



Welcome to [E-XFL.COM](https://www.e-xfl.com)

### Understanding [Embedded - FPGAs \(Field Programmable Gate Array\)](#)

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

### Applications of Embedded - FPGAs

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

#### Details

|                                |   |
|--------------------------------|---|
| Product Status                 | Active  |
| Number of LABs/CLBs            | -   |
| Number of Logic Elements/Cells | -   |
| Total RAM Bits                 | 110592  |
| Number of I/O                  | 177   |
| Number of Gates                | 600000  |
| Voltage - Supply               | 1.14V ~ 1.575V  |
| Mounting Type                  | Surface Mount   |
| Operating Temperature          | -40°C ~ 100°C (TJ)  |
| Package / Case                 | 144-LBGA  |
| Supplier Device Package        | 144-FPBGA (13x13)   |
| Purchase URL                   | <a href="https://www.e-xfl.com/product-detail/microchip-technology/m1a3p600l-1fg144i">https://www.e-xfl.com/product-detail/microchip-technology/m1a3p600l-1fg144i</a> |

## Flash Families Support the Flash\*Freeze Feature

The low power flash FPGAs listed in Table 2-1 support the Flash\*Freeze feature and the functions described in this document.

**Table 2-1 • 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                            |
| 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 2-1. 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 2-1. 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*.

- There will be added skew and clock insertion delay due to the clock gating circuit. The user should analyze external setup/hold times carefully. The user should also ensure the additional skew across the clock gating filter circuit is accounted for in any paths where the launch register is driven from the filter input clock and captured by a register driven by the gated clock filter output clock.

## Power Analysis

SmartPower identifies static and dynamic power consumption problems quickly within a design. It provides a hierarchical view, allowing users to drill down and estimate the power consumption of individual components or events. SmartPower analyzes power consumption for nets, gates, I/Os, memories, clocks, cores, clock domains, power supply rails, peak power during a clock cycle, and switching transitions.

SmartPower generates detailed hierarchical reports of the dynamic power consumption of a design for easy inspection. These reports include design-level power summary, average switching activity, and ambient and junction temperature readings. Enter the target clock and data frequencies for a design, and let SmartPower perform a detailed and accurate power analysis. SmartPower supports importing files in the VCD (Value-Change Dump) format as specified in the IEEE 1364 standard. It also supports the Synopsys® Switching Activity Interchange Format (SAIF) standard. Support for these formats lets designers generate switching activity information in a variety of simulators and then import this information directly into SmartPower.

For portable or battery-operated applications, a power profile feature enables you to measure power and battery life, based on a sequence of operational modes of the design. In most portable and battery-operated applications, the system is seldom fully "on" 100 percent of the time. "On" is a combination of fully active, standby, sleep, or other functional modes. SmartPower allows users to create a power profile for a design by specifying operational modes and the percent of time the device will run in each of the modes. Power is calculated for each of the modes, and total power is calculated based on the weighted average of all modes.

SmartPower also provides an estimated battery life based on the power profile. The current capacity for a given battery is entered and used to estimate the life of the battery. The result is an accurate and realistic indication of battery life.

More information on SmartPower can be found on the Microsemi SoC Products Group website:  
<http://www.microsemi.com/soc/products/software/libero/smartpower.aspx>.

## Additional Power Conservation Techniques

IGLOO, IGLOO nano, IGLOO PLUS, ProASIC3L, and RT ProASIC3 FPGAs provide many ways to inherently conserve power; however, there are also several design techniques that can be used to reduce power on the board.

- Microsemi recommends that the designer use the minimum number of I/O banks possible and tie any unused power supplies (such as  $V_{CCPLL}$ ,  $V_{CCI}$ ,  $V_{MV}$ , and  $V_{PUMP}$ ) to ground.
- Leave unused I/O ports floating. Unused I/Os are configured by the software as follows:
  - Output buffer is disabled (with tristate value of high impedance)
  - Input buffer is disabled (with tristate value of high impedance)
- Use the lowest available voltage I/O standard, the lowest drive strength, and the slowest slew rate to reduce I/O switching contribution to power consumption.
- Advanced and pro I/O banks may consume slightly higher static current than standard and standard plus banks—avoid using advanced and pro banks whenever practical.
  - The small static power benefit obtained by avoiding advanced or pro I/O banks is usually negligible compared to the benefit of using a low power I/O standard.
- Deselect RAM blocks that are not being used.
- Only enable read and write ports on RAM blocks when they are needed.
- Gating clocks LOW offers improved static power of RAM blocks.
- Drive the FF port of RAM blocks with the Flash\_Freeze\_Enabled signal from the Flash\*Freeze management IP.
- Drive inputs to the full voltage level so that all transistors are turned on or off completely.

## Global Resource Support in Flash-Based Devices

The flash FPGAs listed in Table 3-1 support the global resources and the functions described in this document.

**Table 3-1 • 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 PLUS           | IGLOO FPGAs with enhanced I/O capabilities  |
|          | IGLOO nano           | The industry's lowest-power, smallest-size solution   |
| 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® 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 products as listed in Table 3-1. Where the information applies to only one product line or limited devices, these exclusions will be explicitly stated.

