



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.

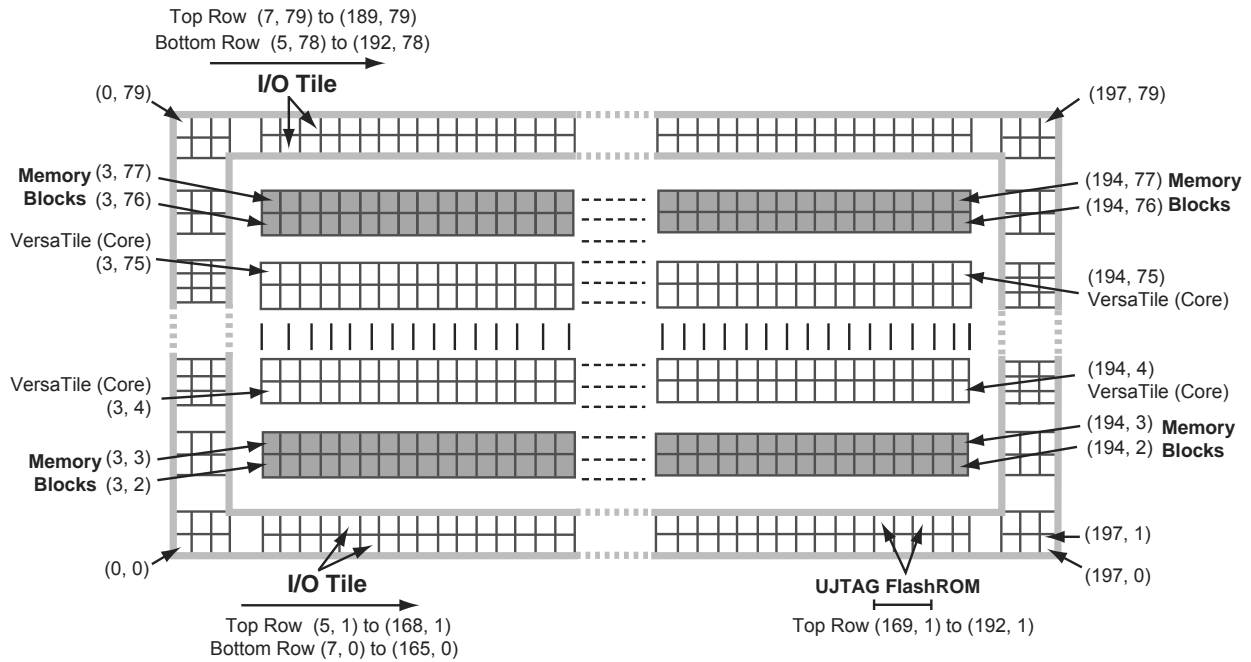
#### Details

Product Status	Obsolete
Number of LABs/CLBs	-
Number of Logic Elements/Cells	-
Total RAM Bits	18432
Number of I/O	71
Number of Gates	60000
Voltage - Supply	1.425V ~ 1.575V
Mounting Type	Surface Mount
Operating Temperature	-20°C ~ 85°C (TJ)
Package / Case	100-TQFP
Supplier Device Package	100-VQFP (14x14)
Purchase URL	<a href="https://www.e-xfl.com/product-detail/microchip-technology/a3pn060-z1vq100">https://www.e-xfl.com/product-detail/microchip-technology/a3pn060-z1vq100</a>

PLL Core Specifications	84
Functional Description	85
Software Configuration	96
Detailed Usage Information	104
Recommended Board-Level Considerations	112
Conclusion	113
Related Documents	113
List of Changes	113
<b>5 FlashROM in Microsemi's Low Power Flash Devices</b>	<b>117</b>
Introduction	117
Architecture of User Nonvolatile FlashROM	117
FlashROM Support in Flash-Based Devices	118
FlashROM Applications	120
FlashROM Security	121
Programming and Accessing FlashROM	122
FlashROM Design Flow	124
Custom Serialization Using FlashROM	129
Conclusion	130
Related Documents	130
List of Changes	130
<b>6 SRAM and FIFO Memories in Microsemi's Low Power Flash Devices</b>	<b>131</b>
Introduction	131
Device Architecture	131
SRAM/FIFO Support in Flash-Based Devices	134
SRAM and FIFO Architecture	135
Memory Blocks and Macros	135
Initializing the RAM/FIFO	148
Software Support	154
Conclusion	157
List of Changes	157
<b>7 I/O Structures in nano Devices</b>	<b>159</b>
Introduction	159
Low Power Flash Device I/O Support	161
nano Standard I/Os	162
I/O Architecture	164
I/O Standards	166
Wide Range I/O Support	166
I/O Features	167
Simultaneously Switching Outputs (SSOs) and Printed Circuit Board Layout	176
I/O Software Support	177
User I/O Naming Convention	178
I/O Bank Architecture and CCC Naming Conventions	179
Board-Level Considerations	181
Conclusion	182
Related Documents	183
List of Changes	183

