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	Active
Number of LABs/CLBs	-
Number of Logic Elements/Cells	-
Total RAM Bits	516096
Number of I/O	620
Number of Gates	3000000
Voltage - Supply	1.14V ~ 1.575V
Mounting Type	Surface Mount
Operating Temperature	-40°C ~ 100°C (TJ)
Package / Case	896-BGA
Supplier Device Package	896-FBGA (31x31)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/a3pe3000l-fg896i

Flash*Freeze Mode Device Behavior

*Entering Flash*Freeze Mode*

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

PLLs

- If an embedded PLL is used, entering Flash*Freeze mode will automatically power down the PLL.
- The PLL output clocks will stop toggling within 1 µs after the assertion of the FF pin in type 1, or after both FF pin and LSICC signal are asserted in type 2. At the same time, I/Os will transition into the state specified in Table 2-6 on page 29. The user design must ensure it is safe to enter Flash*Freeze mode.

I/Os and Globals

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

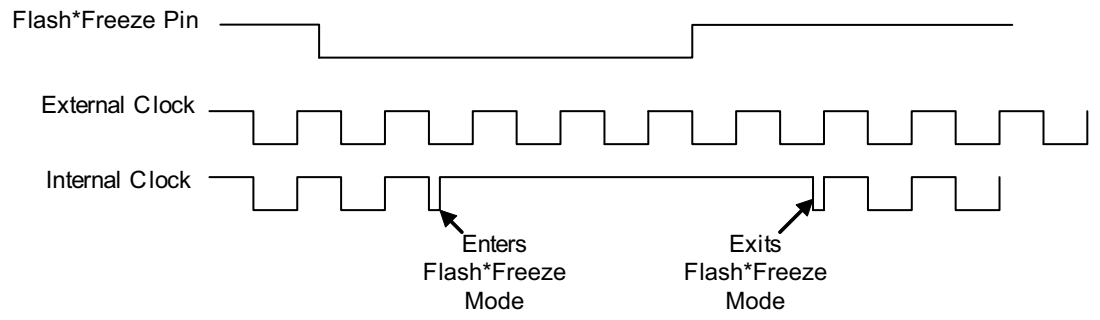


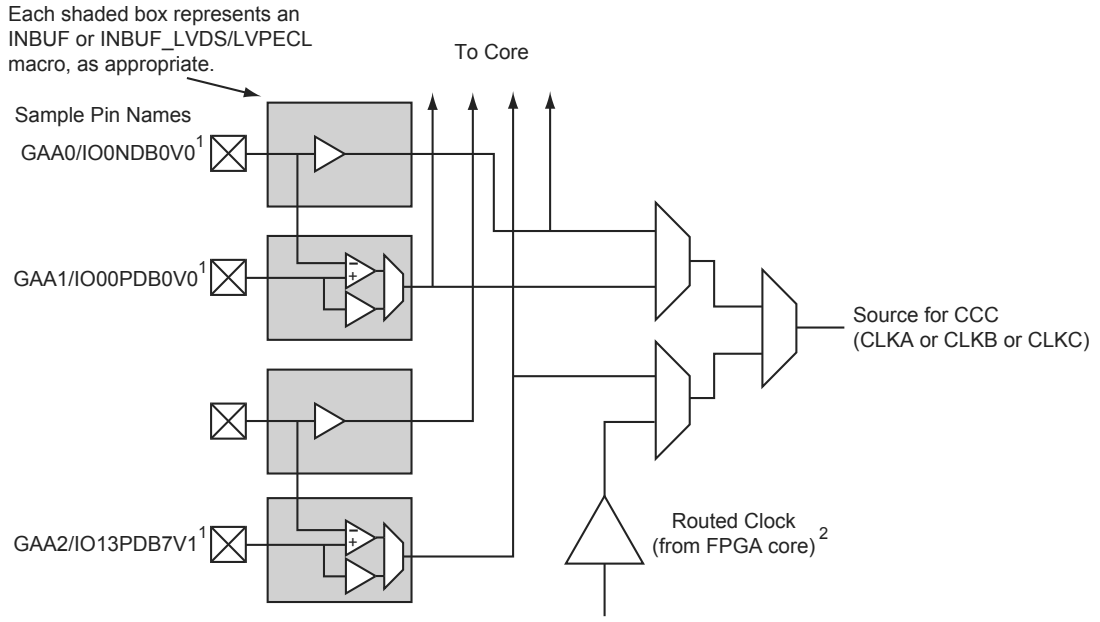
***Figure 2-5* • Narrow Clock Pulses During Flash*Freeze Entrance and Exit**

- I/O banks are not all deactivated simultaneously when entering Flash*Freeze mode. This can cause clocks and inputs to become disabled at different times, resulting in unexpected data being captured.
- Upon entering Flash*Freeze mode, all inputs and globals become tied High internally (except when an input hold state is used on IGLOO nano or IGLOO PLUS devices). If any of these signals are driven Low or tied Low externally, they will experience a Low to High transition internally when entering Flash*Freeze mode.
- Upon entering type 2 Flash*Freeze mode, ensure the LSICC signal (active High) does not de-assert. This can prevent the device from entering Flash*Freeze mode.
- Asynchronous input to output paths may experience output glitches. For example, on a direct in-to-out path, if the current state is '0' and the input bank turns off first, the input and then the output will transition to '1' before the output enters its Flash*Freeze state. This can be prevented by using latches in asynchronous in-to-out paths.
- The above situations can cause glitches or invalid data to be clocked into and preserved in the device. Refer to the "Flash*Freeze Design Guide" section on page 34 for solutions.