### **ProASIC3 Terminology**

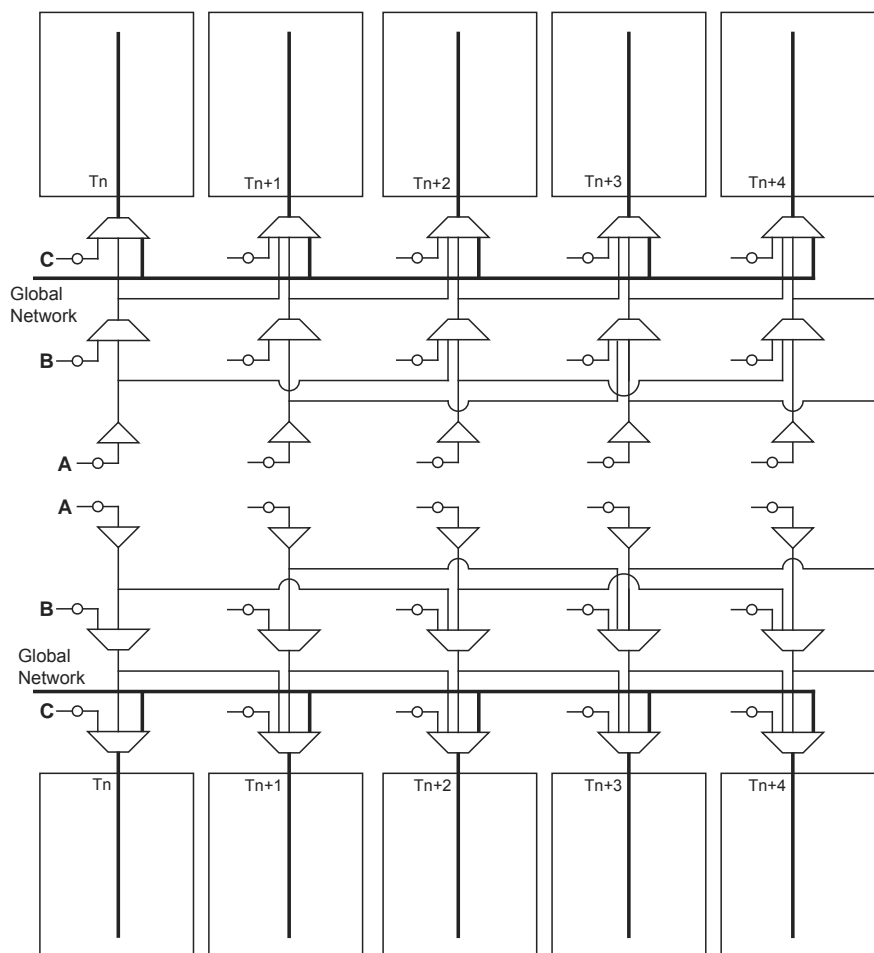
In documentation, the terms ProASIC3 series and ProASIC3 devices refer to all of the ProASIC3 devices as listed in Table 3-1. Where the information applies to only one product line or limited devices, these exclusions will be explicitly stated.

To further understand the differences between the IGLOO and ProASIC3 devices, refer to the *Industry's Lowest Power FPGAs Portfolio*.



## Spine Access

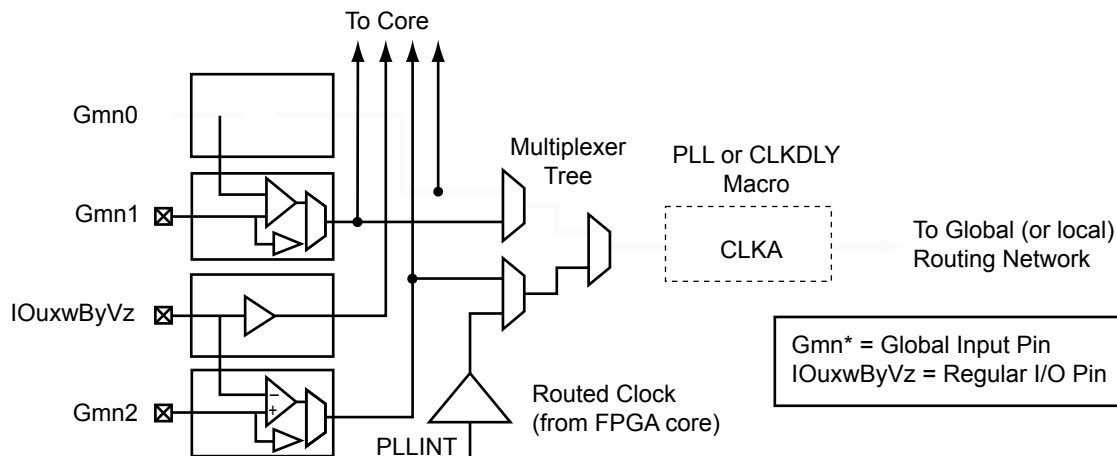
The physical location of each spine is identified by the letter T (top) or B (bottom) and an accompanying number ( $Tn$  or  $Bn$ ). The number  $n$  indicates the horizontal location of the spine; 1 refers to the first spine on the left side of the die. Since there are six chip spines in each spine tree, there are up to six spines available for each combination of T (or B) and  $n$  (for example, six  $T1$  spines). Similarly, there are three quadrant spines available for each combination of T (or B) and  $n$  (for example, four  $T1$  spines), as shown in Figure 3-7.



**Figure 3-7 • Chip Global Aggregation**

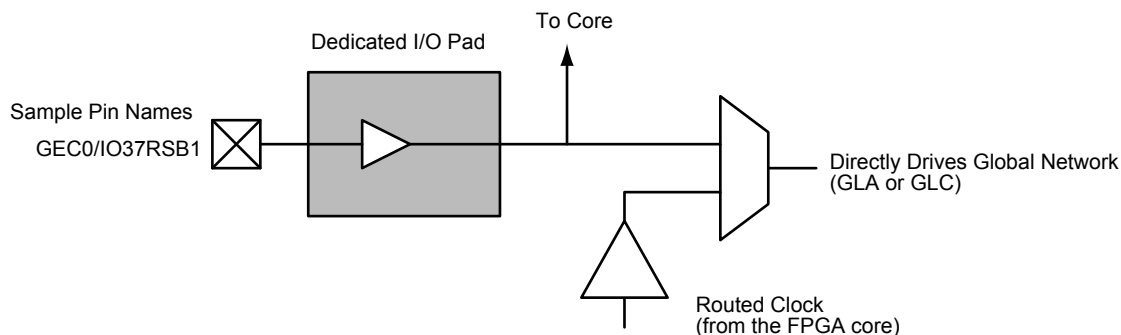
A spine is also called a local clock network, and is accessed by the dedicated global MUX architecture. These MUXes define how a particular spine is driven. Refer to Figure 3-8 on page 60 for the global MUX architecture. The MUXes for each chip global spine are located in the middle of the die. Access to the top and bottom chip global spine is available from the middle of the die. There is no control dependency between the top and bottom spines. If a top spine,  $T1$ , of a chip global network is assigned to a net,  $B1$  is not wasted and can be used by the global clock network. The signal assigned only to the top or bottom spine cannot access the middle two rows of the architecture. However, if a spine is using the top and bottom at the same time ( $T1$  and  $B1$ , for instance), the previous restriction is lifted.

The MUXes for each quadrant global spine are located in the north and south sides of the die. Access to the top and bottom quadrant global spines is available from the north and south sides of the die. Since the MUXes for quadrant spines are located in the north and south sides of the die, you should not try to drive  $T1$  and  $B1$  quadrant spines from the same signal.



