



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

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

Embedded - FPGAs, or Field Programmable Gate Arrays, are advanced integrated circuits that offer unparalleled flexibility and performance for digital systems. Unlike traditional fixed-function logic devices, FPGAs can be programmed and reprogrammed to execute a wide array of logical operations, enabling customized functionality tailored to specific applications. This reprogrammability allows developers to iterate designs quickly and implement complex functions without the need for custom hardware.

Applications of Embedded - FPGAs

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

Details

Product Status	Obsolete
Number of LABs/CLBs	-
Number of Logic Elements/Cells	-
Total RAM Bits	147456
Number of I/O	300
Number of Gates	1000000
Voltage - Supply	1.14V ~ 1.575V
Mounting Type	Surface Mount
Operating Temperature	-40°C ~ 100°C (TJ)
Package / Case	484-BGA
Supplier Device Package	484-FPBGA (23x23)
Purchase URL	https://www.e-xfl.com/product-detail/microsemi/m1a3p1000l-fg484i

CCC Locations

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

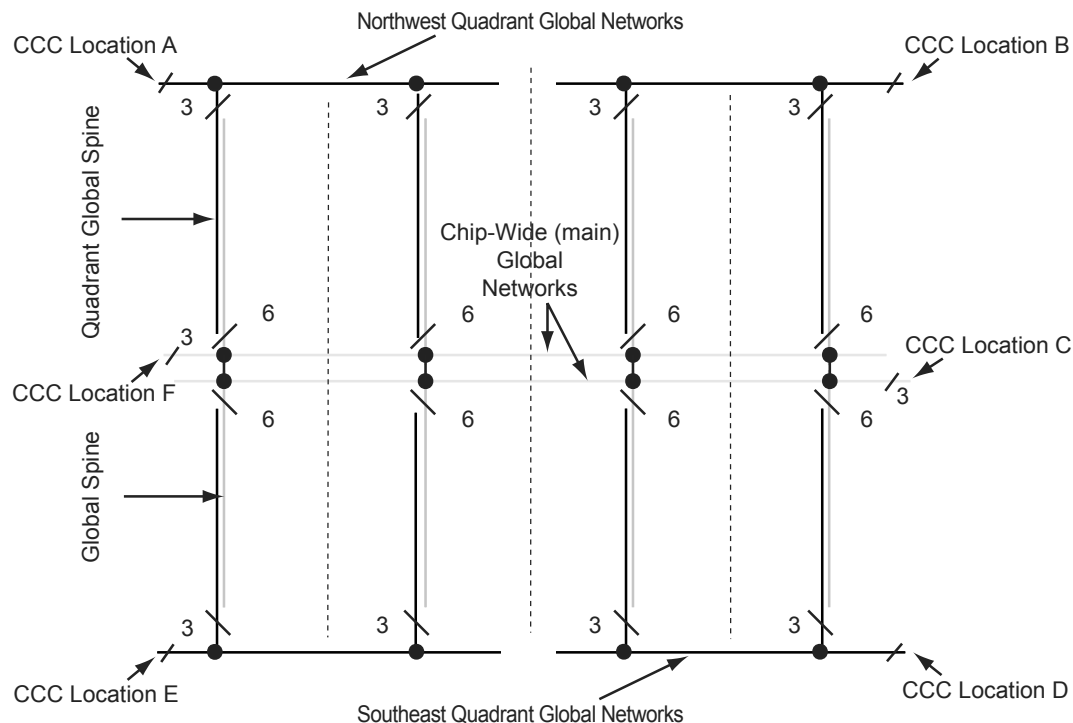


Figure 4-13 • Global Network Architecture for 60 k Gate Devices and Above

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

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

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 4-17 and Figure 4-18). 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).

