Welcome to E-XFL.COM

Understanding Embedded - FPGAs (Field Programmable Gate Array)

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

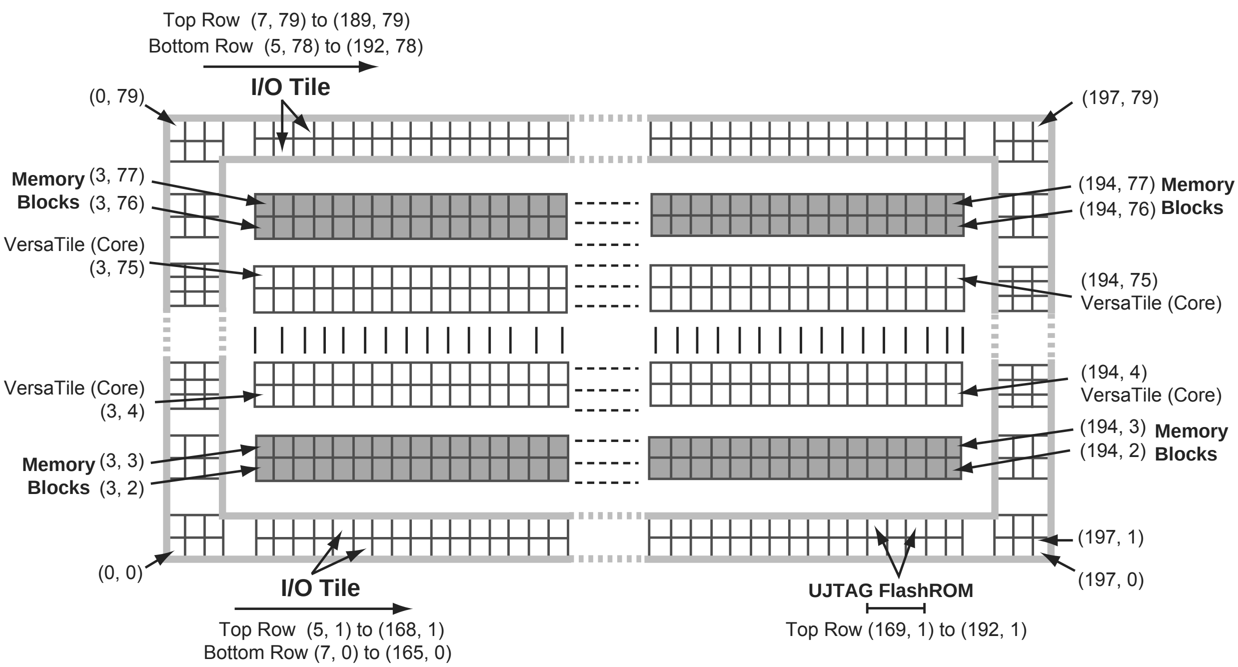
Applications of Embedded - FPGAs

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

Details	
Product Status	Obsolete
Number of LABs/CLBs	-
Number of Logic Elements/Cells	-
Total RAM Bits	36864
Number of I/O	151
Number of Gates	250000
Voltage - Supply	1.14V ~ 1.575V
Mounting Type	Surface Mount
Operating Temperature	-40°C ~ 100°C (TJ)
Package / Case	208-BFQFP
Supplier Device Package	208-PQFP (28x28)
Purchase URL	https://www.e-xfl.com/product-detail/microsemi/a3p250l-1pq208i

***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, AGLE600, A3P600, and A3PE600**

Flash Families Support the Flash*Freeze Feature

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

Table 2-1 • **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	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

*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 2-1. 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 2-1. 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*.

Date	Changes	Page
51900147-2/5.07	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.	24
	Figure 2-2 • Flash*Freeze Mode Type 1 – Timing Diagram was updated.	25
	Information about ULSICC was added to the "Prototyping for IGLOO and ProASIC3L Devices Using ProASIC3" section.	2-21
51900147-1/3.07	In the "Flash*Freeze Mode" section, "active high" was changed to "active low."	24
	The "Prototyping for IGLOO and ProASIC3L Devices Using ProASIC3" section was updated with information concerning the Flash*Freeze pin.	2-21

