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/a3pn125-2vqg100i

Email: info@E-XFL.COM

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

1 – FPGA Array Architecture in Low Power Flash Devices

Device Architecture

Advanced Flash Switch

Unlike SRAM FPGAs, the low power flash devices use a live-at-power-up ISP flash switch as their programming element. Flash cells are distributed throughout the device to provide nonvolatile, reconfigurable programming to connect signal lines to the appropriate VersaTile inputs and outputs. In the flash switch, two transistors share the floating gate, which stores the programming information (Figure 1-1). One is the sensing transistor, which is only used for writing and verification of the floating gate voltage. The other is the switching transistor. The latter is used to connect or separate routing nets, or to configure VersaTile logic. It is also used to erase the floating gate. Dedicated high-performance lines are connected as required using the flash switch for fast, low-skew, global signal distribution throughout the device core. Maximum core utilization is possible for virtually any design. The use of the flash switch technology also removes the possibility of firm errors, which are increasingly common in SRAM-based FPGAs.



Figure 1-1 • Flash-Based Switch

FPGA Array Architecture in Low Power Flash Devices



Note: Flash*Freeze technology only applies to IGLOOe devices.

Figure 1-7 • IGLOOe and ProASIC3E Device Architecture Overview (AGLE600 device is shown)

I/O State of Newly Shipped Devices

Devices are shipped from the factory with a test design in the device. The power-on switch for VCC is OFF by default in this test design, so I/Os are tristated by default. Tristated means the I/O is not actively driven and floats. The exact value cannot be guaranteed when it is floating. Even in simulation software, a tristate value is marked as unknown. Due to process variations and shifts, tristated I/Os may float toward High or Low, depending on the particular device and leakage level.

If there is concern regarding the exact state of unused I/Os, weak pull-up/pull-down should be added to the floating I/Os so their state is controlled and stabilized.

Config. Bits	Signal	Name	Description	
83	RXCSEL ¹	CLKC input selection	Select the CLKC input clock source between RC oscillator and crystal oscillator (refer to Table 4-16 on page 94). ²	
82	RXBSEL ¹	CLKB input selection	Select the CLKB input clock source between RC oscillator and crystal oscillator (refer to Table 4-16 on page 94). ²	
81	RXASEL ¹	CLKA input selection	Select the CLKA input clock source between RC oscillator and crystal oscillator (refer to Table 4-16 on page 94). ²	
80	RESETEN	Reset Enable	Enables (active high) the synchronization of PLL output dividers after dynamic reconfiguration (SUPDATE). The Reset Enable signal is READ-ONLY.	
79	DYNCSEL	Clock Input C Dynamic Select	Configures clock input C to be sent to GLC for dynamic control. ²	
78	DYNBSEL	Clock Input B Dynamic Select	Configures clock input B to be sent to GLB for dynamic control. ²	
77	DYNASEL	Clock Input A Dynamic Select	Configures clock input A for dynamic PLL configuration. ²	
<76:74>	VCOSEL[2:0]	VCO Gear Control	Three-bit VCO Gear Control for four frequency ranges (refer to Table 4-19 on page 95 and Table 4-20 on page 95).	
73	STATCSEL	MUX Select on Input C	MUX selection for clock input C ²	
72	STATBSEL	MUX Select on Input B	MUX selection for clock input B ²	
71	STATASEL	MUX Select on Input A	MUX selection for clock input A ²	
<70:66>	DLYC[4:0]	YC Output Delay	Sets the output delay value for YC.	
<65:61>	DLYB[4:0]	YB Output Delay	Sets the output delay value for YB.	
<60:56>	DLYGLC[4:0]	GLC Output Delay	Sets the output delay value for GLC.	
<55:51>	DLYGLB[4:0]	GLB Output Delay	Sets the output delay value for GLB.	
<50:46>	DLYGLA[4:0]	Primary Output Delay	Primary GLA output delay	
45	XDLYSEL	System Delay Select	When selected, inserts System Delay in the feedback path in Figure 4-20 on page 85.	
<44:40>	FBDLY[4:0]	Feedback Delay	Sets the feedback delay value for the feedback element in Figure 4-20 on page 85.	
<39:38>	FBSEL[1:0]	Primary Feedback Delay Select	Controls the feedback MUX: no delay, includ programmable delay element, or use extern feedback.	
<37:35>	OCMUX[2:0]	Secondary 2 Output Select	Selects from the VCO's four phase outputs for GLC/YC.	
<34:32>	OBMUX[2:0]	Secondary 1 Output Select	Selects from the VCO's four phase outputs for GLB/YB.	