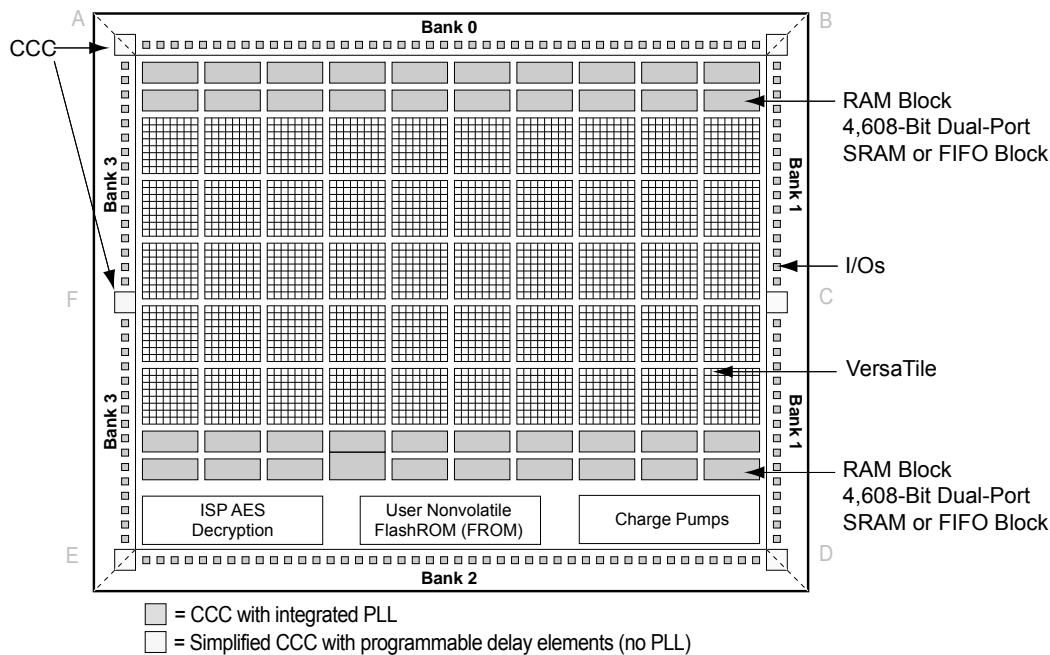


Figure 4-17 • CCC Locations in Fusion Family Devices (AFS090, AFS250, M1AFS250)

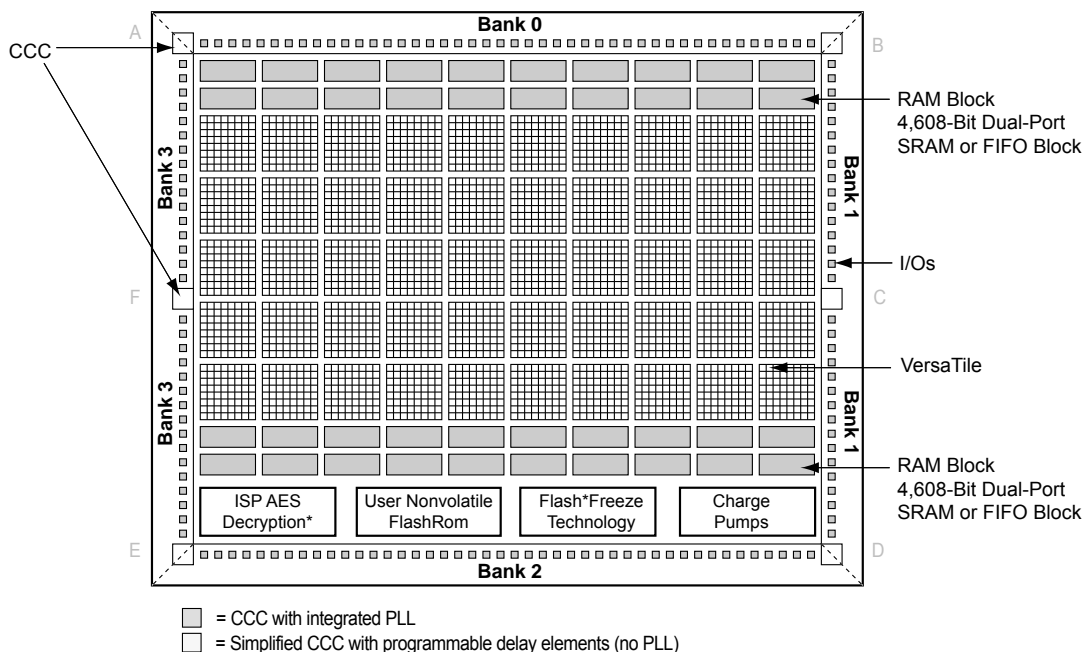


Figure 4-18 • CCC Locations in Fusion Family Devices (except AFS090, AFS250, M1AFS250)

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 4-24. 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 4-24 • Internal Feedback with Selectable System Delay

Figure 4-25 shows the controllable Feedback Delay. If set properly in conjunction with the fixed System Delay, the total output delay can be advanced significantly.

