and the TDI is momentarily driven to ground. In response to the HIGH TRST signal, the circuit responds and pulls the core voltage to 1.5 V. After 100 ms, TRST is briefly driven LOW by the FlashPro software. This is expected behavior that ensures the device JTAG state machine is in Reset prior to programming. The TRST remains HIGH for the duration of the programming. It can be seen in Figure 14-3 that the V_{CC} core voltage signal remains at 1.5 V for approximately 50 ms before information starts passing through on TDI and TMS. This confirms that the voltage switching circuit drives the V_{CC} core supply voltage to 1.5 V prior to programming.

3. V_{CC} switches from 1.5 V to 1.2 V when TRST is LOW.

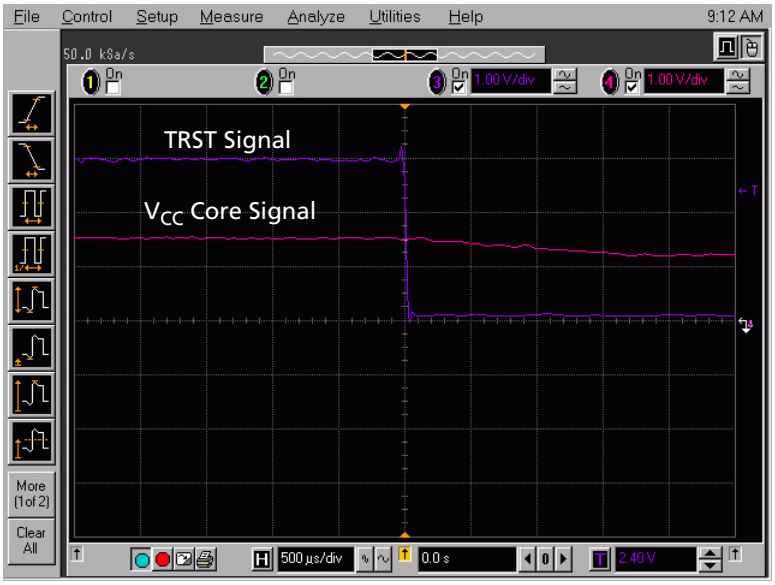


***Figure 14-4* • TRST Toggled LOW**

In Figure 14-4, the TRST signal and the V_{CC} core voltage signal are labeled. As the TRST is pulled to ground, the core voltage is observed to switch from 1.5 V to 1.2 V. The observed fall time is approximately 2 ms.

DirectC

The above analysis is based on FlashPro3 but there are other solutions to ISP, such as DirectC. DirectC is a microprocessor program that can be run in-system to program Actel flash devices. For FlashPro3, TRST is the most convenient control signal to use for the recommended circuit. However, for DirectC, users may use any signal to control the FET. For example, the DirectC code can be edited so that a separate non-JTAG signal can be asserted from the microcontroller that signals the board that it is about to start programming the device. After asserting the N-Channel Digital FET control signal, the programming algorithm must allow sufficient time for the supply to rise to 1.5 V before initiating DirectC programming. As seen in Figure 14-3 on page 14-5, 50 ms is adequate time. Depending on the size of the PCB and the capacitance on the V_{CC} supply, results may vary from system to system. Actel recommends using a conservative value for the wait time to make sure that the V_{CC} core voltage is at the right level.

Conclusion

Actel's IGLOO and ProASIC3L low-power FPGAs offer 1.2 V core operation; however, they must be programmed with a core voltage of 1.5 V. Since the device can have either a 1.2 V or a 1.5 V normal operation, there must be a way for the core voltage to switch from 1.2 V to 1.5 V, which is required during in-system programming. The circuit explained in this document illustrates one simple, cost-effective way of handling this requirement. A JTAG signal from the FlashPro3 programmer allows the circuit to sense when programming is in progress, enabling it to switch to the correct core voltage.

Part Number and Revision Date

This document was previously published as an application note describing features and functions of the device, and as such has now been incorporated into the device handbook format. No technical changes have been made to the content.

Part Number 51700094-028-1

Revised October 2008

List of Changes

The following table lists critical changes that were made in the current version of the chapter.

Previous Version	Changes in Current Version (v1.1)	Page
v1.0 (August 2008)	The "Actel's Flash Families Support Voltage Switching Circuit" section was revised to include new families and make the information more concise.	14-2

Boundary Scan and UJTAG

Actel
POWER MATTERS

15 – Boundary Scan in Low-Power Flash Devices

Boundary Scan

Low-power flash devices are compatible with IEEE Standard 1149.1, which defines a hardware architecture and the set of mechanisms for boundary scan testing. JTAG operations are used during boundary scan testing.

The basic boundary scan logic circuit is composed of the TAP controller, test data registers, and instruction register (Figure 15-2 on page 15-4).

Low-power flash devices support three types of test data registers: bypass, device identification, and boundary scan. The bypass register is selected when no other register needs to be accessed in a device. This speeds up test data transfer to other devices in a test data path. The 32-bit device identification register is a shift register with four fields (LSB, ID number, part number, and version). The boundary scan register observes and controls the state of each I/O pin. Each I/O cell has three boundary scan register cells, each with serial-in, serial-out, parallel-in, and parallel-out pins.

TAP Controller State Machine

The TAP controller is a 4-bit state machine (16 states) that operates as shown in Figure 15-1.

The 1s and 0s represent the values that must be present on TMS at a rising edge of TCK for the given state transition to occur. IR and DR indicate that the instruction register or the data register is operating in that state.

The TAP controller receives two control inputs (TMS and TCK) and generates control and clock signals for the rest of the test logic architecture. On power-up, the TAP controller enters the Test-Logic-Reset state. To guarantee a reset of the controller from any of the possible states, TMS must remain HIGH for five TCK cycles. The TRST pin can also be used to asynchronously place the TAP controller in the Test-Logic-Reset state.

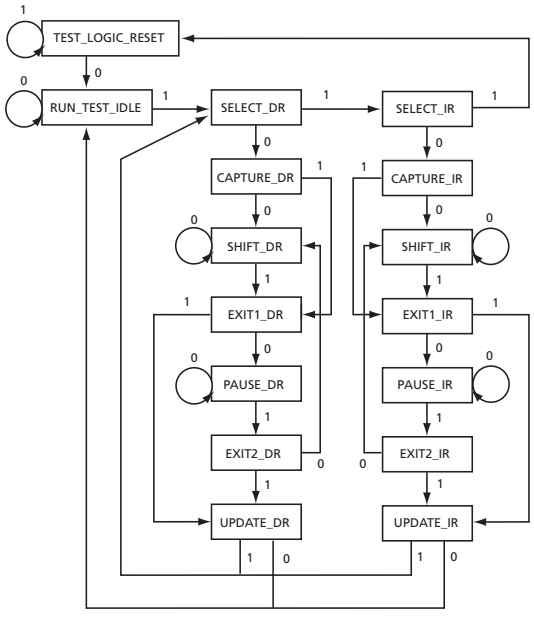


Figure 15-1 • **TAP Controller State Machine**

Actel's Flash Families Support the JTAG Feature

The low-power flash families listed in Table 15-1 support the JTAG feature and the functions described in this document.

Table 15-1 • **Low-Power Flash Families**

Product Line	Family*	Description
Fusion	Fusion	Mixed-signal FPGA integrating ProASIC®3 FPGA fabric, programmable analog block, support for ARM® Cortex™-M1 soft processors and flash memory into a monolithic device
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 PLUS	IGLOO FPGAs with enhanced I/O capabilities
ProASIC3	ProASIC3	Low-power, high-performance 1.5 V FPGAs
	ProASIC3E	Higher density ProASIC3 FPGAs with six PLLs and additional I/O standards
	ProASIC3L	ProASIC3 FPGAs supporting 1.2 V to 1.5 V with Flash*Freeze technology
	Automotive ProASIC3	ProASIC3 FPGAs qualified for automotive applications
	Military ProASIC3/EL	Military temperature A3PE600L, A3P1000, and A3PE3000L
	RT ProASIC3	Radiation-tolerant RT3PE600L and RT3PE3000L

*Note: *The family names link to the appropriate datasheet, including product brief, DC and switching characteristics, and packaging information.*

IGLOO Terminology

In documentation, the terms IGLOO families and IGLOO devices refer to all of the IGLOO products as listed in Table 15-1. Where the information applies to only one family or limited devices, these exclusions will be explicitly stated.

ProASIC3 Terminology

In documentation, the terms ProASIC3 families and ProASIC3 devices refer to all of the ProASIC3 families as listed in Table 15-1. Where the information applies to only one family or limited devices, these exclusions will be explicitly stated.

To further understand the differences between the IGLOO and ProASIC3 families, refer to the *Industry's Lowest Power FPGAs Portfolio*.

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 15-2).

Table 15-2 • **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 15-2 on page 15-4), 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" description in *Pin Descriptions* for pull-up/-down recommendations for TDO and TCK pins.

