**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 | 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 | 256-LBGA |
| Supplier Device Package | 256-FPBGA (17x17) |
| Purchase URL | https://www.e-xfl.com/product-detail/microsemi/m1a3p600l-fgg256i |

standard for CLKBUF is LVTTL in the current Microsemi Libero® System-on-Chip (SoC) and Designer software.

*Table 3-9 •* **I/O Standards within CLKBUF**

| Name | Description |
|------|-------------|
| CLKBUF_LVCMOS5 | LVCMOS clock buffer with 5.0 V CMOS voltage level |
| CLKBUF_LVCMOS33 | LVCMOS clock buffer with 3.3 V CMOS voltage level |
| CLKBUF_LVCMOS25 | LVCMOS clock buffer with 2.5 V CMOS voltage level[1] |
| CLKBUF_LVCMOS18 | LVCMOS clock buffer with 1.8 V CMOS voltage level |
| CLKBUF_LVCMOS15 | LVCMOS clock buffer with 1.5 V CMOS voltage level |
| CLKBUF_LVCMOS12 | LVCMOS clock buffer with 1.2 V CMOS voltage level |
| CLKBUF_PCI | PCI clock buffer |
| CLKBUF_PCIX | PCIX clock buffer |
| CLKBUF_GTL25 | GTL clock buffer with 2.5 V CMOS voltage level[1] |
| CLKBUF_GTL33 | GTL clock buffer with 3.3 V CMOS voltage level[1] |
| CLKBUF_GTLP25 | GTL+ clock buffer with 2.5 V CMOS voltage level[1] |
| CLKBUF_GTLP33 | GTL+ clock buffer with 3.3 V CMOS voltage level[1] |
| CLKBUF_ HSTL _I | HSTL Class I clock buffer[1] |
| CLKBUF_ HSTL _II | HSTL Class II clock buffer[1] |
| CLKBUF_SSTL2_I | SSTL2 Class I clock buffer[1] |
| CLKBUF_SSTL2_II | SSTL2 Class II clock buffer[1] |
| CLKBUF_SSTL3_I | SSTL3 Class I clock buffer[1] |
| CLKBUF_SSTL3_II | SSTL3 Class II clock buffer[1] |

*Notes:*

1. *Supported in only the IGLOOe, ProASIC3E, AFS600, and AFS1500 devices*

2. *By default, the CLKBUF macro uses the 3.3 V LVTTL I/O technology.*

The current synthesis tool libraries only infer the CLKBUF or CLKINT macros in the netlist. All other global macros must be instantiated manually into your HDL code. The following is an example of CLKBUF_LVCMOS25 global macro instantiations that you can copy and paste into your code:

## *VHDL*

```
component clkbuf_lvcmos25
  port (pad : in std_logic; y : out std_logic);
end component

begin
-- concurrent statements
u2 : clkbuf_lvcmos25 port map (pad =>  ext_clk, y => int_clk);
end
```

## *Verilog*

```
module design (_____);

input _____;
output _____;

clkbuf_lvcmos25 u2 (.y(int_clk), .pad(ext_clk);

endmodule
```

# Simple Design Example

Consider a design consisting of six building blocks (shift registers) and targeted for an A3PE600-PQ208 (Figure 3-16 on page 68). The example design consists of two PLLs (PLL1 has GLA only; PLL2 has both GLA and GLB), a global reset (ACLR), an enable (EN_ALL), and three external clock domains (QCLK1, QCLK2, and QCLK3) driving the different blocks of the design. Note that the PQ208 package only has two PLLs (which access the chip global network). Because of fanout, the global reset and enable signals need to be assigned to the chip global resources. There is only one free chip global for the remaining global (QCLK1, QCLK2, QCLK3). Place two of these signals on the quadrant global resource. The design example demonstrates manually assignment of QCLK1 and QCLK2 to the quadrant global using the PDC command.
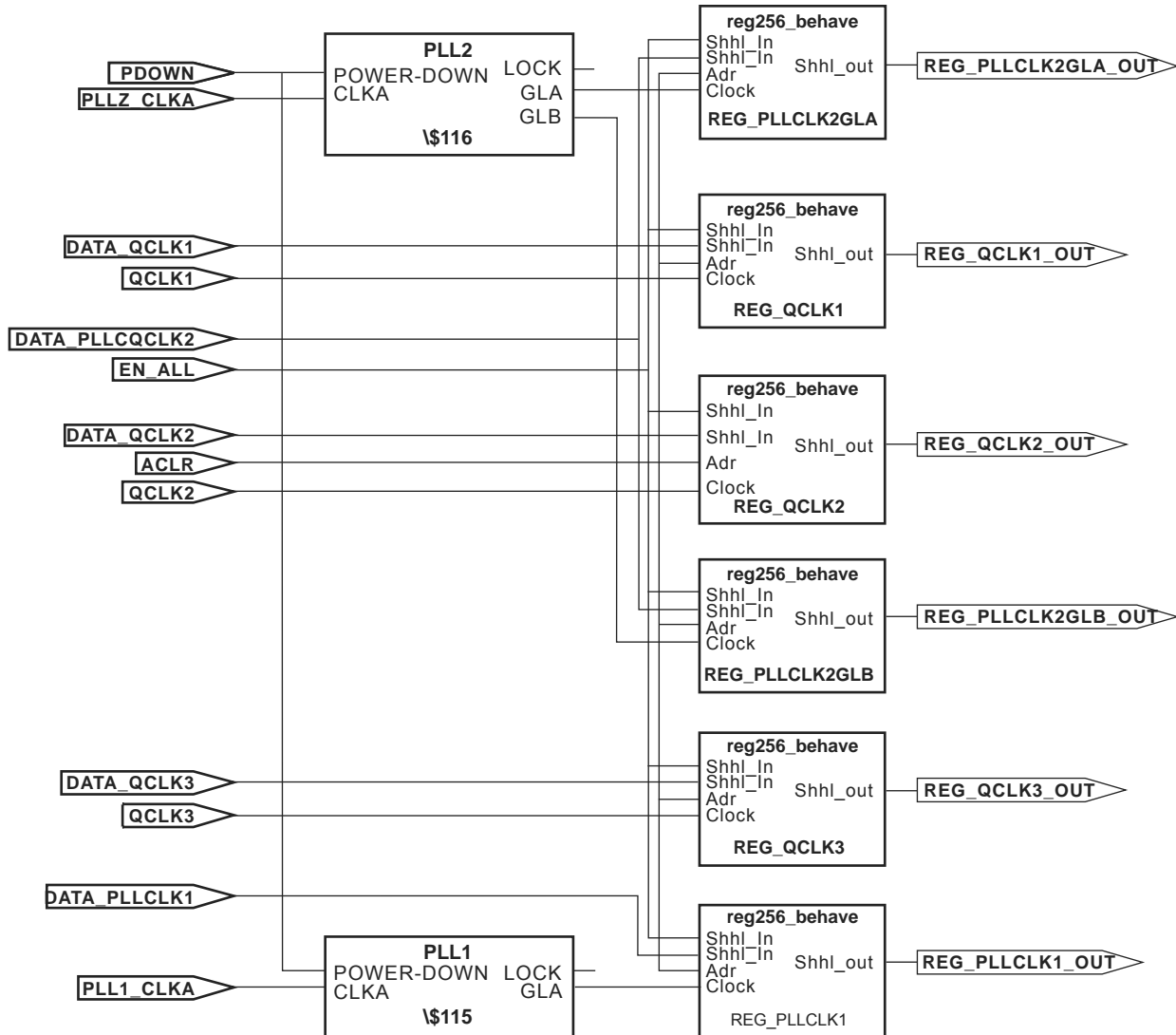