Figure 4-25 • Internal Feedback with Selectable Feedback Delay

Figure 4-31 • 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

Conclusion

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

Related Documents

Application Notes

Board-Level Considerations

http://www.microsemi.com/soc/documents/ALL_AC276_AN.pdf

Datasheets

Fusion Family of Mixed Signal FPGAs

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

User's Guides

IGLOO, ProASIC3, SmartFusion, and Fusion Macro Library Guide

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

List of Changes

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

Date	Changes	Page
August 2012	The "Implementing EXTFB in ProASIC3/E Devices" section is new (SAR 36647).	86
	Table 4-7 • Delay Values in Libero SoC Software per Device Family was added to the "Clock Delay Adjustment" section (SAR 22709).	102
	The "Phase Adjustment" section was rewritten to explain better why the visual CCC shows both the actual phase and the actual delay that is equivalent to this phase shift (SAR 29647).	103
	The hyperlink for the <i>Board-Level Considerations</i> application note was corrected (SAR 36663)	128, 129
December 2011	Figure 4-20 • PLL Block Diagram, Figure 4-22 • CCC Block Control Bits – Graphical Representation of Assignments, and Table 4-12 • MUXA, MUXB, MUXC were revised to change the phase shift assignments for PLLs 4 through 7 (SAR 33791).	101, 105, 109
June 2011	The description for RESETEN in Table 4-8 • Configuration Bit Descriptions for the CCC Blocks was revised. The phrase "and should not be modified via dynamic configuration" was deleted because RESETEN is read only (SAR 25949).	106
July 2010	This chapter is no longer published separately with its own part number and version but is now part of several FPGA fabric user's guides.	N/A
	Notes were added where appropriate to point out that IGLOO nano and ProASIC3 nano devices do not support differential inputs (SAR 21449).	N/A

Date	Changes	Page
v1.2 (June 2008)	The following changes were made to the family descriptions in Figure 4-1 • Overview of the CCCs Offered in Fusion, IGLOO, and ProASIC3: <ul style="list-style-type: none"> • ProASIC3L was updated to include 1.5 V. • The number of PLLs for ProASIC3E was changed from five to six. 	77
v1.1 (March 2008)	Table 4-1 • Flash-Based FPGAs and the associated text were updated to include the IGLOO PLUS family. The "IGLOO Terminology" section and "ProASIC3 Terminology" section are new.	79
	The "Global Input Selections" section was updated to include 15 k gate devices as supported I/O types for globals, for CCC only.	87
	Table 4-5 • Number of CCCs by Device Size and Package was revised to include ProASIC3L, IGLOO PLUS, A3P015, AGL015, AGLP030, AGLP060, and AGLP125.	94
	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.	97
v1.0 (January 2008)	Information about unlocking the PLL was removed from the "Dynamic PLL Configuration" section.	103
	In the "Dynamic PLL Configuration" section, information was added about running Layout and determining the exact setting of the ports.	116
	In Table 4-8 • 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.	106

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).

		Byte Number in Page															
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Page Number	7																
	6																
	5																
	4																
	3																
	2																
	1																
	0																

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 138). 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 Microsemi flash-based lock technology (FlashLock®). Together, they provide unmatched levels of security in a programmable logic device. This security applies to both the FPGA core and FlashROM content. These devices use the 128-bit AES (Rijndael) algorithm to encrypt programming files for secure transmission to the on-chip AES decryption core. The same algorithm is then used to decrypt the programming file. This key size provides approximately 3.4×10^{38} possible 128-bit keys. A computing system that could find a DES key in a second would take approximately 149 trillion years to crack a 128-bit AES key. The 128-bit FlashLock feature in low power flash devices works via a FlashLock security Pass Key mechanism, where the user locks or unlocks the device with a user-defined key. Refer to the "Security in Low Power Flash Devices" section on page 301.

If the device is locked with certain security settings, functions such as device read, write, and erase are disabled. This unique feature helps to protect against invasive and noninvasive attacks. Without the correct Pass Key, access to the FPGA is denied. To gain access to the FPGA, the device first must be unlocked using the correct Pass Key. During programming of the FlashROM or the FPGA core, you can generate the security header programming file, which is used to program the AES key and/or FlashLock Pass Key. The security header programming file can also be generated independently of the FlashROM and FPGA core content. The FlashLock Pass Key is not stored in the FlashROM.

Low power flash devices with AES-based security allow for secure remote field updates over public networks such as the Internet, and ensure that valuable intellectual property (IP) remains out of the hands of IP thieves. Figure 5-5 shows this flow diagram.