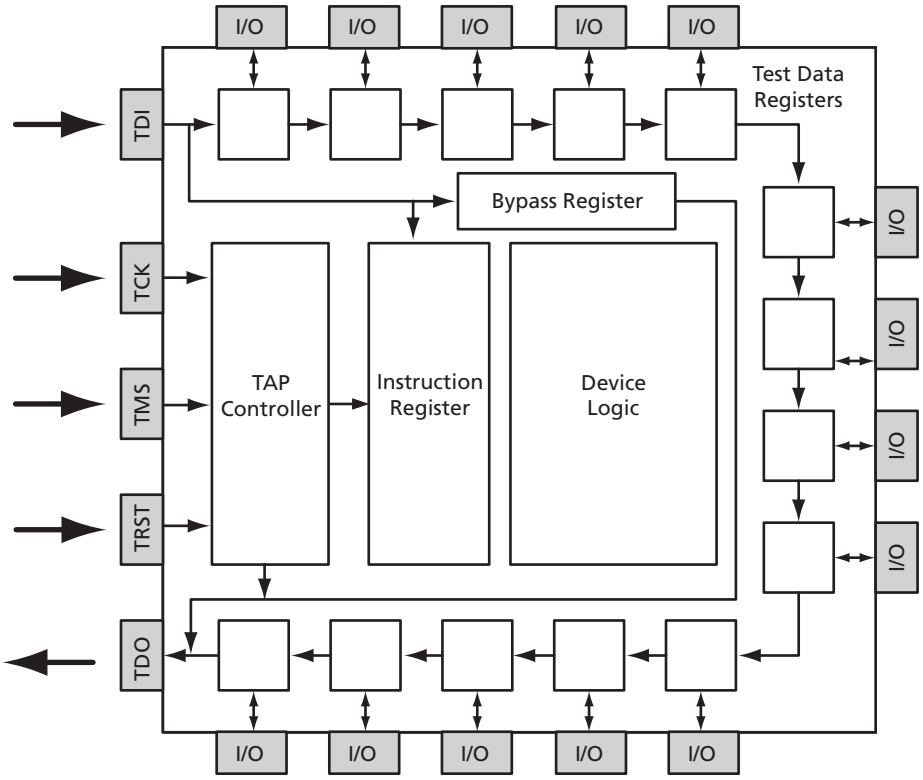


Figure 15-2 • **Boundary Scan Chain**

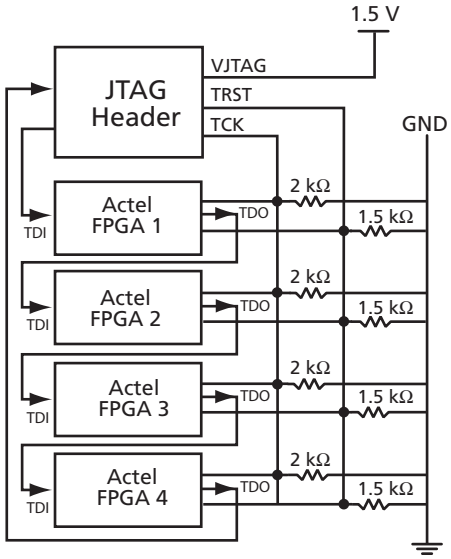
Board Level Recommendations

Table 15-3 gives pull-down recommendations for the TRST and TCK pins.

Table 15-3 • **TRST and TCK Pull-Down Recommendations**

V_{JTAG}	Tie-Off Resistance*
V_{JTAG} at 3.3 V	200 Ω to 1 kΩ
V_{JTAG} at 2.5 V	200 Ω to 1 kΩ
V_{JTAG} at 1.8 V	500 Ω to 1 kΩ
V_{JTAG} at 1.5 V	500 Ω to 1 kΩ

** Equivalent parallel resistance if more than one device is on JTAG chain (Figure 15-3)*



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

***Figure 15-3* • Parallel Resistance on JTAG Chain of Devices**

Related Documents

Handbook Documents

Pin Descriptions

http://www.actel.com/documents/LPD_PinDescriptions_HBs.pdf

Part Number and Revision Date

This document contains content extracted from the Device Architecture section of the datasheet. To improve usability for customers, the device architecture information has now been split into handbook sections, which also include usage information. No technical changes were made to the content unless explicitly listed.

Part Number 51700094-019-3

Revised October 2008

List of Changes

The following table lists critical changes that were made in the current version of the chapter.

Previous Version	Changes in Current Version (v1.3)	Page
v1.2 (June 2008)	The "Boundary Scan Support in Low-Power Devices" section was revised to include new families and make the information more concise.	15-3
v1.1 (March 2008)	The following changes were made to the family descriptions in Table 15-1 · Low-Power Flash Families: • ProASIC3L was updated to include 1.5 V. • The number of PLLs for ProASIC3E was changed from five to six.	15-2
v1.0 (January 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.	15-2

16 – UJTAG Applications in Actel's Low-Power Flash Devices

Introduction

In Fusion, IGLOO,® and ProASIC®3 devices, there is bidirectional access from the JTAG port to the core VersaTiles during normal operation of the device (Figure 16-1). User JTAG (UJTAG) is the ability for the design to use the JTAG ports for access to the device for updates, etc. While regular JTAG is used, the UJTAG tiles, located at the southeast area of the die, are directly connected to the JTAG Test Access Port (TAP) Controller in normal operating mode. As a result, all the functional blocks of the device, such as Clock Conditioning Circuits (CCC) with PLLs, SRAM blocks, embedded FlashROM, flash memory blocks, and I/O tiles, can be reached via the JTAG ports. The UJTAG functionality is available by instantiating the UJTAG macro directly in the source code of a design. Access to the FPGA core VersaTiles from the JTAG ports enables users to implement different applications using the TAP Controller (JTAG port). This document introduces the UJTAG tile functionality and discusses a few application examples. However, the possible applications are not limited to what is presented in this document. UJTAG can serve different purposes in many designs as an elementary or auxiliary part of the design. For detailed usage information, refer to *Boundary Scan in Low-Power Flash Devices*.

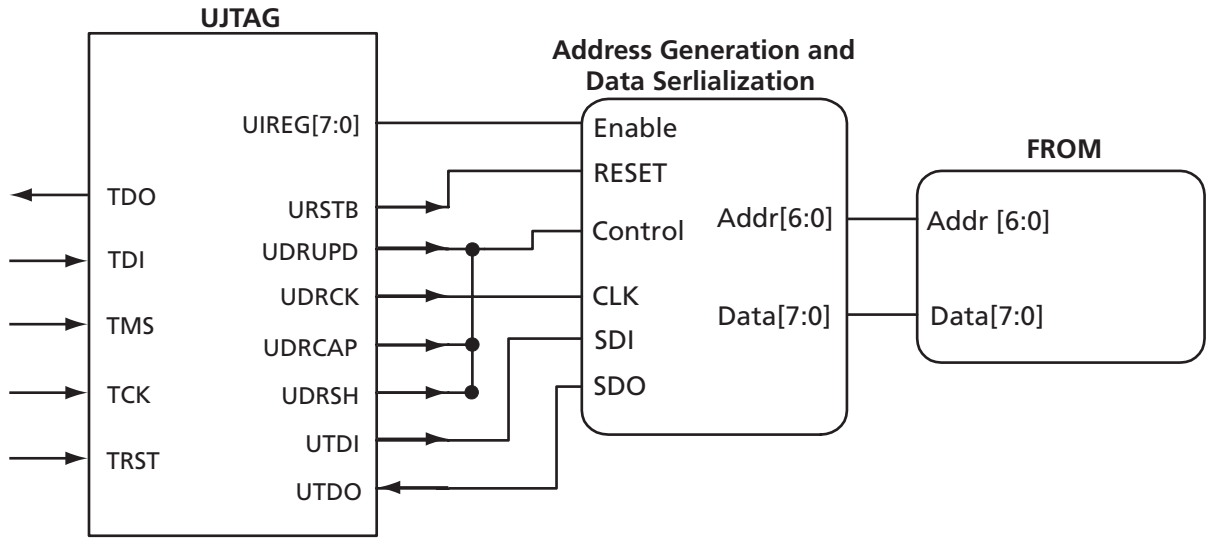


***Figure 16-1* • Block Diagram of Using UJTAG to Read FlashROM Contents**

UJTAG Support in Low-Power Devices

The low-power flash families listed in Table 16-1 support the UJTAG feature and the functions described in this document.