***Figure 3-19 •*** **Block Diagram of the Global Management Example Design**

# List of Changes

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

| Date | Changes | Page |
|---|---|---|
| July 2010 | This chapter is no longer published separately with its own part number and version but is now part of several FPGA fabric user's guides. | N/A |
| | Notes were added where appropriate to point out that IGLOO nano and ProASIC3 nano devices do not support differential inputs (SAR 21449). | N/A |
| | The "Global Architecture" section and "VersaNet Global Network Distribution" section were revised for clarity (SARs 20646, 24779). | 47, 49 |
| | The "I/O Banks and Global I/Os" section was moved earlier in the document, renamed to "Chip and Quadrant Global I/Os", and revised for clarity. Figure 3-4 • Global Connections Details, Figure 3-6 • Global Inputs, Table 3-2 • Chip Global Pin Name, and Table 3-3 • Quadrant Global Pin Name are new (SARs 20646, 24779). | 51 |
| | The "Clock Aggregation Architecture" section was revised (SARs 20646, 24779). | 57 |
| | Figure 3-7 • Chip Global Aggregation was revised (SARs 20646, 24779). | 59 |
| | The "Global Macro and Placement Selections" section is new (SARs 20646, 24779). | 64 |
| v1.4 (December 2008) | The "Global Architecture" section was updated to include 10 k devices, and to include information about VersaNet global support for IGLOO nano devices. | 47 |
| | The Table 3-1 • Flash-Based FPGAs was updated to include IGLOO nano and ProASIC3 nano devices. | 48 |
| | The "VersaNet Global Network Distribution" section was updated to include 10 k devices and to note an exception in global lines for nano devices. | 49 |
| | Figure 3-2 • Simplified VersaNet Global Network (30 k gates and below) is new. | 50 |
| | The "Spine Architecture" section was updated to clarify support for 10 k and nano devices. | 57 |
| | Table 3-4 • Globals/Spines/Rows for IGLOO and ProASIC3 Devices was updated to include IGLOO nano and ProASIC3 nano devices. | 57 |
| | The figure in the CLKBUF_LVDS/LVPECL row of Table 3-8 • Clock Macros was updated to change CLKBIBUF to CLKBUF. | 62 |
| v1.3 (October 2008) | A third bullet was added to the beginning of the "Global Architecture" section: In Fusion devices, the west CCC also contains a PLL core. In the two larger devices (AFS600 and AFS1500), the west and east CCCs each contain a PLL. | 47 |
| | The "Global Resource Support in Flash-Based Devices" section was revised to include new families and make the information more concise. | 48 |
| | Table 3-4 • Globals/Spines/Rows for IGLOO and ProASIC3 Devices was updated to include A3PE600/L in the device column. | 57 |
| | Table note 1 was revised in Table 3-9 • I/O Standards within CLKBUF to include AFS600 and AFS1500. | 63 |
| v1.2 (June 2008) | The following changes were made to the family descriptions in Table 3-1 • 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. | 48 |

## IGLOOe and ProASIC3E CCC Locations

IGLOOe and ProASIC3E devices have six CCCs—one in each of the four corners and one each in the middle of the east and west sides of the device (Figure 4-15).