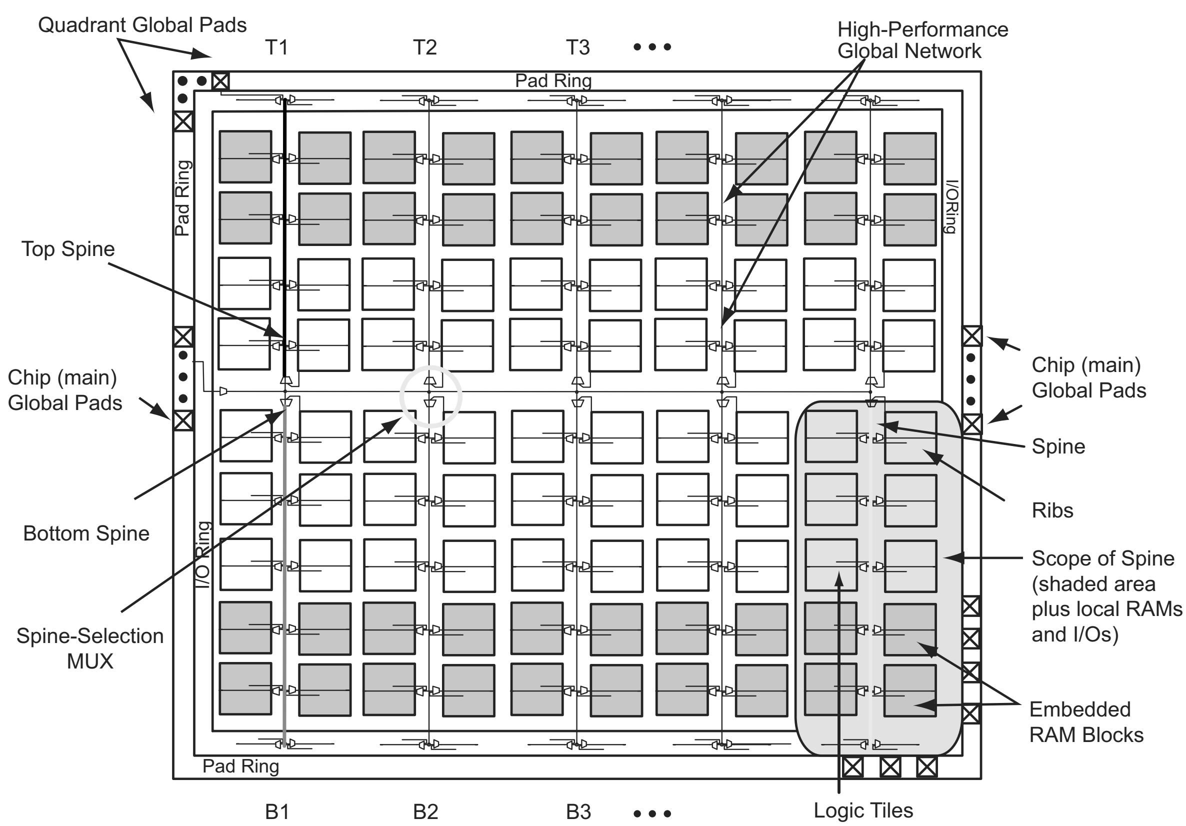
VersaNet Global Network Distribution

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 (except for nano 10 k, 15 k, and 20 k gate devices) that are distributed from the center of the FPGA array. In addition, each device (except the 10 k through 30 k gate device) has four quadrant global networks, each consisting of three quadrant global net resources. These quadrant global networks can only drive a signal inside their own quadrant. Each 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 an overview of the VersaNet global network and device architecture for devices 60 k and above. Figure 3-2 and Figure 3-3 on page 50 show simplified VersaNet global networks.

The VersaNet global networks are segmented and consist of 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 57.

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 clock 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 10 k through 30 k gate devices

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

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 4-12 for an example illustration of the connections, shown in red).