Table 16-1 • **Low-Power Flash Families**

Product Line	Family*	Description
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
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 PLUS	IGLOO FPGAs with enhanced I/O capabilities
ProASIC3	ProASIC3	Low-power, high-performance 1.5 V FPGAs
	ProASIC3E	Higher density ProASIC3 FPGAs with six PLLs and additional I/O standards
	ProASIC3L	ProASIC3 FPGAs supporting 1.2 V to 1.5 V with Flash*Freeze technology
	Automotive ProASIC3	ProASIC3 FPGAs qualified for automotive applications
	Military ProASIC3/EL	Military temperature A3PE600L, A3P1000, and A3PE3000L
	RT ProASIC3	Radiation-tolerant RT3PE600L and RT3PE3000L

*Note: *The family names link to the appropriate datasheet, including product brief, DC and switching characteristics, and packaging information.*

IGLOO Terminology

In documentation, the terms IGLOO families and IGLOO devices refer to all of the IGLOO products as listed in Table 16-1. Where the information applies to only one family or limited devices, these exclusions will be explicitly stated.

ProASIC3 Terminology

In documentation, the terms ProASIC3 families and ProASIC3 devices refer to all of the ProASIC3 families as listed in Table 16-1. Where the information applies to only one family or limited devices, these exclusions will be explicitly stated.

To further understand the differences between the IGLOO and ProASIC3 families, refer to the *Industry's Lowest Power FPGAs Portfolio*.

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 16-2. 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 16-3 on page 16-4 illustrates the correct connection of the UJTAG macro to the user design netlist. Actel Designer software will automatically connect these ports to the TAP during place-and-route. Table 16-2 gives the port descriptions for the rest of the UJTAG ports:

Table 16-2 • **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

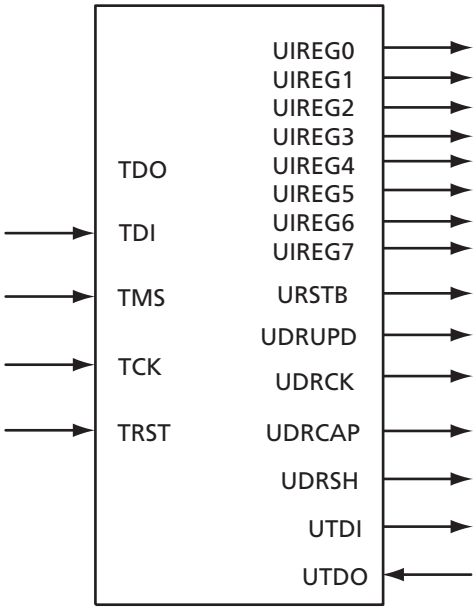
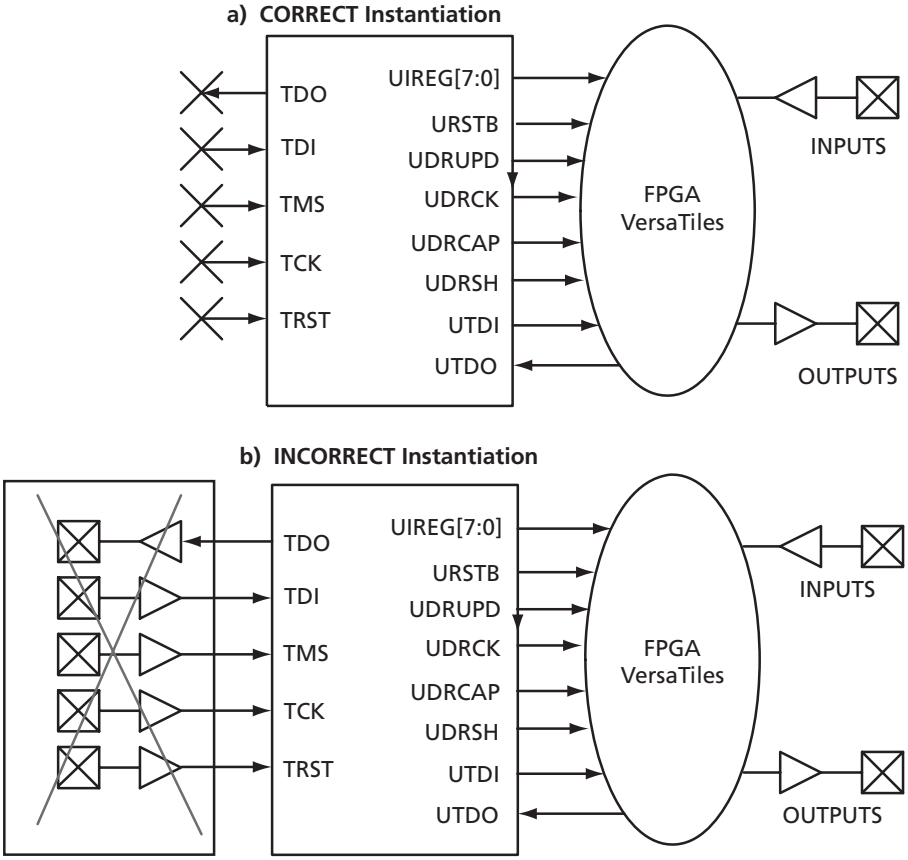


Figure 16-2 • **UJTAG Tile Block Symbol**



Note: Do not connect JTAG pins (TDO, TDI, TMS, TCK, or TRST) to I/Os in the design.

Figure 16-3 • **Connectivity Method of UJTAG Macro**

UJTAG Operation

There are a few basic functions of the UJTAG macro that users must understand before designing with it. The most important fundamental concept of the UJTAG design is its connection with the TAP Controller state machine.

TAP Controller State Machine

The 16 states of the Tap Controller state machine are shown in Figure 16-4 on page 16-5. The 1s and 0s, shown adjacent to the state transitions, represent the TMS values that must be present at the time of a rising TCK edge for a state transition to occur. In the states that include the letters "IR," the instruction register operates; in the states that contain the letters "DR," the test data register operates. The TAP Controller receives two control inputs, TMS and TCK, and generates control and clock signals for the rest of the test logic.

On power-up (or the assertion of TRST), the TAP Controller enters the Test-Logic-Reset state. To reset the controller from any other state, TMS must be held HIGH for at least five TCK cycles. After reset, the TAP state changes at the rising edge of TCK, based on the value of TMS.

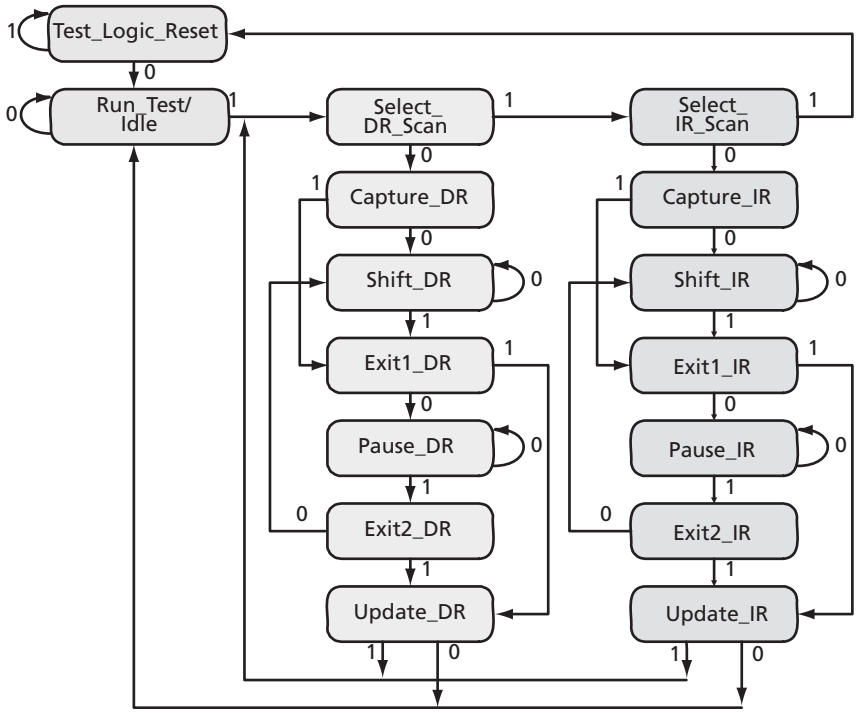


***Figure 16-4* • TAP Controller State Diagram**

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.

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 16-3 lists the 81 configuration bits in the CCC.

Table 16-3 • **Configuration Bits of Fusion, IGLOO, and ProASIC3 CCC Blocks**

Bit Number	Control Function
80	RESET ENABLE
79	DYNCSEL
78	DYNBSEL
77	DYNASEL
<76:74>	VCOSEL [2:0]
73	STATCSEL
72	STATBSEL
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 Actel 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).

Fine Tuning

In some applications, design constants or parameters need to be modified after programming the original design. The tuning process can be done using the UJTAG tile without reprogramming the device with new values. If the parameters or constants of a design are stored in distributed registers or embedded SRAM blocks, the new values can be shifted onto the JTAG TAP Controller pins, replacing the old values. The UJTAG tile is used as the "bridge" for data transfer between the JTAG pins and the FPGA VersaTiles or SRAM logic. Figure 16-5 shows a flow chart example for fine-tuning application steps using the UJTAG tile.