All six CCCs are integrated with PLLs, except in PQFP-208 package devices. PQFP-208 package devices also have six CCCs, of which two include PLLs and four are simplified CCCs. The CCCs with PLLs are implemented in the middle of the east and west sides of the device (middle right and middle left). The simplified CCCs without PLLs are located in the four corners of the device (Figure 4-16).
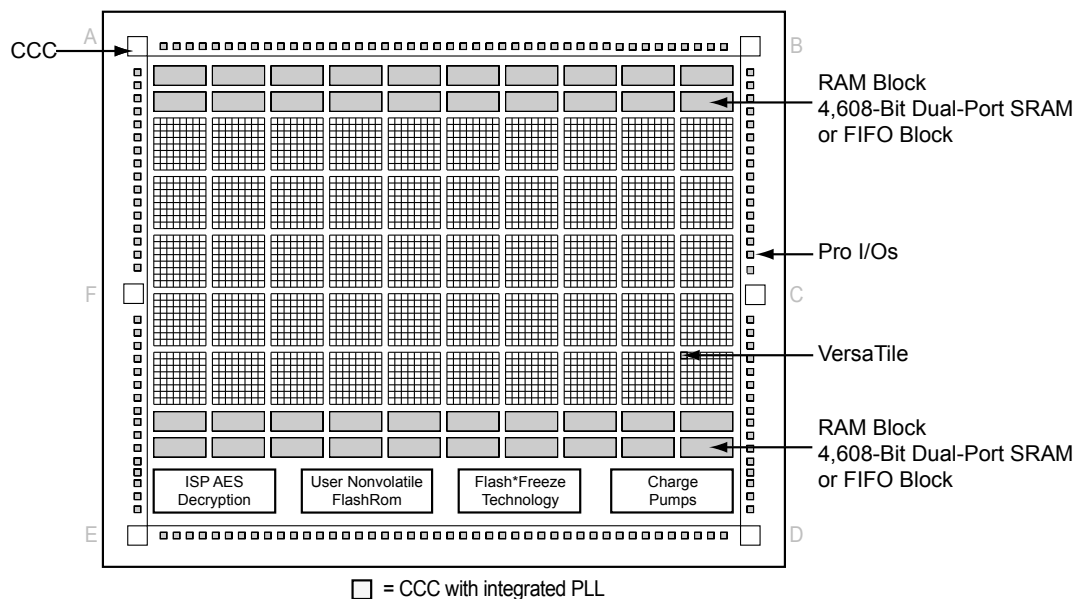


*Figure 4-15 •* **CCC Locations in IGLOOe and ProASIC3E Family Devices (except PQFP-208 package)**
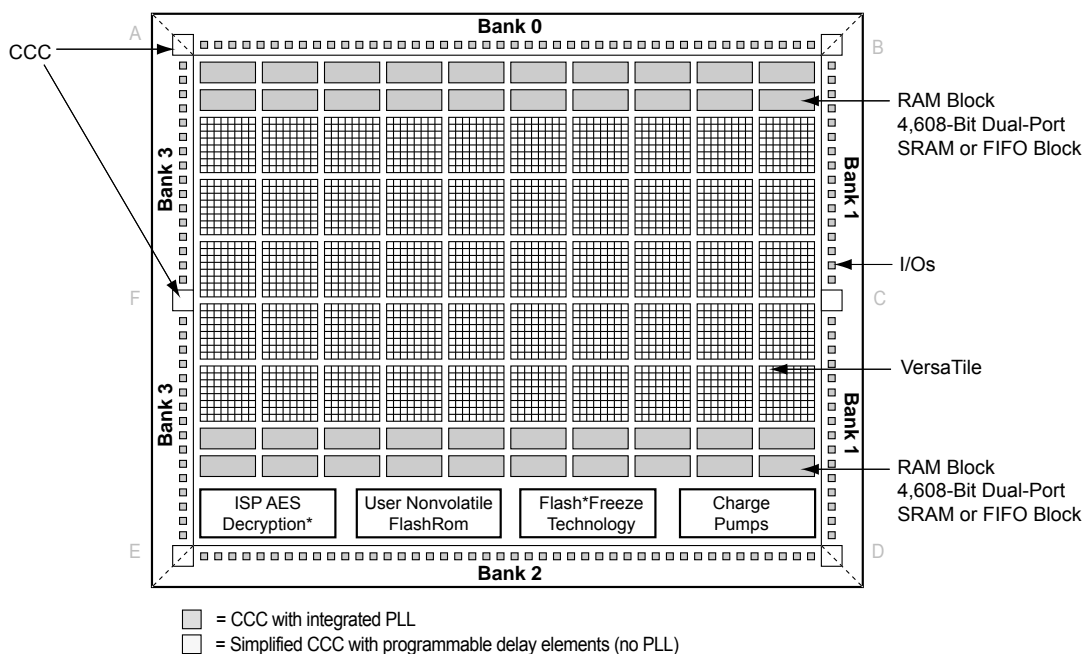


*Figure 4-16 •* **CCC Locations in ProASIC3E Family Devices (PQFP-208 package)**

## Phase Adjustment

The four phases available (0, 90, 180, 270) are phases with respect to VCO (PLL output). The VCO is divided to achieve the user's CCC required output frequency (GLA, YB/GLB, YC/GLC). The division happens after the selection of the VCO phase. The effective phase shift is actually the VCO phase shift divided by the output divider. This is why the visual CCC shows both the actual achievable phase and more importantly the actual delay that is equivalent to the phase shift that can be achieved.

## Dynamic PLL Configuration

The CCCs can be configured both statically and dynamically.

In addition to the ports available in the Static CCC, the Dynamic CCC has the dynamic shift register signals that enable dynamic reconfiguration of the CCC. With the Dynamic CCC, the ports CLKB and CLKC are also exposed. All three clocks (CLKA, CLKB, and CLKC) can be configured independently.

