# E·XFL



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	36864
Number of I/O	71
Number of Gates	125000
Voltage - Supply	1.425V ~ 1.575V
Mounting Type	Surface Mount
Operating Temperature	-40°C ~ 100°C (TJ)
Package / Case	100-TQFP
Supplier Device Package	100-VQFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/a3pn125-2vq100i

Email: info@E-XFL.COM

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



Figure 2-1 • CCC/PLL Macro

## User Low Static (Idle) Mode

User Low Static (Idle) mode is an advanced feature supported by ProASIC3/E devices to reduce static (idle) power consumption. Entering and exiting this mode is made possible using the ULSICC macro by setting its value to disable/enable the User Low Static (Idle) mode. Under typical operating conditions, characterization results show up to 25% reduction of the static (idle) power consumption. The greatest power savings in terms of percentage are seen in the smaller members of the ProASIC3 family. The active-high control signal for User Low Static (Idle) mode can be generated by internal or external logic. When the device is operating in User Low Static (Idle) mode, FlashROM functionality is temporarily disabled to save power. If FlashROM functionality is needed, the device can exit User Low Static mode temporarily and re-enter the mode once the functionality is no longer needed.

To utilize User Low Static (Idle) mode, simply instantiate the ULSICC macro (Table 2-2 on page 24) in your design, and connect the input port to either an internal logic signal or a device package pin, as illustrated in Figure 2-2 on page 24 or Figure 2-3 on page 25, respectively. The attribute is used so the Synplify<sup>®</sup> synthesis tool will not optimize the instance with no output port.

This mode can be used to lower standard static (idle) power consumption when the FlashROM feature is not needed. Configuring the device to enter User Low Static (Idle) mode is beneficial when the FPGA enters and exits static mode frequently and lowering power consumption as much as possible is desired. The device is still functional, and data is retained in this state so the device can enter and exit this mode quickly, resulting in reduced total power consumption. The device can also stay in User Low Static mode when the FlashROM feature is not used in the device.

Global Resources in Low Power Flash Devices

#### Table 3-5 • Globals/Spines/Rows for IGLOO PLUS Devices

IGLOO PLUS Devices	Chip Globals	Quadrant Globals (4×3)	Clock Trees	Globals/ Spines per Tree	Total Spines per Device	VersaTiles in Each Tree	Total VersaTiles	Rows in Each Spine
AGLP030	6	0	2	9	18	384*	792	12
AGLP060	6	12	4	9	36	384*	1,584	12
AGLP125	6	12	8	9	72	384*	3,120	12

Note: \*Clock trees that are located at far left and far right will support more VersaTiles.

Table 3-6 • Globa	als/Spines/Rows	for Fusion	Devices

Fusion Device	Chip Globals	Quadrant Globals (4×3)	Clock Trees	Globals/ Spines per Tree	Total Spines per Device	VersaTiles in Each Tree	Total VersaTiles	Rows in Each Spine
AFS090	6	12	6	9	54	384	2,304	12
AFS250	6	12	8	9	72	768	6,144	24
AFS600	6	12	12	9	108	1,152	13,824	36
AFS1500	6	12	20	9	180	1,920	38,400	60