Date	Changes	Page
v1.2 (continued)	Figure 2-3 • Flash*Freeze Mode Type 2 – Controlled by Flash*Freeze Pin and Internal Logic (LSICC signal) was updated.	27
	Figure 2-4 • Flash*Freeze Mode Type 2 – Timing Diagram was revised to show deasserting LSICC after the device has exited Flash*Freeze mode.	27
	The "IGLOO nano and IGLOO PLUS I/O State in Flash*Freeze Mode" section was added to include information for IGLOO PLUS devices. Table 2-6 • IGLOO nano and IGLOO PLUS Flash*Freeze Mode (type 1 and type 2)—I/O Pad State is new.	28, 29
	The "During Flash*Freeze Mode" section was revised to include a new bullet pertaining to output behavior for IGLOO PLUS. The bullet on JTAG operation was revised to provide more detail.	31
	Figure 2-6 • Controlling Power-On/-Off State Using Microprocessor and Power FET and Figure 2-7 • Controlling Power-On/-Off State Using Microprocessor and Voltage Regulator were updated to include IGLOO PLUS.	33, 33
	The first sentence of the "Shutdown Mode" section was updated to list the devices for which it is supported.	32
	The first paragraph of the "Power-Up/-Down Behavior" section was revised. The second sentence was changed to, "The I/Os remain tristated until the last voltage supply (V_{CC} or V_{CCI}) is powered to its activation level." The word "activation" replaced the word "functional." The sentence, "During power-down, device I/Os become tristated once the first power supply (V_{CC} or V_{CCI}) drops below its deactivation voltage level" was revised. The word "deactivation" replaced the word "brownout."	33
	The "Prototyping for IGLOO and ProASIC3L Devices Using ProASIC3" section was revised to state that prototyping in ProASIC3 does not apply for the IGLOO PLUS family.	2-21
	Table 2-8 • Prototyping/Migration Solutions, Table 2-9 • Device Migration—IGLOO Supported Packages in ProASIC3 Devices, and Table 2-10 • Device Migration—ProASIC3L Supported Packages in ProASIC3 Devices were updated with a table note stating that device migration is not supported for IGLOO PLUS devices.	2-21, 2-23
	The text following Table 2-10 • Device Migration—ProASIC3L Supported Packages in ProASIC3 Devices was moved to a new section: the "Flash*Freeze Design Guide" section.	34
v1.1 (February 2008)	Table 2-1 • Flash-Based FPGAs was updated to remove the ProASIC3, ProASIC3E, and Automotive ProASIC3 families, which were incorrectly included.	22
v1.0 (January 2008)	Detailed descriptions of low power modes are described in the advanced datasheets. This application note was updated to describe how to use the features in an IGLOO/e application.	N/A
	Figure 2-1 • Flash*Freeze Mode Type 1 – Controlled by the Flash*Freeze Pin was updated.	25
	Figure 2-2 • Flash*Freeze Mode Type 1 – Timing Diagram is new.	25
	Steps 4 and 5 are new in the "Flash*Freeze Type 2: Control by Dedicated Flash*Freeze Pin and Internal Logic" section.	26

Figure 3-5 shows more detailed global input connections. It shows the global input pins connection to the northwest quadrant global networks. 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



Note: Differential inputs are not supported for IGLOO nano or ProASIC3 nano devices.

Figure 3-5 • **Global I/O Overview**

Global Macro and Placement Selections

Low power flash devices provide the flexibility of choosing one of the three global input pad locations available to connect to a global / quadrant global network. For 60K gate devices and above, if the single-ended I/O standard is chosen, there is flexibility to choose one of the global input pads (the first, second, and fourth input). Once chosen, the other I/O locations are used as regular I/Os. If the differential I/O standard is chosen, the first and second inputs are considered as paired, and the third input is paired with a regular I/O. The user then has the choice of selecting one of the two sets to be used as the global input source. There is also the option to allow an internal clock signal to feed the global network. A multiplexer tree selects the appropriate global input for routing to the desired location. Note that the global I/O pads do not need to feed the global network; they can also be used as regular I/O pads.

Hardwired I/O Clock Source

Hardwired I/O refers to global input pins that are hardwired to the multiplexer tree, which directly accesses the global network. These global input pins have designated pin locations and are indicated with the I/O naming convention Gmn (m refers to any one of the positions where the global buffers is available, and n refers to any one of the three global input MUXes and the pin number of the associated global location, m). Choosing this option provides the benefit of directly connecting to the global buffers, which provides less delay. See Figure 3-11 for an example illustration of the connections, shown in red. If a CLKBUF macro is initiated, the clock input can be placed at one of nine dedicated global input pin locations: GmA0, GmA1, GmA2, GmB0, GmB1, GmB2, GmC0, GmC1, or GmC2. Note that the placement of the global will determine whether you are using chip global or quadrant global. For example, if the CLKBIF is placed in one of the GF pin locations, it will use the chip global network; if the CLKBIF is placed in one of the GA pin locations, it will use quadrant global network. This is shown in Figure 3-12 on page 65 and Figure 3-13 on page 65.