In Figure 16-5, the TMS signal sets the TAP Controller state machine to the appropriate states. The flow mainly consists of two steps: a) shifting the defined instruction and b) shifting the new data. If the target parameter is constantly used in the design, the new data can be shifted into a temporary shift register from UTDI. The UDRSH output of UJTAG can be used as a shift-enable signal, and UDRCK is the shift clock to the shift register. Once the shift process is completed and the TAP Controller state is moved to the Update_DR state, the UDRUPD output of the UJTAG can latch the new parameter value from the temporary register into a permanent location. This avoids any interruption or malfunctioning during the serial shift of the new value.

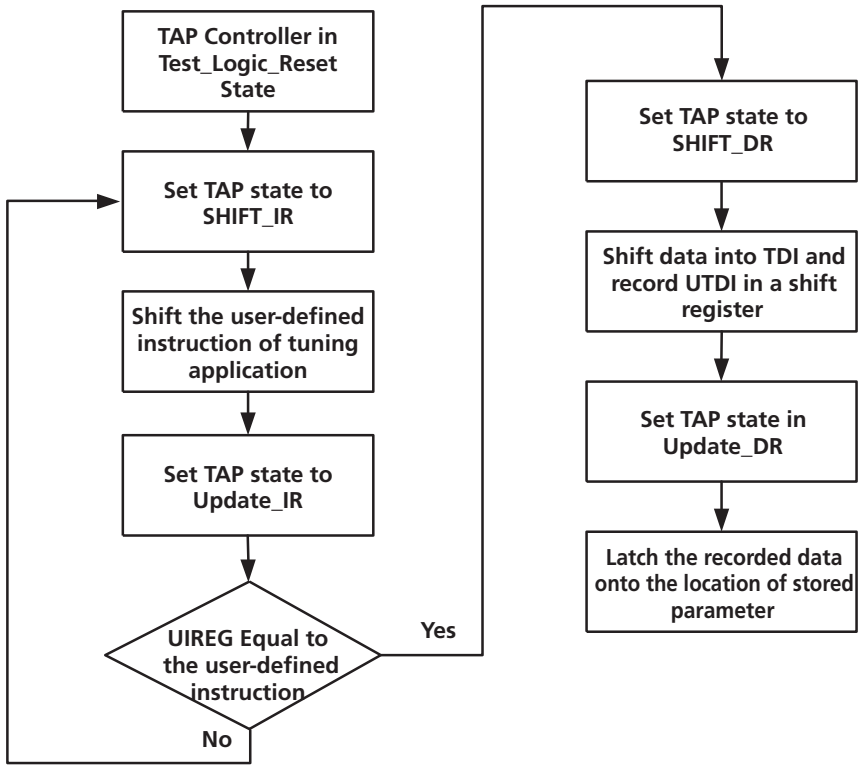


Figure 16-5 • **Flow Chart Example of Fine-Tuning an Application Using UJTAG**

Silicon Testing and Debugging

In many applications, the design needs to be tested, debugged, and verified on real silicon or in the final embedded application. To debug and test the functionality of designs, users may need to monitor some internal logic (or nets) during device operation. The approach of adding design test pins to monitor the critical internal signals has many disadvantages, such as limiting the number of user I/Os. Furthermore, adding external I/Os for test purposes may require additional or dedicated board area for testing and debugging.

The UJTAG tiles of low-power flash devices offer a flexible and cost-effective solution for silicon test and debug applications. In this solution, the signals under test are shifted out to the TDO pin of the TAP Controller. The main advantage is that all the test signals are monitored from the TDO pin; no pins or additional board-level resources are required. Figure 16-6 illustrates this technique. Multiple test nets are brought into an internal MUX architecture. The selection of the MUX is done using the contents of the TAP Controller instruction register, where individual instructions (values from 16 to 127) correspond to different signals under test. The selected test signal can be synchronized with the rising or falling edge of TCK (optional) and sent out to UTDO to drive the TDO output of JTAG.

The test and debug procedure is not limited to the example in Figure 16-5 on page 16-7. Users can customize the debug and test interface to make it appropriate for their applications. For example, multiple test signals can be registered and then sent out through UTDO, each at a different edge of TCK. In other words, *n* signals are sampled with an F_{TCK} / *n* sampling rate. The bandwidth of the information sent out to TDO is always proportional to the frequency of TCK.

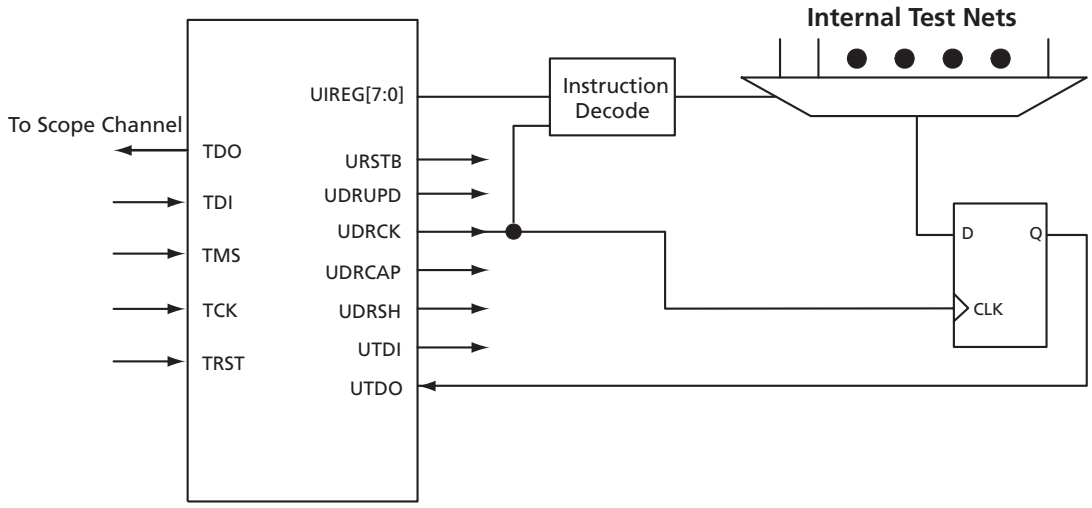


Figure 16-6 • **UJTAG Usage Example in Test and Debug Applications**

SRAM Initialization

Users can also initialize embedded SRAMs of the low-power flash devices. The initialization of the embedded SRAM blocks of the design can be done using UJTAG tiles, where the initialization data is imported using the TAP Controller. Similar functionality is available in ProASIC^{PLUS} devices using JTAG. The guidelines for implementation and design examples are given in the *RAM Initialization and ROM Emulation in ProASIC^{PLUS} Devices* application note.

SRAMs are volatile by nature; data is lost in the absence of power. Therefore, the initialization process should be done at each power-up if necessary.

FlashROM Read-Back Using JTAG

The low-power flash architecture contains a dedicated nonvolatile FlashROM block, which is formatted into eight 128-bit pages. For more information on FlashROM, refer to *FlashROM in Actel's Low-Power Flash Devices*. The contents of FlashROM are available to the VersaTiles during normal operation through a read operation. As a result, the UJTAG macro can be used to provide the FlashROM contents to the JTAG port during normal operation. Figure 16-7 illustrates a simple block diagram of using UJTAG to read the contents of FlashROM during normal operation.

The FlashROM read address can be provided from outside the FPGA through the TDI input or can be generated internally using the core logic. In either case, data serialization logic is required (Figure 16-7) and should be designed using the VersaTile core logic. FlashROM contents are read asynchronously in parallel from the flash memory and shifted out in a synchronous serial format to TDO. Shifting the serial data out of the serialization block should be performed while the TAP is in UDRSH mode. The coordination between TCK and the data shift procedure can be done using the TAP state machine by monitoring UDRSH, UDRCAP, and UDRUPD.