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

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



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

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

---

## 3 – Global Resources in Low Power Flash Devices

---

### Introduction

IGLOO, Fusion, and ProASIC3 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 contain spines (the vertical branches of the global network tree) and ribs that can reach all the VersaTiles inside their region. This allows users the flexibility to create low-skew local clock 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 the "Clock Conditioning Circuits in Low Power Flash Devices and Mixed Signal FPGAs" section on page 61. This chapter 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 global 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 (except the PQ208 package, which has only 2 PLLs).
- In IGLOO, IGLOO nano, IGLOO PLUS, ProASIC3, and ProASIC3L devices, the west CCC contains a PLL core (except in 10 k through 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.

Refer to Table 4-6 on page 84 for details. 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 chip global lines on each side of the chip (six chip 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 10 k through 30 k gate devices).

The nano 10 k, 15 k, and 20 k devices support four VersaNet global resources, and 30 k devices support six global resources. The 10 k through 30 k devices have simplified CCCs called CCC-GLs.

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.

**Note:** Microsemi recommends that you choose the appropriate global pin and use the appropriate global resource so you can realize these benefits.

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



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 Low Power Flash Devices and Mixed Signal FPGAs" section on page 61. Microsemi 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 3-5 on page 36). 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 45) 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 3-20).

For example, if you want to limit the CLK1\_c signal to the left half of the chip and want to 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 3-20 • 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

### User's Guides

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

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

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 the "UJTAG Applications in Microsemi's Low Power Flash Devices" section on page 297 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 the "Global Resources in Low Power Flash Devices" section on page 31.

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 68 for more information.

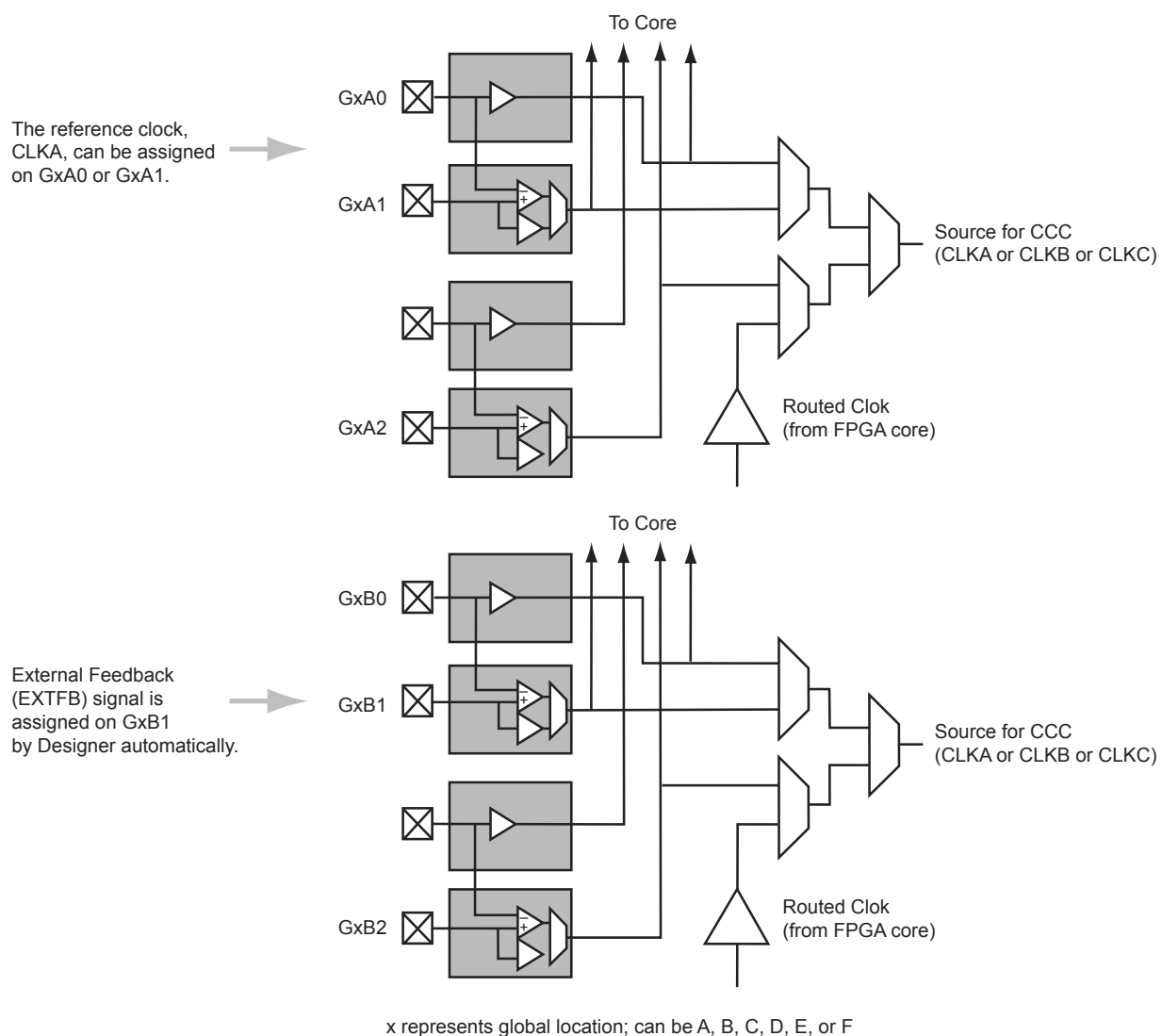
Each global buffer, as well as the PLL reference clock, can be driven from one of the following:

