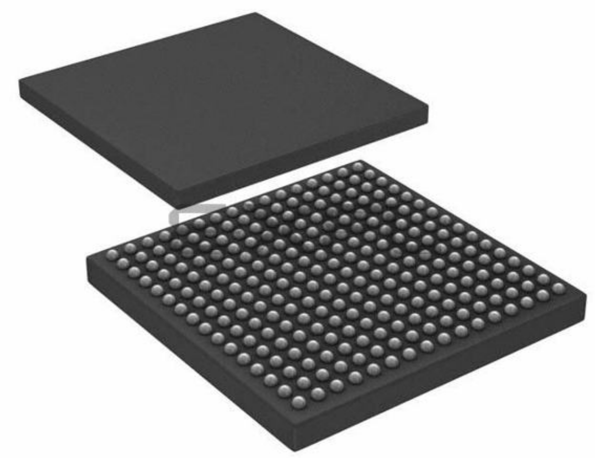


Welcome to E-XFL.COM

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/a3p600l-fg256i

Figure 2-1 shows the concept of FF pin control in Flash*Freeze mode type 1.

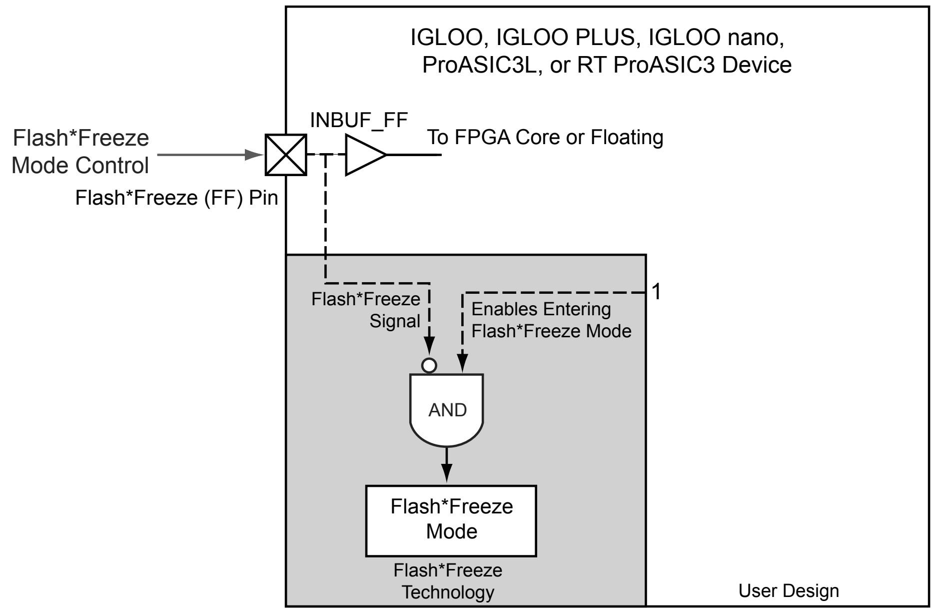


Figure 2-1 **• Flash*Freeze Mode Type 1 – Controlled by the Flash*Freeze Pin**

Figure 2-2 shows the timing diagram for entering and exiting Flash*Freeze mode type 1.

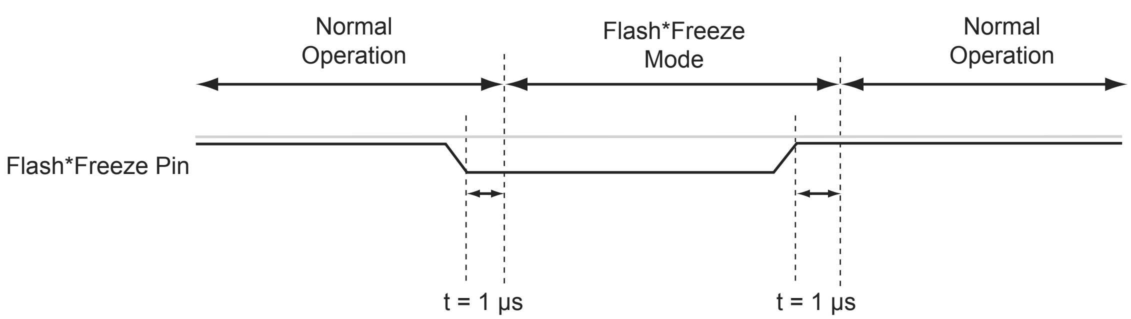


Figure 2-2 **• Flash*Freeze Mode Type 1 – Timing Diagram**

Sleep and Shutdown Modes

Sleep Mode

IGLOO, IGLOO nano, IGLOO PLUS, ProASIC3L, and RT ProASIC3 FPGAs support Sleep mode when device functionality is not required. In Sleep mode, V_{CC} (core voltage), V_{JTAG} (JTAG DC voltage), and VPUMP (programming voltage) are grounded, resulting in the FPGA core being turned off to reduce power consumption. While the device is in Sleep mode, the rest of the system can still be operating and driving the input buffers of the device. The driven inputs do not pull up the internal power planes, and the current draw is limited to minimal leakage current.

Table 2-7 shows the power supply status in Sleep mode.

Table 2-7 • **Sleep Mode—Power Supply Requirement for IGLOO, IGLOO nano, IGLOO PLUS, ProASIC3L, and RT ProASIC3 Devices**

Power Supplies	Power Supply State
VCC	Powered off
VCCI = VMV	Powered on
VJTAG	Powered off
VPUMP	Powered off

Refer to the "Power-Up/-Down Behavior" section on page 33 for more information about I/O states during Sleep mode and the timing diagram for entering and exiting Sleep mode.

Shutdown Mode

Shutdown mode is supported for all IGLOO nano and IGLOO PLUS devices as well the following IGLOO/e devices: AGL015, AGL030, AGLE600, AGLE3000, and A3PE3000L. Shutdown mode can be used by turning off all power supplies when the device function is not needed. Cold-sparing and hot-insertion features enable these devices to be powered down without turning off the entire system. When power returns, the live-at-power-up feature enables operation of the device after reaching the voltage activation point.