Table 4-8 • Configuration Bit Descriptions for the CCC Blocks (continued)

Notes:

1. The <88:81> configuration bits are only for the Fusion dynamic CCC.

 This value depends on the input clock source, so Layout must complete before these bits can be set. After completing Layout in Designer, generate the "CCC_Configuration" report by choosing Tools > Report > CCC_Configuration. The report contains the appropriate settings for these bits.



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

```
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 84) 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 109.

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>Vib\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 precompiled in the $ModelSim^{\mbox{\sc B}}$ Microsemi simulation tool.

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

Date	Changes	Page		
v1.4 (December 2008)	The"CCC Support in Microsemi's Flash Devices" section was updated to include IGLOO nano and ProASIC3 nano devices.	63		
	Figure 4-2 • CCC Options: Global Buffers with No Programmable Delay was revised to add the CLKBIBUF macro.			
	The description of the reference clock was revised in Table 4-2 • Input and Output Description of the CLKDLY Macro.			
	Figure 4-7 • Clock Input Sources (30 k gates devices and below) is new. Figure 4-8 • Clock Input Sources Including CLKBUF, CLKBUF_LVDS/LVPECL, and CLKINT (60 k gates devices and above) applies to 60 k gate devices and above.	72		
	The "IGLOO and ProASIC3" section was updated to include information for IGLOO nano devices.	73		
	A note regarding Fusion CCCs was added to Figure 4-9 • Illustration of Hardwired I/O (global input pins) Usage for IGLOO and ProASIC3 devices 60 k Gates and Larger and the name of the figure was changed from Figure 4-8 • Illustration of Hardwired I/O (global input pins) Usage. Figure 4-10 • Illustration of Hardwired I/O (global input pins) Usage for IGLOO and ProASIC3 devices 30 k Gates and Smaller is new.			
	Table 4-5 • Number of CCCs by Device Size and Package was updated to include IGLOO nano and ProASIC3 nano devices. Entries were added to note differences for the CS81, CS121, and CS201 packages.			
	The "Clock Conditioning Circuits without Integrated PLLs" section was rewritten.			
	The "IGLOO and ProASIC3 CCC Locations" section was updated for nano devices.	81		
	Figure 4-13 • CCC Locations in the 15 k and 30 k Gate Devices was deleted.	4-20		
v1.3 (October 2008)	This document was updated to include Fusion and RT ProASIC3 device information. Please review the document very carefully.	N/A		
	The "CCC Support in Microsemi's Flash Devices" section was updated.	63		
	In the "Global Buffer with Programmable Delay" section, the following sentence was changed from: "In this case, the I/O must be placed in one of the dedicated global I/O locations."	64		
	To "In this case, the software will automatically place the dedicated global I/O in the			
	Figure 4-4 • CCC Options: Global Buffers with PLL was updated to include OADIVRST and OADIVHALF.	67		
	In Figure 4-6 • CCC with PLL Block "fixed delay" was changed to "programmable delay".	67		
	Table 4-3 • Input and Output Signals of the PLL Block was updated to include OADIVRST and OADIVHALF descriptions.	68		
	Table 4-8 • Configuration Bit Descriptions for the CCC Blocks was updated to include configuration bits 88 to 81. Note 2 is new. In addition, the description for bit <76:74> was updated.	90		
	Table 4-16 • Fusion Dynamic CCC Clock Source Selection and Table 4-17 • Fusion Dynamic CCC NGMUX Configuration are new.	94		
	Table 4-18 • Fusion Dynamic CCC Division by Half Configuration and Table 4-19 • Configuration Bit <76:75> / VCOSEL<2:1> Selection for All Families are new.	95		

FlashROM Security