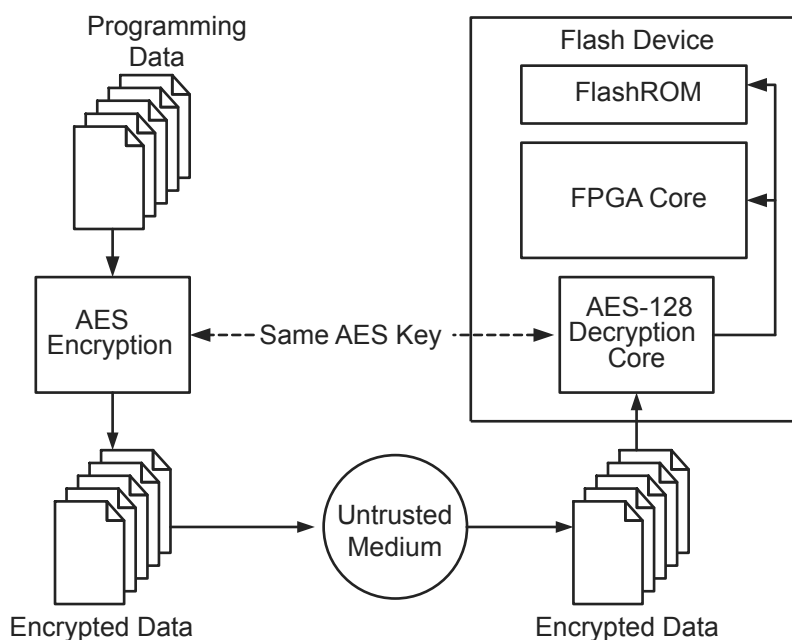


Figure 5-5 • Programming FlashROM Using AES

Figure 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. 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.

SRAM/FIFO Support in Flash-Based Devices

The flash FPGAs listed in Table 6-1 support SRAM and FIFO blocks and the functions described in this document.

Table 6-1 • Flash-Based FPGAs

Series	Family*	Description
IGLOO	IGLOO	Ultra-low power 1.2 V to 1.5 V FPGAs with Flash*Freeze technology
	IGLOOe	Higher density IGLOO FPGAs with six PLLs and additional I/O standards
	IGLOO nano	The industry's lowest-power, smallest-size solution
	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
	ProASIC3 nano	Lowest-cost solution with enhanced I/O capabilities
	ProASIC3L	ProASIC3 FPGAs supporting 1.2 V to 1.5 V with Flash*Freeze technology
	RT ProASIC3	Radiation-tolerant RT3PE600L and RT3PE3000L
	Military ProASIC3/EL	Military temperature A3PE600L, A3P1000, and A3PE3000L
	Automotive ProASIC3	ProASIC3 FPGAs qualified for automotive applications
Fusion	Fusion	Mixed signal FPGA integrating ProASIC3 FPGA fabric, programmable analog block, support for ARM® Cortex™-M1 soft processors, and flash memory into a monolithic device

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

IGLOO Terminology

In documentation, the terms IGLOO series and IGLOO devices refer to all of the IGLOO devices as listed in Table 6-1. Where the information applies to only one product line or limited devices, these exclusions will be explicitly stated.

ProASIC3 Terminology

In documentation, the terms ProASIC3 series and ProASIC3 devices refer to all of the ProASIC3 devices as listed in Table 6-1. Where the information applies to only one product line or limited devices, these exclusions will be explicitly stated.