Using Sleep and Shutdown Modes in the System

Depending on the power supply and the components used in an application, there are many ways to power on or off the power supplies connected to the device. For example, Figure 2-6 shows how a microprocessor can be used to control a power FET. Microsemi recommends that power FETs with low resistance be used to perform the switching action.

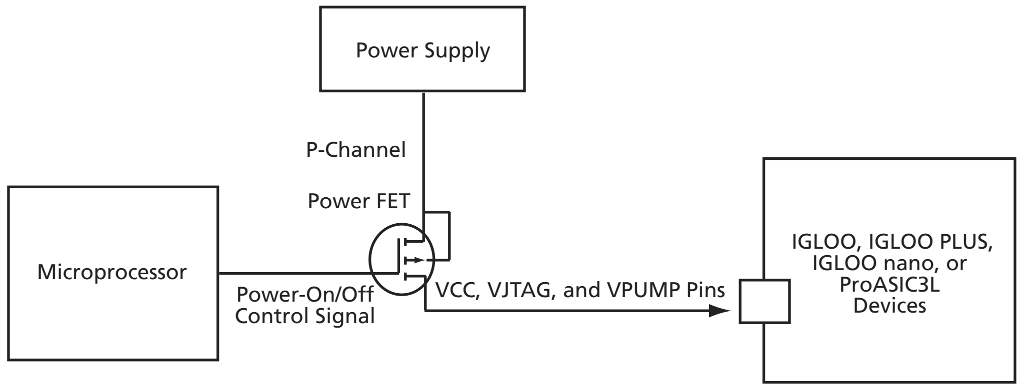


Figure 2-6 **• Controlling Power-On/-Off State Using Microprocessor and Power FET**

Figure 2-7 shows how a microprocessor can be used with a voltage regulator's shutdown pin to turn on or off the power supplies connected to the device.

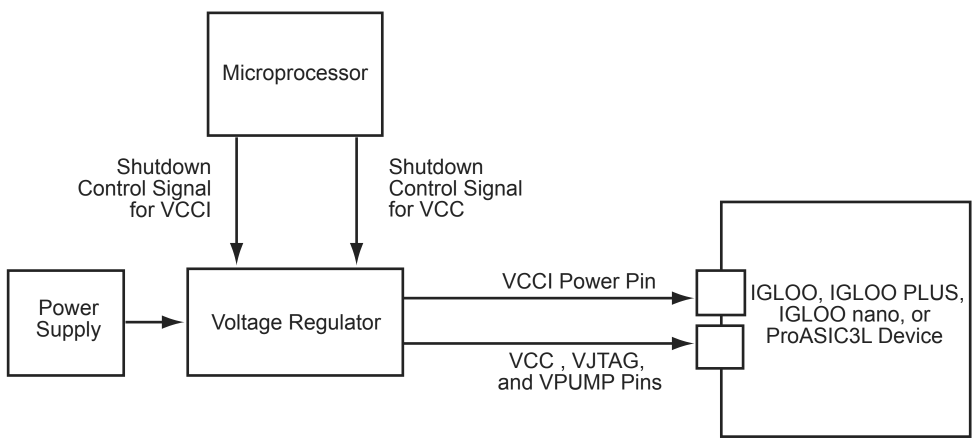
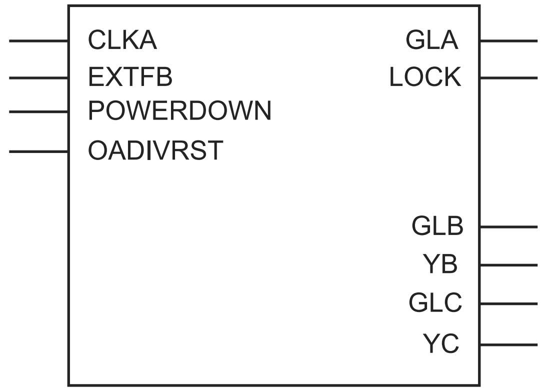


Figure 2-7 **• Controlling Power-On/-Off State Using Microprocessor and Voltage Regulator**

Power-Up/-Down Behavior

By design, all IGLOO, IGLOO nano, IGLOO PLUS, ProASIC3L, and RT ProASIC3 I/Os are in tristate mode before device power-up. The I/Os remain tristated until the last voltage supply (V_{CC} or V_{CCI}) is powered to its activation level. After the last supply reaches its functional level, the outputs exit the tristate mode and drive the logic at the input of the output buffer. The behavior of user I/Os is independent of the V_{CC} and V_{CCI} sequence or the state of other voltage supplies of the FPGA (V_{PUMP} and V_{JTAG}). During power-down, device I/Os become tristated once the first power supply (V_{CC} or V_{CCI}) drops below its deactivation voltage level. The I/O behavior during power-down is also independent of voltage supply sequencing.

Figure 2-8 on page 34 shows a timing diagram when the V_{CC} power supply crosses the activation and deactivation trip points in a typical application when the V_{CC} power supply ramp-rate is 100 µs (ramping from 0 V to 1.5 V in this example). This is the timing diagram for the FPGA entering and exiting Sleep mode, as this function is dependent on powering V_{CC} down or up. Depending on the ramp-rate of the



Note: OAVDIVRST exists only in the Fusion PLL.

Figure 3-15 **• PLLs in Low Power Flash Devices**

You can use the syn_global_buffers attribute in Synplify to specify a maximum number of global macros to be inserted in the netlist. This can also be used to restrict the number of global buffers inserted. In the Synplicity 8.1 version or newer, a new attribute, syn_global_minfanout, has been added for low power flash devices. This enables you to promote only the high-fanout signal to global. However, be aware that you can only have six signals assigned to chip global networks, and the rest of the global signals should be assigned to quadrant global networks. So, if the netlist has 18 global macros, the remaining 12 global macros should have fanout that allows the instances driven by these globals to be placed inside a quadrant.