Low power flash devices have an on-chip Advanced Encryption Standard (AES) decryption core, combined with an enhanced version of the Microsemi flash-based lock technology (FlashLock[®]). Together, they provide unmatched levels of security in a programmable logic device. This security applies to both the FPGA core and FlashROM content. These devices use the 128-bit AES (Rijndael) algorithm to encrypt programming files for secure transmission to the on-chip AES decryption core. The same algorithm is then used to decrypt the programming file. This key size provides approximately 3.4×10^{38} possible 128-bit keys. A computing system that could find a DES key in a second would take approximately 149 trillion years to crack a 128-bit AES key. The 128-bit FlashLock feature in low power flash devices works via a FlashLock security Pass Key mechanism, where the user locks or unlocks the device with a user-defined key. Refer to the "Security in Low Power Flash Devices" section on page 235.

If the device is locked with certain security settings, functions such as device read, write, and erase are disabled. This unique feature helps to protect against invasive and noninvasive attacks. Without the correct Pass Key, access to the FPGA is denied. To gain access to the FPGA, the device first must be unlocked using the correct Pass Key. During programming of the FlashROM or the FPGA core, you can generate the security header programming file, which is used to program the AES key and/or FlashLock Pass Key. The security header programming file can also be generated independently of the FlashROM and FPGA core content. The FlashLock Pass Key is not stored in the FlashROM.

Low power flash devices with AES-based security allow for secure remote field updates over public networks such as the Internet, and ensure that valuable intellectual property (IP) remains out of the hands of IP thieves. Figure 5-5 shows this flow diagram.



Figure 5-5 • Programming FlashROM Using AES

SRAM and FIFO Memories in Microsemi's Low Power Flash Devices



Notes:

2. Flash*Freeze is supported in all IGLOO devices and the ProASIC3L devices.

Figure 6-1 • IGLOO and ProASIC3 Device Architecture Overview

^{1.} AES decryption not supported in 30 k gate devices and smaller.

I/O Software Support

In Microsemi's Libero software, default settings have been defined for the various I/O standards supported. Changes can be made to the default settings via the use of attributes; however, not all I/O attributes are applicable for all I/O standards.

	SLEW			OUT_LOAD (output only)				
I/O Standard	(output only)	OUT_DRIVE (output only)	RES_PULL	IGLOO nano	ProASIC 3 nano	Schmitt Trigger	Hold State	Combine Register
LVTTL/ LVCMOS3.3	1	✓ (8)	1	1	1	1	1	1
LVCMOS2.5	1	✓ (8)	1	1	1	1	1	1
LVCMOS1.8	1	✓ (4)	1	1	1	1	1	1
LVCMOS1.5	1	✓ (2)	1	1	1	1	1	1
LVCMOS1.2	1	✓ (2)	<i>✓</i>	1	-	1	1	1
Software Defaults	HIGH	Refer to numbers in parentheses in above cells.	None	All Devices: 5 pF	10 pF or 35 pF*	Off	Off	No

Table 7-15 • nano I/O Attributes vs. I/O Standard Applications

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

Software-Controlled I/O Attributes

Users may modify these programmable I/O attributes using the I/O Attribute Editor. Modifying an I/O attribute may result in a change of state in Designer. Table 8-2 details which steps have to be re-run as a function of modified I/O attribute.

	Designer States ¹						
I/O Attribute	Compile	Layout	Fuse	Timing	Power		
Slew Control ²	No	No	Yes	Yes	Yes		
Output Drive (mA)	No	No	Yes	Yes	Yes		
Skew Control	No	No	Yes	Yes	Yes		
Resistor Pull	No	No	Yes	Yes	Yes		
Input Delay	No	No	Yes	Yes	Yes		
Schmitt Trigger	No	No	Yes	Yes	Yes		
OUT_LOAD	No	No	No	Yes	Yes		
COMBINE_REGISTER	Yes	Yes	N/A	N/A	N/A		

Table 8-2 • Designer State (resulting from I/O attribute modification)

Notes:

1. No = Remains the same, Yes = Re-run the step, N/A = Not applicable

2. Skew control does not apply to IGLOO nano, IGLOO PLUS, and ProASIC3 nano devices.

3. Programmable input delay is applicable only for ProASIC3E, ProASIC3EL, RT ProASIC3, and IGLOOe devices.

I/O Software Control in Low Power Flash Devices

Output Buffers

There are two variations: Regular and Special.

If the **Regular** variation is selected, only the Width (1 to 128) needs to be entered. The default value for Width is 1.

The Special variation has Width, Technology, Output Drive, and Slew Rate options.

Bidirectional Buffers

There are two variations: Regular and Special.