The CCC block is fully configurable. The following two sources can act as the CCC configuration bits.
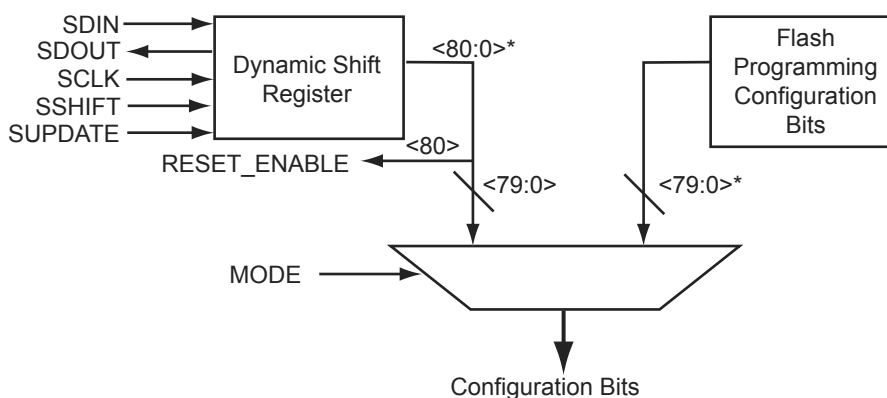
### Flash Configuration Bits

The flash configuration bits are the configuration bits associated with programmed flash switches. These bits are used when the CCC is in static configuration mode. Once the device is programmed, these bits cannot be modified. They provide the default operating state of the CCC.

### Dynamic Shift Register Outputs

This source does not require core reprogramming and allows core-driven dynamic CCC reconfiguration. When the dynamic register drives the configuration bits, the user-defined core circuit takes full control over SDIN, SDOUT, SCLK, SSHIFT, and SUPDATE. The configuration bits can consequently be dynamically changed through shift and update operations in the serial register interface. Access to the logic core is accomplished via the dynamic bits in the specific tiles assigned to the PLLs.

Figure 4-21 illustrates a simplified block diagram of the MUX architecture in the CCCs.



*Note:    *For Fusion, bit <88:81> is also needed.*

***Figure 4-21 •** The CCC Configuration MUX Architecture*

The selection between the flash configuration bits and the bits from the configuration register is made using the MODE signal shown in Figure 4-21. If the MODE signal is logic HIGH, the dynamic shift register configuration bits are selected. There are 81 control bits to configure the different functions of the CCC.

# Recommended Board-Level Considerations

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

## PLL Power Supply Decoupling Scheme

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

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

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

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

Recommended 100 nF capacitor:

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

Recommended 10 nF capacitor:

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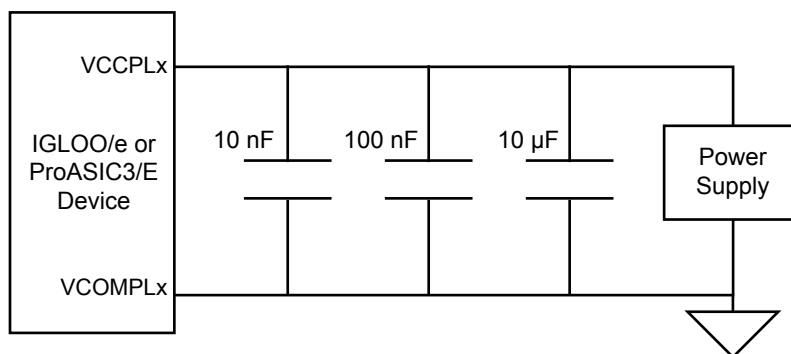- Digi-Key part number: BC1252TR-ND



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

# 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 | • High performance (Table 7-6 on page 180)<br>• Electrostatic discharge (ESD) protection<br>• I/O register combining option |
| Single-Ended Transmitter Features | • Hot-swap:<br>  – 30K gate devices: hot-swap in every mode<br>  – All other IGLOO and ProASIC3 devices: no hot-swap<br>• Output slew rate: 2 slew rates (except 30K gate devices)<br>• Weak pull-up and pull-down resistors<br>• Output drive: 3 drive strengths<br>• Programmable output loading<br>• 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)<br>• LVTTL/LVCMOS 3.3 V outputs compatible with 5 V TTL inputs |
| Single-Ended Receiver Features | • 5 V–input–tolerant receiver (Table 7-12 on page 193)<br>• Separate ground plane for GNDQ pin and power plane for VMV pin are used for input buffer to reduce output-induced noise. |
| Differential Receiver Features—250K through 1M Gate Devices | • Separate ground plane for GNDQ pin and power plane for VMV pin are used for input buffer to reduce output-induced noise. |
| CMOS-Style LVDS, B-LVDS, M-LVDS, or LVPECL Transmitter | • Two I/Os and external resistors are used to provide a CMOS-style LVDS, DDR LVDS, B-LVDS, and M-LVDS/LVPECL transmitter solution.<br>• High slew rate<br>• Weak pull-up and pull-down resistors<br>• Programmable output loading |

# I/O Banks and I/O Standards Compatibility

I/Os are grouped into I/O voltage banks.

Each I/O voltage bank has dedicated I/O supply and ground voltages (VMV/GNDQ for input buffers and $V_{CCI}$/GND for output buffers). Because of these dedicated supplies, only I/Os with compatible standards can be assigned to the same I/O voltage bank. Table 8-3 on page 217 shows the required voltage compatibility values for each of these voltages.

There are eight I/O banks (two per side).

Every I/O bank is divided into minibanks. Any user I/O in a VREF minibank (a minibank is the region of scope of a VREF pin) can be configured as a VREF pin (Figure 8-2). Only one $V_{REF}$ pin is needed to control the entire $V_{REF}$ minibank. The location and scope of the $V_{REF}$ minibanks can be determined by the I/O name. For details, see the user I/O naming conventions for "IGLOOe and ProASIC3E" on page 245. Table 8-5 on page 217 shows the I/O standards supported by IGLOOe and ProASIC3E devices, and the corresponding voltage levels.

I/O standards are compatible if they comply with the following:

- Their VCCI and VMV values are identical.
- Both of the standards need a VREF, and their VREF values are identical.
- All inputs and disabled outputs are voltage tolerant up to 3.3 V.