Global Promotion and Demotion Using PDC

The HDL source file or schematic is the preferred place for defining which signals should be assigned to a clock network using clock macro instantiation. This method is preferred because it is guaranteed to be honored by the synthesis tools and Designer software and stop any replication on this net by the synthesis tool. Note that a signal with fanout may have logic replication if it is not promoted to global during synthesis. In that case, the user cannot promote that signal to global using PDC. See Synplicity Help for details on using this attribute. To help you with global management, Designer allows you to promote a signal to a global network or demote a global macro to a regular macro from the user netlist using the compile options and/or PDC commands.

The following are the PDC constraints you can use to promote a signal to a global network:

1. PDC syntax to promote a regular net to a chip global clock:
   ```
   assign_global_clock –net netname
   ```
   The following will happen during promotion of a regular signal to a global network:
   – If the net is external, the net will be driven by a CLKINT inserted automatically by Compile.
   – The I/O macro will not be changed to CLKBUF macros.
   – If the net is an internal net, the net will be driven by a CLKINT inserted automatically by Compile.
2. PDC syntax to promote a net to a quadrant clock:
   ```
   assign_local_clock –net netname –type quadrant UR|UL|LR|LL
   ```
   This follows the same rule as the chip global clock network.

The following PDC command demotes the clock nets to regular nets.

```
unassign_global_clock -net netname
```

Step 1

Run Synthesis with default options. The Synplicity log shows the following device utilization:

Cell usage:

	cell count	area	count*area
DFN1E1C1	1536	2.0	3072.0
BUFF	278	1.0	278.0
INBUF	10	0.0	0.0
VCC	9	0.0	0.0
GND	9	0.0	0.0
OUTBUF	6	0.0	0.0
CLKBUF	3	0.0	0.0
PLL	2	0.0	0.0
TOTAL	1853		3350.0

Step 2

Run Compile with the **Promote regular nets whose fanout is greater than** option selected in Designer; you will see the following in the Compile report:

```
Device utilization report:
==========================
CORE                Used:   1536  Total:  13824   (11.11%)
IO (W/ clocks)      Used:     19  Total:    147   (12.93%)
Differential IO     Used:      0  Total:     65    (0.00%)
GLOBAL              Used:      8  Total:     18   (44.44%)
PLL                 Used:      2  Total:      2  (100.00%)
RAM/FIFO            Used:      0  Total:     24    (0.00%)
FlashROM            Used:      0  Total:      1    (0.00%)
……………………
The following nets have been assigned to a global resource:
Fanout  Type          Name
--------------------------
1536    INT_NET       Net   : EN_ALL_c
                      Driver: EN_ALL_pad_CLKINT
                      Source: AUTO PROMOTED
1536    SET/RESET_NET Net   : ACLR_c
                      Driver: ACLR_pad_CLKINT
                      Source: AUTO PROMOTED
256     CLK_NET       Net   : QCLK1_c
                      Driver: QCLK1_pad_CLKINT
                      Source: AUTO PROMOTED
256     CLK_NET       Net   : QCLK2_c
                      Driver: QCLK2_pad_CLKINT
                      Source: AUTO PROMOTED
256     CLK_NET       Net   : QCLK3_c
                      Driver: QCLK3_pad_CLKINT
                      Source: AUTO PROMOTED
256     CLK_NET       Net   : $1N14
                      Driver: $1I5/Core
                      Source: ESSENTIAL
256     CLK_NET       Net   : $1N12
                      Driver: $1I6/Core
                      Source: ESSENTIAL
256     CLK_NET       Net   : $1N10
                      Driver: $1I6/Core
                      Source: ESSENTIAL
```

Designer will promote five more signals to global due to high fanout. There are eight signals assigned to global networks.