Note: Fusion CCCs have additional source selections (RCOSC, XTAL).

**Figure 4-9 • Illustration of Hardwired I/O (global input pins) Usage for IGLOO and ProASIC3 devices 60 k Gates and Larger**



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

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

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

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

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

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

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

## 5 – FlashROM in Microsemi’s Low Power Flash Devices

### Introduction

The Fusion, IGLOO, and ProASIC3 families of low power flash-based devices have a dedicated nonvolatile FlashROM memory of 1,024 bits, which provides a unique feature in the FPGA market. The FlashROM can be read, modified, and written using the JTAG (or UJTAG) interface. It can be read but not modified from the FPGA core. Only low power flash devices contain on-chip user nonvolatile memory (NVM).

### Architecture of User Nonvolatile FlashROM

Low power flash devices have 1 kbit of user-accessible nonvolatile flash memory on-chip that can be read from the FPGA core fabric. The FlashROM is arranged in eight banks of 128 bits (16 bytes) during programming. The 128 bits in each bank are addressable as 16 bytes during the read-back of the FlashROM from the FPGA core. Figure 5-1 shows the FlashROM logical structure.

The FlashROM can only be programmed via the IEEE 1532 JTAG port. It cannot be programmed directly from the FPGA core. When programming, each of the eight 128-bit banks can be selectively reprogrammed. The FlashROM can only be reprogrammed on a bank boundary. Programming involves an automatic, on-chip bank erase prior to reprogramming the bank. The FlashROM supports synchronous read. The address is latched on the rising edge of the clock, and the new output data is stable after the falling edge of the same clock cycle. For more information, refer to the timing diagrams in the DC and Switching Characteristics chapter of the appropriate datasheet. The FlashROM can be read on byte boundaries. The upper three bits of the FlashROM address from the FPGA core define the bank being accessed. The lower four bits of the FlashROM address from the FPGA core define which of the 16 bytes in the bank is being accessed.

|                                  |   | Byte Number in Bank |    |    |    |    |    |   |   | 4 LSB of ADDR (READ) |   |   |   |   |   |   |   |
|----------------------------------|---|---------------------|----|----|----|----|----|---|---|----------------------|---|---|---|---|---|---|---|
|                                  |   | 15                  | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7                    | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bank Number 3 MSB of ADDR (READ) | 7 |                     |    |    |    |    |    |   |   |                      |   |   |   |   |   |   |   |
|                                  | 6 |                     |    |    |    |    |    |   |   |                      |   |   |   |   |   |   |   |
|                                  | 5 |                     |    |    |    |    |    |   |   |                      |   |   |   |   |   |   |   |
|                                  | 4 |                     |    |    |    |    |    |   |   |                      |   |   |   |   |   |   |   |
|                                  | 3 |                     |    |    |    |    |    |   |   |                      |   |   |   |   |   |   |   |
|                                  | 2 |                     |    |    |    |    |    |   |   |                      |   |   |   |   |   |   |   |
|                                  | 1 |                     |    |    |    |    |    |   |   |                      |   |   |   |   |   |   |   |
|                                  | 0 |                     |    |    |    |    |    |   |   |                      |   |   |   |   |   |   |   |

Figure 5-1 • FlashROM Architecture

## Programming and Accessing FlashROM

The FlashROM content can only be programmed via JTAG, but it can be read back selectively through the JTAG programming interface, the UJTAG interface, or via direct FPGA core addressing. The pages of the FlashROM can be made secure to prevent read-back via JTAG. In that case, read-back on these secured pages is only possible by the FPGA core fabric or via UJTAG.

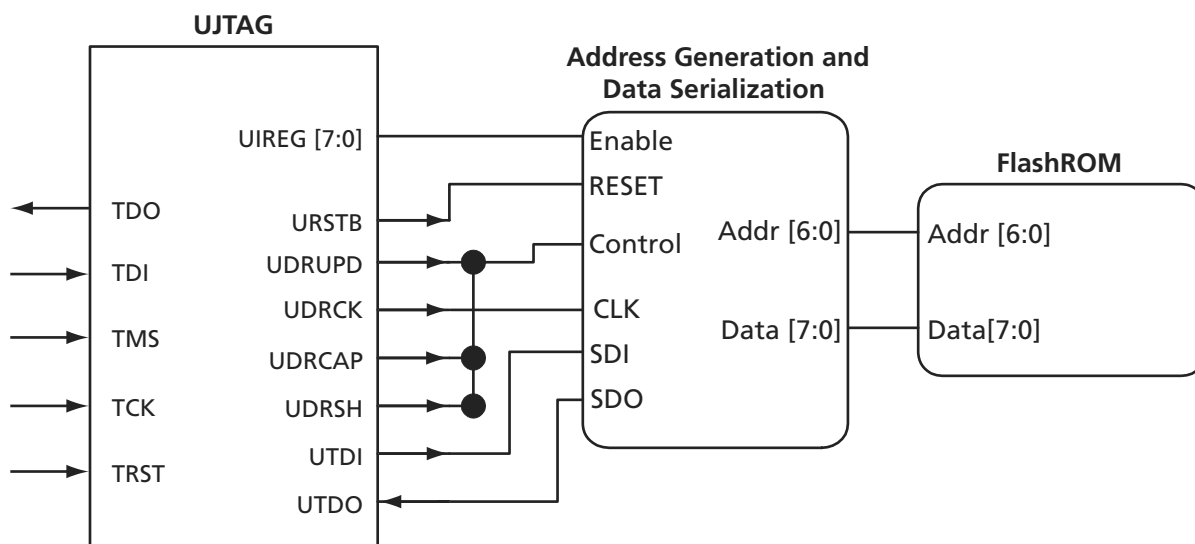
A 7-bit address from the FPGA core defines which of the eight pages (three MSBs) is being read, and which of the 16 bytes within the selected page (four LSBs) are being read. The FlashROM content can be read on a random basis; the access time is 10 ns for a device supporting commercial specifications. The FPGA core will be powered down during writing of the FlashROM content. FPGA power-down during FlashROM programming is managed on-chip, and FPGA core functionality is not available during programming of the FlashROM. Table 5-2 summarizes various FlashROM access scenarios.

