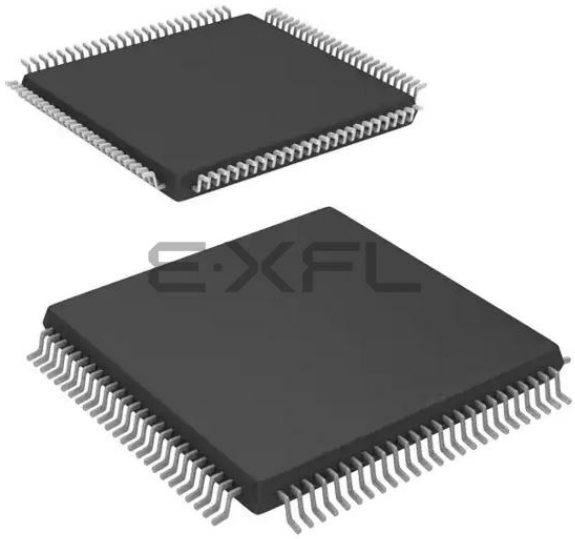


Welcome to E-XFL.COM

## Understanding Embedded - FPGAs (Field Programmable Gate Array)

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

## Applications of Embedded - FPGAs

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

| Details | |
|---|---|
| Product Status | Obsolete |
| Number of LABs/CLBs | - |
| Number of Logic Elements/Cells | - |
| Total RAM Bits | 36864 |
| Number of I/O | 71 |
| Number of Gates | 125000 |
| Voltage - Supply | 1.425V ~ 1.575V |
| Mounting Type | Surface Mount |
| Operating Temperature | -20°C ~ 85°C (TJ) |
| Package / Case | 100-TQFP |
| Supplier Device Package | 100-VQFP (14x14) |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/a3pn125-z1vq100 |

# FPGA Array Architecture Support

The flash FPGAs listed in Table 1-1 support the architecture features described in this document.

***Table 1-1* • Flash-Based FPGAs**

| Series | Family* | Description |
|---|---|---|
| IGLOO® | IGLOO | Ultra-low power 1.2 V to 1.5 V FPGAs with Flash*Freeze technology |
| | IGLOOe | Higher density IGLOO FPGAs with six PLLs and additional I/O standards |
| | IGLOO nano | The industry's lowest-power, smallest-size solution |
| | IGLOO PLUS | IGLOO FPGAs with enhanced I/O capabilities |
| ProASIC®3 | ProASIC3 | Low power, high-performance 1.5 V FPGAs |
| | ProASIC3E | Higher density ProASIC3 FPGAs with six PLLs and additional I/O standards |
| | ProASIC3 nano | Lowest-cost solution with enhanced I/O capabilities |
| | ProASIC3L | ProASIC3 FPGAs supporting 1.2 V to 1.5 V with Flash*Freeze technology |
| | RT ProASIC3 | Radiation-tolerant RT3PE600L and RT3PE3000L |
| | Military ProASIC3/EL | Military temperature A3PE600L, A3P1000, and A3PE3000L |
| | Automotive ProASIC3 | ProASIC3 FPGAs qualified for automotive applications |
| Fusion | Fusion | Mixed signal FPGA integrating ProASIC3 FPGA fabric, programmable analog block, support for ARM® Cortex™-M1 soft processors, and flash memory into a monolithic device |