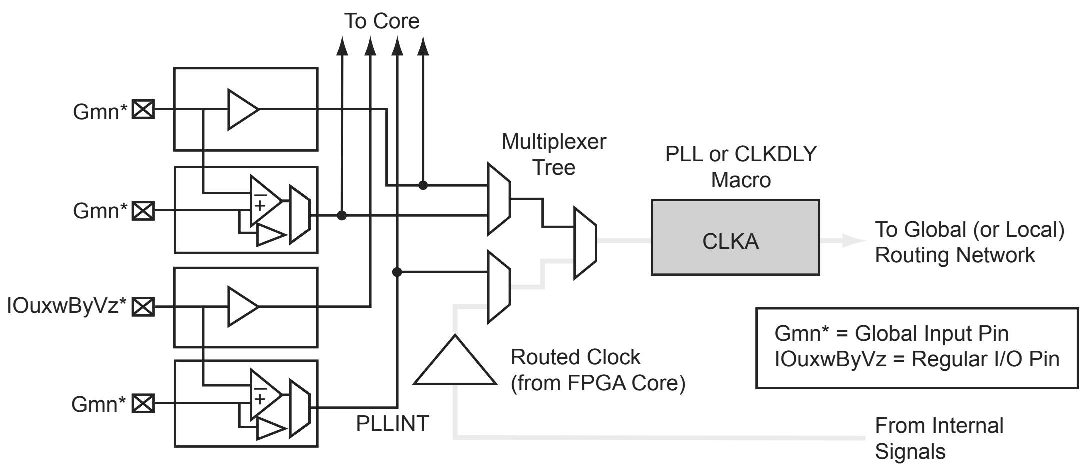


Figure 4-12 **• Illustration of Core Logic Usage**

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.

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	**IGLOO**			
A3PN010	AGLN010	All	0	2
A3PN015	AGLN015	All	0	2
A3PN020	AGLN020	All	0	2
	AGLN060	CS81	0	6
A3PN060	AGLN060	All other packages	1	5
	AGLN125	CS81	0	6
A3PN125	AGLN125	All other packages	1	5
	AGLN250	CS81	0	6
A3PN250	AGLN250	All other packages	1	5
A3P015	AGL015	All	0	2
A3P030	AGL030/AGLP030	All	0	2
	AGL060/AGLP060	CS121/CS201	0	6
A3P060	AGL060/AGLP060	All other packages	1	5
A3P125	AGL125/AGLP125	All	1	5
A3P250/L	AGL250	All	1	5
A3P400	AGL400	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

Note: nano 10 k, 15 k, and 20 k offer 6 global MUXes instead of CCCs.

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 4-34 • **Cascade PLL Configuration**

Using internal feedback, we know from EQ 4-1 on page 102 that the maximum achievable output frequency from the primary output is

$$f_{GLA} = f_{CLKA} \times m / (n \times u) = 2\text{ MHz} \times 128 / (1 \times 1) = 256\text{ MHz}$$

EQ 4-5

Figure 4-35 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.

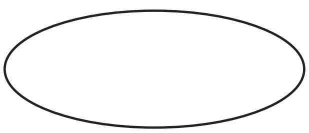


Figure 4-35 • **First-Stage PLL Showing Input of 2 MHz and Output of 256 MHz**

A second PLL can be connected serially to achieve the required frequency. EQ 4-1 on page 102 to EQ 4-3 on page 102 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

$$f_{GLC2} = f_{YC2} = f_{CLKA1} \times m_1 \times m_2 / (n_1 \times n_2 \times w_1 \times w2) \text{ – Secondary 2 PLL Output Clock(s)}$$

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):

$$f_{output} = f_{GLA2} = f_{GLA} \times m_2 / (n_2 \times u_2) = 256\text{ MHz} \times 70 / (64 \times 1) = 280\text{ MHz}$$

EQ 4-9

Figure 4-36 on page 127 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.

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: • 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

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.