ProASIC3 nano FPGA Fabric User's Guide



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

## List of Changes

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
	Notes were added where appropriate to point out that IGLOO nano and ProASIC3 nano devices do not support differential inputs (SAR 21449).	N/A
	The "Global Architecture" section and "VersaNet Global Network Distribution" section were revised for clarity (SARs 20646, 24779).	31, 33
	The "I/O Banks and Global I/Os" section was moved earlier in the document, renamed to "Chip and Quadrant Global I/Os", and revised for clarity. Figure 3-4 • Global Connections Details, Figure 3-6 • Global Inputs, Table 3-2 • Chip Global Pin Name, and Table 3-3 • Quadrant Global Pin Name are new (SARs 20646, 24779).	35
	The "Clock Aggregation Architecture" section was revised (SARs 20646, 24779).	41
	Figure 3-7 • Chip Global Aggregation was revised (SARs 20646, 24779).	43
	The "Global Macro and Placement Selections" section is new (SARs 20646, 24779).	48
v1.4 (December 2008)	The "Global Architecture" section was updated to include 10 k devices, and to include information about VersaNet global support for IGLOO nano devices.	31
	The Table 3-1 • Flash-Based FPGAs was updated to include IGLOO nano and ProASIC3 nano devices.	32
	The "VersaNet Global Network Distribution" section was updated to include 10 k devices and to note an exception in global lines for nano devices.	33
	Figure 3-2 • Simplified VersaNet Global Network (30 k gates and below) is new.	34
	The "Spine Architecture" section was updated to clarify support for 10 k and nano devices.	41
	Table 3-4 • Globals/Spines/Rows for IGLOO and ProASIC3 Devices was updated to include IGLOO nano and ProASIC3 nano devices.	41
	The figure in the CLKBUF_LVDS/LVPECL row of Table 3-8 • Clock Macros was updated to change CLKBIBUF to CLKBUF.	46
v1.3 (October 2008)	A third bullet was added to the beginning of the "Global Architecture" section: In Fusion devices, the west CCC also contains a PLL core. In the two larger devices (AFS600 and AFS1500), the west and east CCCs each contain a PLL.	31
	The "Global Resource Support in Flash-Based Devices" section was revised to include new families and make the information more concise.	32
	Table 3-4 • Globals/Spines/Rows for IGLOO and ProASIC3 Devices was updated to include A3PE600/L in the device column.	41
	Table note 1 was revised in Table 3-9 • I/O Standards within CLKBUF to include AFS600 and AFS1500.	47
v1.2 (June 2008)	The following changes were made to the family descriptions in Table 3-1 • Flash-Based FPGAs:	32
	ProASIC3L was updated to include 1.5 V.	
	The number of PLLs for ProASIC3E was changed from five to six.	

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

## Microsemi

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







Figure 4-10 • Illustration of Hardwired I/O (global input pins) Usage for IGLOO and ProASIC3 devices 30 k Gates and Smaller

### 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



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

```
DLYGLC[4:0] 00000
DLYYB[4:0] 00000
DLYYC[4:0] 00000
VCOSEL[2:0] 100
```

Primary Clock Frequency 33.000 Primary Clock Phase Shift 0.000 Primary Clock Output Delay from CLKA 1.695

Secondaryl Clock Frequency 40.000 Secondaryl Clock Phase Shift 0.000 Secondaryl Clock Global Output Delay from CLKB 0.200

Secondary2 Clock Frequency 50.000 Secondary2 Clock Phase Shift 0.000 Secondary2 Clock Global Output Delay from CLKC 0.200

\*\*\*\*\*

NAME	SDIN	VALUE	TYPE
FINDIV	[6:0]	0000101	EDIT
FBDIV	[13:7]	0100000	EDIT
OADIV	[18:14]	00100	EDIT
OBDIV	[23:19]	00000	EDIT
OCDIV	[28:24]	00000	EDIT
OAMUX	[31:29]	100	EDIT
OBMUX	[34:32]	000	EDIT
OCMUX	[37:35]	000	EDIT
FBSEL	[39:38]	01	EDIT
FBDLY	[44:40]	00000	EDIT
XDLYSEL	[45]	0	EDIT
DLYGLA	[50:46]	00000	EDIT
DLYGLB	[55:51]	00000	EDIT
DLYGLC	[60:56]	00000	EDIT
DLYYB	[65:61]	00000	EDIT
DLYYC	[70:66]	00000	EDIT
STATASEL	[71]	X	MASKED
STATBSEL	[72]	X	MASKED
STATCSEL	[73]	X	MASKED
VCOSEL	[76:74]	100	EDIT
DYNASEL	[77]	X	MASKED
DYNBSEL	[78]	X	MASKED
DYNCSEL	[79]	X	MASKED
RESETEN	[80]	1	READONLY

Below is the resultant Verilog HDL description of a legal dynamic PLL core configuration generated by SmartGen:

module dyn\_pll\_macro(POWERDOWN, CLKA, LOCK, GLA, GLB, GLC, SDIN, SCLK, SSHIFT, SUPDATE, MODE, SDOUT, CLKB, CLKC);

input POWERDOWN, CLKA; output LOCK, GLA, GLB, GLC; input SDIN, SCLK, SSHIFT, SUPDATE, MODE; output SDOUT; input CLKB, CLKC; wire VCC, GND; VCC VCC\_1\_net(.Y(VCC));

GND GND\_1\_net(.Y(GND));