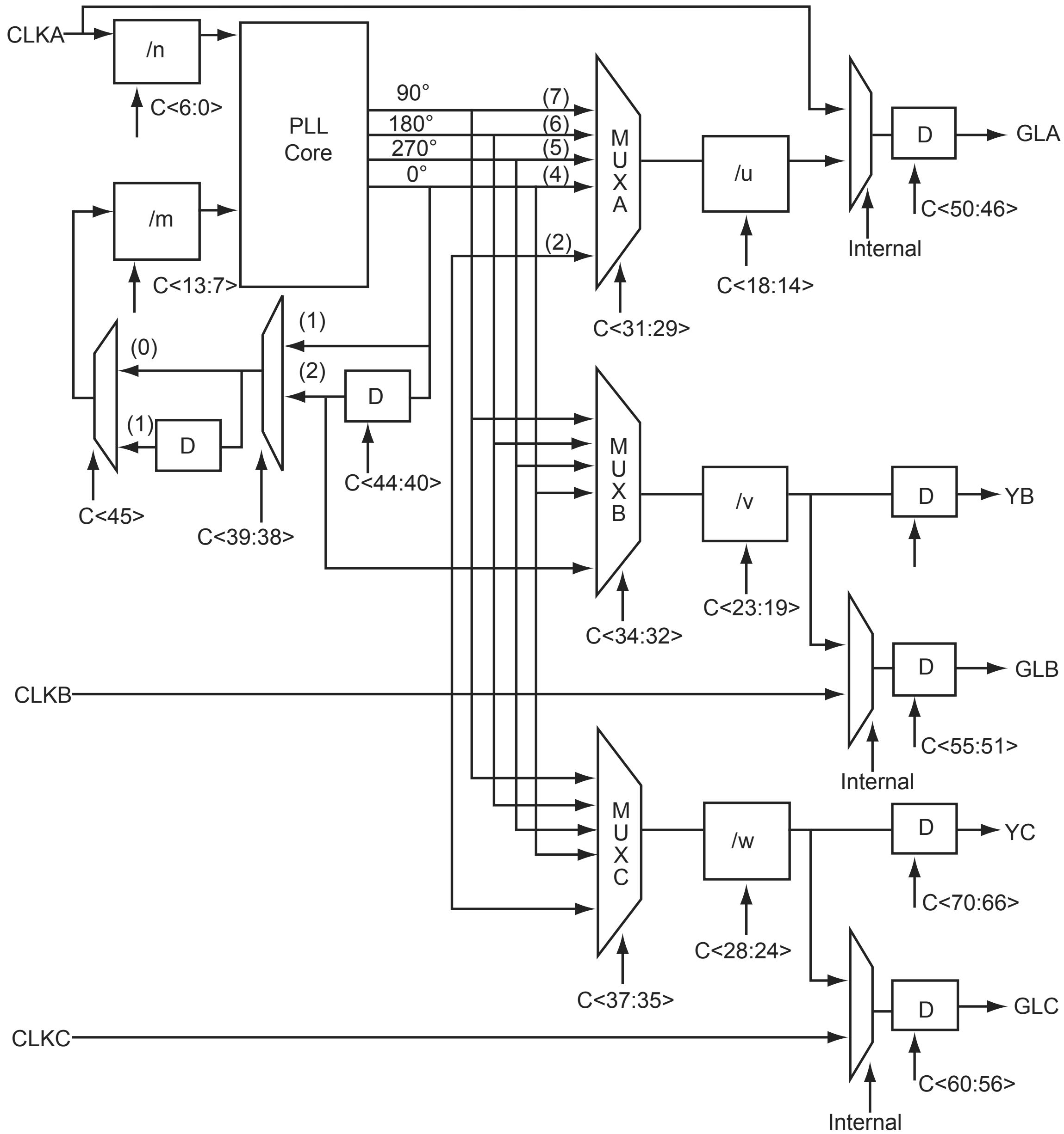


Figure 4-22 **• CCC Block Control Bits – Graphical Representation of Assignments**

Loading the Configuration Register

The most important part of CCC dynamic configuration is to load the shift register properly with the configuration bits. There are different ways to access and load the configuration shift register:

- JTAG interface
- Logic core
- Specific I/O tiles

JTAG Interface

The JTAG interface requires no additional I/O pins. The JTAG TAP controller is used to control the loading of the CCC configuration shift register.

Low power flash devices provide a user interface macro between the JTAG pins and the device core logic. This macro is called UJTAG. A user should instantiate the UJTAG macro in his design to access the configuration register ports via the JTAG pins.

For more information on CCC dynamic reconfiguration using UJTAG, refer to the "UJTAG Applications in Microsemi's Low Power Flash Devices" section on page 363.

Logic Core

If the logic core is employed, the user must design a module to provide the configuration data and control the shifting and updating of the CCC configuration shift register. In effect, this is a user-designed TAP controller, which requires additional chip resources.

Specific I/O Tiles

If specific I/O tiles are used for configuration, the user must provide the external equivalent of a TAP controller. This does not require additional core resources but does use pins.

Shifting the Configuration Data

To enter a new configuration, all 81 bits must shift in via SDIN. After all bits are shifted, SSHIFT must go LOW and SUPDATE HIGH to enable the new configuration. For simulation purposes, bits <71:73> and <77:80> are "don't care."

The SUPDATE signal must be LOW during any clock cycle where SSHIFT is active. After SUPDATE is asserted, it must go back to the LOW state until a new update is required.

PLL Configuration Bits Description

***Table 4-8* • Configuration Bit Descriptions for the CCC Blocks**

Config. Bits	Signal	Name	Description
<88:87>	GLMUXCFG [1:0][1]	NGMUX configuration	The configuration bits specify the input clocks to the NGMUX (refer to Table 4-17 on page 110).[2]
86	OCDIVHALF[1]	Division by half	When the PLL is bypassed, the 100 MHz RC oscillator can be divided by the divider factor in Table 4-18 on page 111.
85	OBDIVHALF[1]	Division by half	When the PLL is bypassed, the 100 MHz RC oscillator can be divided by a 0.5 factor (refer to Table 4-18 on page 111).
84	OADIVHALF[1]	Division by half	When the PLL is bypassed, the 100 MHz RC oscillator can be divided by certain 0.5 factor (refer to Table 4-16 on page 110).

Notes:

1. *The <88:81> configuration bits are only for the Fusion dynamic CCC.*
2. *This value depends on the input clock source, so Layout must complete before these bits can be set. After completing Layout in Designer, generate the "CCC_Configuration" report by choosing **Tools** > **Report** > **CCC_Configuration**. The report contains the appropriate settings for these bits.*

Figure 4-36 • **Second-Stage PLL Showing Input of 256 MHz from First Stage and Final Output of 280 MHz**

Figure 4-37 shows the simulation results, where the first PLL's output period is 3.9 ns (~256 MHz), and the stage 2 (final) output period is 3.56 ns (~280 MHz).

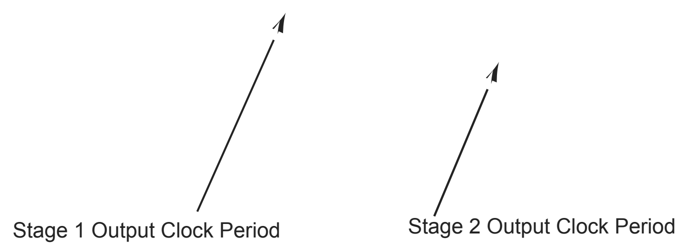


Figure 4-37 • **Model*Sim* Simulation Results**

recommended, since it reduces the complexity of the user interface block and the board-level JTAG driver.

Moreover, using an internal counter for address generation speeds up the initialization procedure, since the user only needs to import the data through the JTAG port.

The designer may use different methods to select among the multiple RAM blocks. Using counters along with demultiplexers is one approach to set the write enable signals. Basically, the number of RAM blocks needing initialization determines the most efficient approach. For example, if all the blocks are initialized with the same data, one enable signal is enough to activate the write procedure for all of them at the same time. Another alternative is to use different opcodes to initialize each memory block. For a small number of RAM blocks, using counters is an optimal choice. For example, a ring counter can be used to select from multiple RAM blocks. The clock driver of this counter needs to be controlled by the address generation process.

Once the addressing of one block is finished, a clock pulse is sent to the (ring) counter to select the next memory block.

Figure 6-9 illustrates a simple block diagram of an interface block between UJTAG and RAM blocks.