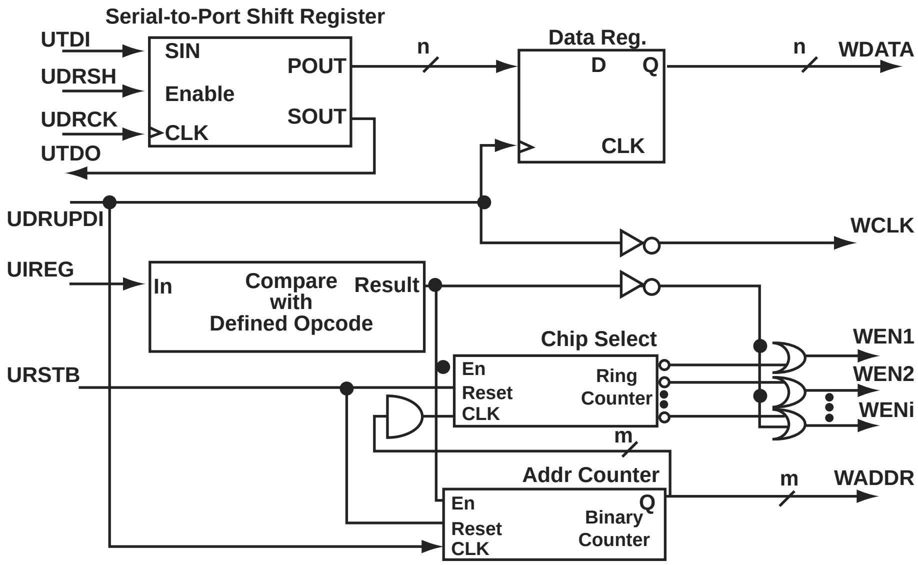


Figure 6-9 • **Block Diagram of a Sample User Interface**

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 166.