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

## *IGLOO Terminology*

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

# Core Architecture

## *VersaTile*

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

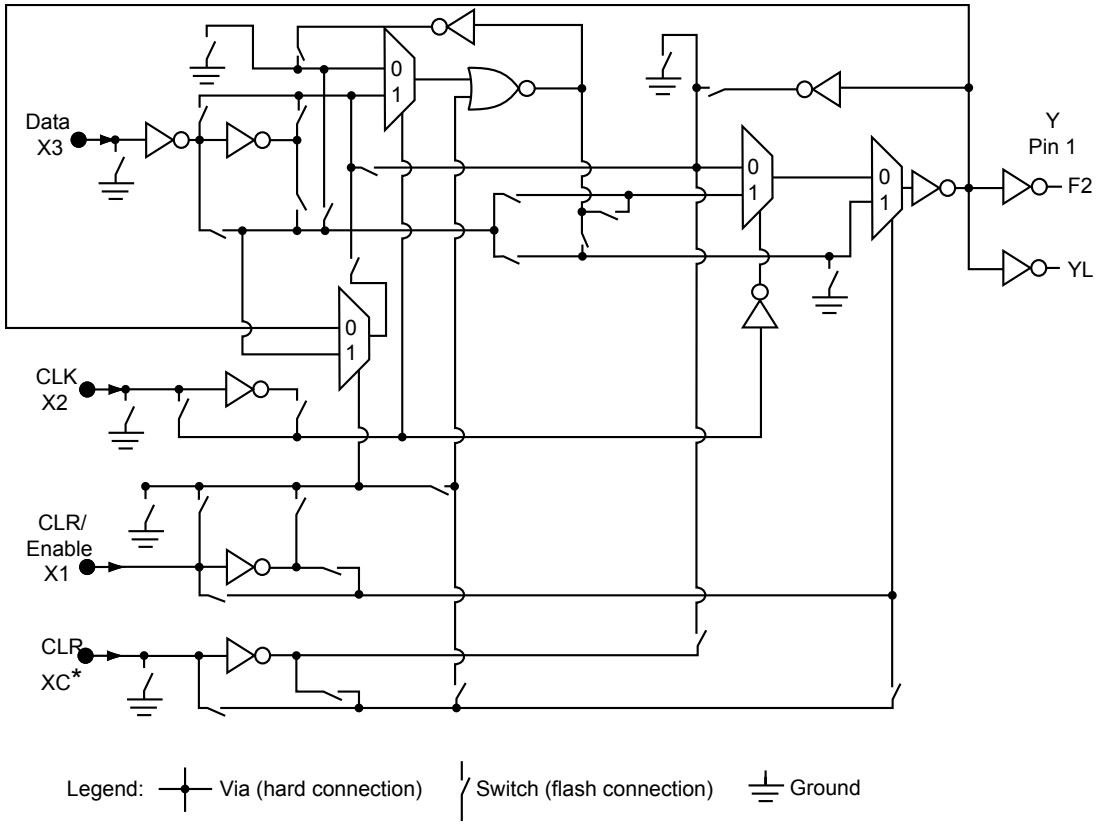
As illustrated in Figure 1-8, there are four inputs in a logic VersaTile cell, and each VersaTile can be configured using the appropriate flash switch connections:

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

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

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

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




* *This input can only be connected to the global clock distribution network.*

***Figure 1-8 •*** **Low Power Flash Device Core VersaTile**

# 2 – Low Power Modes in ProASIC3/E and ProASIC3 nano FPGAs

## Introduction

The demand for low power systems and semiconductors, combined with the strong growth observed for value-based FPGAs, is driving growing demand for low power FPGAs. For portable and battery-operated applications, power consumption has always been the greatest challenge. The battery life of a system and on-board devices has a direct impact on the success of the product. As a result, FPGAs used in these applications should meet low power consumption requirements.

ProASIC®3/E and ProASIC3 nano FPGAs offer low power consumption capability inherited from their nonvolatile and live-at-power-up (LAPU) flash technology. This application note describes the power consumption and how to use different power saving modes to further reduce power consumption for power-conscious electronics design.

## Power Consumption Overview

In evaluating the power consumption of FPGA technologies, it is important to consider it from a system point of view. Generally, the overall power consumption should be based on static, dynamic, inrush, and configuration power. Few FPGAs implement ways to reduce static power consumption utilizing sleep modes.

SRAM-based FPGAs use volatile memory for their configuration, so the device must be reconfigured after each power-up cycle. Moreover, during this initialization state, the logic could be in an indeterminate state, which might cause inrush current and power spikes. More complex power supplies are required to eliminate potential system power-up failures, resulting in higher costs. For portable electronics requiring frequent power-up and -down cycles, this directly affects battery life, requiring more frequent recharging or replacement.

$$\text{SRAM-Based FPGA Total Power Consumption} = P_{static} + P_{dynamic} + P_{inrush} + P_{config} \quad \textit{EQ 1}$$

$$\text{ProASIC3/E Total Power Consumption} = P_{static} + P_{dynamic} \quad \textit{EQ 2}$$

Unlike SRAM-based FPGAs, Microsemi flash-based FPGAs are nonvolatile and do not require power-up configuration. Additionally, Microsemi nonvolatile flash FPGAs are live at power-up and do not require additional support components. Total power consumption is reduced as the inrush current and configuration power components are eliminated.

Note that the static power component can be reduced in flash FPGAs (such as the ProASIC3/E devices) by entering User Low Static mode or Sleep mode. This leads to an extremely low static power component contribution to the total system power consumption.

The following sections describe the usage of Static (Idle) mode to reduce the power component, User Low Static mode to reduce the static power component, and Sleep mode and Shutdown mode to achieve a range of power consumption when the FPGA or system is idle. Table 2-1 on page 22 summarizes the different low power modes offered by ProASIC3/E devices.

## CLKDLY Macro Usage

When a CLKDLY macro is used in a CCC location, the programmable delay element is used to allow the clock delays to go to the global network. In addition, the user can bypass the PLL in a CCC location integrated with a PLL, but use the programmable delay that is associated with the global network by instantiating the CLKDLY macro. The same is true when using programmable delay elements in a CCC location with no PLLs (the user needs to instantiate the CLKDLY macro). There is no difference between the programmable delay elements used for the PLL and the CLKDLY macro. The CCC will be configured to use the programmable delay elements in accordance with the macro instantiated by the user.