**Table 5-2 • FlashROM Read/Write Capabilities by Access Mode**

| Access Mode | FlashROM Read | FlashROM Write |
|-------------|---------------|----------------|
| JTAG        | Yes           | Yes            |
| UJTAG       | Yes           | No             |
| FPGA core   | Yes           | No             |

Figure 5-6 shows the accessing of the FlashROM using the UJTAG macro. This is similar to FPGA core access, where the 7-bit address defines which of the eight pages (three MSBs) is being read and which of the 16 bytes within the selected page (four LSBs) are being read. Refer to the "UJTAG Applications in Microsemi's Low Power Flash Devices" section on page 363 for details on using the UJTAG macro to read the FlashROM.

Figure 5-7 on page 139 and Figure 5-8 on page 139 show the FlashROM access from the JTAG port. The FlashROM content can be read on a random basis. The three-bit address defines which page is being read or updated.



**Figure 5-6 • Block Diagram of Using UJTAG to Read FlashROM Contents**

## Features Supported on Every I/O

Table 7-5 lists all features supported by transmitter/receiver for single-ended and differential I/Os. Table 7-6 on page 180 lists the performance of each I/O technology.

**Table 7-5 • I/O Features**

| Feature   | Description  |
|---|--|
| All I/O   | <ul style="list-style-type: none"> <li>• High performance (Table 7-6 on page 180)</li> <li>• Electrostatic discharge (ESD) protection</li> <li>• I/O register combining option</li> </ul>  |
| Single-Ended Transmitter Features                           | <ul style="list-style-type: none"> <li>• Hot-swap: <ul style="list-style-type: none"> <li>– 30K gate devices: hot-swap in every mode</li> <li>– All other IGLOO and ProASIC3 devices: no hot-swap</li> </ul> </li> <li>• Output slew rate: 2 slew rates (except 30K gate devices)</li> <li>• Weak pull-up and pull-down resistors</li> <li>• Output drive: 3 drive strengths</li> <li>• Programmable output loading</li> <li>• Skew between output buffer enable/disable time: 2 ns delay on rising edge and 0 ns delay on falling edge (see the "Selectable Skew between Output Buffer Enable and Disable Times" section on page 199 for more information)</li> <li>• LVTTTL/LVCMOS 3.3 V outputs compatible with 5 V TTL inputs</li> </ul> |
| Single-Ended Receiver Features                              | <ul style="list-style-type: none"> <li>• 5 V–input–tolerant receiver (Table 7-12 on page 193)</li> <li>• Separate ground plane for GNDQ pin and power plane for VMV pin are used for input buffer to reduce output-induced noise.</li> </ul>   |
| Differential Receiver Features—250K through 1M Gate Devices | <ul style="list-style-type: none"> <li>• Separate ground plane for GNDQ pin and power plane for VMV pin are used for input buffer to reduce output-induced noise.</li> </ul>   |
| CMOS-Style LVDS, B-LVDS, M-LVDS, or LVPECL Transmitter      | <ul style="list-style-type: none"> <li>• Two I/Os and external resistors are used to provide a CMOS-style LVDS, DDR LVDS, B-LVDS, and M-LVDS/LVPECL transmitter solution.</li> <li>• High slew rate</li> <li>• Weak pull-up and pull-down resistors</li> <li>• Programmable output loading</li> </ul>  |

**Table 7-13 • Comparison Table for 5 V–Compliant Receiver Solutions**

| Solution | Board Components   | Speed                    | Current Limitations   |
|----------|--|--------------------------|---|
| 1        | Two resistors  | Low to High <sup>1</sup> | Limited by transmitter's drive strength   |
| 2        | Resistor and Zener 3.3 V   | Medium                   | Limited by transmitter's drive strength   |
| 3        | Bus switch   | High                     | N/A   |
| 4        | Minimum resistor value <sup>2,3,4,5</sup><br>R = 47 $\Omega$ at T <sub>J</sub> = 70°C<br>R = 150 $\Omega$ at T <sub>J</sub> = 85°C<br>R = 420 $\Omega$ at T <sub>J</sub> = 100°C | Medium                   | Maximum diode current at 100% duty cycle, signal constantly at 1<br>52.7 mA at T <sub>J</sub> = 70°C / 10-year lifetime<br>16.5 mA at T <sub>J</sub> = 85°C / 10-year lifetime<br>5.9 mA at T <sub>J</sub> = 100°C / 10-year lifetime<br>For duty cycles other than 100%, the currents can be increased by a factor of 1 / (duty cycle).<br>Example: 20% duty cycle at 70°C<br>Maximum current = (1 / 0.2) × 52.7 mA = 5 × 52.7 mA = 263.5 mA |

Notes:

1. Speed and current consumption increase as the board resistance values decrease.
2. Resistor values ensure I/O diode long-term reliability.
3. At 70°C, customers could still use 420  $\Omega$  on every I/O.
4. At 85°C, a 5 V solution on every other I/O is permitted, since the resistance is lower (150  $\Omega$ ) and the current is higher. Also, the designer can still use 420  $\Omega$  and use the solution on every I/O.
5. At 100°C, the 5 V solution on every I/O is permitted, since 420  $\Omega$  are used to limit the current to 5.9 mA.

## 5 V Output Tolerance

IGLOO and ProASIC3 I/Os must be set to 3.3 V LVTTTL or 3.3 V LVCMOS mode to reliably drive 5 V TTL receivers. It is also critical that there be NO external I/O pull-up resistor to 5 V, since this resistor would pull the I/O pad voltage beyond the 3.6 V absolute maximum value and consequently cause damage to the I/O.

When set to 3.3 V LVTTTL or 3.3 V LVCMOS mode, the I/Os can directly drive signals into 5 V TTL receivers. In fact, VOL = 0.4 V and VOH = 2.4 V in both 3.3 V LVTTTL and 3.3 V LVCMOS modes exceeds the VIL = 0.8 V and VIH = 2 V level requirements of 5 V TTL receivers. Therefore, level 1 and level 0 will be recognized correctly by 5 V TTL receivers.

## Schmitt Trigger

A Schmitt trigger is a buffer used to convert a slow or noisy input signal into a clean one before passing it to the FPGA. Using Schmitt trigger buffers guarantees a fast, noise-free input signal to the FPGA.

The Schmitt trigger is available for the LVTTTL, LVCMOS, and 3.3 V PCI I/O standards.