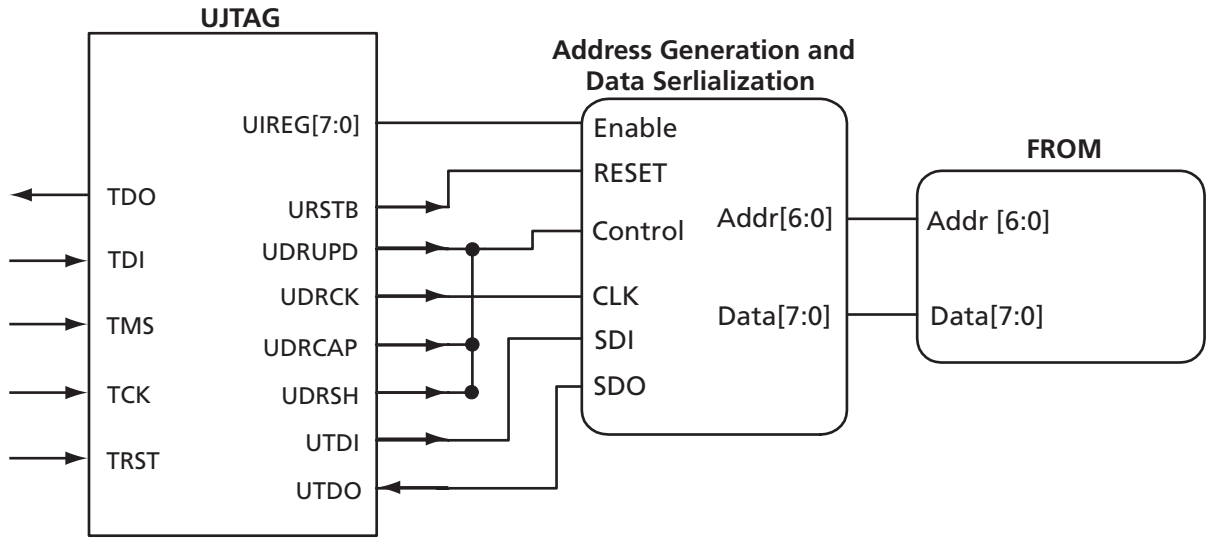


***Figure 16-7* • Block Diagram of Using UJTAG to Read FlashROM Contents**

Conclusion

Actel low-power flash FPGAs offer many unique advantages, such as security, nonvolatility, reprogrammablity, and low power—all in a single chip. In addition, Fusion, IGLOO, and ProASIC3 devices provide access to the JTAG port from core VersaTiles while the device is in normal operating mode. A wide range of available user-defined JTAG opcodes allows users to implement various types of applications, exploiting this feature of these devices. The connection between the JTAG port and core tiles is implemented through an embedded and hardwired UJTAG tile. A UJTAG tile can be instantiated in designs using the UJTAG library cell. This document presents multiple examples of UJTAG applications, such as dynamic reconfiguration, silicon test and debug, fine-tuning of the design, and RAM initialization. Each of these applications offers many useful advantages.

Related Documents

Application Notes

RAM Initialization and ROM Emulation in ProASIC^{PLUS} Devices
http://www.actel.com/documents/APA_RAM_Initd_AN.pdf

Handbook Documents

Boundary Scan in Low-Power Flash Devices
http://www.actel.com/documents/LPD_BoundaryScan_HBs.pdf
FlashROM in Actel's Low-Power Flash Devices
http://www.actel.com/documents/LPD_FlashROM_HBs.pdf

Part Number and Revision Date

This document contains content extracted from the Device Architecture section of the datasheet, combined with content previously published as an application note describing features and functions of the device. To improve usability for customers, the device architecture information has now been combined with usage information, to reduce duplication and possible inconsistencies in published information. No technical changes were made to the datasheet content unless explicitly listed. Changes to the application note content were made only to be consistent with existing datasheet information

Part Number 51700094-020-3

Revised October 2008

List of Changes

The following table lists critical changes that were made in the current version of the chapter.

Previous Version	Changes in Current Version (v1.3)	Page
v1.2 (June 2008)	The "UJTAG Support in Low-Power Devices" section was revised to include new families and make the information more concise.	16-2
	The title of Table 16-3 · Configuration Bits of Fusion, IGLOO, and ProASIC3 CCC Blocks was revised to include Fusion.	16-6
v1.1 (March 2008)	The following changes were made to the family descriptions in Table 16-1 · Low-Power Flash Families: • ProASIC3L was updated to include 1.5 V. • The number of PLLs for ProASIC3E was changed from five to six.	16-2
v1.0 (January 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.	16-2

Board-Level Requirements

Actel
POWER MATTERS

17 – Power-Up/-Down Behavior of Low-Power Flash Devices

Introduction

Actel's low-power flash devices are flash-based FPGAs manufactured on a 0.13 µm process node. These devices offer a single-chip, reprogrammable solution and support Level 0 live at power-up (LAPU) due to their nonvolatile architecture.

Actel's four low-power flash FPGA families are optimized for logic area, I/O features, and performance. IGLOO® devices are optimized for power, making them the industry's lowest power programmable solution. IGLOO PLUS FPGAs offer enhanced I/O features beyond those of the IGLOO ultra-low power solution for I/O-intensive low-power applications. ProASIC3®L FPGAs balance low power with high performance. The ProASIC3 family is Actel's high-performance flash FPGA solution.

Actel's low-power flash devices exhibit very low transient current on each power supply during power-up. The peak value of the transient current depends on the device size, temperature, voltage levels, and power-up sequence.

The following devices can have inputs driven in while the device is not powered:

- IGLOO (AGL015 and AGL030)
- IGLOO PLUS (AGLP030, AGLP060, AGLP125)
- IGLOOe (AGLE600, AGLE3000)
- ProASIC3L (A3PE3000L)
- ProASIC3 (A3P015 and A3P030)
- ProASIC3E (A3PE600, A3PE1500, A3PE3000)
- Military ProASIC3EL (A3PE600L, A3PE3000L, but not A3P1000)
- RT ProASIC3 (RT3PE600L, RT3PE3000L)

The driven I/Os do not pull up power planes, and the current draw is limited to very small leakage current, making them suitable for applications that require cold-sparing. These devices are hot-swappable, meaning they can be inserted in a live power system.[1]

1. *For more details on the levels of hot-swap compatibility in Actel's low-power flash devices, refer to the "Hot-Swap Support" section in the I/O Structures chapter of the handbook for the device you are using.*

Actel's Flash Families Support Power-Up Behavior

The low-power flash families listed in Table 17-1 support power-up behavior and the functions described in this document.

Table 17-1 • **Low-Power Flash Families**

Product Line	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 PLUS	IGLOO FPGAs with enhanced I/O capabilities
ProASIC3	ProASIC3	Low-power, high-performance 1.5 V FPGAs
	ProASIC3E	Higher density ProASIC3 FPGAs with six PLLs and additional I/O standards
	ProASIC3L	ProASIC3 FPGAs supporting 1.2 V to 1.5 V with Flash*Freeze technology
	Automotive ProASIC3	ProASIC3 FPGAs qualified for automotive applications
	Military ProASIC3/EL	Military temperature A3PE600L, A3P1000, and A3PE3000L
	RT ProASIC3	Radiation-tolerant RT3PE600L and RT3PE3000L

*Note: *The family names link to the appropriate datasheet, including product brief, DC and switching characteristics, and packaging information.*

IGLOO Terminology

In documentation, the terms IGLOO families and IGLOO devices refer to all of the IGLOO products as listed in Table 17-1. Where the information applies to only one family or limited devices, these exclusions will be explicitly stated.

ProASIC3 Terminology

In documentation, the terms ProASIC3 families and ProASIC3 devices refer to all of the ProASIC3 families as listed in Table 17-1. Where the information applies to only one family or limited devices, these exclusions will be explicitly stated.

To further understand the differences between the IGLOO and ProASIC3 families, refer to the *Industry's Lowest Power FPGAs Portfolio*.

Power-Up/Down Sequence and Transient Current

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

- V_{CCPLX}
- V_{JTAG}
- V_{CC}: Voltage supply to the FPGA core
  - V_{CC} is 1.5 V ± 0.075 V for IGLOO, IGLOO PLUS, and ProASIC3 devices operating at 1.5 V.
  - V_{CC} is 1.2 V ± 0.06 V for IGLOO, IGLOO PLUS, and ProASIC3L devices operating at 1.2 V.
  - V5 devices will require a 1.5V V_{CC} supply, where as V2 devices can utilize either 1.2 V or 1.5 V V_{CC}.
- V_{CCI}Bx: 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 PLUS devices do not have VMVx supply pins.)

The I/O bank VMV pin must be tied to the V_{CCI} pin within the same bank. Therefore, the supplies that need to be powered up/down during normal operation are V_{CC} and V_{CCI}. These power supplies can be powered up/down in any sequence during normal operation of IGLOO, IGLOO PLUS, ProASIC3L, and ProASIC3 FPGAs. During power-up, I/Os in each bank will remain tristated until the last supply (either V_{CCI}Bx or V_{CC}) 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 Actel's low-power flash devices have no power-up or power-down sequencing requirements, Actel identifies the following power conditions that will result in higher than normal transient current. Use this information to help maximize power savings:

Actel recommends tying V_{CCPLX} to V_{CC} and using proper filtering circuits to decouple V_{CC} noise from the PLL.

a. If V_{CCPLX} is powered up before V_{CC}, a static current of up to 5 mA (typical) per PLL may be measured on V_{CCPLX}.
   i. The current vanishes as soon as V_{CC} reaches V_{CCPLX} voltage level.
   ii. The same current is observed at power-down (V_{CC} before V_{CCPLX}).
b. If V_{CCPLX} is powered up simultaneously or after V_{CC}:
   i. Actel's low-power flash devices exhibit very low transient current on V_{CC}. For ProASIC3 devices, the maximum transient current on V_{CC} does not exceed the maximum standby current specified in the device datasheet.