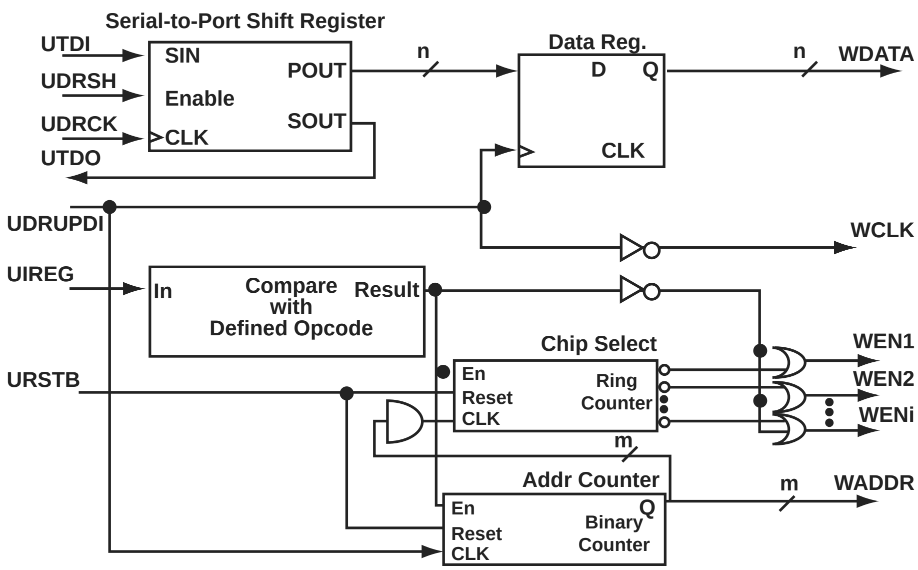


Figure 6-9 • **Block Diagram of a Sample User Interface**

In the circuit shown in Figure 6-9, the shift register is enabled by the UDRSH output of the UJTAG macro. The counters and chip select outputs are controlled by the value of the TAP Instruction Register. The comparison block compares the UIREG value with the "start initialization" opcode value (defined by the user). If the result is true, the counters start to generate addresses and activate the WEN inputs of appropriate RAM blocks.

The UDRUPD output of the UJTAG macro, also shown in Figure 6-9, is used for generating the write clock (WCLK) and synchronizing the data register and address counter with WCLK. UDRUPD is HIGH when the TAP Controller is in the Data Register Update state, which is an indication of completing the loading of one data word. Once the TAP Controller goes into the Data Register Update state, the UDRUPD output of the UJTAG macro goes HIGH. Therefore, the pipeline register and the address counter place the proper data and address on the outputs of the interface block. Meanwhile, WCLK is defined as the inverted UDRUPD. This will provide enough time (equal to the UDRUPD HIGH time) for the data and address to be placed at the proper ports of the RAM block before the rising edge of WCLK. The inverter is not required if the RAM blocks are clocked at the falling edge of the write clock. An example of this is described in the "Example of RAM Initialization" section on page 166.

Example: For a bus consisting of 20 equidistant loads, the terminations given in EQ 1 provide the required differential voltage, in worst-case industrial operating conditions, at the farthest receiver:

R_S = 60 Ω, R_T = 70 Ω, given Z_O = 50 Ω (2") and Z_{stub} = 50 Ω (~1.5").

EQ 1

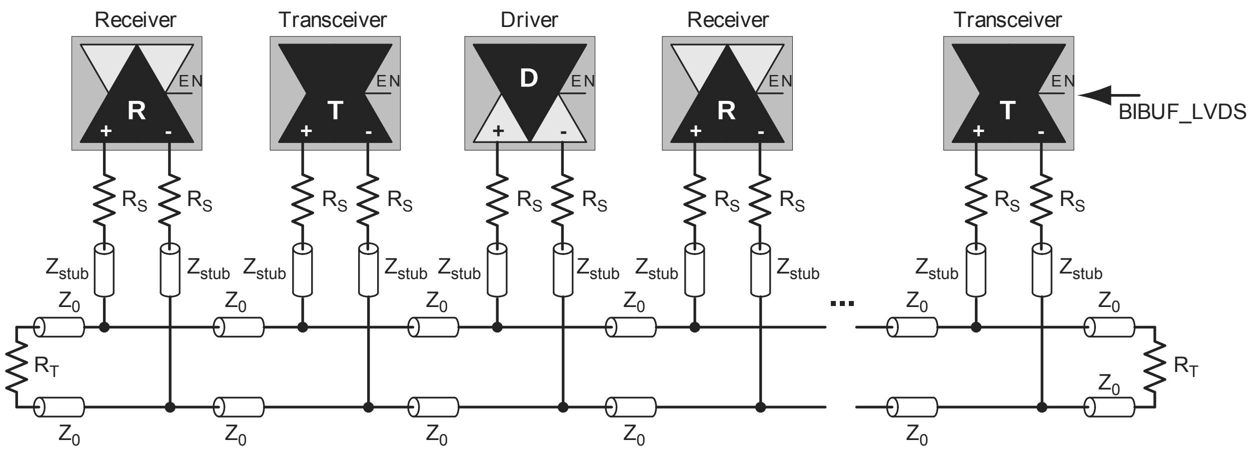


Figure 7-8 **• A B-LVDS/M-LVDS Multipoint Application Using LVDS I/O Buffers**

- In Active and Static modes:
  - Input buffers with pull-up, driven Low
  - Input buffers with pull-down, driven High
  - Bidirectional buffers with pull-up, driven Low
  - Bidirectional buffers with pull-down, driven High
  - Output buffers with pull-up, driven Low
  - Output buffers with pull-down, driven High
  - Tristate buffers with pull-up, driven Low
  - Tristate buffers with pull-down, driven High
- In Flash*Freeze mode:
  - Input buffers with pull-up, driven Low
  - Input buffers with pull-down, driven High
  - Bidirectional buffers with pull-up, driven Low
  - Bidirectional buffers with pull-down, driven High

Electrostatic Discharge Protection

Low power flash devices are tested per JEDEC Standard JESD22-A114-B.

These devices contain clamp diodes at every I/O, global, and power pad. Clamp diodes protect all device pads against damage from ESD as well as from excessive voltage transients.