- 3 dedicated single-ended I/Os using a hardwired connection
- 2 dedicated differential I/Os using a hardwired connection (not supported for IGLOO nano or ProASIC3 nano devices)
- The FPGA core

## Implementing EXTFB in ProASIC3/E Devices

When the external feedback (EXTFB) signal of the PLL in the ProASIC3/E devices is implemented, the phase detector of the PLL core receives the reference clock (CLKA) and EXTFB as inputs. EXTFB must be sourced as an INBUF macro and located at the global/chip clock location associated with the target PLL by Designer software. EXTFB cannot be sourced from the FPGA fabric.

The following example shows CLKA and EXTFB signals assigned to two global I/Os in the same global area of ProASIC3E device.



**Figure 4-5 • CLKA and EXTFB Assigned to Global I/Os**

## Phase Adjustment

The four phases available (0, 90, 180, 270) are phases with respect to VCO (PLL output). The VCO is divided to achieve the user's CCC required output frequency (GLA, YB/GLB, YC/GLC). The division happens after the selection of the VCO phase. The effective phase shift is actually the VCO phase shift divided by the output divider. This is why the visual CCC shows both the actual achievable phase and more importantly the actual delay that is equivalent to the phase shift that can be achieved.

## 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.

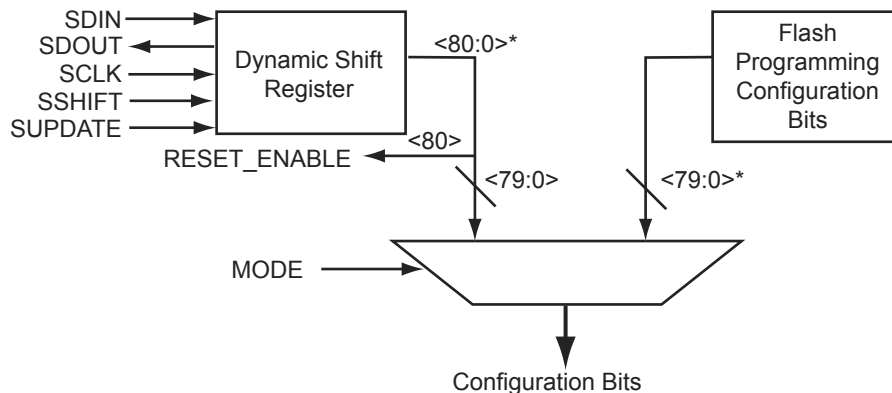
### Flash Configuration Bits

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 4-21 illustrates a simplified block diagram of the MUX architecture in the CCCs.