The Regular variation has Enable Polarity (Active High, Active Low) in addition to the Width option.

The **Special** variation has Width, Technology, Output Drive, Slew Rate, and Resistor Pull-Up/-Down options.

Tristate Buffers

Same as Bidirectional Buffers.

DDR

There are eight variations: DDR with Regular Input Buffers, Special Input Buffers, Regular Output Buffers, Special Output Buffers, Regular Tristate Buffers, Special Tristate Buffers, Regular Bidirectional Buffers, and Special Bidirectional Buffers.

These variations resemble the options of the previous I/O macro. For example, the Special Input Buffers variation has Width, Technology, Voltage Level, and Resistor Pull-Up/-Down options. DDR is not available on IGLOO PLUS devices.

- 4. Once the desired configuration is selected, click the **Generate** button. The Generate Core window opens (Figure 8-4).
- 5. Enter a name for the macro. Click **OK**. The core will be generated and saved to the appropriate location within the project files (Figure 8-5 on page 191).

Figure 8-4 • Generate Core Window

6. Instantiate the I/O macro in the top-level code.

The user must instantiate the DDR_REG or DDR_OUT macro in the design. Use SmartGen to generate both these macros and then instantiate them in your top level. To combine the DDR macros with the I/O, the following rules must be met:

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[®] 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.

If the assignment is not successful, an error message appears in the Output window.

To undo the I/O bank assignments, choose **Undo** from the **Edit** menu. Undo removes the I/O technologies assigned by the IOBA. It does not remove the I/O technologies previously assigned.

To redo the changes undone by the Undo command, choose Redo from the Edit menu.

To clear I/O bank assignments made before using the Undo command, manually unassign or reassign I/O technologies to banks. To do so, choose **I/O Bank Settings** from the **Edit** menu to display the I/O Bank Settings dialog box.

Conclusion

Fusion, IGLOO, and ProASIC3 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 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.

Related Documents

User's Guides

Libero SoC User's Guide http://www.microsemi.com/soc/documents/libero_ug.pdf IGLOO, ProASIC3, SmartFusion, and Fusion Macro Library Guide http://www.microsemi.com/soc/documents/pa3_libguide_ug.pdf SmartGen Core Reference Guide http://www.microsemi.com/soc/documents/genguide_ug.pdf

9 – DDR for Microsemi's Low Power Flash Devices

Introduction

The I/Os in Fusion, IGLOO, and ProASIC3 devices support Double Data Rate (DDR) mode. In this mode, new data is present on every transition (or clock edge) of the clock signal. This mode doubles the data transfer rate compared with Single Data Rate (SDR) mode, where new data is present on one transition (or clock edge) of the clock signal. Low power flash devices have DDR circuitry built into the I/O tiles. I/Os are configured to be DDR receivers or transmitters by instantiating the appropriate special macros (examples shown in Figure 9-4 on page 210 and Figure 9-5 on page 211) and buffers (DDR_OUT or DDR_REG) in the RTL design. This document discusses the options the user can choose to configure the I/Os in this mode and how to instantiate them in the design.

Double Data Rate (DDR) Architecture

Low power flash devices support 350 MHz DDR inputs and outputs. In DDR mode, new data is present on every transition of the clock signal. Clock and data lines have identical bandwidths and signal integrity requirements, making them very efficient for implementing very high-speed systems. High-speed DDR interfaces can be implemented using LVDS (not applicable for IGLOO nano and ProASIC3 nano devices). In IGLOOe, ProASIC3E, AFS600, and AFS1500 devices, DDR interfaces can also be implemented using the HSTL, SSTL, and LVPECL I/O standards. The DDR feature is primarily implemented in the FPGA core periphery and is not tied to a specific I/O technology or limited to any I/O standard.



Figure 9-1 • DDR Support in Low Power Flash Devices

VHDL

