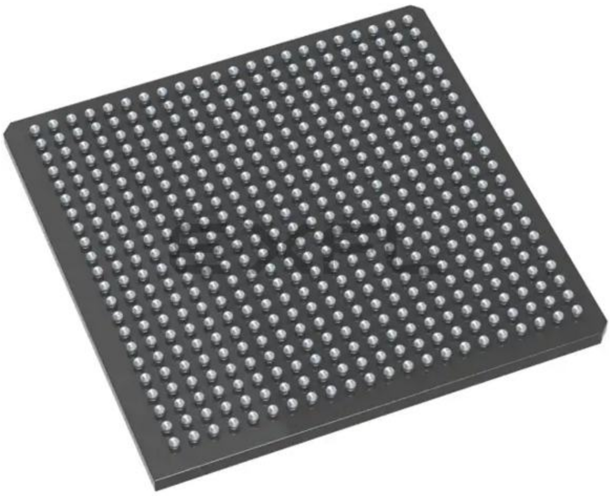
Microchip Technology - A3P600L-FG484 Datasheet



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	Active
Number of LABs/CLBs	-
Number of Logic Elements/Cells	-
Total RAM Bits	110592
Number of I/O	235
Number of Gates	600000
Voltage - Supply	1.14V ~ 1.575V
Mounting Type	Surface Mount
Operating Temperature	0°C ~ 85°C (TJ)
Package / Case	484-BGA
Supplier Device Package	484-FPBGA (23x23)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/a3p600l-fg484

Table 3-3 **• Quadrant Global Pin Name**

I/O Type	Beginning of I/O Name	Notes
Single-Ended	GAAO/IOuxwByVz GAA1/IOuxwByVz GAA2/IOuxwByVz	Only one of the I/Os can be directly connected to a quadrant global at a time
	GABO/IOuxwByVz GAB1/IOuxwByVz GAB2/IOuxwByVz	Only one of the I/Os can be directly connected to a quadrant global at a time.
	GAC0/IOuxwByVz GAC1/IOuxwByVz GAC2/IOuxwByVz	Only one of the I/Os can be directly connected to a quadrant global at a time.
	GBAO/IOuxwByVz GBA1/IOuxwByVz GBA2/IOuxwByVz	Only one of the I/Os can be directly connected to a global at a time.
	GBBO/IOuxwByVz GBB1/IOuxwByVz GBB2/IOuxwByVz	Only one of the I/Os can be directly connected to a global at a time.
	GBC0/IOuxwByVz GBC1/IOuxwByVz GBC2/IOuxwByVz	Only one of the I/Os can be directly connected to a global at a time.
	GDAO/IOuxwByVz GDA1/IOuxwByVz GDA2/IOuxwByVz	Only one of the I/Os can be directly connected to a global at a time.
	GDBO/IOuxwByVz GDB1/IOuxwByVz GDB2/IOuxwByVz	Only one of the I/Os can be directly connected to a global at a time.
	GDC0/IOuxwByVz GDC1/IOuxwByVz GDC2/IOuxwByVz	Only one of the I/Os can be directly connected to a global at a time.
	GEAO/IOuxwByVz GEA1/IOuxwByVz GEA2/IOuxwByVz	Only one of the I/Os can be directly connected to a global at a time.
	GEBO/IOuxwByVz GEB1/IOuxwByVz GEB2/IOuxwByVz	Only one of the I/Os can be directly connected to a global at a time.
	GEC0/IOuxwByVz GEC1/IOuxwByVz GEC2/IOuxwByVz	Only one of the I/Os can be directly connected to a global at a time.

Note: Only one of the I/Os can be directly connected to a quadrant at a time.