All IGLOO and ProASIC3 devices are tested to the Human Body Model (HBM) and the Charged Device Model (CDM).

Each I/O has two clamp diodes. One diode has its positive (P) side connected to the pad and its negative (N) side connected to VCCI. The second diode has its P side connected to GND and its N side connected to the pad. During operation, these diodes are normally biased in the off state, except when transient voltage is significantly above VCCI or below GND levels.

In 30K gate devices, the first diode is always off. In other devices, the clamp diode is always on and cannot be switched off.

By selecting the appropriate I/O configuration, the diode is turned on or off. Refer to Table 7-12 on page 193 for more information about the I/O standards and the clamp diode.

The second diode is always connected to the pad, regardless of the I/O configuration selected.

***Table 7-12* • I/O Hot-Swap and 5 V Input Tolerance Capabilities in IGLOO and ProASIC3 Devices**

I/O Assignment	Clamp Diode [1]		Hot Insertion		5 V Input Tolerance [2]		Input and Output Buffer
	AGL030 and A3P030	Other IGLOO and ProASIC3 Devices	AGL015 and AGL030	Other IGLOO Devices and All ProASIC3	AGL030 and A3P030	Other IGLOO and ProASIC3 Devices	
3.3 V LVTTL/LVCMOS	No	Yes	Yes	No	Yes [2]	Yes [2]	Enabled/Disabled
3.3 V PCI, 3.3 V PCI-X	N/A	Yes	N/A	No	N/A	Yes [2]	Enabled/Disabled
LVCMOS 2.5 V [5]	No	Yes	Yes	No	Yes [2]	Yes [4]	Enabled/Disabled
LVCMOS 2.5 V/5.0 V [6]	N/A	Yes	N/A	No	N/A	Yes [4]	Enabled/Disabled
LVCMOS 1.8 V	No	Yes	Yes	No	No	No	Enabled/Disabled
LVCMOS 1.5 V	No	Yes	Yes	No	No	No	Enabled/Disabled
Differential, LVDS/ B-LVDS/M-LVDS/LVPECL	N/A	Yes	N/A	No	N/A	No	Enabled/Disabled

Notes:

1. *The clamp diode is always off for the AGL030 and A3P030 device and always active for other IGLOO and ProASIC3 devices.*
2. *Can be implemented with an external IDT bus switch, resistor divider, or Zener with resistor.*
3. *Refer to Table 7-8 on page 189 to Table 7-11 on page 190 for device-compliant information.*
4. *Can be implemented with an external resistor and an internal clamp diode.*
5. *The LVCMOS 2.5 V I/O standard is supported by the 30 k gate devices only; select the LVCMOS25 macro.*
6. *The LVCMOS 2.5 V / 5.0 V I/O standard is supported by all IGLOO and ProASIC3 devices except 30K gate devices; select the LVCMOS5 macro.*

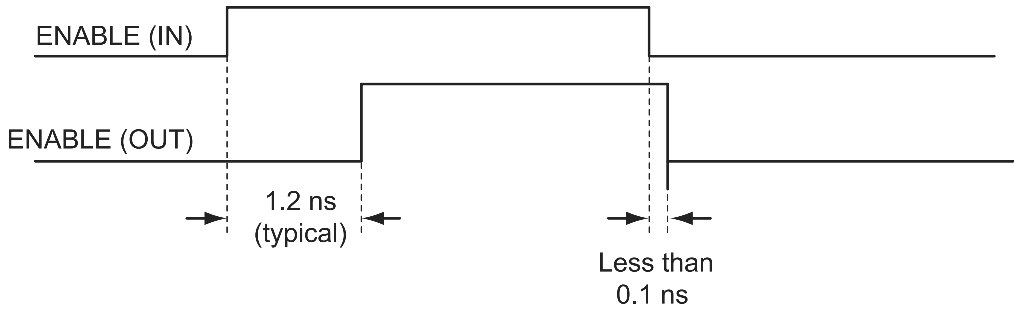


***Figure 7-15* • Timing Diagram (option 2: enables skew circuit)**

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 7-16 presents an example of the skew circuit implementation in a bidirectional communication system. Figure 7-17 on page 201 shows how bus contention is created, and Figure 7-18 on page 201 shows how it can be avoided with the skew circuit.

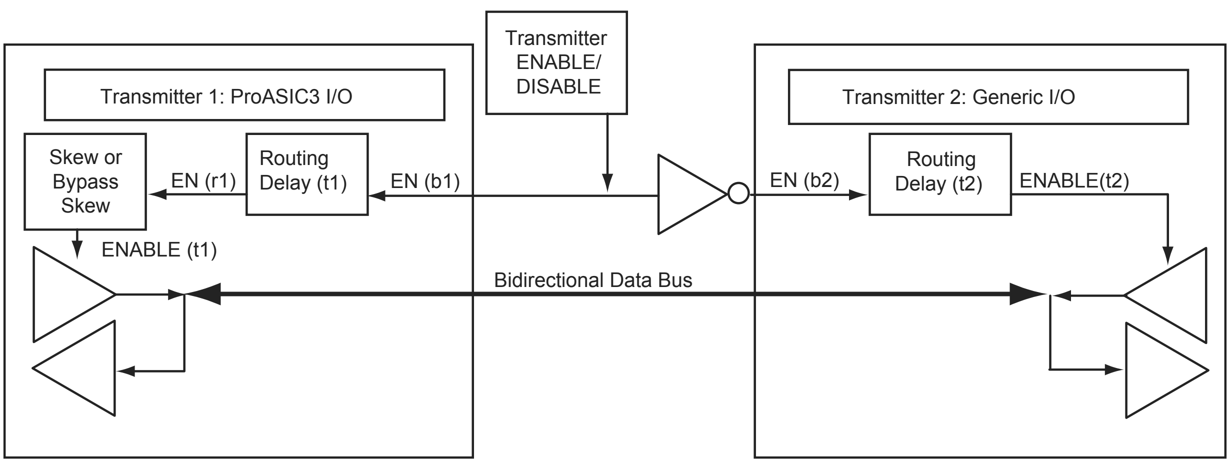


***Figure 7-16* • Example of Implementation of Skew Circuits in Bidirectional Transmission Systems Using IGLOO or ProASIC3 Devices**

I/O Software Support

In Microsemi's Libero software, default settings have been defined for the various I/O standards supported. Changes can be made to the default settings via the use of attributes; however, not all I/O attributes are applicable for all I/O standards. Table 7-17 list the valid I/O attributes that can be manipulated by the user for each I/O standard.