```
library ieee;
use ieee.std_logic_1164.all;
library proasic3; use proasic3.all;
entity DDR_BiDir_HSTL_I_LowEnb is
  port(DataR, DataF, CLR, CLK, Trien : in std_logic; QR, QF : out std_logic;
    PAD : inout std_logic) ;
end DDR_BiDir_HSTL_I_LowEnb;
architecture DEF_ARCH of DDR_BiDir_HSTL_I_LowEnb is
  component INV
   port(A : in std_logic := 'U'; Y : out std_logic) ;
  end component;
  component DDR_OUT
   port(DR, DF, CLK, CLR : in std_logic := 'U'; Q : out std_logic);
  end component;
  component DDR_REG
    port(D, CLK, CLR : in std_logic := 'U'; QR, QF : out std_logic) ;
  end component;
  component BIBUF_HSTL_I
   port(PAD : inout std_logic := 'U'; D, E : in std_logic := 'U'; Y : out std_logic) ;
  end component;
signal TrienAux, D, Q : std_logic ;
begin
```

```
Inv_Tri : INV
port map(A => Trien, Y => TrienAux);
DDR_OUT_0_inst : DDR_OUT
port map(DR => DataR, DF => DataF, CLK => CLK, CLR => CLR, Q => Q);
DDR_REG_0_inst : DDR_REG
port map(D => D, CLK => CLK, CLR => CLR, QR => QR, QF => QF);
BIBUF_HSTL_I_0_inst : BIBUF_HSTL_I
port map(PAD => PAD, D => Q, E => TrienAux, Y => D);
```

end DEF_ARCH;

10 – Programming Flash Devices

Introduction

This document provides an overview of the various programming options available for the Microsemi flash families. The electronic version of this document includes active links to all programming resources, which are available at http://www.microsemi.com/soc/products/hardware/default.aspx. For Microsemi antifuse devices, refer to the *Programming Antifuse Devices* document.

Summary of Programming Support

FlashPro4 and FlashPro3 are high-performance in-system programming (ISP) tools targeted at the latest generation of low power flash devices offered by the SmartFusion,[®] Fusion, IGLOO,[®] and ProASIC[®]3 families, including ARM-enabled devices. FlashPro4 and FlashPro3 offer extremely high performance through the use of USB 2.0, are high-speed compliant for full use of the 480 Mbps bandwidth, and can program ProASIC3 devices in under 30 seconds. Powered exclusively via USB, FlashPro4 and FlashPro3 provide a VPUMP voltage of 3.3 V for programming these devices.

FlashPro4 replaced FlashPro3 in 2010. FlashPro4 supports SmartFusion, Fusion, ProASIC3, and IGLOO devices as well as future generation flash devices. FlashPro4 also adds 1.2 V programming for IGLOO nano V2 devices. FlashPro4 is compatible with FlashPro3; however it adds a programming mode (PROG_MODE) signal to the previously unused pin 4 of the JTAG connector. The PROG_MODE goes high during programming and can be used to turn on a 1.5 V external supply for those devices that require 1.5 V for programming. If both FlashPro3 and FlashPro4 programmers are used for programming the same boards, pin 4 of the JTAG connector must not be connected to anything on the board because FlashPro4 uses pin 4 for PROG_MODE.



Figure 10-1 • FlashPro Programming Setup

Security in Low Power Flash Devices

Figure 11-15 • Programming Fusion Security Settings Only

- 2. Choose the desired security level setting and enter the key(s).
 - The High security level employs FlashLock Pass Key with AES Key protection.
 - The Medium security level employs FlashLock Pass Key protection only.

Figure 11-16 • High Security Level to Implement FlashLock Pass Key and AES Key Protection

Core Voltage Switching Circuit for IGLOO and ProASIC3L In-System Programming

Microsemi's Flash Families Support Voltage Switching Circuit

The flash FPGAs listed in Table 13-1 support the voltage switching circuit feature and the functions described in this document.

Table 13-1 • Flash-Based FPGAs Supporting Voltage Switching Circuit

Series	Family [*]	Description