Table 3-3 **• Quadrant Global Pin Name (continued)**

Differential I/O Pairs	GAAO/IOuxwByVz GAA1/IOuxwByVz	The output of the different pair will drive the global.
	GABO/IOuxwByVz GAB1/IOuxwByVz	The output of the different pair will drive the global.
	GACO/IOuxwByVz GAC1/IOuxwByVz	The output of the different pair will drive the global.
	GBAO/IOuxwByVz GBA1/IOuxwByVz	The output of the different pair will drive the global.
	GBBO/IOuxwByVz GBB1/IOuxwByVz	The output of the different pair will drive the global.
	GBCO/IOuxwByVz GBC1/IOuxwByVz	The output of the different pair will drive the global.
	GDAO/IOuxwByVz GDA1/IOuxwByVz	The output of the different pair will drive the global.
	GDBO/IOuxwByVz GDB1/IOuxwByVz	The output of the different pair will drive the global.
	GDCO/IOuxwByVz GDC1/IOuxwByVz	The output of the different pair will drive the global.
	GEAO/IOuxwByVz GEA1/IOuxwByVz	The output of the different pair will drive the global.
	GEBO/IOuxwByVz GEB1/IOuxwByVz	The output of the different pair will drive the global.
	GECO/IOuxwByVz GEC1/IOuxwByVz	The output of the different pair will drive the global.

Note: Only one of the I/Os can be directly connected to a quadrant at a time.

Unused Global I/O Configuration

The unused clock inputs behave similarly to the unused Pro I/Os. The Microsemi Designer software automatically configures the unused global pins as inputs with pull-up resistors if they are not used as regular I/O.