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

The 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 the "SRAM and FIFO Memories in Microsemi's Low Power Flash Devices" section on page 147 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 (.Shiftin(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));
```

8 – I/O Structures in IGLOOe and ProASIC3E 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. IGLOO[®]e, ProASIC[®]3EL, and ProASIC3E families support Pro 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 Libero SoC software provides an easy way to implement I/O that will result in robust I/O design.

This document first describes the two different I/O types in terms of the standards and features they support. It then explains the individual features and how to implement them in Libero SoC.

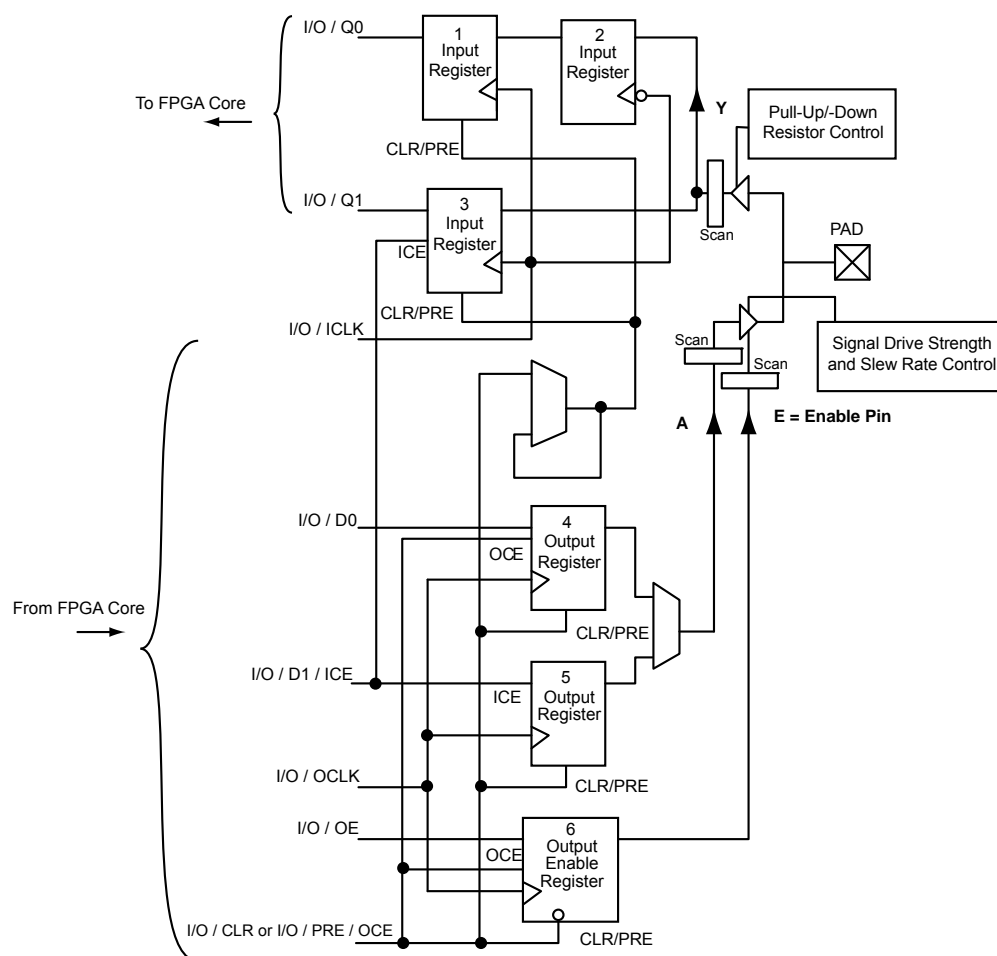


Figure 8-1 • DDR Configured I/O Block Logical Representation

Date	Changes	Page
v1.3 (October 2008)	The "Low Power Flash Device I/O Support" section was revised to include new families and make the information more concise.	214
v1.2 (June 2008)	<p>The following changes were made to the family descriptions in Table 8-1 · Flash-Based FPGAs:</p> <ul style="list-style-type: none"> • ProASIC3L was updated to include 1.5 V. • The number of PLLs for ProASIC3E was changed from five to six. 	214
v1.1 (March 2008)	<p>This document was previously part of <i>I/O Structures in IGLOO and ProASIC3 Devices</i>. To provide information specific to IGLOOe, ProASIC3E, and ProASIC3EL, the content was separated and made into a new document.</p> <p>For information on other low power flash family I/O structures, refer to the following documents:</p> <p><i>I/O Structures in IGLOO and ProASIC3 Devices</i> contains information specific to IGLOO, ProASIC3, and ProASIC3L I/O features.</p> <p><i>I/O Structures in IGLOO PLUS Devices</i> contains information specific to IGLOO PLUS I/O features.</p>	N/A

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.

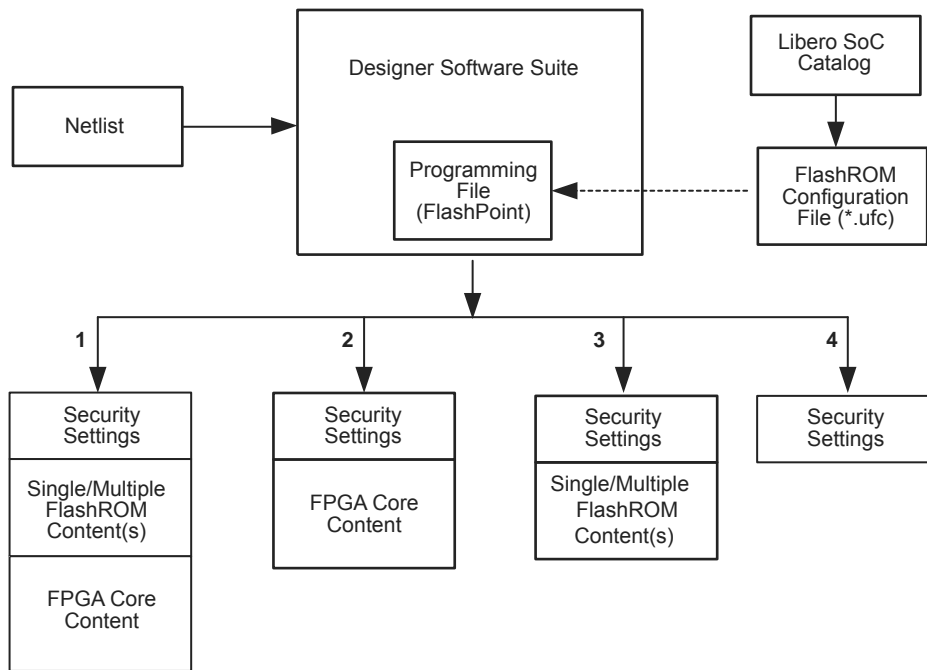


Figure 13-4 • Flexible Programming File Generation for Different Applications

Programming Solution

For device programming, any IEEE 1532-compliant programmer can be used; however, the FlashPro4/3/3X programmer must be used to control the low power flash device's rich security features and FlashROM programming options. The FlashPro4/3/3X programmer is a low-cost portable programmer for the Microsemi flash families. It can also be used with a powered USB hub for parallel programming. General specifications for the FlashPro4/3/3X 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 FlashPro4/3/3X 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 VCC, VCCI, VPUMP, and VJTAG during programming. However, if there is only one device on the target board, the FlashPro4/3/3X programmer can generate the required VPUMP voltage from the USB port.