This feature can be implemented by using a Physical Design Constraints (PDC) command (Table 7-5 on page 179) or by selecting a check box in the I/O Attribute Editor in Designer. The check box is cleared by default.

## Simultaneously Switching Outputs (SSOs) and Printed Circuit Board Layout

Each I/O voltage bank has a separate ground and power plane for input and output circuits (VMV/GNDQ for input buffers and VCCI/GND for output buffers). This isolation is necessary to minimize simultaneous switching noise from the input and output (SSI and SSO). The switching noise (ground bounce and power bounce) is generated by the output buffers and transferred into input buffer circuits, and vice versa.

Since voltage bounce originates on the package inductance, the VMV and VCCI supplies have separate package pin assignments. For the same reason, GND and GNDQ also have separate pin assignments.

The VMV and VCCI pins must be shorted to each other on the board. Also, the GND and GNDQ pins must be shorted to each other on the board. This will prevent unwanted current draw from the power supply.

SSOs can cause signal integrity problems on adjacent signals that are not part of the SSO bus. Both inductive and capacitive coupling parasitics of bond wires inside packages and of traces on PCBs will transfer noise from SSO busses onto signals adjacent to those busses. Additionally, SSOs can produce ground bounce noise and VCCI dip noise. These two noise types are caused by rapidly changing currents through GND and VCCI package pin inductances during switching activities (EQ 2 and EQ 3).

$$\text{Ground bounce noise voltage} = L(\text{GND}) \times di/dt$$

EQ 2

$$\text{VCCI dip noise voltage} = L(\text{VCCI}) \times di/dt$$

EQ 3

Any group of four or more input pins switching on the same clock edge is considered an SSO bus. The shielding should be done both on the board and inside the package unless otherwise described.

In-package shielding can be achieved in several ways; the required shielding will vary depending on whether pins next to the SSO bus are LVTTTL/LVCMOS inputs, LVTTTL/LVCMOS outputs, or GTL/SSTL/HSTL/LVDS/LVPECL inputs and outputs. Board traces in the vicinity of the SSO bus have to be adequately shielded from mutual coupling and inductive noise that can be generated by the SSO bus. Also, noise generated by the SSO bus needs to be reduced inside the package.

PCBs perform an important function in feeding stable supply voltages to the IC and, at the same time, maintaining signal integrity between devices.

Key issues that need to be considered are as follows:

- Power and ground plane design and decoupling network design
- Transmission line reflections and terminations

For extensive data per package on the SSO and PCB issues, refer to the "ProASIC3/E SSO and Pin Placement and Guidelines" chapter of the *ProASIC3 FPGA Fabric User's Guide*.



## 5 V Output Tolerance

IGLOO and ProASIC3 I/Os must be set to 3.3 V LVTTTL or 3.3 V LVCMOS mode to reliably drive 5 V TTL receivers. It is also critical that there be NO external I/O pull-up resistor to 5 V, since this resistor would pull the I/O pad voltage beyond the 3.6 V absolute maximum value and consequently cause damage to the I/O.

When set to 3.3 V LVTTTL or 3.3 V LVCMOS mode, the I/Os can directly drive signals into 5 V TTL receivers. In fact,  $V_{OL} = 0.4\text{ V}$  and  $V_{OH} = 2.4\text{ V}$  in both 3.3 V LVTTTL and 3.3 V LVCMOS modes exceeds the  $V_{IL} = 0.8\text{ V}$  and  $V_{IH} = 2\text{ V}$  level requirements of 5 V TTL receivers. Therefore, level 1 and level 0 will be recognized correctly by 5 V TTL receivers.

## Schmitt Trigger

A Schmitt trigger is a buffer used to convert a slow or noisy input signal into a clean one before passing it to the FPGA. Using Schmitt trigger buffers guarantees a fast, noise-free input signal to the FPGA.

ProASIC3E devices have Schmitt triggers built into their I/O circuitry. The Schmitt trigger is available for the LVTTTL, LVCMOS, and 3.3 V PCI I/O standards.

This feature can be implemented by using a Physical Design Constraints (PDC) command (Table 8-6 on page 218) or by selecting a check box in the I/O Attribute Editor in Designer. The check box is cleared by default.

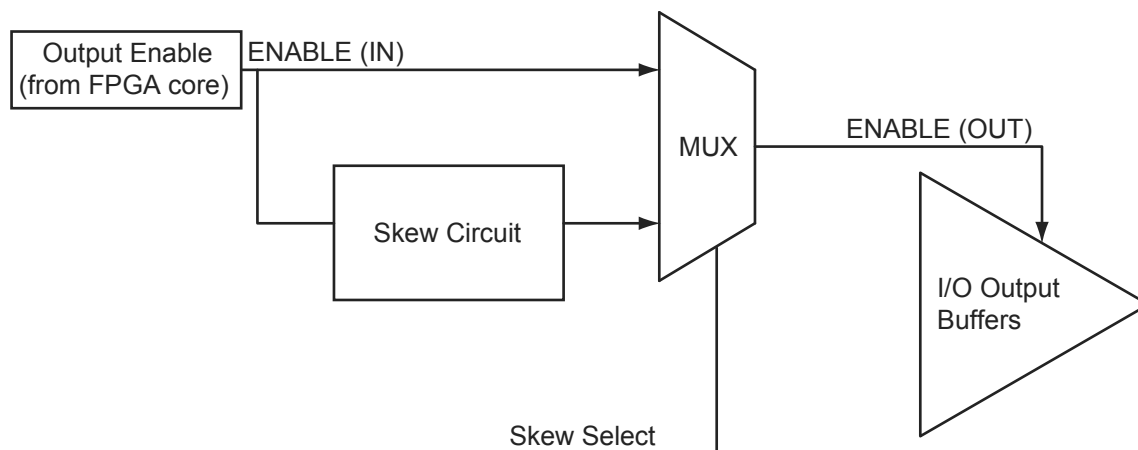
## Selectable Skew between Output Buffer Enable and Disable Times

Low power flash devices have a configurable skew block in the output buffer circuitry that can be enabled to delay output buffer assertion without affecting deassertion time. Since this skew block is only available for the OE signal, the feature can be used in tristate and bidirectional buffers. A typical 1.2 ns delay is added to the OE signal to prevent potential bus contention. Refer to the appropriate family datasheet for detailed timing diagrams and descriptions.