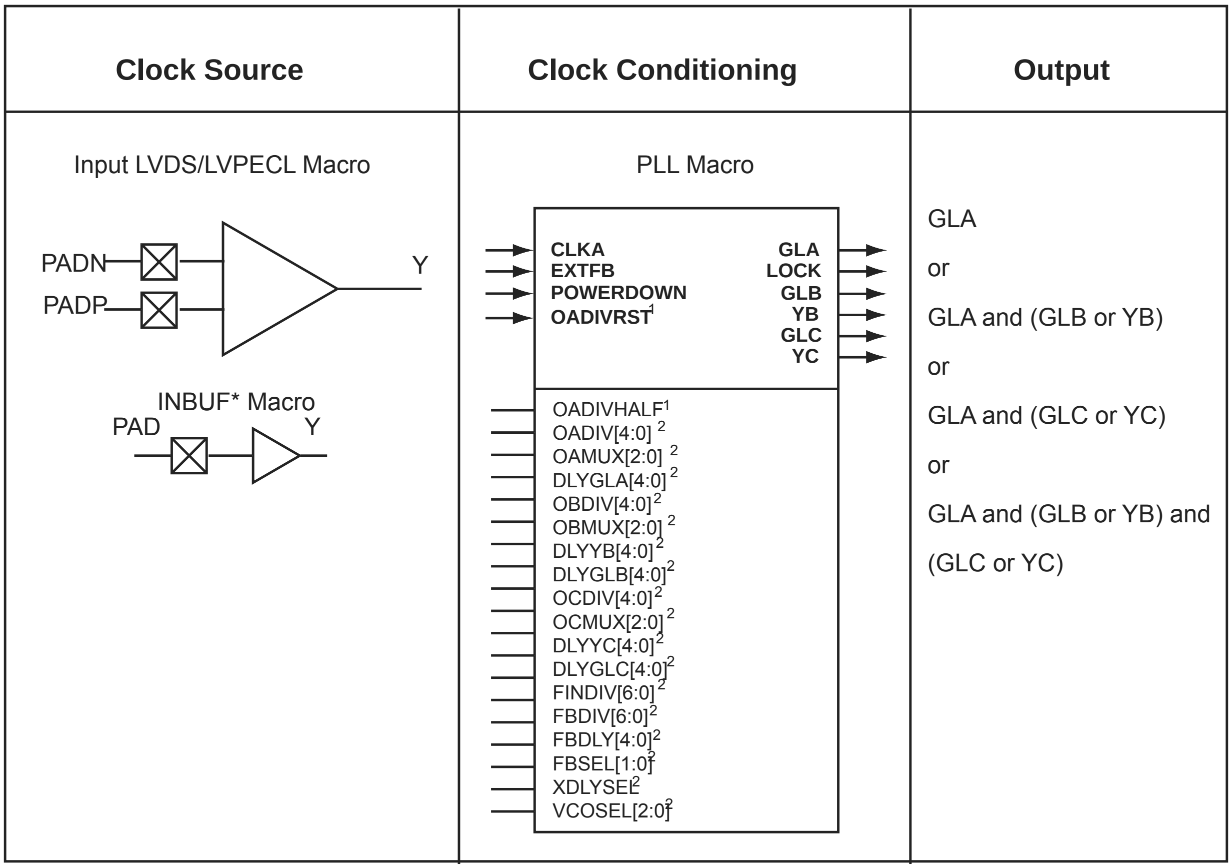
I/O Banks and Global I/O Standards

In low power flash devices, any I/O or internal logic can be used to drive the global network. However, only the global macro placed at the global pins will use the hardwired connection between the I/O and global network. Global signal (signal driving a global macro) assignment to I/O banks is no different from regular I/O assignment to I/O banks with the exception that you are limited to the pin placement location available. Only global signals compatible with both the VCCI and VREF standards can be assigned to the same bank.

Global Buffers with PLL Function

Clocks requiring frequency synthesis or clock adjustments can utilize the PLL core before connecting to the global / quadrant global networks. A maximum of 18 CCC global buffers can be instantiated in a device—three per CCC and up to six CCCs per device. Each PLL core can generate up to three global/quadrant clocks, while a clock delay element provides one.

The PLL functionality of the clock conditioning block is supported by the PLL macro.



Notes:

1. *For Fusion only.*
2. *Refer to the* IGLOO, ProASIC3, SmartFusion, and Fusion Macro Library Guide *for more information.*
3. *For INBUF* driving a PLL macro or CLKDLY macro, the I/O will be hard-routed to the CCC; i.e., will be placed by software to a dedicated Global I/O.*
4. *IGLOO nano and ProASIC3 nano devices do not support differential inputs.*

***Figure 4-4* • CCC Options: Global Buffers with PLL**

The PLL macro provides five derived clocks (three independent) from a single reference clock. The PLL macro also provides power-down input and lock output signals. The additional inputs shown on the macro are configuration settings, which are configured through the use of SmartGen. For manual setting of these bits refer to the *IGLOO, ProASIC3, SmartFusion, and Fusion Macro Library Guide* for details.

Figure 4-6 on page 87 illustrates the various clock output options and delay elements.

```
Primary Clock Output Delay from CLKA -3.020

Secondary1 Clock frequency 40.000
Secondary1 Clock Phase Shift 0.000
Secondary1 Clock Global Output Delay from CLKA 2.515
```

Next, perform simulation in Model*Sim* to verify the correct delays. Figure 4-30 shows the simulation results. The delay values match those reported in the SmartGen PLL Wizard.

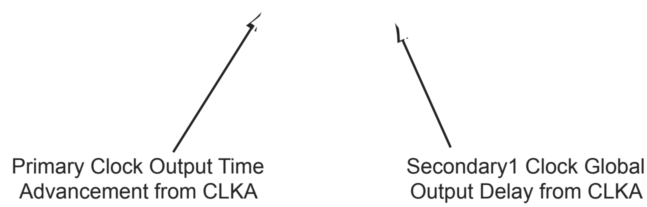


***Figure 4-30* • Model*Sim* Simulation Results**

The timing can also be analyzed using SmartTime in Designer. The user should import the synthesized netlist to Designer, perform Compile and Layout, and then invoke SmartTime. Go to **Tools** > **Options** and change the maximum delay operating conditions to **Typical Case**. Then expand the Clock-to-Out paths of GLA and GLB and the individual components of the path delays are shown. The path of GLA is shown in Figure 4-31 on page 123 displaying the same delay value.