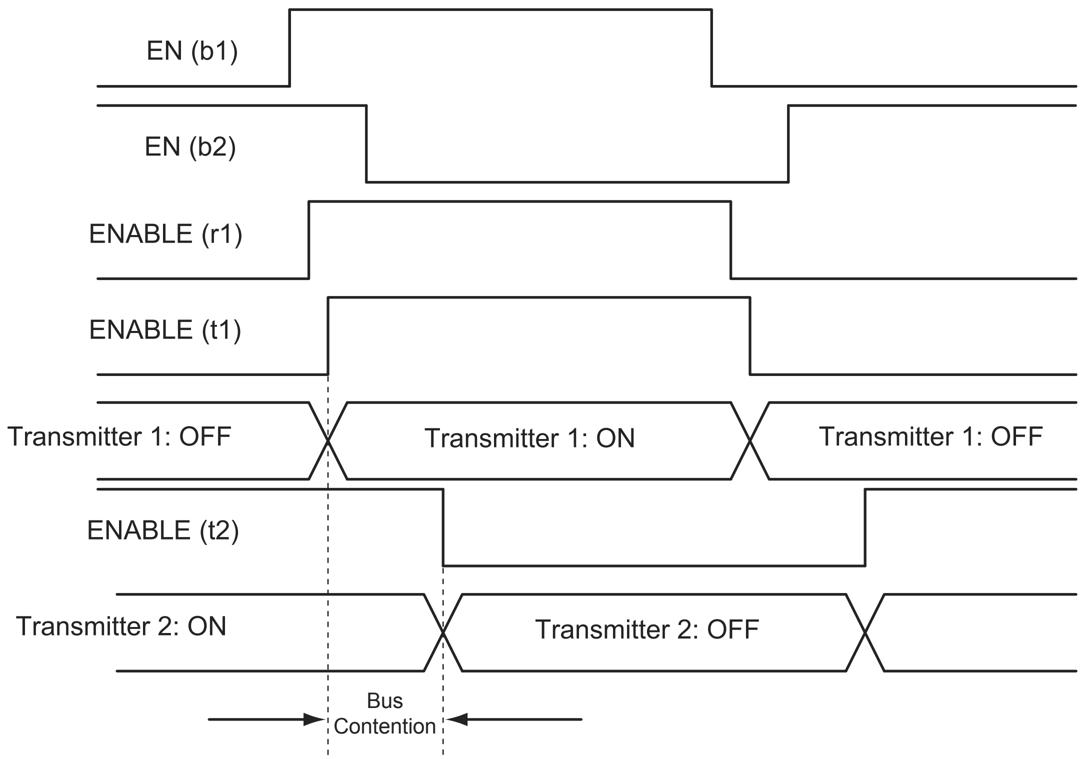


***Figure 7-17* • Timing Diagram (bypasses skew circuit)**

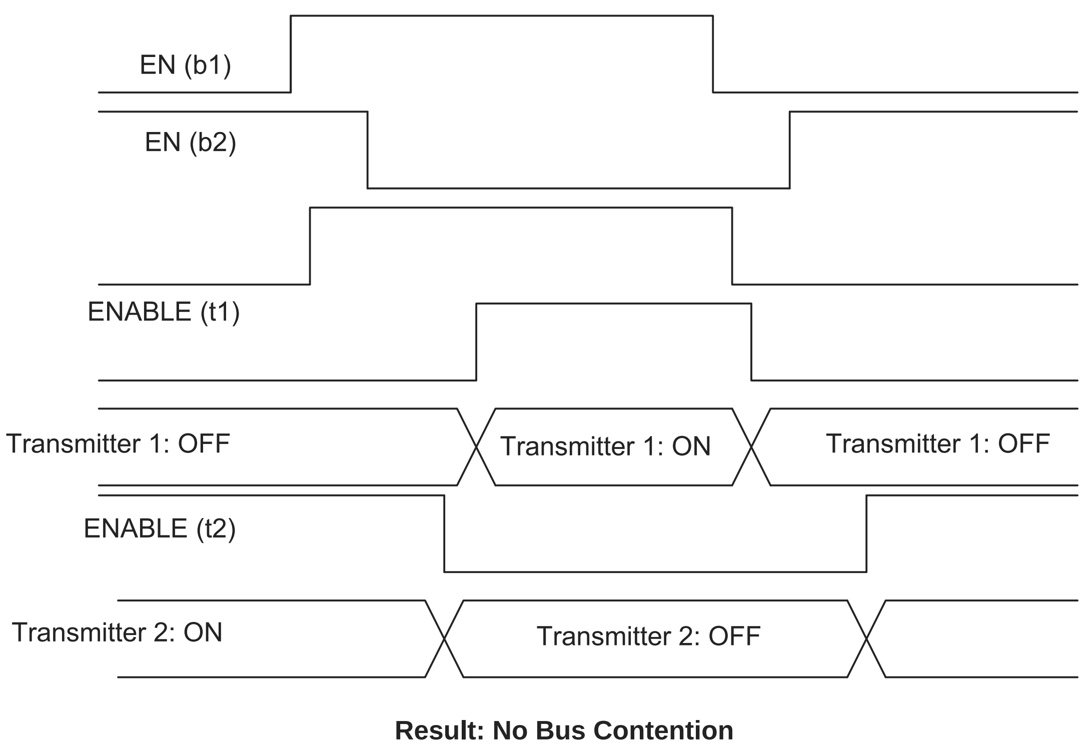


***Figure 7-18* • Timing Diagram (with skew circuit selected)**

Low Power Flash Device I/O Support

The low power flash FPGAs listed in Table 8-1 support I/Os and the functions described in this document.

***Table 8-1* • Flash-Based FPGAs**

Series	Family*	Description
IGLOO	IGLOOe	Higher density IGLOO FPGAs with six PLLs and additional I/O standards
ProASIC3	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
	Military ProASIC3/EL	Military temperature A3PE600L, A3P1000, and A3PE3000L
	RT ProASIC3	Radiation-tolerant RT3PE600L and RT3PE3000L

*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 8-1. 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 8-1. 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*.

Pro I/Os—IGLOOe, ProASIC3EL, and ProASIC3E

Table 8-2 shows the voltages and compatible I/O standards for Pro I/Os. I/Os provide programmable slew rates, drive strengths, and weak pull-up and pull-down circuits. All I/O standards, except 3.3 V PCI and 3.3 V PCI-X, are capable of hot-insertion. 3.3 V PCI and 3.3 V PCI-X can be configured to be 5 V–tolerant. See the "5 V Input Tolerance" section on page 232 for possible implementations of 5 V tolerance. Single-ended input buffers support both the Schmitt trigger and programmable delay options on a per–I/O basis.

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 8-2 **• Supported I/O Standards**

	A3PE600	AGLE600	A3PE1500	A3PE3000/ A3PE3000L	AGLE3000
Single-Ended					
LVTTL/LVCMOS 3.3 V, LVCMOS 2.5 V / 1.8 V / 1.5 V, LVCMOS 2.5/5.0 V, 3.3 V PCI/PCI-X	✓	✓	✓	✓	✓
LVCMOS 1.2 V	–	✓	–	–	✓
Differential					
LVPECL, LVDS, B-LVDS, M-LVDS	✓	✓	✓	✓	✓
Voltage-Referenced					
GTL+ 2.5 V / 3.3 V, GTL 2.5 V / 3.3 V, HSTL Class I and II, SSTL2 Class I and II, SSTL3 Class I and II	✓	✓	✓	✓	✓

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 256 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 9-7). 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 9-7 **• 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 VREF to I/O Banks" section on page 264 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 Bank Resource Usage

This is an important portion of the report. The user must meet the requirements stated in this table. Figure 9-10 shows the I/O Bank Resource Usage table included in the I/O bank report:

Figure 9-10 **• I/O Bank Resource Usage Table**

The example in Figure 9-10 shows that none of the I/O macros is assigned to the bank because more than one VCCI is detected.