## Microsemi

ProASIC3 nano FPGA Fabric User's Guide

```
DYNCCC Core(.CLKA(CLKA), .EXTFB(GND), .POWERDOWN(POWERDOWN), .GLA(GLA), .LOCK(LOCK),
  .CLKB(CLKB), .GLB(GLB), .YB(), .CLKC(CLKC), .GLC(GLC), .YC(), .SDIN(SDIN),
  .SCLK(SCLK), .SSHIFT(SSHIFT), .SUPDATE(SUPDATE), .MODE(MODE), .SDOUT(SDOUT),
  .OADIV0(GND), .OADIV1(GND), .OADIV2(VCC), .OADIV3(GND), .OADIV4(GND), .OAMUX0(GND),
  .OAMUX1(GND), .OAMUX2(VCC), .DLYGLA0(GND), .DLYGLA1(GND), .DLYGLA2(GND),
  .DLYGLA3(GND), .DLYGLA4(GND), .OBDIV0(GND), .OBDIV1(GND), .OBDIV2(GND),
  .OBDIV3(GND), .OBDIV4(GND), .OBMUX0(GND), .OBMUX1(GND), .OBMUX2(GND), .DLYYB0(GND),
  .DLYYB1(GND), .DLYYB2(GND), .DLYYB3(GND), .DLYYB4(GND), .DLYGLB0(GND),
  .DLYGLB1(GND), .DLYGLB2(GND), .DLYGLB3(GND), .DLYGLB4(GND), .OCDIV0(GND),
  .OCDIV1(GND), .OCDIV2(GND), .OCDIV3(GND), .OCDIV4(GND), .OCMUX0(GND), .OCMUX1(GND),
  .OCMUX2(GND), .DLYYC0(GND), .DLYYC1(GND), .DLYYC2(GND), .DLYYC3(GND), .DLYYC4(GND),
  .DLYGLC0(GND), .DLYGLC1(GND), .DLYGLC2(GND), .DLYGLC3(GND), .DLYGLC4(GND),
  .FINDIV0(VCC), .FINDIV1(GND), .FINDIV2(VCC), .FINDIV3(GND), .FINDIV4(GND),
  .FINDIV5(GND), .FINDIV6(GND), .FBDIV0(GND), .FBDIV1(GND), .FBDIV2(GND),
  .FBDIV3(GND), .FBDIV4(GND), .FBDIV5(VCC), .FBDIV6(GND), .FBDLY0(GND), .FBDLY1(GND),
  .FBDLY2(GND), .FBDLY3(GND), .FBDLY4(GND), .FBSEL0(VCC), .FBSEL1(GND),
  .XDLYSEL(GND), .VCOSEL0(GND), .VCOSEL1(GND), .VCOSEL2(VCC));
defparam Core.VCOFREQUENCY = 165.000;
```

endmodule

\*\*\*\*\*

### **Delayed Clock Configuration**

The CLKDLY macro can be generated with the desired delay and input clock source (Hardwired I/O, External I/O, or Core Logic), as in Figure 4-28.

#### Figure 4-28 • Delayed Clock Configuration Dialog Box

After setting all the required parameters, users can generate one or more PLL configurations with HDL or EDIF descriptions by clicking the **Generate** button. SmartGen gives the option of saving session results and messages in a log file:

```
Macro Parameters
*****
                               : delay_macro
Name
Family
                               : ProASIC3
                               : Verilog
Output Format
                               : Delayed Clock
Type
Delay Index
                               : 2
CLKA Source
                               : Hardwired I/O
Total Clock Delay = 0.935 ns.
The resultant CLKDLY macro Verilog netlist is as follows:
module delay_macro(GL,CLK);
output GL;
input CLK;
```

### SRAM Usage

The following descriptions refer to the usage of both RAM4K9 and RAM512X18.

#### Clocking

The dual-port SRAM blocks are only clocked on the rising edge. SmartGen allows falling-edge-triggered clocks by adding inverters to the netlist, hence achieving dual-port SRAM blocks that are clocked on either edge (rising or falling). For dual-port SRAM, each port can be clocked on either edge and by separate clocks by port. Note that for Automotive ProASIC3, the same clock, with an inversion between the two clock pins of the macro, should be used in design to prevent errors during compile.

Low power flash devices support inversion (bubble-pushing) throughout the FPGA architecture, including the clock input to the SRAM modules. Inversions added to the SRAM clock pin on the design schematic or in the HDL code will be automatically accounted for during design compile without incurring additional delay in the clock path.

