



Welcome to E-XFL.COM

Understanding <u>Embedded - FPGAs (Field</u> <u>Programmable Gate Array)</u>

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	Active
Number of LABs/CLBs	-
Number of Logic Elements/Cells	-
Total RAM Bits	147456
Number of I/O	177
Number of Gates	1000000
Voltage - Supply	1.14V ~ 1.575V
Mounting Type	Surface Mount
Operating Temperature	0°C ~ 85°C (TJ)
Package / Case	256-LBGA
Supplier Device Package	256-FPBGA (17x17)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/a3p1000l-1fgg256

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong

Core Architecture

VersaTile

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

As illustrated in Figure 1-8, there are four inputs in a logic VersaTile cell, and each VersaTile can be configured using the appropriate flash switch connections:

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

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

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

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



Figure 1-8 • Low Power Flash Device Core VersaTile

Flash*Freeze Technology and Low Power Modes

Sleep and Shutdown Modes

Sleep Mode

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

Table 2-7 shows the power supply status in Sleep mode.

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

Power Supplies	Power Supply State
VCC	Powered off
VCCI = VMV	Powered on
VJTAG	Powered off
VPUMP	Powered off

Refer to the "Power-Up/-Down Behavior" section on page 33 for more information about I/O states during Sleep mode and the timing diagram for entering and exiting Sleep mode.

Shutdown Mode

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

Microsemi

Flash*Freeze Technology and Low Power Modes

Set/Reset

Since all I/Os and globals are tied High in Flash*Freeze mode (unless hold state is used on IGLOO nano or IGLOO PLUS), Microsemi recommends using active low set/reset at the top-level port. If needed, the signal can be inverted internally.

If the intention is to always set/reset in Flash*Freeze mode, a self set/reset circuit may be implemented to accomplish this, as shown in Figure 2-9. Configure an active High set/reset input pin so it uses the internal pull-up during Flash*Freeze mode, and drives Low during active mode. When the device exits Flash*Freeze mode, the input will transition from High to Low, releasing the set/reset. Note that this circuit may release set/reset before all outputs become active, since outputs are enabled up to 200 ns after inputs when exiting Flash*Freeze mode.



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

I/Os

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

JTAG

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

ULSICC

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

Flash*Freeze Management IP

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

ProASIC3L FPGA Fabric User's Guide



Figure 3-6 shows all nine global inputs for the location A connected to the top left quadrant global network via CCC.

Figure 3-6 • Global Inputs

Since each bank can have a different I/O standard, the user should be careful to choose the correct global I/O for the design. There are 54 global pins available to access 18 global networks. For the single-ended and voltage-referenced I/O standards, you can use any of these three available I/Os to access the global network. For differential I/O standards such as LVDS and LVPECL, the I/O macro needs to be placed on (A0, A1), (B0, B1), (C0, C1), or a similar location. The unassigned global I/Os can be used as regular I/Os. Note that pin names starting with GF and GC are associated with the chip global networks, and GA, GB, GD, and GE are used for quadrant global networks. Table 3-2 on page 54 and Table 3-3 on page 55 show the general chip and quadrant global pin names.



Global Resources in Low Power Flash Devices

	,	,
Differential I/O Pairs	GAAO/IOuxwByVz	The output of the different pair will drive the global.
	GAA1/IOuxwByVz	
	GABO/IOuxwByVz	The output of the different pair will drive the global.
	GAB1/IOuxwByVz	
	GACO/IOuxwByVz	The output of the different pair will drive the global.
	GAC1/IOuxwByVz	
	GBAO/IOuxwByVz	The output of the different pair will drive the global.
	GBA1/IOuxwByVz	
	GBBO/IOuxwByVz	The output of the different pair will drive the global.
	GBB1/IOuxwByVz	
	GBCO/IOuxwByVz	The output of the different pair will drive the global.
	GBC1/IOuxwByVz	
	GDAO/IOuxwByVz	The output of the different pair will drive the global.
	GDA1/IOuxwByVz	
	GDBO/IOuxwByVz	The output of the different pair will drive the global.
	GDB1/IOuxwByVz	
	GDCO/IOuxwByVz	The output of the different pair will drive the global.
	GDC1/IOuxwByVz	
	GEAO/IOuxwByVz	The output of the different pair will drive the global.
	GEA1/IOuxwByVz	
	GEBO/IOuxwByVz	The output of the different pair will drive the global.
	GEB1/IOuxwByVz	
	GECO/IOuxwByVz	The output of the different pair will drive the global.
	GEC1/IOuxwByVz	

Table 3-3 • Quadrant Global Pin Name (continued)

Note: Only one of the I/Os can be directly connected to a quadrant at a time.