I/O Voltage Usage

The I/O Voltage Usage table provides the number of VREF (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, VREF assignments must be made if there are any voltage-referenced I/Os.

Figure 9-11 shows the I/O Voltage Usage table included in the I/O bank report.

Figure 9-11 **• I/O Voltage Usage Table**

The table in Figure 9-11 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 VCCI voltage, their VREF voltages are different. As a result, two I/O banks are needed to assign the VCCI and VREF voltages.

In addition, there are six single-ended I/Os used that have the **same VCCI voltage. Since two banks are already assigned with the same VCCI voltage and there are enough u**nused bonded I/Os in

Related Documents

User’s Guides

FlashPro User's Guide

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

List of Changes

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

Date	Changes	Page
July 2010	This chapter is no longer published separately with its own part number and version but is now part of several FPGA fabric user’s guides.	N/A
v1.5 (August 2009)	The "CoreMP7 Device Security" section was removed from "Security in ARM-Enabled Low Power Flash Devices", since M7-enabled devices are no longer supported.	304
v1.4 (December 2008)	IGLOO nano and ProASIC3 nano devices were added to Table 12-1 • Flash-Based FPGAs.	302
v1.3 (October 2008)	The "Security Support in Flash-Based Devices" section was revised to include new families and make the information more concise.	302
v1.2 (June 2008)	The following changes were made to the family descriptions in Table 12-1 • Flash-Based FPGAs: • ProASIC3L was updated to include 1.5 V. • The number of PLLs for ProASIC3E was changed from five to six.	302
v1.1 (March 2008)	The chapter was updated to include the IGLOO PLUS family and information regarding 15 k gate devices.	N/A
	The "IGLOO Terminology" section and "ProASIC3 Terminology" section are new.	302

Programming Voltage (VPUMP) and VJTAG

Low-power flash devices support on-chip charge pumps, and therefore require only a single 3.3 V programming voltage for the VPUMP pin during programming. When the device is not being programmed, the VPUMP pin can be left floating or can be tied (pulled up) to any voltage between 0 V and 3.6 V[2]. During programming, the target board or the FlashPro4/3/3X programmer can provide VPUMP. FlashPro4/3/3X is capable of supplying VPUMP to a single device. If more than one device is to be programmed using FlashPro4/3/3X on a given board, FlashPro4/3/3X should not be relied on to supply the VPUMP voltage. A FlashPro4/3/3X programmer is not capable of providing reliable VJTAG voltage. The board must supply VJTAG voltage to the device and the VJTAG pin of the programmer header must be connected to the device VJTAG pin. Microsemi recommends that VPUMP[3] and VJTAG power supplies be kept separate with independent filtering capacitors rather than supplying them from a common rail. Refer to the "Board-Level Considerations" section on page 337 for capacitor requirements.

Low power flash device I/Os support a bank-based, voltage-supply architecture that simultaneously supports multiple I/O voltage standards (Table 13-2). 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). Microsemi recommends that TCK be tied to GND through a 200 ohm to 1 Kohm resistor. This prevents a possible totempole current on the input buffer stage. For TDI, TMS, and TRST pins, the devices provide an internal nominal 10 Kohm pull-up resistor. During programming, all I/O pins, except for JTAG interface pins, are tristated and weakly pulled up to VCCI. This isolates the part and prevents the signals from floating. The JTAG interface pins are driven by the FlashPro4/3/3X during programming, including the TRST pin, which is driven HIGH.

Table 13-2 **• Power Supplies**

Power Supply	Programming Mode	Current during Programming
VCC	1.2 V / 1.5 V	< 70 mA
VCCI	1.2 V / 1.5 V / 1.8 V / 2.5 V / 3.3 V (bank-selectable)	I/Os are weakly pulled up.
VJTAG	1.2 V / 1.5 V / 1.8 V / 2.5 V / 3.3 V	< 20 mA
VPUMP	3.15 V to 3.45 V	< 80 mA

Note: *All supply voltages should be at 1.5 V or higher, regardless of the setting during normal operation, except for IGLOO nano, where 1.2 V VCC and VJTAG programming is allowed.*

Nonvolatile Memory (NVM) Programming Voltage

SmartFusion and Fusion devices need stable VCCNVM/VCCENVM[3] (1.5 V power supply to the embedded nonvolatile memory blocks) and VCCOSC/VCCROSC[4] (3.3 V power supply to the integrated RC oscillator). The tolerance of VCCNVM/VCCENVM is ± 5% and VCCOSC/VCCROSC is ± 5%.