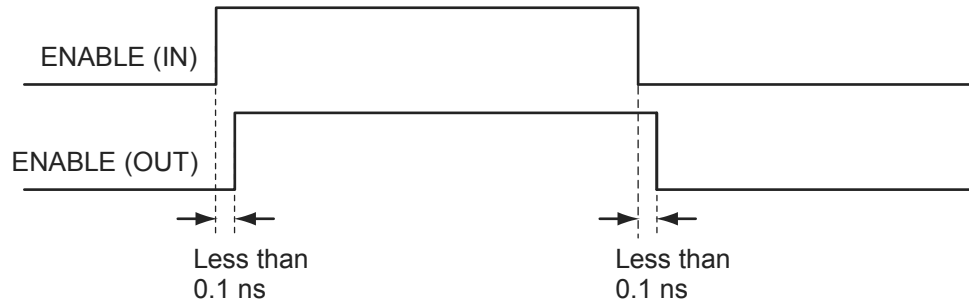
The Skew feature is available for all I/O standards.

This feature can be implemented by using a PDC command (Table 8-6 on page 218) or by selecting a check box in the I/O Attribute Editor in Designer. The check box is cleared by default.

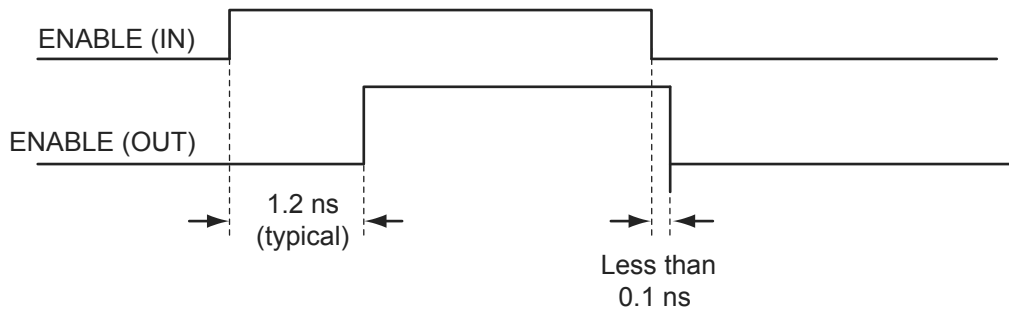
The configurable skew block is used to delay output buffer assertion (enable) without affecting deassertion (disable) time.



**Figure 8-14 • Block Diagram of Output Enable Path**

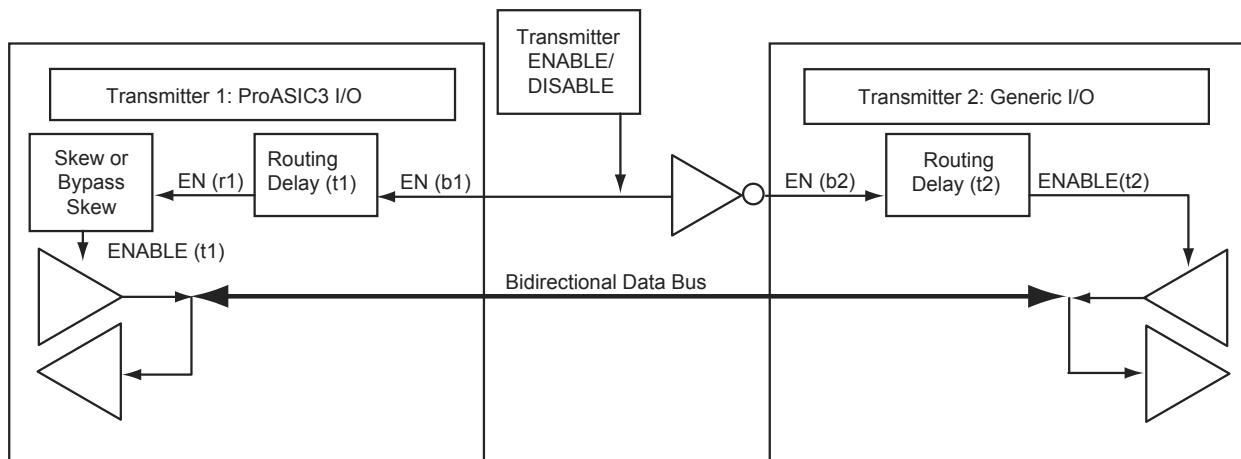


**Figure 8-15 • Timing Diagram (option 1: bypasses skew circuit)**



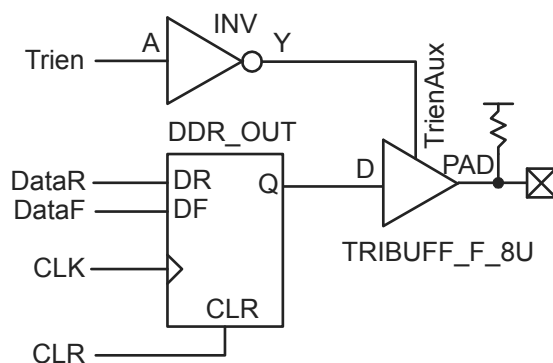
**Figure 8-16 • Timing Diagram (option 2: enables skew circuit)**

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



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

## DDR Tristate Output Register



**Figure 10-7 • DDR Tristate Output Register, LOW Enable, 8 mA, Pull-Up (LVTTL)**

### Verilog

```
module DDR_TriStateBuf_LVTTL_8mA_HighSlew_LowEnb_PullUp(DataR, DataF, CLR, CLK, Trien,
    PAD);

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

    wire TrienAux, Q;

    INV Inv_Tri(.A(Trien),.Y(TrienAux));
    DDR_OUT DDR_OUT_0_inst(.DR(DataR),.DF(DataF),.CLK(CLK),.CLR(CLR),.Q(Q));
    TRIBUFF_F_8U TRIBUFF_F_8U_0_inst(.D(Q),.E(TrienAux),.PAD(PAD));

endmodule
```

### VHDL

```
library ieee;
use ieee.std_logic_1164.all;
library proasic3; use proasic3.all;

entity DDR_TriStateBuf_LVTTL_8mA_HighSlew_LowEnb_PullUp is
    port(DataR, DataF, CLR, CLK, Trien : in std_logic; PAD : out std_logic) ;
end DDR_TriStateBuf_LVTTL_8mA_HighSlew_LowEnb_PullUp;

architecture DEF_ARCH of DDR_TriStateBuf_LVTTL_8mA_HighSlew_LowEnb_PullUp 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 TRIBUFF_F_8U
        port(D, E : in std_logic := 'U'; PAD : out std_logic) ;
    end component;

    signal TrienAux, Q : std_logic ;

begin

    Inv_Tri : INV
        port map(A => Trien, Y => TrienAux);
```

*Note: The settings in this figure are used to show the generation of an AES-encrypted programming file for the FPGA array, FlashROM, and FB contents. One or all locations may be selected for encryption.*

---

**Figure 12-17 • Settings to Program a Device Secured with FlashLock and using AES Encryption**

Choose the **High** security level to reprogram devices using both the FlashLock Pass Key and AES key protection (Figure 12-18 on page 321). Enter the AES key and click **Next**.