Unused Global I/O Configuration

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

I/O Banks and Global I/O Standards

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



Global Resources in Low Power Flash Devices

External I/O or Local signal as Clock Source

External I/O refers to regular I/O pins are labeled with the I/O convention IOuxwByVz. You can allow the external I/O or internal signal to access the global. To allow the external I/O or internal signal to access the global network, you need to instantiate the CLKINT macro. Refer to Figure 3-4 on page 51 for an example illustration of the connections. Instead of using CLKINT, you can also use PDC to promote signals from external I/O or internal signal to the global network. However, it may cause layout issues because of synthesis logic replication. Refer to the "Global Promotion and Demotion Using PDC" section on page 67 for details.



Figure 3-14 • CLKINT Macro

Using Global Macros in Synplicity

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

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

You can control the maximum number of shared instances allowed for the legalization to take place using the Compile Option dialog box shown in Figure 3-17. Refer to Libero SoC / Designer online help for details on the Compile Option dialog box. A large number of shared instances most likely indicates a floorplanning problem that you should address.

Figure 3-17 • Shared Instances in the Compile Option Dialog Box

Designer Flow for Global Assignment

To achieve the desired result, pay special attention to global management during synthesis and placeand-route. The current Synplify tool does not insert more than six global buffers in the netlist by default. Thus, the default flow will not assign any signal to the quadrant global network. However, you can use attributes in Synplify and increase the default global macro assignment in the netlist. Designer v6.2 supports automatic quadrant global assignment, which was not available in Designer v6.1. Layout will make the choice to assign the correct signals to global. However, you can also utilize PDC and perform manual global assignment to overwrite any automatic assignment. The following step-by-step suggestions guide you in the layout of your design and help you improve timing in Designer:

- Run Compile and check the Compile report. The Compile report has global information in the "Device Utilization" section that describes the number of chip and quadrant signals in the design. A "Net Report" section describes chip global nets, quadrant global nets, local clock nets, a list of nets listed by fanout, and net candidates for local clock assignment. Review this information. Note that YB or YC are counted as global only when they are used in isolation; if you use YB only and not GLB, this net is not shown in the global/quadrant nets report. Instead, it appears in the Global Utilization report.
- 2. If some signals have a very high fanout and are candidates for global promotion, promote those signals to global using the compile options or PDC commands. Figure 3-18 on page 70 shows the Globals Management section of the compile options. Select **Promote regular nets whose fanout is greater than** and enter a reasonable value for fanouts.

Simple Design Example

Consider a design consisting of six building blocks (shift registers) and targeted for an A3PE600-PQ208 (Figure 3-16 on page 68). The example design consists of two PLLs (PLL1 has GLA only; PLL2 has both GLA and GLB), a global reset (ACLR), an enable (EN_ALL), and three external clock domains (QCLK1, QCLK2, and QCLK3) driving the different blocks of the design. Note that the PQ208 package only has two PLLs (which access the chip global network). Because of fanout, the global reset and enable signals need to be assigned to the chip global resources. There is only one free chip global for the remaining global (QCLK1, QCLK2, QCLK3). Place two of these signals on the quadrant global resource. The design example demonstrates manually assignment of QCLK1 and QCLK2 to the quadrant global using the PDC command.



Figure 3-19 • Block Diagram of the Global Management Example Design

4 – Clock Conditioning Circuits in Low Power Flash Devices and Mixed Signal FPGAs

Introduction

This document outlines the following device information: Clock Conditioning Circuit (CCC) features, PLL core specifications, functional descriptions, software configuration information, detailed usage information, recommended board-level considerations, and other considerations concerning clock conditioning circuits and global networks in low power flash devices or mixed signal FPGAs.