The two-port SRAM can be clocked on the rising or falling edge of WCLK and RCLK.

If negative-edge RAM and FIFO clocking is selected for memory macros, clock edge inversion management (bubble-pushing) is automatically used within the development tools, without performance penalty.

#### Modes of Operation

There are two read modes and one write mode:

- Read Nonpipelined (synchronous—1 clock edge): In the standard read mode, new data is driven
  onto the RD bus in the same clock cycle following RA and REN valid. The read address is
  registered on the read port clock active edge, and data appears at RD after the RAM access time.
  Setting PIPE to OFF enables this mode.
- Read Pipelined (synchronous—2 clock edges): The pipelined mode incurs an additional clock delay from address to data but enables operation at a much higher frequency. The read address is registered on the read port active clock edge, and the read data is registered and appears at RD after the second read clock edge. Setting PIPE to ON enables this mode.
- Write (synchronous—1 clock edge): On the write clock active edge, the write data is written into the SRAM at the write address when WEN is HIGH. The setup times of the write address, write enables, and write data are minimal with respect to the write clock.

#### **RAM** Initialization

Each SRAM block can be individually initialized on power-up by means of the JTAG port using the UJTAG mechanism. The shift register for a target block can be selected and loaded with the proper bit configuration to enable serial loading. The 4,608 bits of data can be loaded in a single operation.

### **FIFO Features**

The FIFO4KX18 macro is created by merging the RAM block with dedicated FIFO logic (Figure 6-6 on page 142). Since the FIFO logic can only be used in conjunction with the memory block, there is no separate FIFO controller macro. As with the RAM blocks, the FIFO4KX18 nomenclature does not refer to a possible aspect ratio, but rather to the deepest possible data depth and the widest possible data width. FIFO4KX18 can be configured into the following aspect ratios: 4,096×1, 2,048×2, 1,024×4, 512×9, and 256×18. In addition to being fully synchronous, the FIFO4KX18 also has the following features:

- Four FIFO flags: Empty, Full, Almost-Empty, and Almost-Full
- Empty flag is synchronized to the read clock
- Full flag is synchronized to the write clock
- Both Almost-Empty and Almost-Full flags have programmable thresholds
- · Active-low asynchronous reset
- Active-low block enable
- Active-low write enable
- Active-high read enable
- Ability to configure the FIFO to either stop counting after the empty or full states are reached or to allow the FIFO counters to continue

## Low Power Flash Device I/O Support

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

Series	Family <sup>*</sup>	Description
IGLOO	IGLOO nano	Lowest power 1.2 V to 1.5 V FPGAs with Flash*Freeze technology
ProASIC3	ProASIC3 nano	Lowest cost 1.5 V FPGAs with balanced performance

#### Table 7-1 • Flash-Based FPGAs

Note: \*The device name links 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*.



## **Power-Up Behavior**

Low power flash devices are power-up/-down friendly; i.e., no particular sequencing is required for power-up and power-down. This eliminates extra board components for power-up sequencing, such as a power-up sequencer.

During power-up, all I/Os are tristated, irrespective of I/O macro type (input buffers, output buffers, I/O buffers with weak pull-ups or weak pull-downs, etc.). Once I/Os become activated, they are set to the user-selected I/O macros. Refer to the "Power-Up/-Down Behavior of Low Power Flash Devices" section on page 307 for details.

## **Drive Strength**

Low power flash devices have up to four programmable output drive strengths. The user can select the drive strength of a particular output in the I/O Attribute Editor or can instantiate a specialized I/O macro, such as OUTBUF\_S\_8 (slew = low, out\_drive = 8 mA).

The maximum available drive strength is 8 mA per I/O. Though no I/O should be forced to source or sink more than 8 mA indefinitely, I/Os may handle a higher amount of current (refer to the device IBIS model for maximum source/sink current) during signal transition (AC current). Every device package has its own power dissipation limit; hence, power calculation must be performed accurately to determine how much current can be tolerated per I/O within that limit.

## I/O Interfacing

Low power flash devices are 5 V–input– and 5 V–output–tolerant without adding any extra circuitry. Along with other low-voltage I/O macros, this 5 V tolerance makes these devices suitable for many types of board component interfacing.