IGLOO	IGLOO	Ultra-low power 1.2 V to 1.5 V FPGAs with Flash*Freeze technology
	IGLOOe	Higher density IGLOO FPGAs with six PLLs and additional I/O standards
	IGLOO nano	The industry's lowest-power, smallest-size solution
	IGLOO PLUS	IGLOO FPGAs with enhanced I/O capabilities
ProASIC3	ProASIC3L	ProASIC3 FPGAs supporting 1.2 V to 1.5 V with Flash*Freeze technology
	RT ProASIC3	Radiation-tolerant RT3PE600L and RT3PE3000L
	Military ProASIC3/EL	Military temperature A3PE600L, A3P1000, and A3PE3000L

Note: *The device names link to the appropriate datasheet, including product brief, DC and switching characteristics, and packaging information.

IGLOO Terminology

In documentation, the terms IGLOO series and IGLOO devices refer to all of the IGLOO devices as listed in Table 13-1. 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 13-1. 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.*

16 – UJTAG Applications in Microsemi's Low Power Flash Devices

Introduction

In Fusion, IGLOO, and ProASIC3 devices, there is bidirectional access from the JTAG port to the core VersaTiles during normal operation of the device (Figure 16-1). User JTAG (UJTAG) is the ability for the design to use the JTAG ports for access to the device for updates, etc. While regular JTAG is used, the UJTAG tiles, located at the southeast area of the die, are directly connected to the JTAG Test Access Port (TAP) Controller in normal operating mode. As a result, all the functional blocks of the device, such as Clock Conditioning Circuits (CCCs) with PLLs, SRAM blocks, embedded FlashROM, flash memory blocks, and I/O tiles, can be reached via the JTAG ports. The UJTAG functionality is available by instantiating the UJTAG macro directly in the source code of a design. Access to the FPGA core VersaTiles from the JTAG ports enables users to implement different applications using the TAP Controller (JTAG port). This document introduces the UJTAG tile functionality and discusses a few application examples. However, the possible applications are not limited to what is presented in this document. UJTAG can serve different purposes in many designs as an elementary or auxiliary part of the design. For detailed usage information, refer to the "Boundary Scan in Low Power Flash Devices" section on page 291.



Figure 16-1 • Block Diagram of Using UJTAG to Read FlashROM Contents

UJTAG Applications in Microsemi's Low Power Flash Devices



Figure 16-3 • Connectivity Method of UJTAG Macro

UJTAG Operation

There are a few basic functions of the UJTAG macro that users must understand before designing with it. The most important fundamental concept of the UJTAG design is its connection with the TAP Controller state machine.

TAP Controller State Machine

The 16 states of the TAP Controller state machine are shown in Figure 16-4 on page 301. The 1s and 0s, shown adjacent to the state transitions, represent the TMS values that must be present at the time of a rising TCK edge for a state transition to occur. In the states that include the letters "IR," the instruction register operates; in the states that contain the letters "DR," the test data register operates. The TAP Controller receives two control inputs, TMS and TCK, and generates control and clock signals for the rest of the test logic.

On power-up (or the assertion of TRST), the TAP Controller enters the Test-Logic-Reset state. To reset the controller from any other state, TMS must be held HIGH for at least five TCK cycles. After reset, the TAP state changes at the rising edge of TCK, based on the value of TMS.

Brownout Voltage