For more information about I/O and global assignments to I/O banks in a device, refer to the specific pin table for the device in the packaging section of the datasheet, and see the user I/O naming conventions for "IGLOOe and ProASIC3E" on page 245.
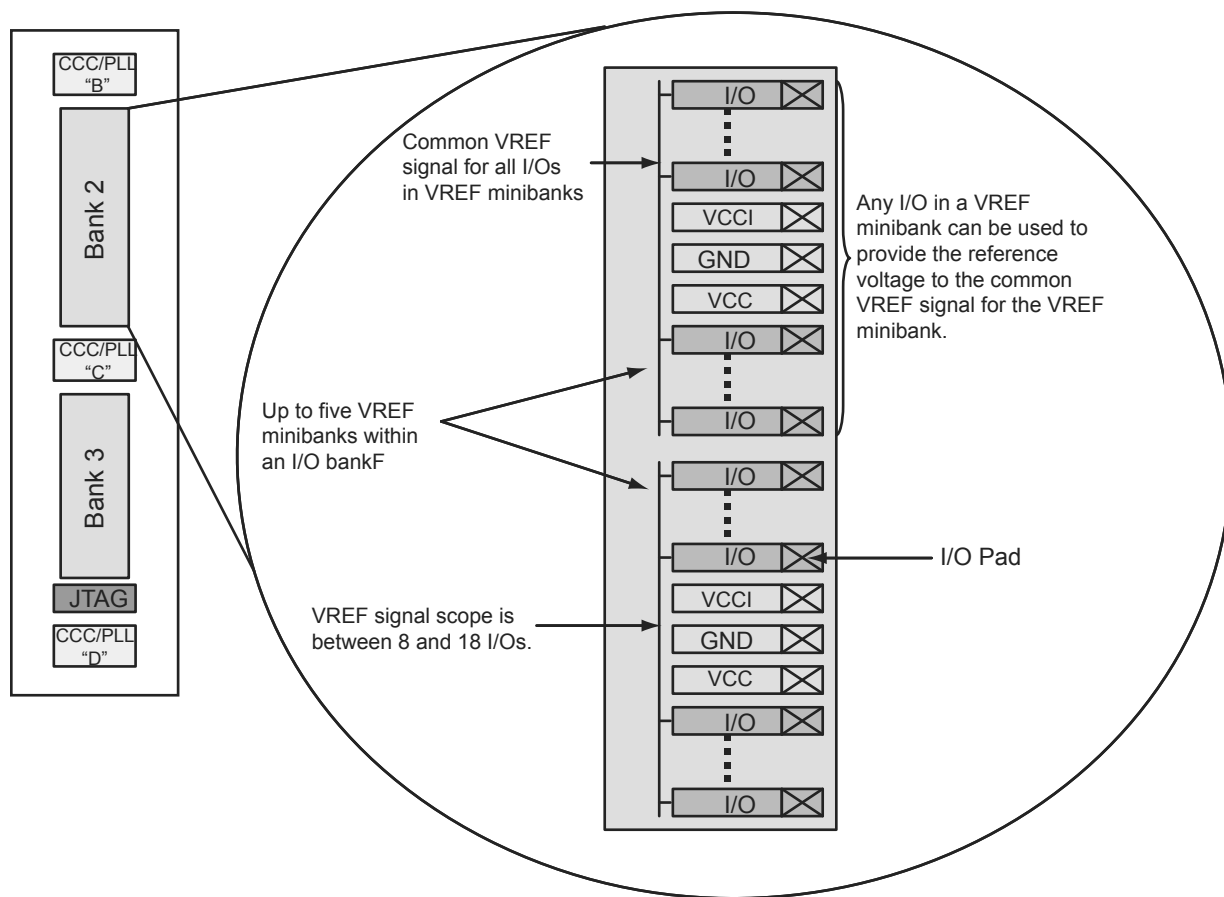


***Figure 8-2 •*** **Typical IGLOOe and ProASIC3E I/O Bank Detail Showing V$_{REF}$ Minibanks**
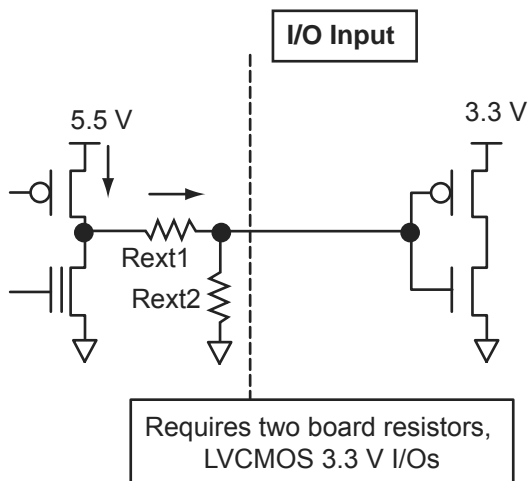
## Solution 1



*Figure 8-10 •* **Solution 1**

### Solution 2

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

This scheme will also work for a 3.3 V PCI/PCI-X configuration, but the internal diode should not be used for clamping, and the voltage must be limited by the external resistors and Zener, as shown in Figure 8-11. Relying on the diode clamping would create an excessive pad DC voltage of 3.3 V + 0.7 V = 4 V.
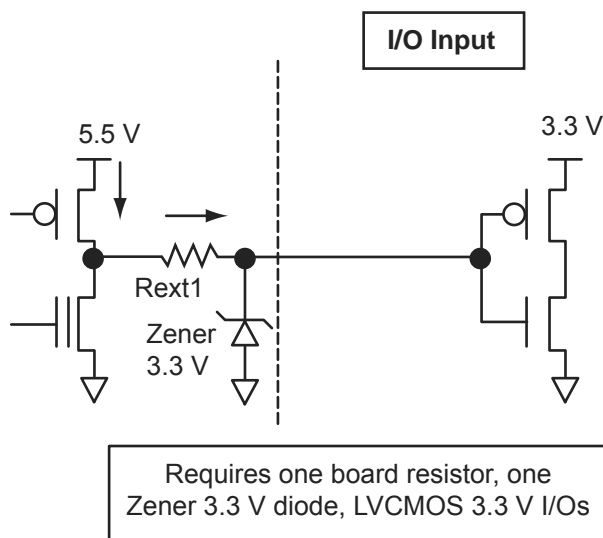
## Solution 2