A device that has already been secured with FlashLock and has an AES key loaded must recognize the AES key to program the device and generate a valid bitstream in authentication. The FlashLock Key is only required to unlock the device and change the security settings.

This is what makes it possible to program in an untrusted environment. The AES key is protected inside the device by the FlashLock Key, so you can only program if you have the correct AES key. In fact, the AES key is not in the programming file either. It is the key used to encrypt the data in the file. The same key previously programmed with the FlashLock Key matches to decrypt the file.

An AES-encrypted file programmed to a device without FlashLock would not be secure, since without FlashLock to protect the AES key, someone could simply reprogram the AES key first, then program with any AES key desired or no AES key at all. This option is therefore not available in the software.

### 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" "FFromStream";
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><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: <ul style="list-style-type: none"> <li>• Flash technology that does not require an external bitstream to program the device</li> <li>• FlashLock Pass Key that secures device content by locking the security settings and preventing access to the device as defined by the user</li> <li>• AES key that allows secure, encrypted device reprogrammability</li> </ul> |

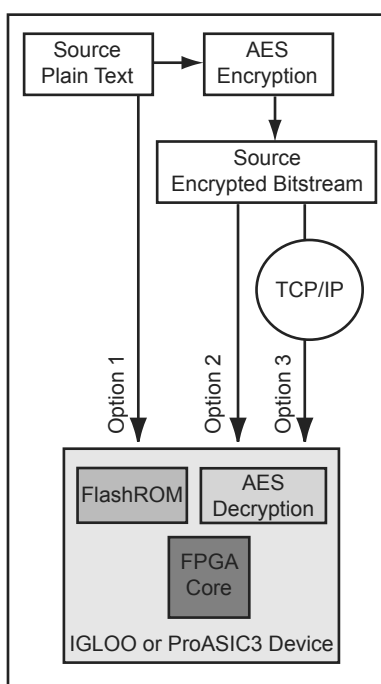
## References

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.

Figure 13-2 shows different applications for ISP programming.

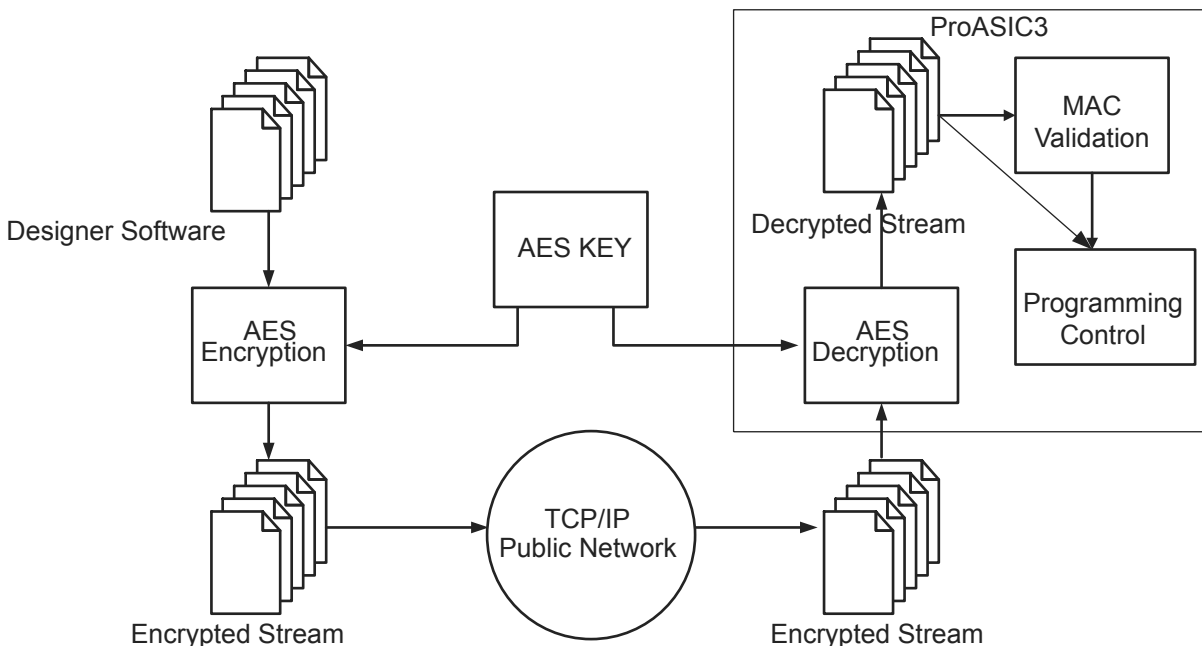
1. In a trusted programming environment, you can program the device using the unencrypted (plaintext) programming file.
2. You can program the AES Key in a trusted programming environment and finish the final programming in an untrusted environment using the AES-encrypted (cipher text) programming file.
3. For the remote ISP updating/reprogramming, the AES Key stored in the device enables the encrypted programming bitstream to be transmitted through the untrusted network connection.

Microsemi low power flash devices also provide the unique Microsemi FlashLock feature, which protects the Pass Key and AES Key. Unless the original FlashLock Pass Key is used to unlock the device, security settings cannot be modified. Microsemi does not support read-back of FPGA core-programmed data; however, the FlashROM contents can selectively be read back (or disabled) via the JTAG port based on the security settings established by the Microsemi Designer software. Refer to the "Security in Low Power Flash Devices" section on page 301 for more information.