Table 7-17 shows some high-level interfacing examples using low power flash devices.

		Clock	I/O			
Interface	Туре	Frequency	Туре	Signals In	Signals Out	Data I/O
GM	Src Sync	125 MHz	LVTTL	8	8	125 Mbps
TBI	Src Sync	125 MHz	LVTTL	10	10	125 Mbps

#### Table 7-17 • nano High-Level Interface

## Conclusion

IGLOO nano and ProASIC3 nano device support for multiple I/O standards minimizes board-level components and makes possible a wide variety of applications. The Microsemi Designer software, integrated with Libero SoC, presents a clear visual display of I/O assignments, allowing users to verify I/O and board-level design requirements before programming the device. The nano device I/O features and functionalities ensure board designers can produce low-cost and low power FPGA applications fulfilling the complexities of contemporary design needs.

## Microsemi

I/O Software Control in Low Power Flash Devices

## Flash FPGAs I/O Support

The 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 <sup>*</sup>	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 <sup>®</sup> Cortex <sup>™</sup> -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 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*.

## *Microsemi*.

I/O Software Control in Low Power Flash Devices

### Instantiating in HDL code

All the supported I/O macros can be instantiated in the top-level HDL code (refer to the *IGLOO*, *ProASIC3*, *SmartFusion*, *and Fusion Macro Library Guide* for a detailed list of all I/O macros). The following is an example:

```
library ieee;
use ieee.std_logic_1164.all;
library proasic3e;
entity TOP is
 port(IN2, IN1 : in std_logic; OUT1 : out std_logic);
end TOP;
architecture DEF_ARCH of TOP is
  component INBUF_LVCMOS5U
    port(PAD : in std_logic := 'U'; Y : out std_logic);
  end component;
  component INBUF_LVCMOS5
   port(PAD : in std_logic := 'U'; Y : out std_logic);
  end component;
  component OUTBUF_SSTL3_II
    port(D : in std_logic := 'U'; PAD : out std_logic);
  end component;
  Other component ....
signal x, y, z.....other signals : std_logic;
begin
  I1 : INBUF_LVCMOS5U
   port map(PAD => IN1, Y =>x);
  12 : INBUF LVCMOS5
   port map(PAD => IN2, Y => y);
  I3 : OUTBUF_SSTL3_II
    port map(D => z, PAD => OUT1);
```

other port mapping ...

end DEF\_ARCH;

## Synthesizing the Design

Libero SoC integrates with the Synplify<sup>®</sup> synthesis tool. Other synthesis tools can also be used with Libero SoC. Refer to the *Libero SoC User's Guide* or Libero online help for details on how to set up the Libero tool profile with synthesis tools from other vendors.

During synthesis, the following rules apply:

- Generic macros:
  - Users can instantiate generic INBUF, OUTBUF, TRIBUF, and BIBUF macros.
  - Synthesis will automatically infer generic I/O macros.
  - The default I/O technology for these macros is LVTTL.
  - Users will need to use the I/O Attribute Editor in Designer to change the default I/O standard if needed (see Figure 8-6 on page 193).
- Technology-specific I/O macros:
  - Technology-specific I/O macros, such as INBUF\_LVCMO25 and OUTBUF\_GTL25, can be instantiated in the design. Synthesis will infer these I/O macros in the netlist.



I/O Software Control in Low Power Flash Devices

### I/O Function

Figure 8-8 shows an example of the I/O Function table included in the I/O bank report:

#### Figure 8-8 • I/O Function Table

This table lists the number of input I/Os, output I/Os, bidirectional I/Os, and differential input and output I/O pairs that use I/O and DDR registers.

Note: IGLOO nano and ProASIC3 nano devices do not support differential inputs.

Certain rules must be met to implement registered and DDR I/O functions (refer to the I/O Structures section of the handbook for the device you are using and the "DDR" section on page 190).

#### I/O Technology

The I/O Technology table (shown in Figure 8-9) gives the values of VCCI and VREF (reference voltage) for all the I/O standards used in the design. The user should assign these voltages appropriately.

Figure 8-9 • I/O Technology Table

### **Device Programmers**

#### Single Device Programmer

Single device programmers are used to program a device before it is mounted on the system board.

The advantage of using device programmers is that no programming hardware is required on the system board. Therefore, no additional components or board space are required.