*Figure 8-11 •* **Solution 2**

# 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 8-2 and EQ 8-3).

Ground bounce noise voltage = L(GND) × di/dt

*EQ 8-2*

VCCI dip noise voltage = L(VCCI) × di/dt

*EQ 8-3*

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 LVTTL/LVCMOS inputs, LVTTL/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*.

# 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 9-2 • Designer State (resulting from I/O attribute modification) were revised to clarify which device families support programmable input delay (SAR 39666). | 253 |
| June 2011 | Figure 9-2 • 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. | 254 |
|  | The following rule was added to the "VREF Rules for the Implementation of Voltage-Referenced I/O Standards" section:<br><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). | 265 |
| 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 9-1 • Flash-Based FPGAs. | 252 |
|  | The notes for Table 9-2 • Designer State (resulting from I/O attribute modification) were revised to indicate that skew control and input delay do not apply to nano devices. | 253 |
| v1.3 (October 2008) | The "Flash FPGAs I/O Support" section was revised to include new families and make the information more concise. | 252 |
| v1.2 (June 2008) | The following changes were made to the family descriptions in Table 9-1 • Flash-Based FPGAs:<br><br>• ProASIC3L was updated to include 1.5 V.<br><br>• The number of PLLs for ProASIC3E was changed from five to six. | 252 |
| 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 9-2 • Designer State (resulting from I/O attribute modification) was updated to include note 2 for IGLOO PLUS. | 253 |

# I/O Cell Architecture

Low power flash devices support DDR in the I/O cells in four different modes: Input, Output, Tristate, and Bidirectional pins. For each mode, different I/O standards are supported, with most I/O standards having special sub-options. For the ProASIC3 nano and IGLOO nano devices, DDR is supported only in the 60 k, 125 k, and 250 k logic densities. Refer to Table 10-2 for a sample of the available I/O options. Additional I/O options can be found in the relevant family datasheet.

*Table 10-2 •* **DDR I/O Options**

| DDR Register Type | I/O Type | I/O Standard | Sub-Options | Comments |
|---|---|---|---|---|
| Receive Register | Input | Normal | None | 3.3 V TTL (default) |
| | | LVCMOS | Voltage | 1.5 V, 1.8 V, 2.5 V, 5 V (1.5 V default) |
| | | | Pull-Up | None (default) |
| | | PCI/PCI-X | None | |
| | | GTL/GTL+ | Voltage | 2.5 V, 3.3 V (3.3 V default) |
| | | HSTL | Class | I / II (I default) |
| | | SSTL2/SSTL3 | Class | I / II (I default) |
| | | LVPECL | None | |
| | | LVDS | None | |
| Transmit Register | Output | Normal | None | 3.3 V TTL (default) |
| | | LVTTL | Output Drive | 2, 4, 6, 8, 12, 16, 24, 36 mA (8 mA default) |
| | | | Slew Rate | Low/high (high default) |
| | | LVCMOS | Voltage | 1.5 V, 1.8 V, 2.5 V, 5 V (1.5 V default) |
| | | PCI/PCI-X | None | |
| | | GTL/GTL+ | Voltage | 1.8 V, 2.5 V, 3.3 V (3.3 V default) |
| | | HSTL | Class | I / II (I default) |
| | | SSTL2/SSTL3 | Class | I / II (I default) |
| | | LVPECL* | None | |
| | | LVDS* | None | |

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

## VHDL

```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;
```
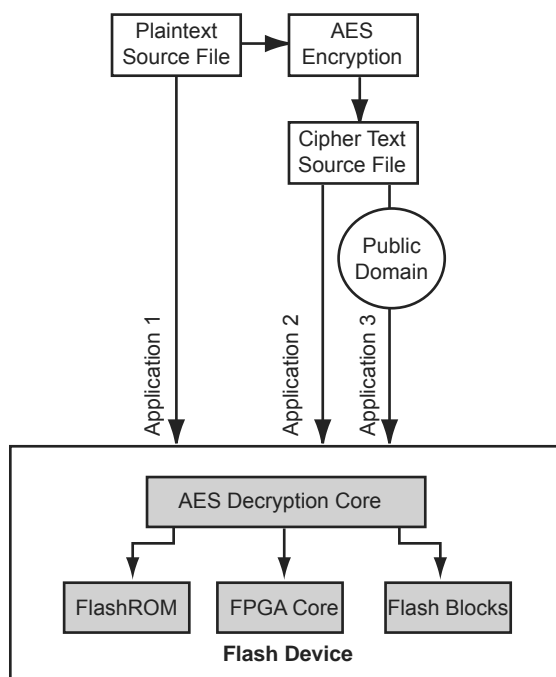
# Security in Action

This section illustrates some applications of the security advantages of Microsemi's devices (Figure 12-6).

.



*Note:    Flash blocks are only used in Fusion devices*

***Figure 12-6 •* Security Options**

*Figure 12-18 •* **Security Level Set High to Reprogram Device with AES Key**

Programming with this file is intended for an unsecured environment. The AES key encrypts the programming file with the same AES key already used in the device and utilizes it to program the device.