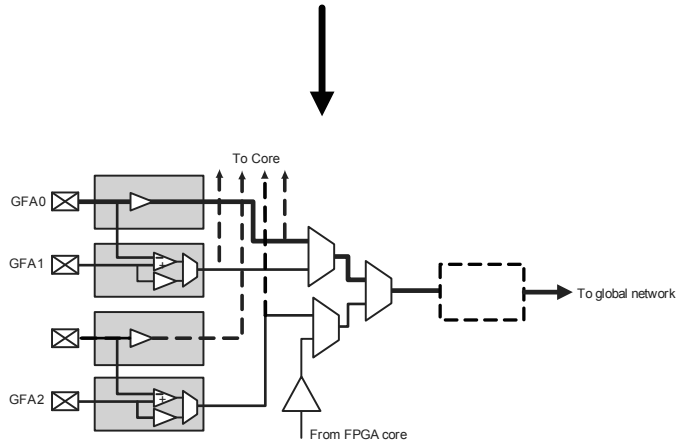
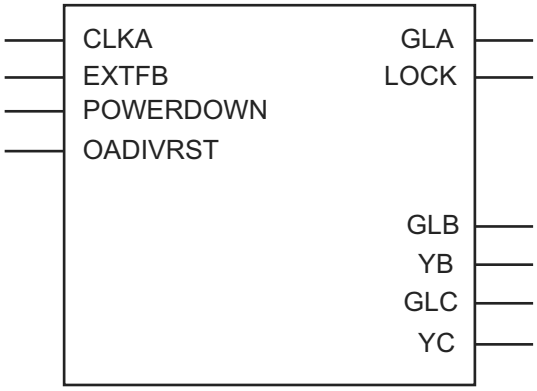


***Figure 3-11* • CLKBUF Macro**



Note: OAVDIVRST exists only in the Fusion PLL.

***Figure 3-15* • PLLs in Low Power Flash Devices**

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

Global Promotion and Demotion Using PDC

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

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

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

   ```
   assign_global_clock –net netname
   ```

   The following will happen during promotion of a regular signal to a global network:

   – If the net is external, the net will be driven by a CLKINT inserted automatically by Compile.
   – The I/O macro will not be changed to CLKBUF macros.
   – If the net is an internal net, the net will be driven by a CLKINT inserted automatically by Compile.

2. PDC syntax to promote a net to a quadrant clock:

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

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

The following PDC command demotes the clock nets to regular nets.

```
unassign_global_clock -net netname
```

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.

```
  wire VCC, GND;

  VCC VCC_1_net(.Y(VCC));
  GND GND_1_net(.Y(GND));
  CLKDLY Inst1(.CLK(CLK), .GL(GL), .DLYGL0(VCC), .DLYGL1(GND), .DLYGL2(VCC),
    .DLYGL3(GND), .DLYGL4(GND));

endmodule
```

Detailed Usage Information

Clock Frequency Synthesis

Deriving clocks of various frequencies from a single reference clock is known as frequency synthesis. The PLL has an input frequency range from 1.5 to 350 MHz. This frequency is automatically divided down to a range between 1.5 MHz and 5.5 MHz by input dividers (not shown in Figure 4-19 on page 100) between PLL macro inputs and PLL phase detector inputs. The VCO output is capable of an output range from 24 to 350 MHz. With dividers before the input to the PLL core and following the VCO outputs, the VCO output frequency can be divided to provide the final frequency range from 0.75 to 350 MHz. Using SmartGen, the dividers are automatically set to achieve the closest possible matches to the specified output frequencies.

Users should be cautious when selecting the desired PLL input and output frequencies and the I/O buffer standard used to connect to the PLL input and output clocks. Depending on the I/O standards used for the PLL input and output clocks, the I/O frequencies have different maximum limits. Refer to the family datasheets for specifications of maximum I/O frequencies for supported I/O standards. Desired PLL input or output frequencies will not be achieved if the selected frequencies are higher than the maximum I/O frequencies allowed by the selected I/O standards. Users should be careful when selecting the I/O standards used for PLL input and output clocks. Performing post-layout simulation can help detect this type of error, which will be identified with pulse width violation errors. Users are strongly encouraged to perform post-layout simulation to ensure the I/O standard used can provide the desired PLL input or output frequencies. Users can also choose to cascade PLLs together to achieve the high frequencies needed for their applications. Details of cascading PLLs are discussed in the "Cascading CCCs" section on page 125.

In SmartGen, the actual generated frequency (under typical operating conditions) will be displayed beside the requested output frequency value. This provides the ability to determine the exact frequency that can be generated by SmartGen, in real time. The log file generated by SmartGen is a useful tool in determining how closely the requested clock frequencies match the user specifications. For example, assume a user specifies 101 MHz as one of the secondary output frequencies. If the best output frequency that could be achieved were 100 MHz, the log file generated by SmartGen would indicate the actual generated frequency.

Simulation Verification

The integration of the generated PLL and CLKDLY modules is similar to any VHDL component or Verilog module instantiation in a larger design; i.e., there is no special requirement that users need to take into account to successfully synthesize their designs.