Adapter modules are purchased with single device programmers to support the FPGA packages used. The FPGA is placed in the adapter module and the programming software is run from a PC. Microsemi supplies the programming software for all of the Microsemi programmers. The software allows for the selection of the correct die/package and programming files. It will then program and verify the device.

Single-site programmers

A single-site programmer programs one device at a time. Microsemi offers Silicon Sculptor 3, built by BP Microsystems, as a single-site programmer. Silicon Sculptor 3 and associated software are available only from Microsemi.

- Advantages: Lower cost than multi-site programmers. No additional overhead for programming on the system board. Allows local control of programming and data files for maximum security. Allows on-demand programming on-site.
- Limitations: Only programs one device at a time.
- Multi-site programmers

Often referred to as batch or gang programmers, multi-site programmers can program multiple devices at the same time using the same programming file. This is often used for large volume programming and by programming houses. The sites often have independent processors and memory enabling the sites to operate concurrently, meaning each site may start programming the same file independently. This enables the operator to change one device while the other sites continue programming, which increases throughput. Multiple adapter modules for the same package are required when using a multi-site programmer. Silicon Sculptor I, II, and 3 programmers can be cascaded to program multiple devices in a chain. Multi-site programmers, such as the BP2610 and BP2710, can also be purchased from BP Microsystems. When using BP Microsystems multi-site programmers, users must use programming adapter modules available only from Microsemi. Visit the Microsemi SoC Products Group website to view the part numbers of the desired adapter module:

http://www.microsemi.com/soc/products/hardware/program\_debug/ss/modules.aspx.

Also when using BP Microsystems programmers, customers must use Microsemi programming software to ensure the best programming result will occur.

- Advantages: Provides the capability of programming multiple devices at the same time. No
  additional overhead for programming on the system board. Allows local control of
  programming and data files for maximum security.
- Limitations: More expensive than a single-site programmer
- Automated production (robotic) programmers

Automated production programmers are based on multi-site programmers. They consist of a large input tray holding multiple parts and a robotic arm to select and place parts into appropriate programming sockets automatically. When the programming of the parts is complete, the parts are removed and placed in a finished tray. The automated programmers are often used in volume programming houses to program parts for which the programming time is small. BP Microsystems part number BP4710, BP4610, BP3710 MK2, and BP3610 are available for this purpose. Auto programmers cannot be used to program RTAX-S devices.

Where an auto-programmer is used, the appropriate open-top adapter module from BP Microsystems must be used.



Programming Flash Devices

## List of Changes

Date	Changes	Page
July 2010	FlashPro4 is a replacement for FlashPro3 and has been added to this chapter. FlashPro is no longer available.	N/A
	The chapter was updated to include SmartFusion devices.	N/A
	The following were deleted:	N/A
	"Live at Power-Up (LAPU) or Boot PROM" section	
	"Design Security" section	
	Table 14-2 • Programming Features for Actel Devices and much of the text in the"Programming Features for Microsemi Devices" section	
	"Programming Flash FPGAs" section	
	"Return Material Authorization (RMA) Policies" section	
	The "Device Programmers" section was revised.	225
	The Independent Programming Centers information was removed from the "Volume Programming Services" section.	226
	Table 10-3 • Programming Solutions was revised to add FlashPro4 and note that FlashPro is discontinued. A note was added for FlashPro Lite regarding power supply requirements.	227
	Most items were removed from Table 10-4 • Programming Ordering Codes, including FlashPro3 and FlashPro.	228
	The "Programmer Device Support" section was deleted and replaced with a reference to the Microsemi SoC Products Group website for the latest information.	228
	The "Certified Programming Solutions" section was revised to add FlashPro4 and remove Silicon Sculptor I and Silicon Sculptor 6X. Reference to <i>Programming and Functional Failure Guidelines</i> was added.	228
	The file type *.pdb was added to the "Use the Latest Version of the Designer Software to Generate Your Programming File (recommended)" section.	229
	Instructions on cleaning and careful insertion were added to the "Perform Routine Hardware Self-Diagnostic Test" section. Information was added regarding testing Silicon Sculptor programmers with an adapter module installed before every programming session verifying their calibration annually.	229
	The "Signal Integrity While Using ISP" section is new.	230
	The "Programming Failure Allowances" section was revised.	230

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

2. Choose the appropriate security level setting and enter a FlashLock Pass Key. The default is the **Medium** security level (Figure 11-12). Click **Next**.