As an example, if the PLL is not used in a particular CCC location, the designer is free to specify up to three CLKDLY macros in the CCC, each of which can have its own input frequency and delay adjustment options. If the PLL core is used, assuming output to only one global clock network, the other two global clock networks are free to be used by either connecting directly from the global inputs or connecting from one or two CLKDLY macros for programmable delay.

The programmable delay elements are shown in the block diagram of the PLL block shown in Figure 4-6 on page 71. Note that any CCC locations with no PLL present contain only the programmable delay blocks going to the global networks (labeled "Programmable Delay Type 2"). Refer to the "Clock Delay Adjustment" section on page 86 for a description of the programmable delay types used for the PLL. Also refer to Table 4-14 on page 94 for Programmable Delay Type 1 step delay values, and Table 4-15 on page 94 for Programmable Delay Type 2 step delay values. CCC locations with a PLL present can be configured to utilize only the programmable delay blocks (Programmable Delay Type 2) going to the global networks A, B, and C.

Global network A can be configured to use only the programmable delay element (bypassing the PLL) if the PLL is not used in the design. Figure 4-6 on page 71 shows a block diagram of the PLL, where the programmable delay elements are used for the global networks (Programmable Delay Type 2).

Table 4-9 to Table 4-15 on page 94 provide descriptions of the configuration data for the configuration bits.

***Table 4-9* • Input Clock Divider, FINDIV[6:0] (/n)**

| FINDIV<6:0> State | Divisor | New Frequency Factor |
|---|---|---|
| 0 | 1 | 1.00000 |
| 1 | 2 | 0.50000 |
| ⋮ | ⋮ | ⋮ |
| 127 | 128 | 0.0078125 |

***Table 4-10* • Feedback Clock Divider, FBDIV[6:0] (/m)**

| FBDIV<6:0> State | Divisor | New Frequency Factor |
|---|---|---|
| 0 | 1 | 1 |
| 1 | 2 | 2 |
| ⋮ | ⋮ | ⋮ |
| 127 | 128 | 128 |

***Table 4-11* • Output Frequency Dividers**
***A* Output Divider, OADIV <4:0> (/u);**
***B* Output Divider, OBDIV <4:0> (/v);**
***C* Output Divider, OCDIV <4:0> (/w)**

| OADIV<4:0>; OBDIV<4:0>; CDIV<4:0> State | Divisor | New Frequency Factor |
|---|---|---|
| 0 | 1 | 1.00000 |
| 1 | 2 | 0.50000 |
| ⋮ | ⋮ | ⋮ |
| 31 | 32 | 0.03125 |

***Table 4-12* • MUXA, MUXB, MUXC**

| OAMUX<2:0>; OBMUX<2:0>; OCMUX<2:0> State | MUX Input Selected |
|---|---|
| 0 | None. Six-input MUX and PLL are bypassed.<br>Clock passes only through global MUX and goes directly into HC ribs. |
| 1 | Not available |
| 2 | PLL feedback delay line output |
| 3 | Not used |
| 4 | PLL VCO 0° phase shift |
| 5 | PLL VCO 270° phase shift |
| 6 | PLL VCO 180° phase shift |
| 7 | PLL VCO 90° phase shift |

***Table 4-13* • 2-Bit Feedback MUX**

| FBSEL<1:0> State | MUX Input Selected |
|---|---|
| 0 | Ground. Used for power-down mode in power-down logic block. |
| 1 | PLL VCO 0° phase shift |
| 2 | PLL delayed VCO 0° phase shift |
| 3 | N/A |

***Table 4-14* • Programmable Delay Selection for Feedback Delay and Secondary Core Output Delays**

| FBDLY<4:0>; DLYYB<4:0>; DLYYC<4:0> State | Delay Value |
|---|---|
| 0 | Typical delay = 600 ps |
| 1 | Typical delay = 760 ps |
| 2 | Typical delay = 920 ps |
| ⋮ | ⋮ |
| 31 | Typical delay = 5.56 ns |

***Table 4-15* • Programmable Delay Selection for Global Clock Output Delays**

| DLYGLA<4:0>; DLYGLB<4:0>; DLYGLC<4:0> State | Delay Value |
|---|---|
| 0 | Typical delay = 225 ps |
| 1 | Typical delay = 760 ps |
| 2 | Typical delay = 920 ps |
| ⋮ | ⋮ |
| 31 | Typical delay = 5.56 ns |