For simulation purposes, users need to refer to the VITAL or Verilog library that includes the functional description and associated timing parameters. Refer to the Software Tools section of the Microsemi SoC Products Group website to obtain the family simulation libraries. If Designer is installed, these libraries are stored in the following locations:

<Designer_Installation_Directory>\lib\vtl\95\proasic3.vhd

<Designer_Installation_Directory>\lib\vtl\95\proasic3e.vhd

<Designer_Installation_Directory>\lib\vlog\proasic3.v

<Designer_Installation_Directory>\lib\vlog\proasic3e.v

For Libero users, there is no need to compile the simulation libraries, as they are conveniently pre-compiled in the Model*Sim*® Microsemi simulation tool.

Recommended Board-Level Considerations

The power to the PLL core is supplied by VCCPLA/B/C/D/E/F (VCCPLx), and the associated ground connections are supplied by VCOMPLA/B/C/D/E/F (VCOMPLx). When the PLLs are not used, the Designer place-and-route tool automatically disables the unused PLLs to lower power consumption. The user should tie unused VCCPLx and VCOMPLx pins to ground. Optionally, the PLL can be turned on/off during normal device operation via the POWERDOWN port (see Table 4-3 on page 84).

PLL Power Supply Decoupling Scheme

The PLL core is designed to tolerate noise levels on the PLL power supply as specified in the datasheets. When operated within the noise limits, the PLL will meet the output peak-to-peak jitter specifications specified in the datasheets. User applications should always ensure the PLL power supply is powered from a noise-free or low-noise power source.

However, in situations where the PLL power supply noise level is higher than the tolerable limits, various decoupling schemes can be designed to suppress noise to the PLL power supply. An example is provided in Figure 4-38. The VCCPLx and VCOMPLx pins correspond to the PLL analog power supply and ground.

Microsemi strongly recommends that two ceramic capacitors (10 nF in parallel with 100 nF) be placed close to the power pins (less than 1 inch away). A third generic 10 µF electrolytic capacitor is recommended for low-frequency noise and should be placed farther away due to its large physical size. Microsemi recommends that a 6.8 µH inductor be placed between the supply source and the capacitors to filter out any low-/medium- and high-frequency noise. In addition, the PCB layers should be controlled so the VCCPLx and VCOMPLx planes have the minimum separation possible, thus generating a good-quality RF capacitor.

For more recommendations, refer to the *Board-Level Considerations* application note.

Recommended 100 nF capacitor:

- Producer BC Components, type X7R, 100 nF, 16 V
- BC Components part number: 0603B104K160BT
- Digi-Key part number: BC1254CT-ND
- Digi-Key part number: BC1254TR-ND

Recommended 10 nF capacitor:

- Surface-mount ceramic capacitor
- Producer BC Components, type X7R, 10 nF, 50 V
- BC Components part number: 0603B103K500BT
- Digi-Key part number: BC1252CT-ND
- Digi-Key part number: BC1252TR-ND

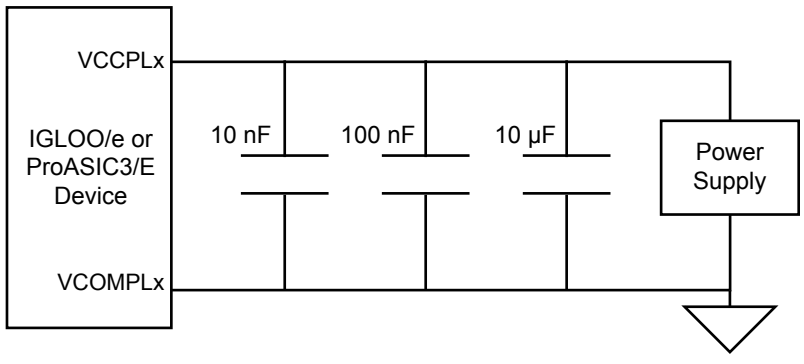


***Figure 4-38* • Decoupling Scheme for One PLL (should be replicated for each PLL used)**

Conclusion

The Fusion, IGLOO, and ProASIC3 families are the only FPGAs that offer on-chip FlashROM support. This document presents information on the FlashROM architecture, possible applications, programming, access through the JTAG and UJTAG interface, and integration into your design. In addition, the Libero tool set enables easy creation and modification of the FlashROM content.

The nonvolatile FlashROM block in the FPGA can be customized, enabling multiple applications.

Additionally, the security offered by the low power flash devices keeps both the contents of FlashROM and the FPGA design safe from system over-builders, system cloners, and IP thieves.

Related Documents

User's Guides

FlashPro User's Guide

http://www.microsemi.com/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
v1.4 (December 2008)	IGLOO nano and ProASIC3 nano devices were added to Table 5-1 • Flash-Based FPGAs.	134
v1.3 (October 2008)	The "FlashROM Support in Flash-Based Devices" section was revised to include new families and make the information more concise.	134
	Figure 5-2 • Fusion Device Architecture Overview (AFS600) was replaced. Figure 5-5 • Programming FlashROM Using AES was revised to change "Fusion" to "Flash Device."	135, 137
	The *FlashPoint User's Guide* was removed from the "User's Guides" section, as its content is now part of the *FlashPro User's Guide*.	146
v1.2 (June 2008)	The following changes were made to the family descriptions in Table 5-1 • Flash-Based FPGAs: • ProASIC3L was updated to include 1.5 V. • The number of PLLs for ProASIC3E was changed from five to six.	134
v1.1 (March 2008)	The chapter was updated to include the IGLOO PLUS family and information regarding 15 k gate devices. The "IGLOO Terminology" section and "ProASIC3 Terminology" section are new.	N/A

Low Power Flash Device I/O Support

The low power flash FPGAs listed in Table 7-1 support I/Os and the functions described in this document.