If you want to select different options for the FPGA and/or FlashROM, this can be set by clicking **Custom Level**. Refer to the "Advanced Options" section on page 256 for different custom security level options and descriptions of each.

Figure 11-12 • Medium Security Level Selected for Low Power Flash Devices

## STAPL vs. DirectC

Programming the low power flash devices is performed using DirectC or the STAPL player. Both tools use the STAPL file as an input. DirectC is a compiled language, whereas STAPL is an interpreted language. Microprocessors will be able to load the FPGA using DirectC much more quickly than STAPL. This speed advantage becomes more apparent when lower clock speeds of 8- or 16-bit microprocessors are used. DirectC also requires less memory than STAPL, since the programming algorithm is directly implemented. STAPL does have one advantage over DirectC—the ability to upgrade. When a new programming algorithm is required, the STAPL user simply needs to regenerate a STAPL file using the latest version of the Designer software and download it to the system. The DirectC user must download the latest version of DirectC from Microsemi, compile everything, and download the result into the system (Figure 14-4).



Figure 14-4 • STAPL vs. DirectC

## **Power-Up/-Down Sequence and Transient Current**

Microsemi's low power flash devices use the following main voltage pins during normal operation:<sup>2</sup>

- VCCPLX
- VJTAG
- VCC: Voltage supply to the FPGA core
  - VCC is 1.5 V ± 0.075 V for IGLOO, IGLOO nano, IGLOO PLUS, and ProASIC3 devices operating at 1.5 V.
  - VCC is 1.2 V ± 0.06 V for IGLOO, IGLOO nano, IGLOO PLUS, and ProASIC3L devices operating at 1.2 V.
  - V5 devices will require a 1.5 V VCC supply, whereas V2 devices can utilize either a 1.2 V or 1.5 V VCC.
- VCCIBx: Supply voltage to the bank's I/O output buffers and I/O logic. Bx is the I/O bank number.
- VMVx: Quiet supply voltage to the input buffers of each I/O bank. x is the bank number. (Note: IGLOO nano, IGLOO PLUS, and ProASIC3 nano devices do not have VMVx supply pins.)

The I/O bank VMV pin must be tied to the VCCI pin within the same bank. Therefore, the supplies that need to be powered up/down during normal operation are VCC and VCCI. These power supplies can be powered up/down in any sequence during normal operation of IGLOO, IGLOO nano, IGLOO PLUS, ProASIC3L, ProASIC3, and ProASIC3 nano FPGAs. During power-up, I/Os in each bank will remain tristated until the last supply (either VCCIBx or VCC) reaches its functional activation voltage. Similarly, during power-down, I/Os of each bank are tristated once the first supply reaches its brownout deactivation voltage.

Although Microsemi's low power flash devices have no power-up or power-down sequencing requirements, Microsemi identifies the following power conditions that will result in higher than normal transient current. Use this information to help maximize power savings:

Microsemi recommends tying VCCPLX to VCC and using proper filtering circuits to decouple VCC noise from the PLL.

a. If VCCPLX is powered up before VCC, a static current of up to 5 mA (typical) per PLL may be measured on VCCPLX.

The current vanishes as soon as VCC reaches the VCCPLX voltage level.

The same current is observed at power-down (VCC before VCCPLX).

- b. If VCCPLX is powered up simultaneously or after VCC:
  - Microsemi's low power flash devices exhibit very low transient current on VCC. For ProASIC3 devices, the maximum transient current on V<sub>CC</sub> does not exceed the maximum standby current specified in the device datasheet.

The source of transient current, also known as inrush current, varies depending on the FPGA technology. Due to their volatile technology, the internal registers in SRAM FPGAs must be initialized before configuration can start. This initialization is the source of significant inrush current in SRAM FPGAs during power-up. Due to the nonvolatile nature of flash technology, low power flash devices do not require any initialization at power-up, and there is very little or no crossbar current through PMOS and NMOS devices. Therefore, the transient current at power-up is significantly less than for SRAM FPGAs. Figure 17-1 on page 310 illustrates the types of power consumption by SRAM FPGAs compared to Microsemi's antifuse and flash FPGAs.

For more information on Microsemi FPGA voltage supplies, refer to the appropriate datasheet located at http://www.microsemi.com/soc/techdocs/ds.