***Table 4-16* • Fusion Dynamic CCC Clock Source Selection**

| RXASEL | DYNASEL | Source of CLKA |
|---|---|---|
| 1 | 0 | RC Oscillator |
| 1 | 1 | Crystal Oscillator |
| **RXBSEL** | **DYNBSEL** | **Source of CLKB** |
| 1 | 0 | RC Oscillator |
| 1 | 1 | Crystal Oscillator |
| **RXBSEL** | **DYNCSEL** | **Source of CLKC** |
| 1 | 0 | RC Oscillator |
| 1 | 1 | Crystal Oscillator |

***Table 4-17* • Fusion Dynamic CCC NGMUX Configuration**

| GLMUXCFG<1:0> | NGMUX Select Signal | Supported Input Clocks to NGMUX |
|---|---|---|
| 00 | 0 | GLA |
| | 1 | GLC |
| 01 | 0 | GLA |
| | 1 | GLINT |
| 10 | 0 | GLC |
| | 1 | GLINT |

# 8 – I/O Software Control in Low Power Flash Devices

Fusion, IGLOO, and ProASIC3 I/Os provide more design flexibility, allowing the user to control specific features by enabling certain I/O standards. Some features are selectable only for certain I/O standards, whereas others are available for all I/O standards. For example, slew control is not supported by differential I/O standards. Conversely, I/O register combining is supported by all I/O standards. For detailed information about which I/O standards and features are available on each device and each I/O type, refer to the I/O Structures section of the handbook for the device you are using.

Figure 8-1 shows the various points in the software design flow where a user can provide input or control of the I/O selection and parameters. A detailed description is provided throughout this document.

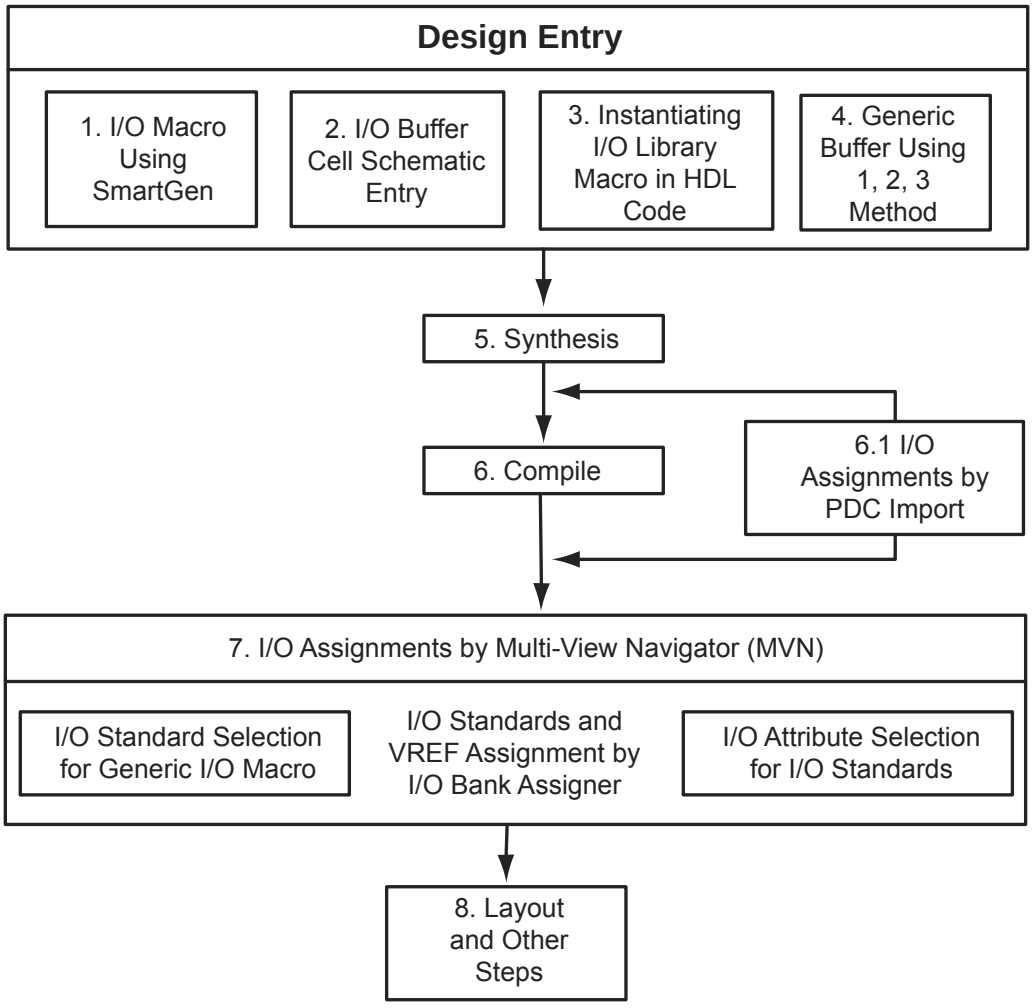



***Figure 8-1* • User I/O Assignment Flow Chart**

### Rules for the DDR I/O Function

- The fanout between an I/O pin (D or Y) and a DDR (DDR_REG or DDR_OUT) macro must be equal to one for the combining to happen on that pin.
- If a DDR_REG macro and a DDR_OUT macro are combined on the same bidirectional I/O, they must share the same clear signal.
- Registers will not be combined in an I/O in the presence of DDR combining on the same I/O.