Single-ended I/O standards in low power flash devices support up to five different drive strengths.

Table 7-17 **• IGLOO and ProASIC3 I/O Attributes vs. I/O Standard Applications**

I/O Standard	SLEW (output only)	OUT_DRIVE (output only)	SKEW (all macros with OE)	RES_PULL	OUT_LOAD (output only)	COMBINE_REGISTER
LVTTL/LVCMOS 3.3 V	✓	✓	✓	✓	✓	✓
LVCMOS 2.5 V	✓	✓	✓	✓	✓	✓
LVCMOS 2.5/5.0 V	✓	✓	✓	✓	✓	✓
LVCMOS 1.8 V	✓	✓	✓	✓	✓	✓
LVCMOS 1.5 V	✓	✓	✓	✓	✓	✓
PCI (3.3 V)			✓		✓	✓
PCI-X (3.3 V)	✓		✓		✓	✓
LVDS, B-LVDS, M-LVDS			✓			✓
LVPECL						✓

Note: Applies to all 30 k gate devices.

Table 7-18 lists the default values for the above selectable I/O attributes as well as those that are preset for that I/O standard. See Table 7-14 on page 203 to Table 7-16 on page 203 for SLEW and OUT_DRIVE settings.

Table 7-18 **• IGLOO and ProASIC3 I/O Default Attributes**

I/O Standards	SLEW (output only)	OUT_DRIVE (output only)	SKEW (tribuf and bibuf only)	RES_PULL	OUT_LOAD (output only)	COMBINE_REGISTER
LVTTL/LVCMOS 3.3 V	See Table 7-14 on page 203 to Table 7-16 on page 203.	See Table 7-14 on page 203 to Table 7-16 on page 203.	Off	None	35 pF	–
LVCMOS 2.5 V			Off	None	35 pF	–
LVCMOS 2.5/5.0 V			Off	None	35 pF	–
LVCMOS 1.8 V			Off	None	35 pF	–
LVCMOS 1.5 V			Off	None	35 pF	–
PCI (3.3 V)			Off	None	10 pF	–
PCI-X (3.3 V)			Off	None	10 pF	–
LVDS, B-LVDS, M-LVDS			Off	None	0 pF	–
LVPECL			Off	None	0 pF	–

Implementing I/Os in Microsemi Software

Microsemi Libero SoC software is integrated with design entry tools such as the SmartGen macro builder, the ViewDraw schematic entry tool, and an HDL editor. It is also integrated with the synthesis and Designer tools. In this section, all necessary steps to implement the I/Os are discussed.

Design Entry

There are three ways to implement I/Os in a design:

1. Use the SmartGen macro builder to configure I/Os by generating specific I/O library macros and then instantiating them in top-level code. This is especially useful when creating I/O bus structures.
2. Use an I/O buffer cell in a schematic design.
3. Manually instantiate specific I/O macros in the top-level code.

If technology-specific macros, such as INBUF_LVCMOS33 and OUTBUF_PCI, are used in the HDL code or schematic, the user will not be able to change the I/O standard later on in Designer. If generic I/O macros are used, such as INBUF, OUTBUF, TRIBUF, CLKBUF, and BIBUF, the user can change the I/O standard using the Designer I/O Attribute Editor tool.

Using SmartGen for I/O Configuration

The SmartGen tool in Libero SoC provides a GUI-based method of configuring the I/O attributes. The user can select certain I/O attributes while configuring the I/O macro in SmartGen. The steps to configure an I/O macro with specific I/O attributes are as follows:

1. Open Libero SoC.
2. On the left-hand side of the Catalog View, select **I/O**, as shown in Figure 9-2.

***Figure 9-2* • SmartGen Catalog**

Security Support in Flash-Based Devices

The flash FPGAs listed in Table 12-1 support the security feature and the functions described in this document.

***Table 12-1* • 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	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.*