Note: \*For Fusion, bit <88:81> is also needed.

**Figure 4-21 • 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 4-21. 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 90. 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. Microsemi 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 4-22 shows a model of the CCC with configurable blocks and switches.

## FlashROM Support in Flash-Based Devices

The flash FPGAs listed in Table 5-1 support the FlashROM feature and the functions described in this document.

**Table 5-1 • 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 5-1. 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 5-1. 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*.

## 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.

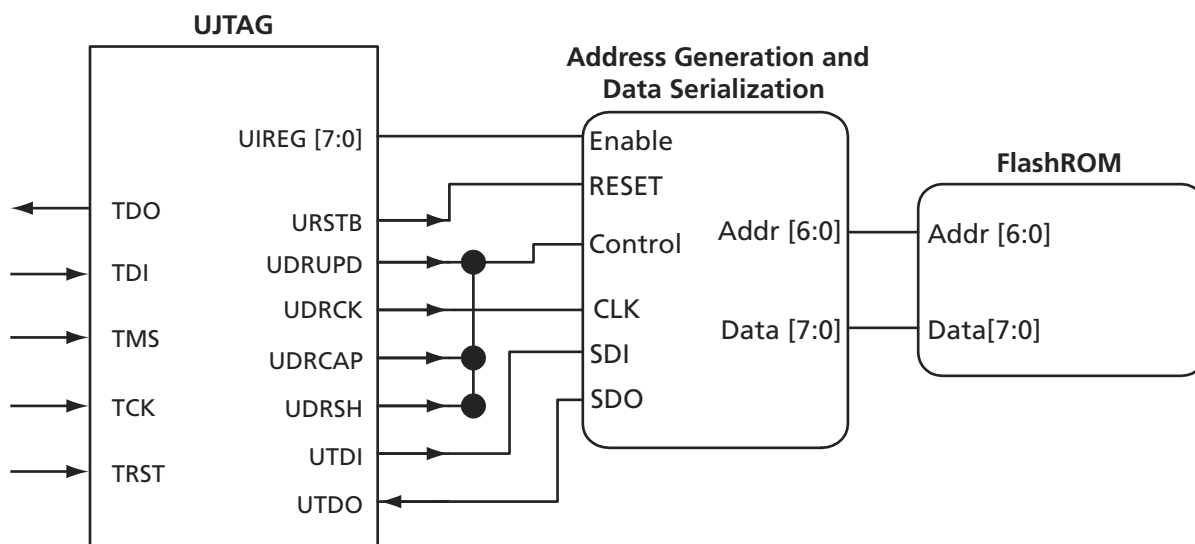
A 7-bit address from the FPGA core defines which of the eight pages (three MSBs) is being read, and which of the 16 bytes within the selected page (four LSBs) are being read. The FlashROM content can be read on a random basis; the access time is 10 ns for a device supporting commercial specifications. 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 the "UJTAG Applications in Microsemi's Low Power Flash Devices" section on page 297 for details on using the UJTAG macro to read the FlashROM.

Figure 5-7 on page 123 and Figure 5-8 on page 123 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**



## 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. Microsemi recommends that you use different file names if you plan to generate multiple MEM files. During simulation, Libero SoC 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.

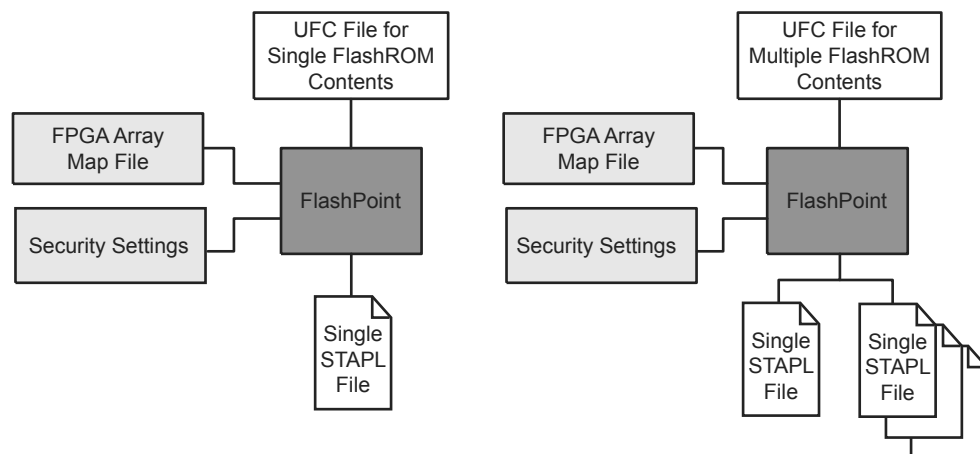
```
.....
UFROM0: UFROM
--generic map(MEMORYFILE => "F:\Appsnotes\FROM\test_designs\testa\smartgen\FROM_a.mem")
--generic map(MEMORYFILE => "F:\Appsnotes\FROM\test_designs\testa\smartgen\FROM_b.mem")
.....
```