Overview of Clock Conditioning Circuitry

In Fusion, IGLOO, and ProASIC3 devices, the CCCs are used to implement frequency division, frequency multiplication, phase shifting, and delay operations. The CCCs are available in six chip locations—each of the four chip corners and the middle of the east and west chip sides. For device-specific variations, refer to the "Device-Specific Layout" section on page 94.

The CCC is composed of the following:

- PLL core
- 3 phase selectors
- 6 programmable delays and 1 fixed delay that advances/delays phase
- 5 programmable frequency dividers that provide frequency multiplication/division (not shown in Figure 4-6 on page 87 because they are automatically configured based on the user's required frequencies)
- · 1 dynamic shift register that provides CCC dynamic reconfiguration capability

Figure 4-1 provides a simplified block diagram of the physical implementation of the building blocks in each of the CCCs.



Figure 4-1 • Overview of the CCCs Offered in Fusion, IGLOO, and ProASIC3



Clock Conditioning Circuits in Low Power Flash Devices and Mixed Signal FPGAs

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.





Microsemi

Clock Conditioning Circuits in Low Power Flash Devices and Mixed Signal FPGAs

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 106. 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.

OADIVHALF / OBDIVHALF / OCDIVHALF	OADIV<4:0> / OBDIV<4:0> / OCDIV<4:0> (in decimal)	Divider Factor	Input Clock Frequency	Output Clock Frequency (MHz)	
1	2	1.5	100 MHz RC	66.7	
	4	2.5	Oscillator	40.0	
	6	3.5		28.6	
	8	4.5	- - - - -	22.2	
	10	5.5		18.2	
	12	6.5		15.4	
	14	7.5		13.3	
	16	8.5		11.8	
	18	9.5		10.5	
	20	10.5		9.5	
	22	11.5		8.7	
	24	12.5		8.0	
	26	13.5	1	13.5 7.4	
	28	14.5]	6.9	
0	0–31	1–32	Other Clock Sources	Depends on other divider settings	

Table 4-18 • Fusion Dynamic CCC Division by Half Configuration

Table 4-19 • Configuration Bit <76:75> / VCOSEL<2:1> Selection for All Families

	VCOSEL[2:1]							
00		01		10		11		
Voltage	Min. (MHz)	Max. (MHz)	Min. (MHz)	Max. (MHz)	Min. (MHz)	Max. (MHz)	Min. (MHz)	Max. (MHz)
IGLOO and IGLOO	IGLOO and IGLOO PLUS							
1.2 V ± 5%	24	35	30	70	60	140	135	160
1.5 V ± 5%	24	43.75	30	87.5	60	175	135	250
ProASIC3L, RT ProASIC3, and Military ProASIC3/L								
1.2 V ± 5%	24	35	30	70	60	140	135	250
1.5 V ± 5%	24	43.75	30	70	60	175	135	350
ProASIC3 and Fusion								
1.5 V ± 5%	24	43.75	33.75	87.5	67.5	175	135	350

Table 4-20 • Configuration Bit <74> / VCOSEL<0> Selection for All Families

VCOSEL[0]	Description
0	Fast PLL lock acquisition time with high tracking jitter. Refer to the corresponding datasheet for specific value and definition.
1	Slow PLL lock acquisition time with low tracking jitter. Refer to the corresponding datasheet for specific value and definition.

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

Microsemi

Clock Conditioning Circuits in Low Power Flash Devices and Mixed Signal FPGAs

global assignments are not allocated properly. See the "Physical Constraints for Quadrant Clocks" section for information on assigning global signals to the quadrant clock networks.

Promoted global signals will be instantiated with CLKINT macros to drive these signals onto the global network. This is automatically done by Designer when the Auto-Promotion option is selected. If the user wishes to assign the signals to the quadrant globals instead of the default chip globals, this can done by using ChipPlanner, by declaring a physical design constraint (PDC), or by importing a PDC file.

Physical Constraints for Quadrant Clocks

If it is necessary to promote global clocks (CLKBUF, CLKINT, PLL, CLKDLY) to quadrant clocks, the user can define PDCs to execute the promotion. PDCs can be created using PDC commands (pre-compile) or the MultiView Navigator (MVN) interface (post-compile). The advantage of using the PDC flow over the MVN flow is that the Compile stage is able to automatically promote any regular net to a global net before assigning it to a quadrant. There are three options to place a quadrant clock using PDC commands:

- Place a clock core (not hardwired to an I/O) into a quadrant clock location.
- Place a clock core (hardwired to an I/O) into an I/O location (set_io) or an I/O module location (set_location) that drives a quadrant clock location.
- Assign a net driven by a regular net or a clock net to a quadrant clock using the following command:

assign_local_clock -net <net name> -type quadrant <quadrant clock region>
where

<net name> is the name of the net assigned to the local user clock region.