The source of transient current, also known as inrush current, varies depending on the FPGA technology. Due to their volatile technology, the internal registers in SRAM FPGAs must be initialized before configuration can start. This initialization is the source of significant inrush current in SRAM FPGAs during power-up. Due to the nonvolatile nature of flash technology, low-power flash devices do not require any initialization at power-up, and there is very little or no crossbar current through PMOS and NMOS devices. Therefore, the transient current at power-up is significantly less than for SRAM FPGAs. Figure 17-1 on page 17-4 illustrates the types of power consumption by SRAM FPGAs compared to Actel's antifuse and flash FPGAs.

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

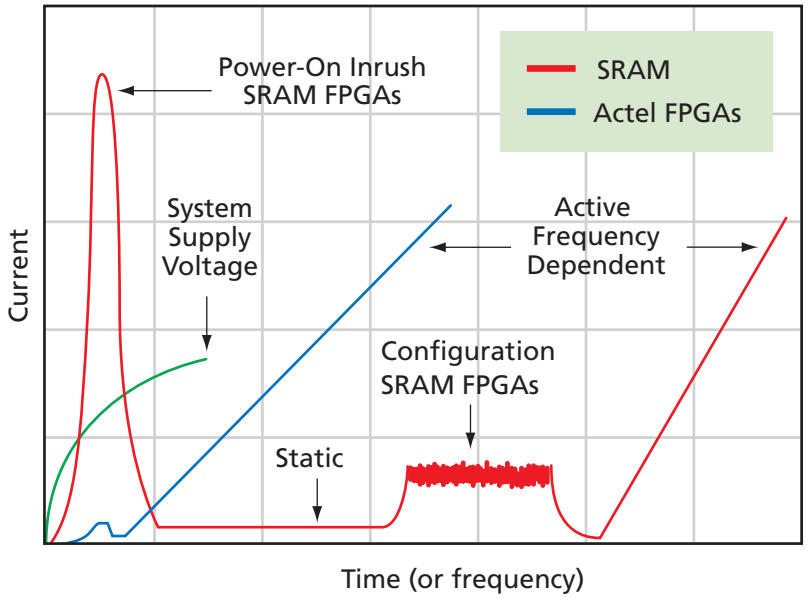


***Figure 17-1* • Types of Power Consumption in SRAM FPGAs and Actel Nonvolatile FPGAs**

Transient Current on V_{CC}

The characterization of the transient current on V_{CC} is performed on nearly all devices within the IGLOO, ProASIC3L, and ProASIC3 families. A sample size of five units is used from each device family member. All the device I/Os are internally pulled down while the transient current measurements are performed. For ProASIC3 devices, the measurements at typical conditions show that the maximum transient current on V_{CC}, when the power supply is powered at ramp-rates ranging from 15 V/ms to 0.15 V/ms, does not exceed the maximum standby current specified in the device datasheets. Refer to *ProASIC3 DC and Switching Characteristics* and *ProASIC3E DC and Switching Characteristics* for more information.

Similarly, IGLOO, IGLOO PLUS, and ProASIC3L devices exhibit very low transient current on V_{CC}. The transient current does not exceed the typical operating current of the device while in active mode. The characterization of AGL600-FG256 V2 and V5 devices has shown that the transient current on V_{CC} is typically in the range of 1–5 mA.

Transient Current on V_{CCI}

The characterization of the transient current on V_{CCI} is performed on nearly all devices within the IGLOO, IGLOO PLUS, ProASIC3L, and ProASIC3 families, similar to V_{CC} transient current measurements. For ProASIC3 devices, the measurements at typical conditions show that the maximum transient current on V_{CCI}, when the power supply is powered at ramp-rates ranging from 33 V/ms to 0.33 V/ms, does not exceed the maximum standby current specified in the device datasheet. Refer to *ProASIC3 DC and Switching Characteristics* and *ProASIC3E DC and Switching Characteristics* for more information.

Similarly, IGLOO, IGLOO PLUS, and ProASIC3L devices exhibit very low transient current on V_{CCI}. The transient current does not exceed the typical operating current of the device while in active mode. The characterization of AGL600-FG256 V2 and V5 devices has shown that the transient current on V_{CCI} is typically in the range of 1–2 mA.

I/O Behavior at Power-Up/-Down

This section discusses the behavior of device I/Os, used and unused, during power-up/-down of V_{CC} and V_{CCI}. As mentioned earlier, VMVx and V_{CCI}Bx are tied together, and therefore, inputs and outputs are powered up/down at the same time.

I/O State during Power-Up/-Down

This section discusses the characteristics of I/O behavior during device power-up and power-down. Before the start of power-up, all I/Os are in tristate mode. The I/Os will remain tristated during power-up until the last voltage supply (V_{CC} or V_{CCI}) is powered to its functional level (power supply functional levels are discussed in the "Power-Up to Functional Time" section on page 17-6). After the last supply reaches the functional level, the outputs will exit the tristate mode and drive the logic at the input of the output buffer. Similarly, the input buffers will pass the external logic into the FPGA fabric once the last supply reaches the functional level. 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}). Figure 17-2 shows the output buffer driving high and its behavior during power-up with 10 kΩ external pull-down. In Figure 17-2, V_{CC} is powered first, and V_{CCI} is powered 5 ms after V_{CC}. Figure 17-3 on page 17-6 shows the state of the I/O when V_{CCI} is powered about 5 ms before V_{CC}. In the circuitry shown in Figure 17-3 on page 17-6, the output is externally pulled down.

During power-down, device I/Os become tristated once the first power supply (V_{CC} or V_{CCI}) drops below its brownout voltage level. The I/O behavior during power-down is also independent of voltage supply sequencing.

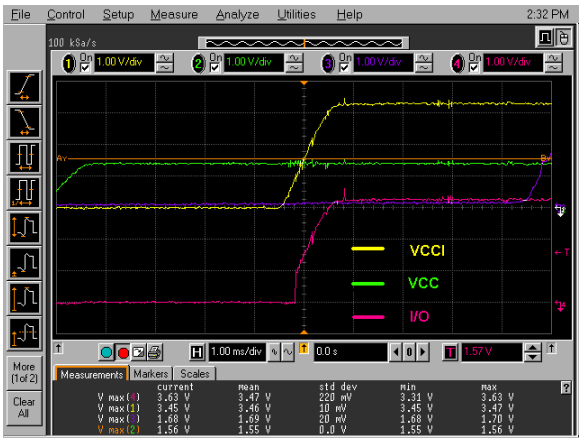


Figure 17-2 • **I/O State when V_{CC} Is Powered before V_{CCI}**

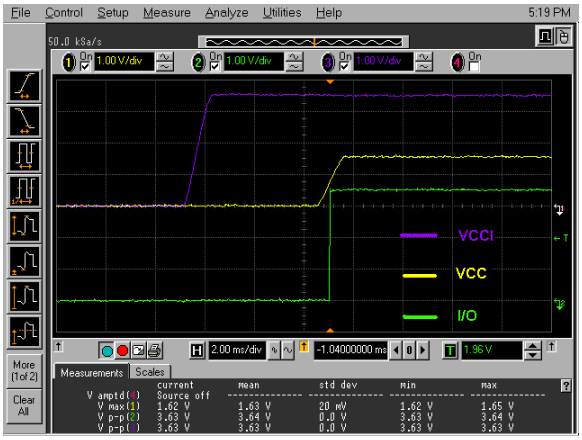


Figure 17-3 • **I/O State when V_{CCI} Is Powered before V_{CC}**

Power-Up to Functional Time

At power-up, device I/Os exit the tristate mode and become functional once the last voltage supply in the power-up sequence (V_{CCI} or V_{CC}) reaches its functional activation level. The power-up–to–functional time is the time it takes for the last supply to power up from zero to its functional level. Note, the functional level of the power supply during power-up may vary slightly within the specification in different ramp-rates. Refer to Table 17-2 for the functional level of the voltage supplies at power-up.

Typical I/O behavior during power-up to functional time is illustrated in Figure 17-2 on page 17-5 and Figure 17-3.

Table 17-2 • **Power-Up Functional Activation Levels for V_{CC} and V_{CCI}**

Device	V_{CC} Functional Activation Level (V)	V_{CCI} Functional Activation Level (V)
ProASIC3, IGLOO, IGLOO PLUS, and ProASIC3L families running at V_{CC} = 1.5 V*	0.85 V ± 0.25 V	0.9 V ± 0.3 V
IGLOO, IGLOO PLUS, and ProASIC3L families running at V_{CC} = 1.2 V*	0.85 V ± 0.2 V	0.9 V ± 0.15 V

Note: V5 devices will require a 1.5 V V_{CC} supply, where as V2 devices can utilize either 1.2 V or 1.5 V V_{CC}.

Actel's low-power flash devices meet Level 0 LAPU; that is, they can be functional prior to V_{CC} reaching the regulated voltage required. This important advantage distinguishes low-power flash devices from their SRAM-based counterparts. SRAM-based FPGAs, due to their volatile technology, require hundreds of milliseconds after power-up to configure the design bitstream before they become functional. Refer to Figure 17-4 on page 17-7 and Figure 17-5 on page 17-8 for more information.