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**

## I/O Software Support

In Microsemi's Libero 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 7-15 • nano 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
				IGLOO nano	ProASIC 3 nano			
LVTTTL/ LVCMOS3.3	✓	✓ (8)	✓	✓	✓	✓	✓	✓
LVCMOS2.5	✓	✓ (8)	✓	✓	✓	✓	✓	✓
LVCMOS1.8	✓	✓ (4)	✓	✓	✓	✓	✓	✓
LVCMOS1.5	✓	✓ (2)	✓	✓	✓	✓	✓	✓
LVCMOS1.2	✓	✓ (2)	✓	✓	–	✓	✓	✓
Software Defaults	HIGH	Refer to numbers in parentheses in above cells.	None	All Devices: 5 pF	10 pF or 35 pF*	Off	Off	No

Note: \*10 pF for A3PN010, A3PN015, and A3PN020; 35 pF for A3PN060, A3PN125, and A3PN250.

## 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 8-17). The IOBA tool automatically assigns technologies and VREF 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 8-16).

---

---

### **Figure 8-16 • 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 8-17.

---

---

### **Figure 8-17 • I/O Bank Assigner Displays Messages in Output Window**

---

**Figure 11-10 • All Silicon Features Selected for IGLOO and ProASIC3 Devices**

---

---

**Figure 11-11 • All Silicon Features Selected for Fusion**

---

*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 11-17 • 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 11-18 on page 255). 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.

An AES-encrypted file programmed to a device without FlashLock would not be secure, since without FlashLock to protect the AES key, someone could simply reprogram the AES key first, then program with any AES key desired or no AES key at all. This option is therefore not available in the software.

---

## 12 – In-System Programming (ISP) of Microsemi's Low Power Flash Devices Using FlashPro4/3/3X

---

### Introduction

Microsemi'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 FlashPro4/3/3X programmers<sup>1</sup>.

IGLOO, ProASIC3, SmartFusion, and Fusion 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. IGLOO, ProASIC3, SmartFusion, and Fusion devices offer the best in-system programming (ISP) solution, FlashLock<sup>®</sup> security features, and AES-decryption-based ISP.

### ISP Architecture

Low power flash devices support ISP via JTAG and require a single VPUMP voltage of 3.3 V during programming. In addition, programming via a microcontroller in a target system is also supported.

Refer to the "Microprocessor Programming of Microsemi's Low Power Flash Devices" chapter of an appropriate FPGA fabric user's guide.

Family-specific support:

- ProASIC3, ProASIC3E, SmartFusion, and Fusion devices support ISP.
- ProASIC3L devices operate using a 1.2 V core voltage; however, programming can be done only 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 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 nano V2 devices can be programmed at 1.2 V core voltage (when using FlashPro4 only) or 1.5 V. IGLOO nano V5 devices are programmed with a VCC core voltage of 1.5 V. Voltage switching is required in-system to switch from a 1.2 V supply (VCC, VCCI, and VJTAG) to 1.5 V for programming. The exception is that V2 devices can be programmed at 1.2 V VCC with FlashPro4.

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

IGLOO, ProASIC3, SmartFusion, and Fusion devices support the JTAG-based IEEE 1532 standard for ISP. To start JTAG operations, the IGLOO device must 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

---

1. *FlashPro4 replaced FlashPro3/3X in 2010 and is backward compatible with FlashPro3/3X as long as there is no connection to pin 4 on the JTAG header on the board. On FlashPro3/3X, there is no connection to pin 4 on the JTAG header; however, pin 4 is used for programming mode (Prog\_Mode) on FlashPro4. When converting from FlashPro3/3X to FlashPro4, users should make sure that JTAG connectors on system boards do not have any connection to pin 4. FlashPro3X supports discrete TCK toggling that is needed to support non-JTAG compliant devices in the chain. This feature is included in FlashPro4.*

## ISP Programming Header Information

The FlashPro4/3/3X programming cable connector can be connected with a 10-pin, 0.1"-pitch programming header. The recommended programming headers are manufactured by AMP (103310-1) and 3M (2510-6002UB). If you have limited board space, you can use a compact programming header manufactured by Samtec (FTSH-105-01-L-D-K). Using this compact programming header, you are required to order an additional header adapter manufactured by Microsemi SoC Products Group (FP3-10PIN-ADAPTER-KIT).