<quadrant clock region> defines which quadrant the net should be assigned to. Quadrant clock regions are defined as UL (upper left), UR (upper right), LL (lower left), and LR (lower right).

Note: If the net is a regular net, the software inserts a CLKINT buffer on the net.

For example:

assign_local_clock -net localReset -type quadrant UR

Keep in mind the following when placing quadrant clocks using MultiView Navigator:

Hardwired I/O–Driven CCCs

• Find the associated clock input port under the Ports tab, and place the input port at one of the Gmn* locations using PinEditor or I/O Attribute Editor, as shown in Figure 4-32.

Figure 4-32 • Port Assignment for a CCC with Hardwired I/O Clock Input

Temporary overshoots are allowed according to the overshoot and undershoot table in the datasheet.



Figure 7-9 • Solution 1

Solution 2

The board-level design must ensure that the reflected waveform at the pad does not exceed the voltage overshoot/undershoot limits provided in the datasheet. This is a requirement to ensure long-term reliability.

This scheme will also work for a 3.3 V PCI/PCI-X configuration, but the internal diode should not be used for clamping, and the voltage must be limited by the external resistors and Zener, as shown in Figure 7-10. Relying on the diode clamping would create an excessive pad DC voltage of 3.3 V + 0.7 V = 4 V.







I/O Structures in IGLOOe and ProASIC3E Devices

Solution 3

The board-level design must ensure that the reflected waveform at the pad does not exceed the voltage overshoot/undershoot limits provided in the datasheet. This is a requirement to ensure long-term reliability.

This scheme will also work for a 3.3 V PCI/PCI-X configuration, but the internal diode should not be used for clamping, and the voltage must be limited by the bus switch, as shown in Figure 8-12. Relying on the diode clamping would create an excessive pad DC voltage of 3.3 V + 0.7 V = 4 V.



Figure 8-12 • Solution 3

Related Documents

Below is a list of related documents, their location on the Microsemi SoC Products Group website, and a brief summary of each document.

Application Notes

Programming Antifuse Devices http://www.microsemi.com/soc/documents/AntifuseProgram_AN.pdf Implementation of Security in Actel's ProASIC and ProASIC^{PLUS} Flash-Based FPGAs http://www.microsemi.com/soc/documents/Flash_Security_AN.pdf

User's Guides

FlashPro Programmers

FlashPro4,¹ FlashPro3, FlashPro Lite, and FlashPro² http://www.microsemi.com/soc/products/hardware/program_debug/flashpro/default.aspx *FlashPro User's Guide* http://www.microsemi.com/soc/documents/FlashPro_UG.pdf The FlashPro User's Guide includes hardware and software setup, self-test instructions, use instructions, and a troubleshooting / error message guide.

Silicon Sculptor 3 and Silicon Sculptor II

http://www.microsemi.com/soc/products/hardware/program_debug/ss/default.aspx

Other Documents

http://www.microsemi.com/soc/products/solutions/security/default.aspx#flashlock The security resource center describes security in Microsemi Flash FPGAs. *Quality and Reliability Guide* http://www.microsemi.com/soc/documents/RelGuide.pdf *Programming and Functional Failure Guidelines* http://www.microsemi.com/soc/documents/FA_Policies_Guidelines_5-06-00002.pdf

^{1.} FlashPro4 replaced FlashPro3 in Q1 2010.

^{2.} FlashPro is no longer available.



Security in Low Power Flash Devices



Figure 12-3 • Block Representation of the AES Decryption Core in a Fusion AFS600 FPGA

Security Features

IGLOO and ProASIC3 devices have two entities inside: FlashROM and the FPGA core fabric. Fusion devices contain three entities: FlashROM, FBs, and the FPGA core fabric. The parts can be programmed or updated independently with a STAPL programming file. The programming files can be AES-encrypted or plaintext. This allows maximum flexibility in providing security to the entire device. Refer to the "Programming Flash Devices" section on page 287 for information on the FlashROM structure.