**Figure 13-2 • Different ISP Use Models**

useless to the thief. To learn more about the low power flash devices' security features, refer to the "Security in Low Power Flash Devices" section on page 301.



**Figure 15-5 • ProASIC3 Device Encryption Flow**

## Conclusion

The Fusion, IGLOO, and ProASIC3 FPGAs are ideal for applications that require field upgrades. The single-chip devices save board space by eliminating the need for EEPROM. The built-in AES with MAC enables transmission of programming data over any network without fear of design theft. Fusion, IGLOO, and ProASIC3 FPGAs are IEEE 1532-compliant and support STAPL, making the target programming software easy to implement.

## SRAM Initialization

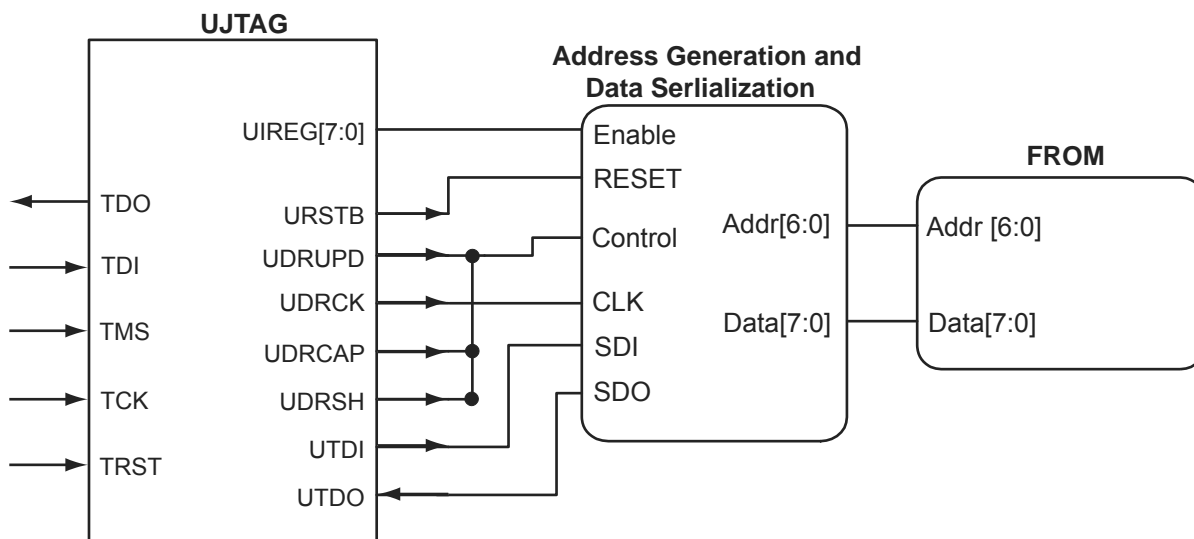
Users can also initialize embedded SRAMs of the low power flash devices. The initialization of the embedded SRAM blocks of the design can be done using UJTAG tiles, where the initialization data is imported using the TAP Controller. Similar functionality is available in ProASIC<sup>PLUS</sup> devices using JTAG. The guidelines for implementation and design examples are given in the *RAM Initialization and ROM Emulation in ProASIC<sup>PLUS</sup> Devices* application note.

SRAMs are volatile by nature; data is lost in the absence of power. Therefore, the initialization process should be done at each power-up if necessary.

## FlashROM Read-Back Using JTAG

The low power flash architecture contains a dedicated nonvolatile FlashROM block, which is formatted into eight 128-bit pages. For more information on FlashROM, refer to the "FlashROM in Microsemi's Low Power Flash Devices" section on page 133. The contents of FlashROM are available to the VersaTiles during normal operation through a read operation. As a result, the UJTAG macro can be used to provide the FlashROM contents to the JTAG port during normal operation. Figure 17-7 illustrates a simple block diagram of using UJTAG to read the contents of FlashROM during normal operation.

The FlashROM read address can be provided from outside the FPGA through the TDI input or can be generated internally using the core logic. In either case, data serialization logic is required (Figure 17-7) and should be designed using the VersaTile core logic. FlashROM contents are read asynchronously in parallel from the flash memory and shifted out in a synchronous serial format to TDO. Shifting the serial data out of the serialization block should be performed while the TAP is in UDRSH mode. The coordination between TCK and the data shift procedure can be done using the TAP state machine by monitoring UDRSH, UDRCAP, and UDRUPD.



**Figure 17-7 • Block Diagram of Using UJTAG to Read FlashROM Contents**



## **B – Product Support**

---

Microsemi SoC Products Group backs its products with various support services, including Customer Service, Customer Technical Support Center, a website, electronic mail, and worldwide sales offices. This appendix contains information about contacting Microsemi SoC Products Group and using these support services.

### **Customer Service**

Contact Customer Service for non-technical product support, such as product pricing, product upgrades, update information, order status, and authorization.

From North America, call 800.262.1060

From the rest of the world, call 650.318.4460

Fax, from anywhere in the world, 650.318.8044

### **Customer Technical Support Center**

Microsemi SoC Products Group staffs its Customer Technical Support Center with highly skilled engineers who can help answer your hardware, software, and design questions about Microsemi SoC Products. The Customer Technical Support Center spends a great deal of time creating application notes, answers to common design cycle questions, documentation of known issues, and various FAQs. So, before you contact us, please visit our online resources. It is very likely we have already answered your questions.

### **Technical Support**

Visit the Customer Support website ([www.microsemi.com/soc/support/search/default.aspx](http://www.microsemi.com/soc/support/search/default.aspx)) for more information and support. Many answers available on the searchable web resource include diagrams, illustrations, and links to other resources on the website.

### **Website**

You can browse a variety of technical and non-technical information on the SoC home page, at [www.microsemi.com/soc](http://www.microsemi.com/soc).

### **Contacting the Customer Technical Support Center**

Highly skilled engineers staff the Technical Support Center. The Technical Support Center can be contacted by email or through the Microsemi SoC Products Group website.

#### **Email**

You can communicate your technical questions to our email address and receive answers back by email, fax, or phone. Also, if you have design problems, you can email your design files to receive assistance. We constantly monitor the email account throughout the day. When sending your request to us, please be sure to include your full name, company name, and your contact information for efficient processing of your request.

The technical support email address is [soc\\_tech@microsemi.com](mailto:soc_tech@microsemi.com).