***Table 7-1* • Flash-Based FPGAs**

Series	Family*	Description
IGLOO	IGLOO	Ultra-low power 1.2 V to 1.5 V FPGAs with Flash*Freeze technology
ProASIC3	ProASIC3	Low power, high-performance 1.5 V FPGAs
	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 7-1. 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 7-1. 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*.

***Table 7-8* • Hot-Swap Level 1**

Description	Cold-swap
Power Applied to Device	No
Bus State	–
Card Ground Connection	–
Device Circuitry Connected to Bus Pins	–
Example Application	System and card with Microsemi FPGA chip are powered down, and the card is plugged into the system. Then the power supplies are turned on for the system but not for the FPGA on the card.
Compliance of IGLOO and ProASIC3 Devices	30 k gate devices: Compliant Other IGLOO/ProASIC3 devices: Compliant if bus switch used to isolate FPGA I/Os from rest of system IGLOOe/ProASIC3E devices: Compliant I/Os can but do not have to be set to hot-insertion mode.

***Table 7-9* • Hot-Swap Level 2**

Description	Hot-swap while reset
Power Applied to Device	Yes
Bus State	Held in reset state
Card Ground Connection	Reset must be maintained for 1 ms before, during, and after insertion/removal.
Device Circuitry Connected to Bus Pins	–
Example Application	In the PCI hot-plug specification, reset control circuitry isolates the card busses until the card supplies are at their nominal operating levels and stable.
Compliance of IGLOO and ProASIC3 Devices	30 k gate devices, all IGLOOe/ProASIC3E devices: Compliant I/Os can but do not have to be set to hot-insertion mode. Other IGLOO/ProASIC3 devices: Compliant

***Table 7-12* • I/O Hot-Swap and 5 V Input Tolerance Capabilities in IGLOO and ProASIC3 Devices**

I/O Assignment	Clamp Diode [1]		Hot Insertion		5 V Input Tolerance [2]		Input and Output Buffer
	AGL030 and A3P030	Other IGLOO and ProASIC3 Devices	AGL015 and AGL030	Other IGLOO Devices and All ProASIC3	AGL030 and A3P030	Other IGLOO and ProASIC3 Devices	
3.3 V LVTTL/LVCMOS	No	Yes	Yes	No	Yes [2]	Yes [2]	Enabled/Disabled
3.3 V PCI, 3.3 V PCI-X	N/A	Yes	N/A	No	N/A	Yes [2]	Enabled/Disabled
LVCMOS 2.5 V [5]	No	Yes	Yes	No	Yes [2]	Yes [4]	Enabled/Disabled
LVCMOS 2.5 V/5.0 V [6]	N/A	Yes	N/A	No	N/A	Yes [4]	Enabled/Disabled
LVCMOS 1.8 V	No	Yes	Yes	No	No	No	Enabled/Disabled
LVCMOS 1.5 V	No	Yes	Yes	No	No	No	Enabled/Disabled
Differential, LVDS/ B-LVDS/M-LVDS/LVPECL	N/A	Yes	N/A	No	N/A	No	Enabled/Disabled

Notes:

1. *The clamp diode is always off for the AGL030 and A3P030 device and always active for other IGLOO and ProASIC3 devices.*
2. *Can be implemented with an external IDT bus switch, resistor divider, or Zener with resistor.*
3. *Refer to Table 7-8 on page 189 to Table 7-11 on page 190 for device-compliant information.*
4. *Can be implemented with an external resistor and an internal clamp diode.*
5. *The LVCMOS 2.5 V I/O standard is supported by the 30 k gate devices only; select the LVCMOS25 macro.*
6. *The LVCMOS 2.5 V / 5.0 V I/O standard is supported by all IGLOO and ProASIC3 devices except 30K gate devices; select the LVCMOS5 macro.*

5 V Input and Output Tolerance

IGLOO and ProASIC3 devices are both 5 V-input– and 5 V–output–tolerant if certain I/O standards are selected. Table 7-5 on page 179 shows the I/O standards that support 5 V input tolerance. Only 3.3 V LVTTL/LVCMOS standards support 5 V output tolerance. Refer to the appropriate family datasheet for the detailed description and configuration information.

This feature is not shown in the I/O Attribute Editor.

5 V Input Tolerance

I/Os can support 5 V input tolerance when LVTTL 3.3 V, LVCMOS 3.3 V, LVCMOS 2.5 V, and LVCMOS 2.5 V / 5.0 V configurations are used (see Table 7-12 on page 193). There are four recommended solutions for achieving 5 V receiver tolerance (see Figure 7-9 on page 195 to Figure 7-12 on page 197 for details of board and macro setups). All the solutions meet a common requirement of limiting the voltage at the input to 3.6 V or less. In fact, the I/O absolute maximum voltage rating is 3.6 V, and any voltage above 3.6 V may cause long-term gate oxide failures.

Solution 1

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

This 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 two external resistors as explained below. Relying on the diode clamping would create an excessive pad DC voltage of 3.3 V + 0.7 V = 4 V.