## Reprogramming Devices

Previously programmed devices can be reprogrammed using the steps in the "Generation of the Programming File in a Trusted Environment—Application 1" section on page 313 and "Generation of Security Header Programming File Only—Application 2" section on page 316. In the case where a FlashLock Pass Key has been programmed previously, the user must generate the new programming file with a FlashLock Pass Key that matches the one previously programmed into the device. The software will check the FlashLock Pass Key in the programming file against the FlashLock Pass Key in the device. The keys must match before the device can be unlocked to perform further programming with the new programming file.

Figure 12-10 on page 314 and Figure 12-11 on page 314 show the option **Programming previously secured device(s)**, which the user should select before proceeding. Upon going to the next step, the user will be notified that the same FlashLock Pass Key needs to be entered, as shown in Figure 12-19 on page 322.

# Microsemi

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

## Introduction

The IGLOO® and ProASIC®3L families offer devices that can be powered by either 1.5 V or, in the case of V2 devices, a core supply voltage anywhere in the range of 1.2 V to 1.5 V, in 50 mV increments.

Since IGLOO and ProASIC3L devices are flash-based, they can be programmed and reprogrammed multiple times in-system using Microsemi FlashPro3. FlashPro3 uses the JTAG standard interface (IEEE 1149.1) and STAPL file (defined in JESD 71 to support programming of programmable devices using IEEE 1149.1) for in-system configuration/programming (IEEE 1532) of a device. Programming can also be executed by other methods, such as an embedded microcontroller that follows the same standards above.

All IGLOO and ProASIC3L devices must be programmed with the VCC core voltage at 1.5 V. Therefore, applications using IGLOO or ProASIC3L devices powered by a 1.2 V supply must switch the core supply to 1.5 V for in-system programming.

The purpose of this document is to describe an easy-to-use and cost-effective solution for switching the core supply voltage from 1.2 V to 1.5 V during in-system programming for IGLOO and ProASIC3L devices.

# List of Changes

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

| Date | Changes | Page |
|---|---|---|
| July 2010 | This chapter is no longer published separately with its own part number and version but is now part of several FPGA fabric user's guides. | N/A |
| v1.1 (October 2008) | The "Introduction" was revised to include information about the core supply voltage range of operation in V2 devices. | 341 |
| | IGLOO nano device support was added to Table 14-1 • Flash-Based FPGAs Supporting Voltage Switching Circuit. | 342 |
| | The "Circuit Description" section was updated to include IGLOO PLUS core operation from 1.2 V to 1.5 V in 50 mV increments. | 343 |
| v1.0 (August 2008) | The "Microsemi's Flash Families Support Voltage Switching Circuit" section was revised to include new families and make the information more concise. | 342 |

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.
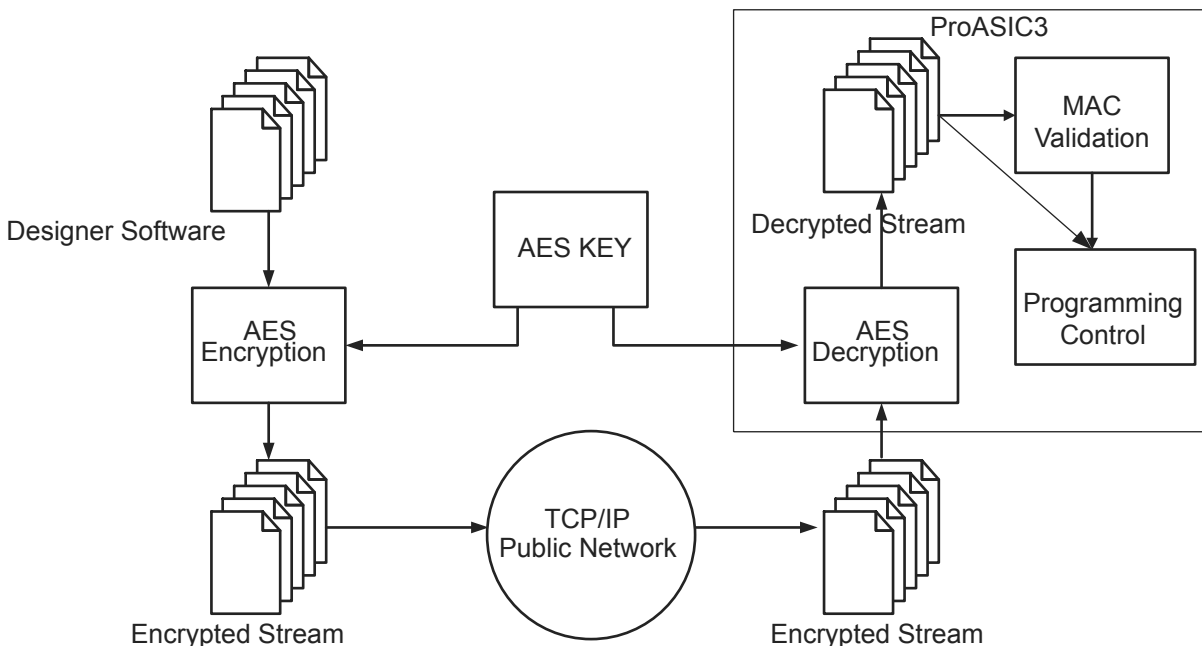


***Figure 15-5 •*** **ProASIC3 Device Encryption Flow**

# 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.

## Silicon Testing and Debugging

In many applications, the design needs to be tested, debugged, and verified on real silicon or in the final embedded application. To debug and test the functionality of designs, users may need to monitor some internal logic (or nets) during device operation. The approach of adding design test pins to monitor the critical internal signals has many disadvantages, such as limiting the number of user I/Os. Furthermore, adding external I/Os for test purposes may require additional or dedicated board area for testing and debugging.