## Using the I/O Buffer Schematic Cell

Libero SoC software includes the ViewDraw schematic entry tool. Using ViewDraw, the user can insert any supported I/O buffer cell in the top-level schematic. Figure 8-5 shows a top-level schematic with different I/O buffer cells. When synthesized, the netlist will contain the same I/O macro.

***Figure 8-5* • I/O Buffer Schematic Cell Usage**

## Automatically Assigning Technologies to I/O Banks

The I/O Bank Assigner (IOBA) tool runs automatically when you run Layout. You can also use this tool from within the MultiView Navigator (Figure 8-17). The IOBA tool automatically assigns technologies and VREF pins (if required) to every I/O bank that does not currently have any technologies assigned to it. This tool is available when at least one I/O bank is unassigned.

To automatically assign technologies to I/O banks, choose I/O Bank Assigner from the **Tools** menu (or click the I/O Bank Assigner's toolbar button, shown in Figure 8-16).

***Figure 8-16 •*** **I/O Bank Assigner's Toolbar Button**

Messages will appear in the Output window informing you when the automatic I/O bank assignment begins and ends. If the assignment is successful, the message "I/O Bank Assigner completed successfully" appears in the Output window, as shown in Figure 8-17.

***Figure 8-17 •*** **I/O Bank Assigner Displays Messages in Output Window**

# List of Changes

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

| Date | Changes | Page |
|---|---|---|
| August 2012 | The notes in Table 8-2 • Designer State (resulting from I/O attribute modification) were revised to clarify which device families support programmable input delay (SAR 39666). | 187 |
| June 2011 | Figure 8-2 • SmartGen Catalog was updated (SAR 24310). Figure 8-3 • Expanded I/O Section and the step associated with it were deleted to reflect changes in the software. | 188 |
| | The following rule was added to the "VREF Rules for the Implementation of Voltage-Referenced I/O Standards" section:<br>Only minibanks that contain input or bidirectional I/Os require a VREF. A VREF is not needed for minibanks composed of output or tristated I/Os (SAR 24310). | 199 |
| July 2010 | Notes were added where appropriate to point out that IGLOO nano and ProASIC3 nano devices do not support differential inputs (SAR 21449). | N/A |
| v1.4 (December 2008) | IGLOO nano and ProASIC3 nano devices were added to Table 8-1 • Flash-Based FPGAs. | 186 |
| | The notes for Table 8-2 • Designer State (resulting from I/O attribute modification) were revised to indicate that skew control and input delay do not apply to nano devices. | 187 |
| v1.3 (October 2008) | The "Flash FPGAs I/O Support" section was revised to include new families and make the information more concise. | 186 |
| v1.2 (June 2008) | The following changes were made to the family descriptions in Table 8-1 • Flash-Based FPGAs:<br>• ProASIC3L was updated to include 1.5 V.<br>• The number of PLLs for ProASIC3E was changed from five to six. | 186 |
| v1.1 (March 2008) | This document was previously part of the *I/O Structures in IGLOO and ProASIC3 Devices* document. The content was separated and made into a new document. | N/A |
| | Table 8-2 • Designer State (resulting from I/O attribute modification) was updated to include note 2 for IGLOO PLUS. | 187 |