This solution requires two board resistors, as demonstrated in Figure 7-9 on page 195. Here are some examples of possible resistor values (based on a simplified simulation model with no line effects and 10 Ω transmitter output resistance, where Rtx_out_high = (VCCI – VOH) / I_{OH} and Rtx_out_low = VOL / I_{OL}).

Example 1 (high speed, high current):

Rtx_out_high = Rtx_out_low = 10 Ω

R1 = 36 Ω (±5%), P(r1)min = 0.069 Ω

R2 = 82 Ω (±5%), P(r2)min = 0.158 Ω

Imax_tx = 5.5 V / (82 × 0.95 + 36 × 0.95 + 10) = 45.04 mA

t_{RISE} = t_{FALL} = 0.85 ns at C_pad_load = 10 pF (includes up to 25% safety margin)

t_{RISE} = t_{FALL} = 4 ns at C_pad_load = 50 pF (includes up to 25% safety margin)

Example 2 (low–medium speed, medium current):

Rtx_out_high = Rtx_out_low = 10 Ω

R1 = 220 Ω (±5%), P(r1)min = 0.018 Ω

R2 = 390 Ω (±5%), P(r2)min = 0.032 Ω

Imax_tx = 5.5 V / (220 × 0.95 + 390 × 0.95 + 10) = 9.17 mA

t_{RISE} = t_{FALL} = 4 ns at C_pad_load = 10 pF (includes up to 25% safety margin)

t_{RISE} = t_{FALL} = 20 ns at C_pad_load = 50 pF (includes up to 25% safety margin)

Other values of resistors are also allowed as long as the resistors are sized appropriately to limit the voltage at the receiving end to 2.5 V < Vin (rx) < 3.6 V when the transmitter sends a logic 1. This range of Vin_dc(rx) must be assured for any combination of transmitter supply (5 V ± 0.5 V), transmitter output resistance, and board resistor tolerances.

IGLOOe and ProASIC3E

For devices requiring Level 3 and/or Level 4 compliance, the board drivers connected to the I/Os must have 10 kΩ (or lower) output drive resistance at hot insertion, and 1 kΩ (or lower) output drive resistance at hot removal. This resistance is the transmitter resistance sending a signal toward the I/O, and no additional resistance is needed on the board. If that cannot be assured, three levels of staging can be used to achieve Level 3 and/or Level 4 compliance. Cards with two levels of staging should have the following sequence:

- Grounds
- Powers, I/Os, and other pins

Cold-Sparing Support

Cold-sparing refers to the ability of a device to leave system data undisturbed when the system is powered up, while the component itself is powered down, or when power supplies are floating.

Cold-sparing is supported on ProASIC3E devices only when the user provides resistors from each power supply to ground. The resistor value is calculated based on the decoupling capacitance on a given power supply. The RC constant should be greater than 3 µs.

To remove resistor current during operation, it is suggested that the resistor be disconnected (e.g., with an NMOS switch) from the power supply after the supply has reached its final value. Refer to the "Power-Up/-Down Behavior of Low Power Flash Devices" section on page 373 for details on cold-sparing.

Cold-sparing means that a subsystem with no power applied (usually a circuit board) is electrically connected to the system that is in operation. This means that all input buffers of the subsystem must present very high input impedance with no power applied so as not to disturb the operating portion of the system.

The 30 k gate devices fully support cold-sparing, since the I/O clamp diode is always off (see Table 8-13 on page 231). If the 30 k gate device is used in applications requiring cold-sparing, a discharge path from the power supply to ground should be provided. This can be done with a discharge resistor or a switched resistor. This is necessary because the 30 k gate devices do not have built-in I/O clamp diodes.

For other IGLOOe and ProASIC3E devices, since the I/O clamp diode is always active, cold-sparing can be accomplished either by employing a bus switch to isolate the device I/Os from the rest of the system or by driving each I/O pin to 0 V. If the resistor is chosen, the resistor value must be calculated based on decoupling capacitance on a given power supply on the board (this decoupling capacitance is in parallel with the resistor). The RC time constant should ensure full discharge of supplies before cold-sparing functionality is required. The resistor is necessary to ensure that the power pins are discharged to ground every time there is an interruption of power to the device.

IGLOOe and ProASIC3E devices support cold-sparing for all I/O configurations. Standards, such as PCI, that require I/O clamp diodes can also achieve cold-sparing compliance, since clamp diodes get disconnected internally when the supplies are at 0 V.

When targeting low power applications, I/O cold-sparing may add additional current if a pin is configured with either a pull-up or pull-down resistor and driven in the opposite direction. A small static current is induced on each I/O pin when the pin is driven to a voltage opposite to the weak pull resistor. The current is equal to the voltage drop across the input pin divided by the pull resistor. Refer to the "Detailed I/O DC Characteristics" section of the appropriate family datasheet for the specific pull resistor value for the corresponding I/O standard.

For example, assuming an LVTTL 3.3 V input pin is configured with a weak pull-up resistor, a current will flow through the pull-up resistor if the input pin is driven LOW. For LVTTL 3.3 V, the pull-up resistor is ~45 kΩ, and the resulting current is equal to 3.3 V / 45 kΩ = 73 µA for the I/O pin. This is true also when a weak pull-down is chosen and the input pin is driven High. This current can be avoided by driving the input Low when a weak pull-down resistor is used and driving it High when a weak pull-up resistor is used.