Remote Upgrade via TCP/IP

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

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

Hardware Requirement

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

Security

Encrypted Programming

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

Message Authentication Code (MAC) Validation/Authentication

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

Low power flash devices with AES and MAC are superior to devices with only DES or 3DES encryption. Because the MAC verifies the correctness of the data, the FPGA is protected from erroneous loading of invalid programming data that could damage a device (Figure 15-5 on page 355).

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

Figure 18-3 • I/O State when VCCI Is Powered before VCC

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 (VCCI or VCC) 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 that the functional level of the power supply during power-up may vary slightly within the specification at different ramp-rates. Refer to Table 18-2 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 18-2 on page 377 and Figure 18-3.

Table 18-2 • Power-Up Functional Activation Levels for VCC and VCCI

Device	VCC Functional Activation Level (V)	VCCI Functional Activation Level (V)
ProASIC3, ProASIC3 nano, IGLOO, IGLOO nano, IGLOO PLUS, and ProASIC3L devices running at VCC = 1.5 V*	0.85 V \pm 0.25 V	0.9 V \pm 0.3 V
IGLOO, IGLOO nano, IGLOO PLUS, and ProASIC3L devices running at VCC = 1.2 V*	0.85 V \pm 0.2 V	0.9 V \pm 0.15 V

Note: *V5 devices will require a 1.5 V VCC supply, whereas V2 devices can utilize either a 1.2 V or 1.5 V VCC.

Microsemi'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 18-4 on page 379 and Figure 18-5 on page 380 for more information.

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 VCC and VCCI 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. VCCI should not exceed 3.6 V, per datasheet specifications.

As described in the "Power-Up to Functional Time" section on page 378, Microsemi's low power flash I/Os are tristated before and during power-up until the last voltage supply (VCC or VCCI) is powered up past its functional level. Furthermore, applying voltage to the FPGA I/Os does not pull up VCC or VCCI and, therefore, does not partially power up the device. Table 18-4 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 18-4 • Cold-Sparing Test Results for A3PE600 Devices

Device I/O	Residual Voltage (V)		Leakage Current
	VCC	VCCI	
Input	0	0.003	<1 μ A
Output	0	0.003	<1 μ A

VCCI 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 nano
- All IGLOO PLUS
- All IGLOOe
- ProASIC3L: A3PE3000L
- ProASIC3: A3P015 and A3P030
- All ProASIC3 nano
- All ProASIC3E
- Military ProASIC3EL: A3PE600L and A3PE3000L
- RT ProASIC3: RT3PE600L and RT3PE3000L

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