The UJTAG tiles of low power flash devices offer a flexible and cost-effective solution for silicon test and debug applications. In this solution, the signals under test are shifted out to the TDO pin of the TAP Controller. The main advantage is that all the test signals are monitored from the TDO pin; no pins or additional board-level resources are required. Figure 17-6 illustrates this technique. Multiple test nets are brought into an internal MUX architecture. The selection of the MUX is done using the contents of the TAP Controller instruction register, where individual instructions (values from 16 to 127) correspond to different signals under test. The selected test signal can be synchronized with the rising or falling edge of TCK (optional) and sent out to UTDO to drive the TDO output of JTAG.

For flash devices, TDO (the output) is configured as low slew and the highest drive strength available in the technology and/or device. Here are some examples:

1. If the device is A3P1000 and VCCI is 3.3 V, TDO will be configured as LVTTL 3.3 V output, 24 mA, low slew.

2. If the device is AGLN020 and VCCI is 1.8 V, TDO will be configured as LVCMOS 1.8 V output, 4 mA, low slew.

3. If the device is AGLE300 and VCCI is 2.5 V, TDO will be configured as LVCMOS 2.5 V output, 24 mA, low slew.

The test and debug procedure is not limited to the example in Figure 17-5 on page 369. Users can customize the debug and test interface to make it appropriate for their applications. For example, multiple test signals can be registered and then sent out through UTDO, each at a different edge of TCK. In other words, $n$ signals are sampled with an $F_{TCK} / n$ sampling rate. The bandwidth of the information sent out to TDO is always proportional to the frequency of TCK.
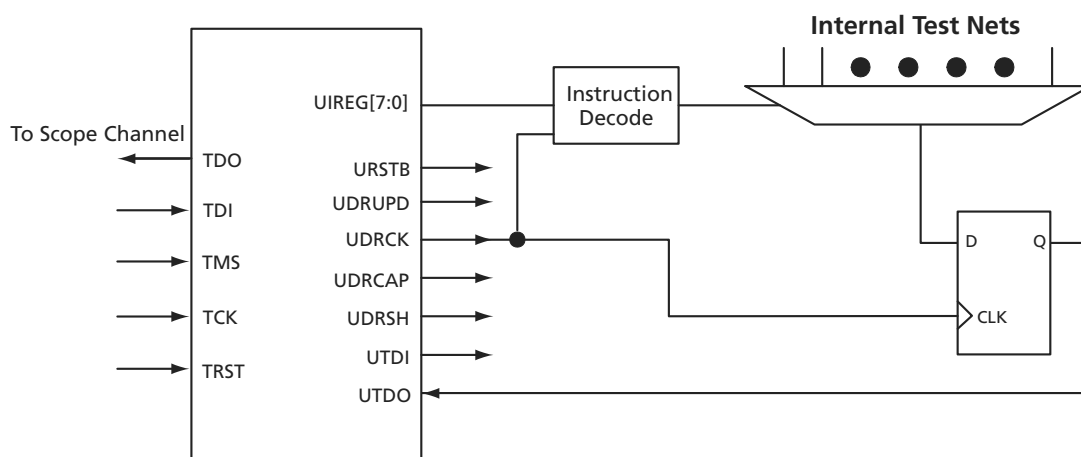


*Figure 17-6 •* **UJTAG Usage Example in Test and Debug Applications**

## 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$^{PLUS}$ devices using JTAG. The guidelines for implementation and design examples are given in the *RAM Initialization and ROM Emulation in ProASIC$^{PLUS}$ 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.
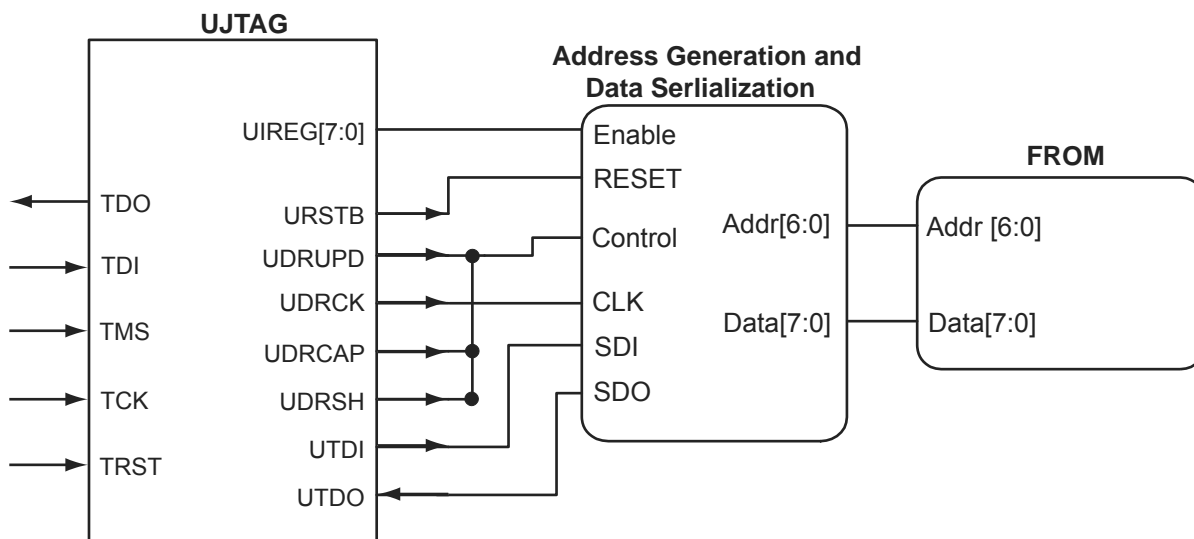


*Figure 17-7 •* **Block Diagram of Using UJTAG to Read FlashROM Contents**