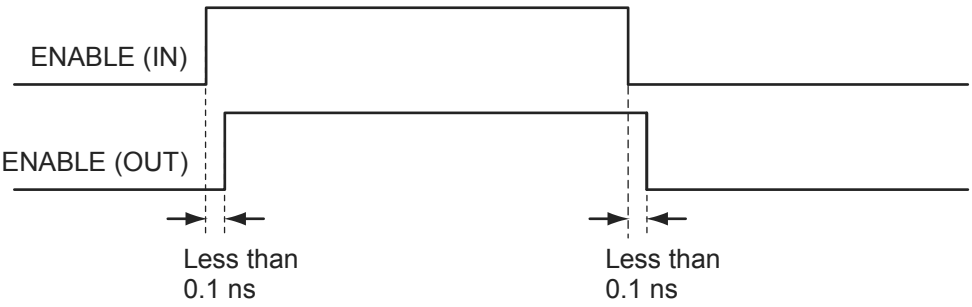


***Figure 8-15* • Timing Diagram (option 1: bypasses skew circuit)**

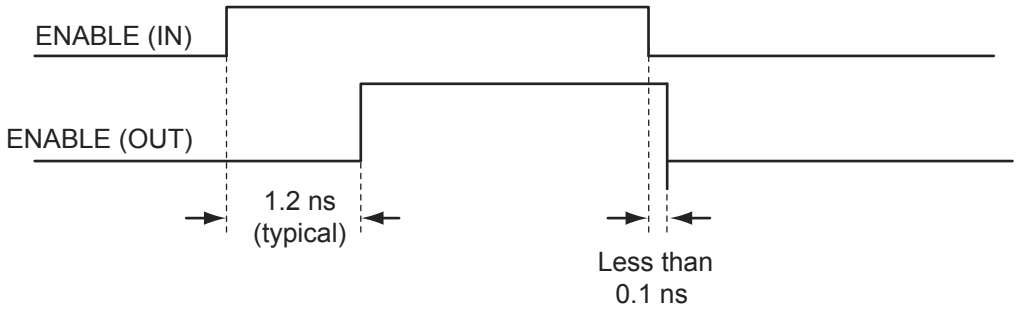


***Figure 8-16* • Timing Diagram (option 2: enables skew circuit)**

At the system level, the skew circuit can be used in applications where transmission activities on bidirectional data lines need to be coordinated. This circuit, when selected, provides a timing margin that can prevent bus contention and subsequent data loss and/or transmitter over-stress due to transmitter-to-transmitter current shorts. Figure 8-17 presents an example of the skew circuit implementation in a bidirectional communication system. Figure 8-18 on page 238 shows how bus contention is created, and Figure 8-19 on page 238 shows how it can be avoided with the skew circuit.

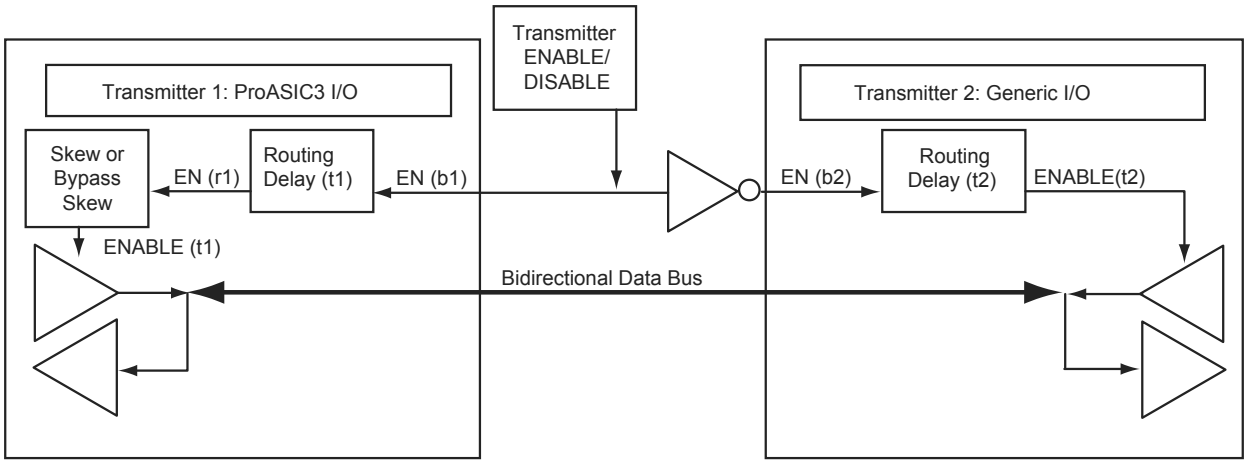


***Figure 8-17* • Example of Implementation of Skew Circuits in Bidirectional Transmission Systems Using IGLOO or ProASIC3 Devices**