Unlike SRAM-based FPGA devices, which require a separate boot PROM to store programming data, low power flash devices are nonvolatile, and the secured configuration data is stored in on-chip flash cells that are part of the FPGA fabric. Once programmed, this data is an inherent part of the FPGA array and does not need to be loaded at system power-up. SRAM-based FPGAs load the configuration bitstream upon power-up; therefore, the configuration is exposed and can be read easily.

The built-in FPGA core, FBs, and FlashROM support programming files encrypted with the 128-bit AES (FIPS-192) block ciphers. The AES key is stored in dedicated, on-chip flash memory and can be programmed before the device is shipped to other parties (allowing secure remote field updates).

Security in ARM-Enabled Low Power Flash Devices

There are slight differences between the regular flash devices and the $\text{ARM}^{\$}$ -enabled flash devices, which have the M1 and M7 prefix.

The AES key is used by Microsemi and preprogrammed into the device to protect the ARM IP. As a result, the design is encrypted along with the ARM IP, according to the details below.

16 – Boundary Scan in Low Power Flash Devices

Boundary Scan

Low power flash devices are compatible with IEEE Standard 1149.1, which defines a hardware architecture and the set of mechanisms for boundary scan testing. JTAG operations are used during boundary scan testing.

The basic boundary scan logic circuit is composed of the TAP controller, test data registers, and instruction register (Figure 16-2 on page 360).

Low power flash devices support three types of test data registers: bypass, device identification, and boundary scan. The bypass register is selected when no other register needs to be accessed in a device. This speeds up test data transfer to other devices in a test data path. The 32-bit device identification register is a shift register with four fields (LSB, ID number, part number, and version). The boundary scan register observes and controls the state of each I/O pin. Each I/O cell has three boundary scan register cells, each with serial-in, serial-out, parallel-in, and parallel-out pins.

TAP Controller State Machine

The TAP controller is a 4-bit state machine (16 states) that operates as shown in Figure 16-1.

The 1s and 0s represent the values that must be present on TMS at a rising edge of TCK for the given state transition to occur. IR and DR indicate that the instruction register or the data register is operating in that state.

The TAP controller receives two control inputs (TMS and TCK) and generates control and clock signals for the rest of the test logic architecture. On power-up, the TAP controller enters the Test-Logic-Reset state. To guarantee a reset of the controller from any of the possible states, TMS must remain HIGH for five TCK cycles. The TRST pin can also be used to asynchronously place the TAP controller in the Test-Logic-Reset state.



Figure 16-1 • TAP Controller State Machine

I/O Behavior at Power-Up/-Down

This section discusses the behavior of device I/Os, used and unused, during power-up/-down of V_{CC} and V_{CCI}. As mentioned earlier, VMVx and V_{CCI}Bx are tied together, and therefore, inputs and outputs are powered up/down at the same time.

I/O State during Power-Up/-Down

This section discusses the characteristics of I/O behavior during device power-up and power-down. Before the start of power-up, all I/Os are in tristate mode. The I/Os will remain tristated during power-up until the last voltage supply (VCC or VCCI) is powered to its functional level (power supply functional levels are discussed in the "Power-Up to Functional Time" section on page 378). After the last supply reaches the functional level, the outputs will exit the tristate mode and drive the logic at the input of the output buffer. Similarly, the input buffers will pass the external logic into the FPGA fabric once the last supply reaches the functional level. The behavior of user I/Os is independent of the VCC and VCCI sequence or the state of other voltage supplies of the FPGA (VPUMP and VJTAG). Figure 18-2 shows the output buffer driving HIGH and its behavior during power-up with 10 k Ω external pull-down. In Figure 18-2, VCC is powered first, and VCCI is powered 5 ms after VCC. Figure 18-3 on page 378 shows the state of the I/O when VCCI is powered about 5 ms before VCC. In the circuitry shown in Figure 18-3 on page 378, the output is externally pulled down.

During power-down, device I/Os become tristated once the first power supply (VCC or VCCI) drops below its brownout voltage level. The I/O behavior during power-down is also independent of voltage supply sequencing.

Figure 18-2 • I/O State when VCC Is Powered before VCCI