Unstable supply voltage on these pins can cause an NVM programming failure due to NVM page corruption. The NVM page can also be corrupted if the NVM reset pin has noise. This signal must be tied off properly.

Microsemi recommends installing the following capacitors[5] on the VCCNVM/VCCENVM and VCCOSC/VCCROSC pins:

- Add one bypass capacitor of 10 µF for each power supply plane followed by an array of decoupling capacitors of 0.1 µF.
- Add one 0.1 µF capacitor near each pin.

2. *During sleep mode in IGLOO devices connect VPUMP to GND.*
3. *VPUMP has to be quiet for successful programming. Therefore VPUMP must be separate and required capacitors must be installed close to the FPGA VPUMP pin.*
4. *VCCROSC is for SmartFusion.*
5. *The capacitors cannot guarantee reliable operation of the device if the board layout is not done properly.*

Microprocessor Programming Support in Flash Devices

The flash-based FPGAs listed in Table 15-1 support programming with a microprocessor and the functions described in this document.

Table 15-1 **• 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 15-1. Where the information applies to only one device 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 15-1. Where the information applies to only one device 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*.

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

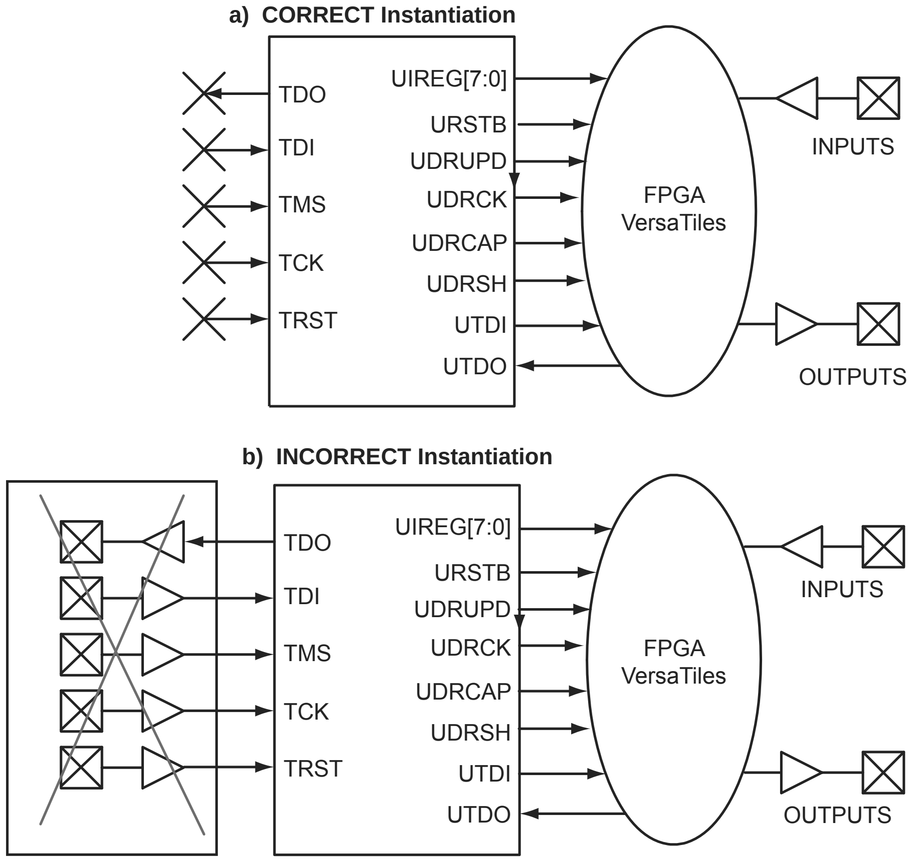
As an additional security measure, the devices are equipped with AES decryption. AES works in two steps. The first step is to program a key into the devices in a secure or trusted programming center (such as Microsemi SoC Products Group In-House Programming (IHP) center). The second step is to encrypt 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



Note: Do not connect JTAG pins (TDO, TDI, TMS, TCK, or TRST) to I/Os in the design.

Figure 17-3 **• 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 17-4 on page 367. 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.