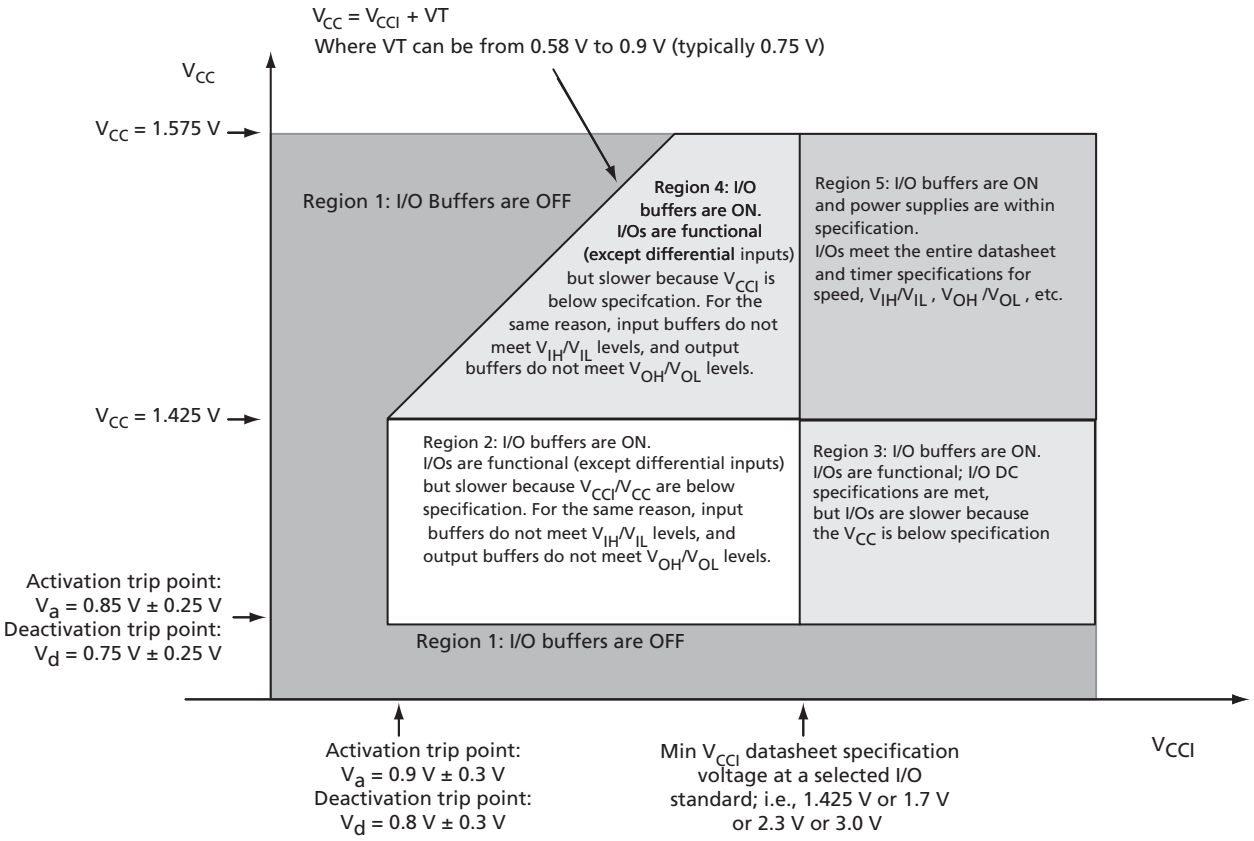


Figure 17-4 • **I/O State as a Function of V_{CCI} and V_{CC} Voltage Levels for IGLOO V5, IGLOO PLUS V5, ProASIC3L, and ProASIC3 Devices, Running at V_{CC} = 1.5 V ± 0.075 V**

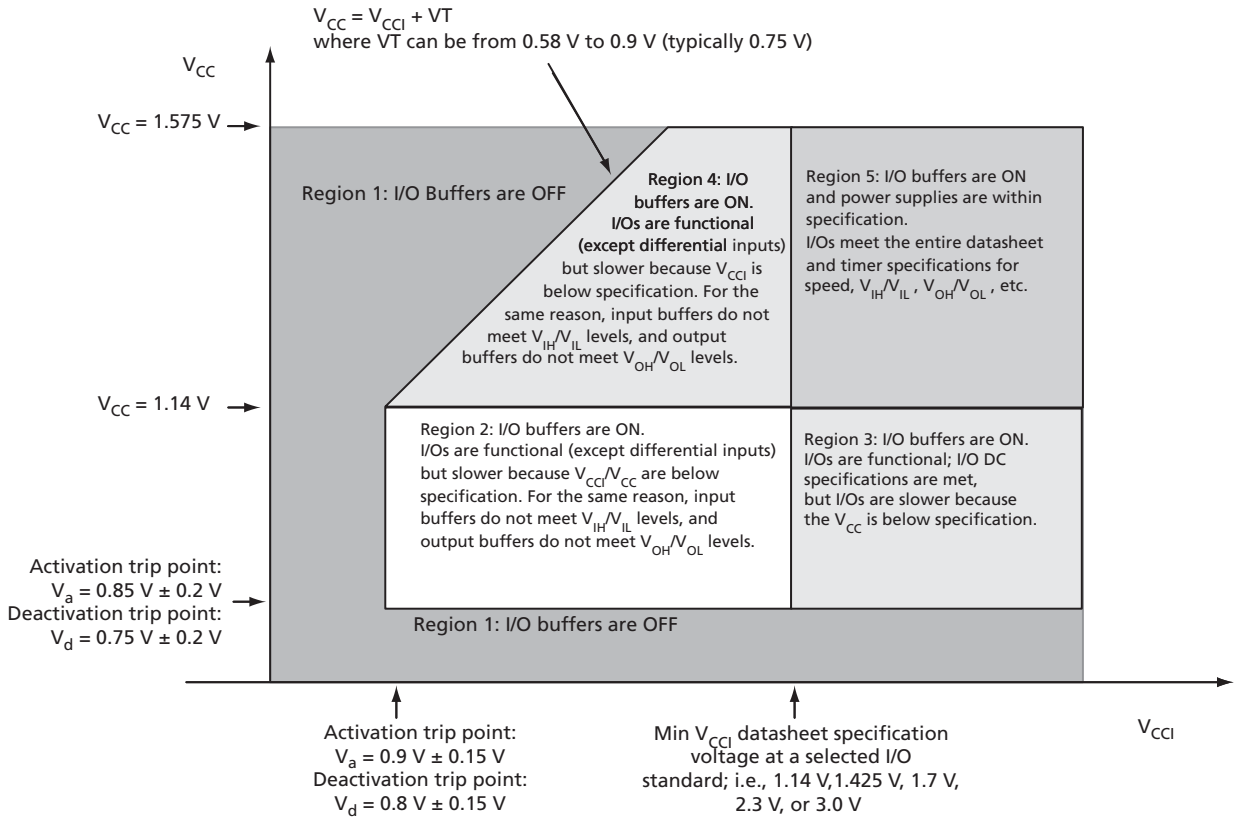


Figure 17-5 • **I/O State as a Function of V_{CCI} and V_{CC} Voltage Levels for IGLOO V2, IGLOO PLUS V2, and ProASIC3L Devices, Running at V_{CC} = 1.2 V ± 0.06 V**

Brownout Voltage

Brownout is a condition in which the voltage supplies are lower than normal, causing the device to malfunction as a result of insufficient power. In general, Actel does not guarantee the functionality of the design inside the flash FPGA if voltage supplies are below their minimum recommended operating condition. Actel has performed measurements to characterize the brownout levels of FPGA power supplies. Refer to Table 17-3 for device-specific brownout deactivation levels. For the purpose of characterization, a direct path from the device input to output is monitored while voltage supplies are lowered gradually. The brownout point is defined as the voltage level at which the output stops following the input. Characterization tests performed on several IGLOO, ProASIC3L, and ProASIC3 devices in typical operating conditions showed the brownout voltage levels to be within the specification.

During device power-down, the device I/Os become tristated once the first supply in the power-down sequence drops below its brownout deactivation voltage.

Table 17-3 • **Brownout Deactivation Levels for V_{CC} and V_{CCI}**

Devices	V_{CC} Brownout Deactivation Level (V)	V_{CCI} Brownout Deactivation Level (V)
ProASIC3, IGLOO, IGLOO PLUS and ProASIC3L families running at V_{CC} = 1.5 V	0.75 V ± 0.25 V	0.8 V ± 0.3 V
IGLOO, IGLOO PLUS, and ProASIC3L families running at V_{CC} = 1.2 V	0.75 V ± 0.2 V	0.8 V ± 0.15 V

PLL Behavior at Brownout Condition

When PLL power supply voltage and/or V_{CC} levels drop below the V_{CC} brownout levels mentioned above for 1.5 V and 1.2 V devices, the PLL output lock signal goes low and/or the output clock is lost. The following sections explain PLL behavior during and after the brownout condition.