Existing ProASIC<sup>PLUS</sup> family customers who are using the Samtec Small Programming Header (FTSH-113-01-L-D-K) and are planning to migrate to IGLOO or ProASIC3 devices can also use FP3-10PIN-ADAPTER-KIT.

**Table 12-3 • Programming Header Ordering Codes**

Manufacturer	Part Number	Description
AMP	103310-1	10-pin, 0.1"-pitch cable header (right-angle PCB mount angle)
3M	2510-6002UB	10-pin, 0.1"-pitch cable header (straight PCB mount angle)
Samtec	FTSH-113-01-L-D-K	Small programming header supported by FlashPro and Silicon Sculptor
Samtec	FTSH-105-01-L-D-K	Compact programming header
Samtec	FFSD-05-D-06.00-01-N	10-pin cable with 50 mil pitch sockets; included in FP3-10PIN-ADAPTER-KIT.
Microsemi	FP3-10PIN-ADAPTER-KIT	Transition adapter kit to allow FP3 to be connected to a micro 10-pin header (50 mil pitch). Includes a 6 inch Samtec FFSD-05-D-06.00-01-N cable in the kit. The transition adapter board was previously offered as FP3-26PIN-ADAPTER and includes a 26-pin adapter for design transitions from ProASIC <sup>PLUS</sup> based boards to ProASIC3 based boards.

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

*Note: \*Prog\_Mode on FlashPro4 is an output signal that goes High during device programming and returns to Low when programming is complete. This signal can be used to drive a system to provide a 1.5 V programming signal to IGLOO nano, ProASIC3L, and RT ProASIC3 devices that can run with 1.2 V core voltage but require 1.5 V for programming. IGLOO nano V2 devices can be programmed at 1.2 V core voltage (when using FlashPro4 only), but IGLOO nano V5 devices are programmed with a VCC core voltage of 1.5 V.*

**Figure 12-5 • Programming Header (top view)**

## List of Changes

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

Date	Changes	Page
August 2012	In the "Boundary Scan Chain" section, the reference made to the datasheet for pull-up/-down recommendations was changed to mention TCK and TRST pins rather than TDO and TCK pins. TDO is an output, so no pull resistor is needed (SAR 35937).	293
	The "Advanced Boundary Scan Register Settings" section is new (SAR 38432).	295
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
	Table 15-3 • TRST and TCK Pull-Down Recommendations was revised to add VJTAG at 1.2 V.	294
v1.4 (December 2008)	IGLOO nano and ProASIC3 nano devices were added to Table 15-1 • Flash-Based FPGAs.	292
v1.3 (October 2008)	The "Boundary Scan Support in Low Power Devices" section was revised to include new families and make the information more concise.	293
v1.2 (June 2008)	The following changes were made to the family descriptions in Table 15-1 • Flash-Based FPGAs: <ul style="list-style-type: none"> <li>ProASIC3L was updated to include 1.5 V.</li> <li>The number of PLLs for ProASIC3E was changed from five to six.</li> </ul>	292
v1.1 (March 2008)	The chapter was updated to include the IGLOO PLUS family and information regarding 15 k gate devices.	N/A
	The "IGLOO Terminology" section and "ProASIC3 Terminology" section are new.	292



Revision (month/year)	Chapter Affected	List of Changes (page number)
Revision 1 (continued)	"In-System Programming (ISP) of Microsemi's Low Power Flash Devices Using FlashPro4/3/3X" was revised.	273
	"Core Voltage Switching Circuit for IGLOO and ProASIC3L In-System Programming" was revised.	281
	"Boundary Scan in Low Power Flash Devices" was revised.	296
Revision 0 (April 2010)	The ProASIC3 nano Low Power Flash FPGAs Handbook was divided into two parts to create the ProASIC3 nano Datasheet ProASIC3 nano Device Family User's Guide.	N/A