# 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.

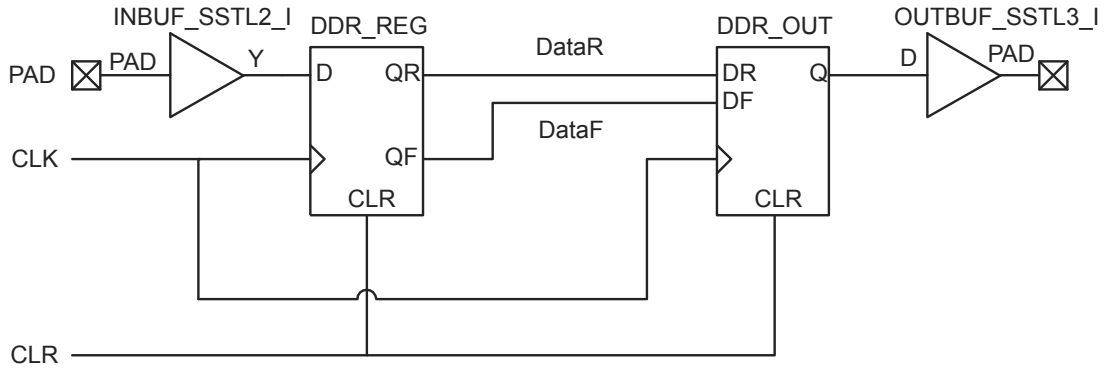



***Figure 9-1* • DDR Support in Low Power Flash Devices**

***Table 9-2* • DDR I/O Options (continued)**

| DDR Register Type | I/O Type | I/O Standard | Sub-Options | Comments |
|---|---|---|---|---|
| Transmit Register (continued) | Tristate Buffer | Normal | Enable Polarity | Low/high (low default) |
| | | LVTTL | Output Drive | 2, 4, 6, 8, 12,16, 24, 36 mA (8 mA default) |
| | | | Slew Rate | Low/high (high default) |
| | | | Enable Polarity | Low/high (low default) |
| | | | Pull-Up/-Down | None (default) |
| | | LVCMOS | Voltage | 1.5 V, 1.8 V, 2.5 V, 5 V (1.5 V default) |
| | | | Output Drive | 2, 4, 6, 8, 12, 16, 24, 36 mA (8 mA default) |
| | | | Slew Rate | Low/high (high default) |
| | | | Enable Polarity | Low/high (low default) |
| | | | Pull-Up/-Down | None (default) |
| | | PCI/PCI-X | Enable Polarity | Low/high (low default) |
| | | GTL/GTL+ | Voltage | 1.8 V, 2.5 V, 3.3 V (3.3 V default) |
| | | | Enable Polarity | Low/high (low default) |
| | | HSTL | Class | I / II (I default) |
| | | | Enable Polarity | Low/high (low default) |
| | | SSTL2/SSTL3 | Class | I / II (I default) |
| | | | Enable Polarity | Low/high (low default) |
| | Bidirectional Buffer | Normal | Enable Polarity | Low/high (low default) |
| | | LVTTL | Output Drive | 2, 4, 6, 8, 12, 16, 24, 36 mA (8 mA default) |
| | | | Slew Rate | Low/high (high default) |
| | | | Enable Polarity | Low/high (low default) |
| | | | Pull-Up/-Down | None (default) |
| | | LVCMOS | Voltage | 1.5 V, 1.8 V, 2.5 V, 5 V (1.5 V default) |
| | | | Enable Polarity | Low/high (low default) |
| | | | Pull-Up | None (default) |
| | | PCI/PCI-X | None | |
| | | | Enable Polarity | Low/high (low default) |
| | | GTL/GTL+ | Voltage | 1.8 V, 2.5 V, 3.3 V (3.3 V default) |
| | | | Enable Polarity | Low/high (low default) |
| | | HSTL | Class | I / II (I default) |
| | | | Enable Polarity | Low/high (low default) |
| | | SSTL2/SSTL3 | Class | I / II (I default) |
| | | | Enable Polarity | Low/high (low default) |

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

## DDR Input Register

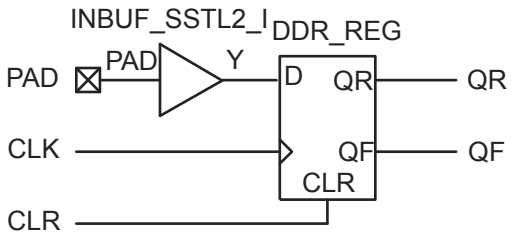



***Figure 9-5* • DDR Input Register (SSTL2 Class I)**

The corresponding structural representations, as generated by SmartGen, are shown below:

### *Verilog*

```
module DDR_InBuf_SSTL2_I(PAD,CLR,CLK,QR,QF);

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

wire Y;

  INBUF_SSTL2_I INBUF_SSTL2_I_0_inst(.PAD(PAD),.Y(Y));
  DDR_REG DDR_REG_0_inst(.D(Y),.CLK(CLK),.CLR(CLR),.QR(QR),.QF(QF));

endmodule
```

### *VHDL*

```
library ieee;
use ieee.std_logic_1164.all;
--The correct library will be inserted automatically by SmartGen
library proasic3; use proasic3.all;
--library fusion; use fusion.all;
--library igloo; use igloo.all;

entity DDR_InBuf_SSTL2_I is
  port(PAD, CLR, CLK : in std_logic;  QR, QF : out std_logic) ;
end DDR_InBuf_SSTL2_I;

architecture DEF_ARCH of  DDR_InBuf_SSTL2_I is

  component INBUF_SSTL2_I
    port(PAD : in std_logic := 'U'; Y : out std_logic) ;
  end component;

  component DDR_REG
    port(D, CLK, CLR : in std_logic := 'U'; QR, QF : out std_logic) ;
  end component;

signal Y : std_logic ;

begin

  INBUF_SSTL2_I_0_inst : INBUF_SSTL2_I
  port map(PAD => PAD, Y => Y);
  DDR_REG_0_inst : DDR_REG
  port map(D => Y, CLK => CLK, CLR => CLR, QR => QR, QF => QF);

end DEF_ARCH;
```

### 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.<br>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:<br>• Flash technology that does not require an external bitstream to program the device<br>• FlashLock Pass Key that secures device content by locking the security settings and preventing access to the device as defined by the user<br>• 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.