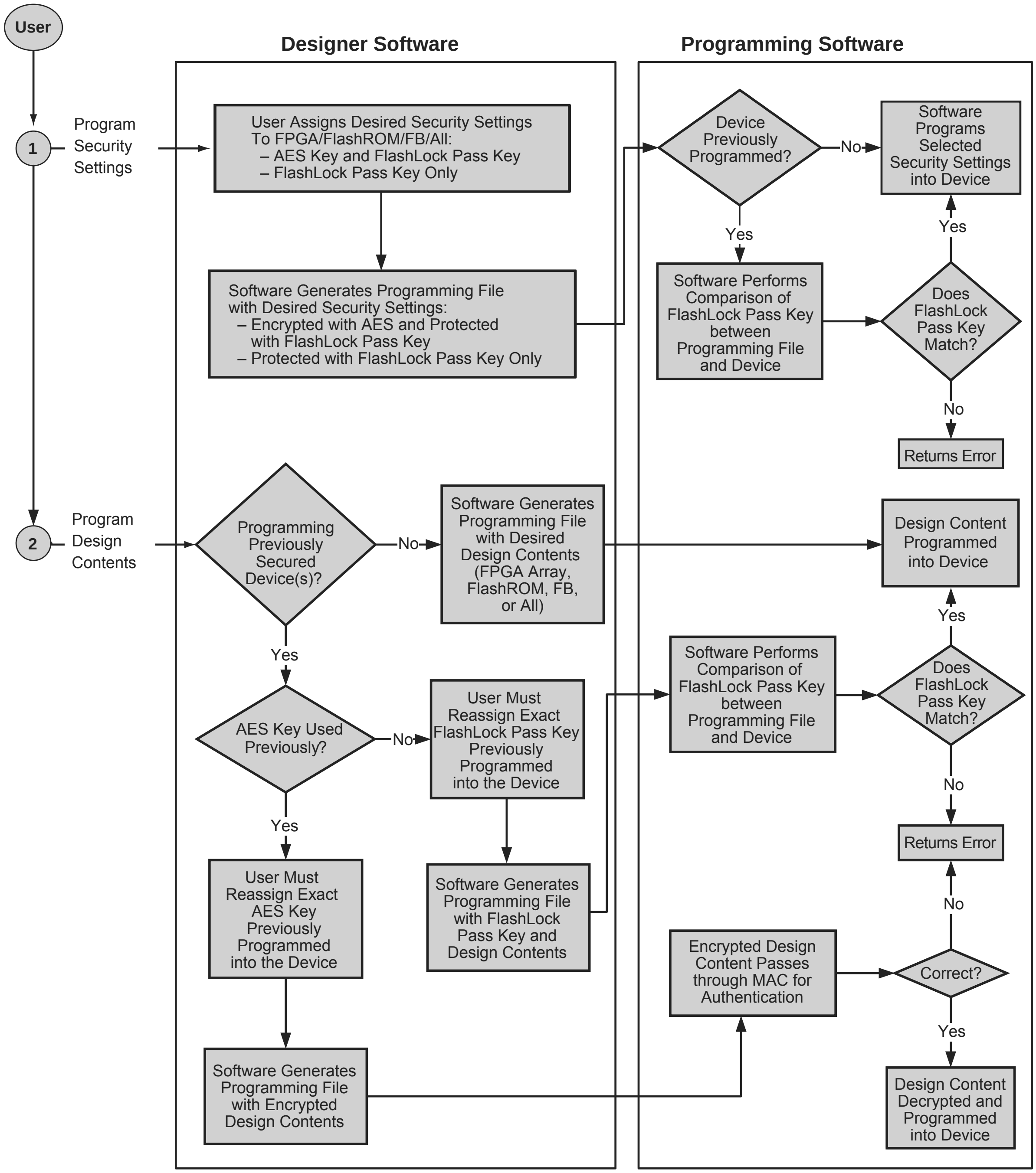
IGLOO Terminology

In documentation, the terms IGLOO series and IGLOO devices refer to all of the IGLOO devices as listed in Table 12-1. 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 12-1. 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*.



Note: *If programming the Security Header only, just perform sub-flow 1.*
If programming design content only, just perform sub-flow 2.

***Figure 12-9* • Security Programming Flows**

Figure 12-10 **• All Silicon Features Selected for IGLOO and ProASIC3 Devices**

Figure 12-11 **• All Silicon Features Selected for Fusion**

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

Introduction

In Fusion, IGLOO, and ProASIC3 devices, there is bidirectional access from the JTAG port to the core VersaTiles during normal operation of the device (Figure 17-1). User JTAG (UJTAG) is the ability for the design to use the JTAG ports for access to the device for updates, etc. While regular JTAG is used, the UJTAG tiles, located at the southeast area of the die, are directly connected to the JTAG Test Access Port (TAP) Controller in normal operating mode. As a result, all the functional blocks of the device, such as Clock Conditioning Circuits (CCCs) with PLLs, SRAM blocks, embedded FlashROM, flash memory blocks, and I/O tiles, can be reached via the JTAG ports. The UJTAG functionality is available by instantiating the UJTAG macro directly in the source code of a design. Access to the FPGA core VersaTiles from the JTAG ports enables users to implement different applications using the TAP Controller (JTAG port). This document introduces the UJTAG tile functionality and discusses a few application examples. However, the possible applications are not limited to what is presented in this document. UJTAG can serve different purposes in many designs as an elementary or auxiliary part of the design. For detailed usage information, refer to the "Boundary Scan in Low Power Flash Devices" section on page 357.

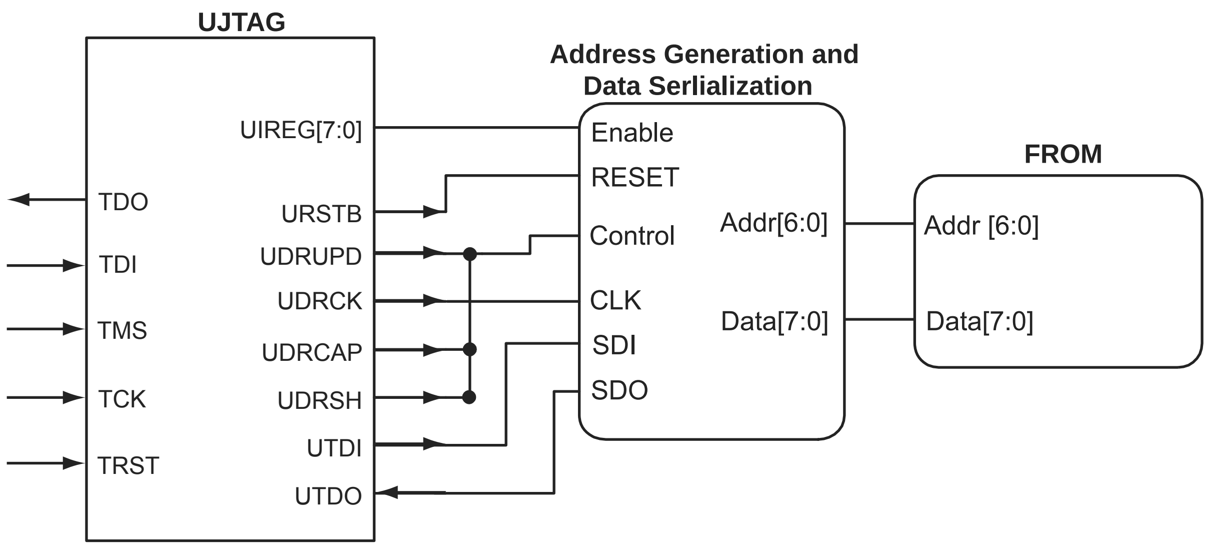


Figure 17-1 • **Block Diagram of Using UJTAG to Read FlashROM Contents**