Brownout is a condition in which the voltage supplies are lower than normal, causing the device to malfunction as a result of insufficient power. In general, Microsemi does not guarantee the functionality of the design inside the flash FPGA if voltage supplies are below their minimum recommended operating condition. Microsemi has performed measurements to characterize the brownout levels of FPGA power supplies. Refer to Table 17-3 for device-specific brownout deactivation levels. For the purpose of characterization, a direct path from the device input to output is monitored while voltage supplies are lowered gradually. The brownout point is defined as the voltage level at which the output stops following the input. Characterization tests performed on several IGLOO, ProASIC3L, and ProASIC3 devices in typical operating conditions showed the brownout voltage levels to be within the specification.

During device power-down, the device I/Os become tristated once the first supply in the power-down sequence drops below its brownout deactivation voltage.

Table 17-3 • Brownout Deactivation Levels for VCC and VCCI

Devices	VCC Brownout Deactivation Level (V)	VCCI Brownout Deactivation Level (V)
ProASIC3, ProASIC3 nano, IGLOO, IGLOO nano, IGLOO PLUS and ProASIC3L devices running at VCC = 1.5 V	0.75 V ± 0.25 V	0.8 V ± 0.3 V
IGLOO, IGLOO nano, IGLOO PLUS, and ProASIC3L devices running at VCC = 1.2 V	0.75 V ± 0.2 V	0.8 V ± 0.15 V

PLL Behavior at Brownout Condition

When PLL power supply voltage and/or V_{CC} levels drop below the V_{CC} brownout levels mentioned above for 1.5 V and 1.2 V devices, the PLL output lock signal goes LOW and/or the output clock is lost. The following sections explain PLL behavior during and after the brownout condition.

VCCPLL and VCC Tied Together

In this condition, both VCC and VCCPLL drop below the 0.75 V (\pm 0.25 V or \pm 0.2 V) brownout level. During the brownout recovery, once VCCPLL and VCC reach the activation point (0.85 \pm 0.25 V or \pm 0.2 V) again, the PLL output lock signal may still remain LOW with the PLL output clock signal toggling. If this condition occurs, there are two ways to recover the PLL output lock signal:

- 1. Cycle the power supplies of the PLL (power off and on) by using the PLL POWERDOWN signal.
- 2. Turn off the input reference clock to the PLL and then turn it back on.

Only VCCPLL Is at Brownout

In this case, only VCCPLL drops below the 0.75 V (\pm 0.25 V or \pm 0.2 V) brownout level and the VCC supply remains at nominal recommended operating voltage (1.5 V \pm 0.075 V for 1.5 V devices and 1.2 V \pm 0.06 V for 1.2 V devices). In this condition, the PLL behavior after brownout recovery is similar to initial power-up condition, and the PLL will regain lock automatically after VCCPLL is ramped up above the activation level (0.85 \pm 0.25 V or \pm 0.2 V). No intervention is necessary in this case.

Only VCC Is at Brownout

In this condition, VCC drops below the 0.75 V (\pm 0.25 V or \pm 0.2 V) brownout level and VCCPLL remains at nominal recommended operating voltage (1.5 V \pm 0.075 V for 1.5 V devices and 1.2 V \pm 0.06 V for 1.2 V devices). During the brownout recovery, once VCC reaches the activation point again (0.85 \pm 0.25 V or \pm 0.2 V), the PLL output lock signal may still remain LOW with the PLL output clock signal toggling. If this condition occurs, there are two ways to recover the PLL output lock signal:

- 1. Cycle the power supplies of the PLL (power off and on) by using the PLL POWERDOWN signal.
- 2. Turn off the input reference clock to the PLL and then turn it back on.

It is important to note that Microsemi recommends using a monotonic power supply or voltage regulator to ensure proper power-up behavior.