# 12 – In-System Programming (ISP) of Microsemi's Low Power Flash Devices Using FlashPro4/3/3X

## Introduction

Microsemi's low power flash devices are all in-system programmable. This document describes the general requirements for programming a device and specific requirements for the FlashPro4/3/3X programmers[1].

IGLOO, ProASIC3, SmartFusion, and Fusion devices offer a low power, single-chip, live-at-power-up solution with the ASIC advantages of security and low unit cost through nonvolatile flash technology. Each device contains 1 kbit of on-chip, user-accessible, nonvolatile FlashROM. The FlashROM can be used in diverse system applications such as Internet Protocol (IP) addressing, user system preference storage, device serialization, or subscription-based business models. IGLOO, ProASIC3, SmartFusion, and Fusion devices offer the best in-system programming (ISP) solution, FlashLock® security features, and AES-decryption-based ISP.

## ISP Architecture

Low power flash devices support ISP via JTAG and require a single VPUMP voltage of 3.3 V during programming. In addition, programming via a microcontroller in a target system is also supported.

Refer to the "Microprocessor Programming of Microsemi's Low Power Flash Devices" chapter of an appropriate FPGA fabric user's guide.

Family-specific support:

- ProASIC3, ProASIC3E, SmartFusion, and Fusion devices support ISP.
- ProASIC3L devices operate using a 1.2 V core voltage; however, programming can be done only at 1.5 V. Voltage switching is required in-system to switch from a 1.2 V core to 1.5 V core for programming.
- IGLOO and IGLOOe V5 devices can be programmed in-system when the device is using a 1.5 V supply voltage to the FPGA core.
- IGLOO nano V2 devices can be programmed at 1.2 V core voltage (when using FlashPro4 only) or 1.5 V. IGLOO nano V5 devices are programmed with a VCC core voltage of 1.5 V. Voltage switching is required in-system to switch from a 1.2 V supply (VCC,VCCI, and VJTAG) to 1.5 V for programming. The exception is that V2 devices can be programmed at 1.2 V VCC with FlashPro4.

IGLOO devices cannot be programmed in-system when the device is in Flash*Freeze mode. The device should exit Flash*Freeze mode and be in normal operation for programming to start. Programming operations in IGLOO devices can be achieved when the device is in normal operating mode and a 1.5 V core voltage is used.

### JTAG 1532

IGLOO, ProASIC3, SmartFusion, and Fusion devices support the JTAG-based IEEE 1532 standard for ISP. To start JTAG operations, the IGLOO device must exit Flash*Freeze mode and be in normal operation before starting to send JTAG commands to the device. As part of this support, when a device is in an unprogrammed state, all user I/O pins are disabled. This is achieved by keeping the global IO_EN

1. *FlashPro4 replaced FlashPro3/3X in 2010 and is backward compatible with FlashPro3/3X as long as there is no connection to pin 4 on the JTAG header on the board. On FlashPro3/3X, there is no connection to pin 4 on the JTAG header; however, pin 4 is used for programming mode (Prog_Mode) on FlashPro4. When converting from FlashPro3/3X to FlashPro4, users should make sure that JTAG connectors on system boards do not have any connection to pin 4. FlashPro3X supports discrete TCK toggling that is needed to support non-JTAG compliant devices in the chain. This feature is included in FlashPro4.*

# Security in ARM-Enabled Low Power Flash Devices

There are slight differences between the regular flash device and the ARM-enabled flash devices, which have the M1 prefix.