***Figure 10-11* • DDR Input/Output Cells as Seen by ChipPlanner for IGLOO/e Devices**

Verilog

```
module Inbuf_ddr(PAD,CLR,CLK,QR,QF);

input PAD, CLR, CLK;
output  QR, QF;

wire Y;

  DDR_REG DDR_REG_0_inst(.D(Y), .CLK(CLK), .CLR(CLR), .QR(QR), .QF(QF));
  INBUF INBUF_0_inst(.PAD(PAD), .Y(Y));

endmodule

module Outbuf_ddr(DataR,DataF,CLR,CLK,PAD);

input DataR, DataF, CLR, CLK;
output  PAD;

wire Q, VCC;

  VCC VCC_1_net(.Y(VCC));
  DDR_OUT DDR_OUT_0_inst(.DR(DataR), .DF(DataF), .CLK(CLK), .CLR(CLR), .Q(Q));
  OUTBUF OUTBUF_0_inst(.D(Q), .PAD(PAD));

endmodule
```

Generating Programming Files

Generation of the Programming File in a Trusted Environment—Application 1

As discussed in the "Application 1: Trusted Environment" section on page 309, in a trusted environment, the user can choose to program the device with plaintext bitstream content. It is possible to use plaintext for programming even when the FlashLock Pass Key option has been selected. In this application, it is not necessary to employ AES encryption protection. For AES encryption settings, refer to the next sections.

The generated programming file will include the security setting (if selected) and the plaintext programming file content for the FPGA array, FlashROM, and/or FBs. These options are indicated in Table 12-2 and Table 12-3.

***Table 12-2* • IGLOO and ProASIC3 Plaintext Security Options, No AES**

Security Protection	FlashROM Only	FPGA Core Only	Both FlashROM and FPGA
No AES / no FlashLock	✓	✓	✓
FlashLock only	✓	✓	✓
AES and FlashLock	–	–	–

***Table 12-3* • Fusion Plaintext Security Options**

Security Protection	FlashROM Only	FPGA Core Only	FB Core Only	All
No AES / no FlashLock	✓	✓	✓	✓
FlashLock	✓	✓	✓	✓
AES and FlashLock	–	–	–	–

Note: For all instructions, the programming of Flash Blocks refers to Fusion only.

For this scenario, generate the programming file as follows:

1. Select the **Silicon features to be programmed** (Security Settings, FPGA Array, FlashROM, Flash Memory Blocks), as shown in Figure 12-10 on page 314 and Figure 12-11 on page 314. Click **Next**.

   If **Security Settings** is selected (i.e., the FlashLock security Pass Key feature), an additional dialog will be displayed to prompt you to select the security level setting. If no security setting is selected, you will be directed to Step 3.

STAPL File with AES Encryption

- Does not contain AES key / FlashLock Key information
- Intended for transmission through web or service to unsecured locations for programming

```
============================================
NOTE "CREATOR" "Designer Version: 6.1.1.108";
NOTE "DEVICE" "A3PE600";
NOTE "PACKAGE" "208 PQFP";
NOTE "DATE" "2005/04/08";
NOTE "STAPL_VERSION" "JESD71";
NOTE "IDCODE" "$123261CF";
NOTE "DESIGN" "counter32";
NOTE "CHECKSUM" "$EF57";
NOTE "SAVE_DATA" "FRomStream";
NOTE "SECURITY" "ENCRYPT FROM CORE ";
NOTE "ALG_VERSION" "1";
NOTE "MAX_FREQ" "20000000";
NOTE "SILSIG" "$00000000";
```

Conclusion

The new and enhanced security features offered in Fusion, IGLOO, and ProASIC3 devices provide state-of-the-art security to designs programmed into these flash-based devices. Microsemi low power flash devices employ the encryption standard used by NIST and the U.S. government—AES using the 128-bit Rijndael algorithm.

The combination of an on-chip AES decryption engine and FlashLock technology provides the highest level of security against invasive attacks and design theft, implementing the most robust and secure ISP solution. These security features protect IP within the FPGA and protect the system from cloning, wholesale “black box” copying of a design, invasive attacks, and explicit IP or data theft.

Glossary

Term	Explanation
Security Header programming file	Programming file used to program the FlashLock Pass Key and/or AES key into the device to secure the FPGA, FlashROM, and/or FBs.
AES (encryption) key	128-bit key defined by the user when the AES encryption option is set in the Microsemi Designer software when generating the programming file.
FlashLock Pass Key	128-bit key defined by the user when the FlashLock option is set in the Microsemi Designer software when generating the programming file. The FlashLock Key protects the security settings programmed to the device. Once a device is programmed with FlashLock, whatever settings were chosen at that time are secure.
FlashLock	The combined security features that protect the device content from attacks. These features are the following: • Flash technology that does not require an external bitstream to program the device • FlashLock Pass Key that secures device content by locking the security settings and preventing access to the device as defined by the user • AES key that allows secure, encrypted device reprogrammability

References

National Institute of Standards and Technology. “ADVANCED ENCRYPTION STANDARD (AES) Questions and Answers.” 28 January 2002 (10 January 2005).
See http://csrc.nist.gov/archive/aes/index1.html for more information.

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).	359
	The "Advanced Boundary Scan Register Settings" section is new (SAR 38432).	361
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 16-3 • TRST and TCK Pull-Down Recommendations was revised to add VJTAG at 1.2 V.	360
v1.4 (December 2008)	IGLOO nano and ProASIC3 nano devices were added to Table 16-1 • Flash-Based FPGAs.	358
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.	359
v1.2 (June 2008)	The following changes were made to the family descriptions in Table 16-1 • Flash-Based FPGAs: • ProASIC3L was updated to include 1.5 V. • The number of PLLs for ProASIC3E was changed from five to six.	358
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.	358

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 kΩ 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**