V_{CCPLL} and V_{CC} Tied Together

In this condition, both V_{CC} and V_{CCPLL} drop below the 0.75 V (± 0.25 V or ± 0.2 V) brownout level. During the brownout recovery, once V_{CCPLL} and V_{CC} reach the activation point (0.85 ± 0.25 V or ± 0.2 V) again, the PLL output lock signal may still remain low with the PLL output clock signal toggling. If this condition occurs, there are two ways to recover the PLL output lock signal:

1. Cycle the power supplies of the PLL (power off and on) by using the PLL POWERDOWN signal.
2. Turn off the input reference clock to the PLL and then turn it back on.

Only V_{CCPLL} is at Brownout

In this case, only V_{CCPLL} drops below the 0.75 V (± 0.25 V or ± 0.2 V) brownout level and the V_{CC} supply remains at nominal recommended operating voltage (1.5 V ± 0.075 V for 1.5 V devices and 1.2 V ± 0.06 V for 1.2 V devices). In this condition, the PLL behavior after brownout recovery is similar to initial power-up condition, and the PLL will regain lock automatically after V_{CCPLL} is ramped up above the activation level (0.85 ± 0.25 V or ± 0.2 V). No intervention is necessary in this case.

Only V_{CC} Is at Brownout

In this condition, V_{CC} drops below the 0.75 V (± 0.25 V or ± 0.2 V) brownout level and V_{CCPLL} remains at nominal recommended operating voltage (1.5 V ± 0.075 V for 1.5 V devices and 1.2 V ± 0.06 V for 1.2 V devices). During the brownout recovery, once V_{CC} reaches the activation point again (0.85 ± 0.25 V or ± 0.2 V), the PLL output lock signal may still remain low with the PLL output clock signal toggling. If this condition occurs, there are two ways to recover the PLL output lock signal:

1. Cycle the power supplies of the PLL (power off and on) by using the PLL POWERDOWN signal.
2. Turn off the input reference clock to the PLL and then turn it back on.

It is important to note that Actel recommends using a monotonic power supply or voltage regulator to ensure proper power-up behavior.

Internal Pull-Up and Pull-Down

Low-power flash device I/Os are equipped with internal weak pull-up/-down resistors that can be used by designers. If used, these internal pull-up/-down resistors will be activated during power-up, once both V_{CC} and V_{CCI} are above their functional activation level. Similarly, during power-down, these internal pull-up/-down resistors will turn off once the first supply voltage falls below its brownout deactivation level.

Cold-Sparing

In cold-sparing applications, voltage can be applied to device I/Os before and during power-up. Cold-sparing applications rely on three important characteristics of the device:

1. I/Os must be tristated before and during power-up.
2. Voltage applied to the I/Os must not power up any part of the device.
3. V_{CCI} should not exceed 3.6 V, per datasheet specifications.

As described in the "Power-Up to Functional Time" section on page 17-6, Actel's low-power flash I/Os are tristated before and during power-up until the last voltage supply (V_{CC} or V_{CCI}) is powered up past its functional level. Furthermore, applying voltage to the FPGA I/Os does not pull up V_{CC} or V_{CCI} and, therefore, does not partially power up the device. Table 17-4 includes the cold-sparing test results on A3PE600-PQ208 devices. In this test, leakage current on the device I/O and residual voltage on the power supply rails were measured while voltage was applied to the I/O before power-up.

Table 17-4 • **Cold-Sparing Test Results for A3PE600 Devices**

Device I/O	Residual Voltage (V)		Leakage Current
	V_{CC}	V_{CCI}	
Input	0	0.003	<1 µA
Output	0	0.003	<1 µA

V_{CCI} must not exceed 3.6 V, as stated in the datasheet specification. Therefore, ProASIC3E devices meet all three requirements stated earlier in this section and are suitable for cold-sparing applications.

The following devices and families support cold-sparing:

- IGLOO: AGL015 and AGL030
- All IGLOO PLUS
- All IGLOOe
- ProASIC3L: A3PE3000L
- ProASIC3: A3P015 and A3P030
- All ProASIC3E
- Military ProASIC3EL: A3PE600L and A3PE3000L
- RT ProASIC3: RT3PE600L and RT3PE3000L

The following devices and families do not support cold-sparing:

- IGLOO: AGL060, AGL125, AGL250, AGL600, AGL1000
- ProASIC3: A3P060, A3P125, A3P250, A3P400, A3P600, A3P1000
- ProASIC3L: A3P250L, A3P600L, A3P1000L
- Military ProASIC3: A3P1000

Hot-Swapping

Hot-swapping is the operation of hot insertion or hot removal of a card in a powered-up system. The I/Os need to be configured in hot-insertion mode if hot-swapping compliance is required. For more details on the levels of hot-swap compatibility in low-power flash devices, refer to the "Hot-Swap Support" section in the I/O Structures chapter of the handbook for the device you are using.

The following devices and families support hot-swapping:

- IGLOO: AGL015 and AGL030
- All IGLOO PLUS
- All IGLOOe
- ProASIC3L: A3PE3000L
- ProASIC3: A3P015 and A3P030
- All ProASIC3E
- Military ProASIC3EL: A3PE600L and A3PE3000L
- RT ProASIC3: RT3PE600L and RT3PE3000L

The following devices and families do not support hot-swapping:

- IGLOO: AGL060, AGL125, AGL250, AGL600, AGL1000
- ProASIC3: A3P060, A3P125, A3P250, A3P400, A3P600, A3P1000
- ProASIC3L: A3P250L, A3P600L, A3P1000L
- Military ProASIC3: A3P1000

Conclusion

Actel's low-power flash FPGAs provide an excellent programmable logic solution for a broad range of applications. In addition to high performance, low cost, security, nonvolatility, and single chip, they are live at power-up (meet Level 0 of the LAPU classification) and offer clear and easy-to-use power-up/-down characteristics. Unlike SRAM FPGAs, low-power flash devices do not require any specific power-up/-down sequencing and have extremely low power-up inrush current in any power-up sequence. Actel low-power flash FPGAs also support both cold-sparing and hot-swapping for applications requiring these capabilities.

Related Documents

Datasheets

ProASIC3 DC and Switching Characteristics

http://www.actel.com/documents/PA3GenSpecs_DS.pdf

ProASIC3E DC and Switching Characteristics

http://www.actel.com/documents/PA3EGenSpecs_DS.pdf

Handbook Documents

I/O Structures in IGLOO PLUS Devices

http://www.actel.com/documents/IGLOOPLUS_IO_HBs.pdf

I/O Structures in IGLOO and ProASIC3 Devices

http://www.actel.com/documents/IGLOO_PA3_IO_HBs.pdf

I/O Structures in IGLOOe and ProASIC3E Devices

http://www.actel.com/documents/IGLOOe_PA3E_IO_HBs.pdf

Part Number and Revision Date

This document was a previously published handbook chapter that only discussed ProASIC3/E (Power-Up Behavior of ProASIC3/E Devices, part number: 51700094-021), and has now been updated to include data for IGLOO and ProASIC3L families.

Part Number 51700094-027-1

Revised October 2008

List of Changes

The following table lists critical changes that were made in the current version of the chapter.

Previous Version	Changes in Current Version (v1.1)	Page
v1.0 (August 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.	17-1
	The "Actel's Flash Families Support Power-Up Behavior" section was revised to include new families and make the information more concise.	17-2
	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.	17-10
	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.	17-11
v1.3 (March 2008)	This document was revised, renamed, and assigned a new part number. It now includes data for the IGLOO and ProASIC3L families.	N/A
v1.2 (January 2008) 51700094-021-2	The "Handbook Documents" section was updated to include the three different I/O Structure handbook chapters.	17-12
v1.1 (January 2008) 51700094-021-1	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 V_{CC} brownout levels (0.75 V ± 0.25 V), the PLL output lock signal goes low and/or the output clock is lost."	17-9
v1.0 (January 2008) 51700094-021-0	The "PLL Behavior at Brownout Condition" section was added.	17-9

www.actel.com

Actel Corporation
2061 Stierlin Court
Mountain View, CA
94043-4655 USA
Phone 650.318.4200
Fax 650.318.4600

Actel Europe Ltd.
River Court,Meadows Business Park
Station Approach, Blackwater
Camberley Surrey GU17 9AB
United Kingdom
Phone +44 (0) 1276 609 300
Fax +44 (0) 1276 607 540

Actel Japan
EXOS Ebisu Buillding 4F
1-24-14 Ebisu Shibuya-ku
Tokyo 150 Japan
Phone +81.03.3445.7671
Fax +81.03.3445.7668
http://jp.actel.com

Actel Hong Kong
Room 2107, China Resources Building
26 Harbour Road
Wanchai, Hong Kong
Phone +852 2185 6460
Fax +852 2185 6488
www.actel.com.cn

51700102-004-8/10.08