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

## Cortex-M1 and Cortex-M3 Device Security

Cortex-M1–enabled and Cortex-M3 devices are shipped with the following security features:

- FPGA array enabled for AES-encrypted programming and verification
- FlashROM enabled for AES-encrypted write and verify
- Embedded Flash Memory enabled for AES encrypted write

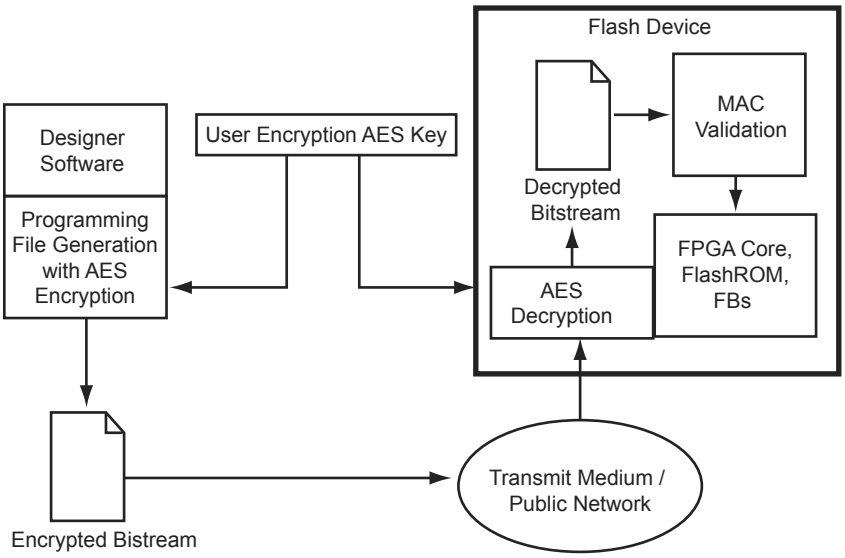



***Figure 12-1* • AES-128 Security Features**

# Boundary Scan Support in Low Power Devices

The information in this document applies to all Fusion, IGLOO, and ProASIC3 devices. For IGLOO, IGLOO PLUS, and ProASIC3L devices, the Flash*Freeze pin must be deasserted for successful boundary scan operations. Devices cannot enter JTAG mode directly from Flash*Freeze mode.

# Boundary Scan Opcodes

Low power flash devices support all mandatory IEEE 1149.1 instructions (EXTEST, SAMPLE/PRELOAD, and BYPASS) and the optional IDCODE instruction (Table 15-2).

***Table 15-2* • Boundary Scan Opcodes**

| | Hex Opcode |
|---|---|
| EXTEST | 00 |
| HIGHZ | 07 |
| USERCODE | 0E |
| SAMPLE/PRELOAD | 01 |
| IDCODE | 0F |
| CLAMP | 05 |
| BYPASS | FF |

# Boundary Scan Chain

The serial pins are used to serially connect all the boundary scan register cells in a device into a boundary scan register chain (Figure 15-2 on page 294), which starts at the TDI pin and ends at the TDO pin. The parallel ports are connected to the internal core logic I/O tile and the input, output, and control ports of an I/O buffer to capture and load data into the register to control or observe the logic state of each I/O.

Each test section is accessed through the TAP, which has five associated pins: TCK (test clock input), TDI, TDO (test data input and output), TMS (test mode selector), and TRST (test reset input). TMS, TDI, and TRST are equipped with pull-up resistors to ensure proper operation when no input data is supplied to them. These pins are dedicated for boundary scan test usage. Refer to the "JTAG Pins" section in the "Pin Descriptions and Packaging" chapter of the appropriate device datasheet for pull-up/-down recommendations for TCK and TRST pins. Pull-down recommendations are also given in Table 15-3 on page 294

***Figure 17-3* • I/O State when VCCI Is Powered before VCC**

## Power-Up to Functional Time

At power-up, device I/Os exit the tristate mode and become functional once the last voltage supply in the power-up sequence (VCCI or VCC) reaches its functional activation level. The power-up–to–functional time is the time it takes for the last supply to power up from zero to its functional level. Note that the functional level of the power supply during power-up may vary slightly within the specification at different ramp-rates. Refer to Table 17-2 for the functional level of the voltage supplies at power-up.

Typical I/O behavior during power-up–to–functional time is illustrated in Figure 17-2 on page 311 and Figure 17-3.

***Table 17-2* • Power-Up Functional Activation Levels for VCC and VCCI**

| Device | VCC Functional Activation Level (V) | VCCI Functional Activation Level (V) |
|---|---|---|
| ProASIC3, ProASIC3 nano, IGLOO, IGLOO nano, IGLOO PLUS, and ProASIC3L devices running at VCC = 1.5 V* | 0.85 V ± 0.25 V | 0.9 V ± 0.3 V |
| IGLOO, IGLOO nano, IGLOO PLUS, and ProASIC3L devices running at VCC = 1.2 V* | 0.85 V ± 0.2 V | 0.9 V ± 0.15 V |

*Note: *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.*

Microsemi's low power flash devices meet Level 0 LAPU; that is, they can be functional prior to $V_{CC}$ reaching the regulated voltage required. This important advantage distinguishes low power flash devices from their SRAM-based counterparts. SRAM-based FPGAs, due to their volatile technology, require hundreds of milliseconds after power-up to configure the design bitstream before they become functional. Refer to Figure 17-4 on page 313 and Figure 17-5 on page 314 for more information.