FlashROM Design Flow

The Microsemi Libero System-on-Chip (SoC) software has extensive FlashROM support, including FlashROM generation, instantiation, simulation, and programming. Figure 5-9 shows the user flow diagram. In the design flow, there are three main steps:

1. FlashROM generation and instantiation in the design
2. Simulation of FlashROM design
3. Programming file generation for FlashROM design

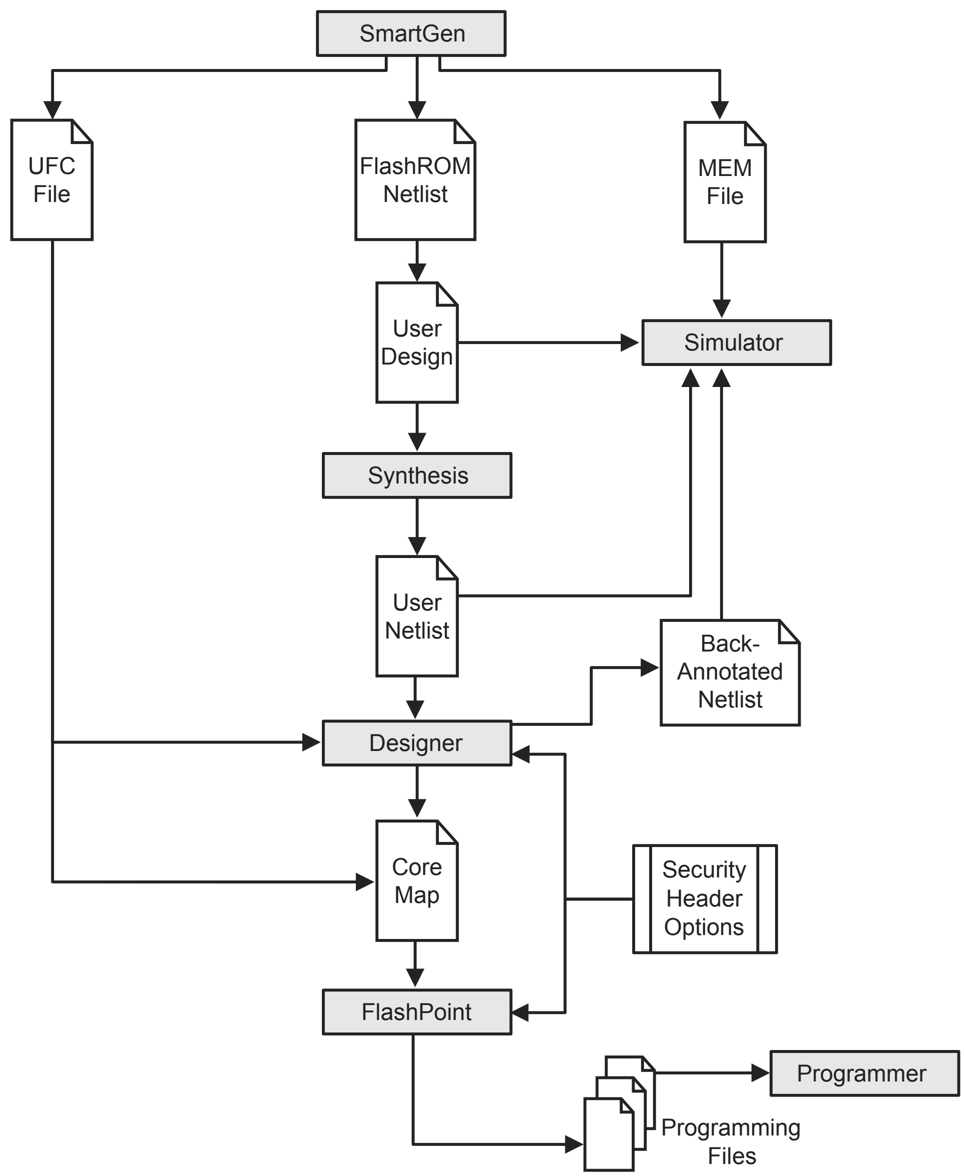


Figure 5-9 **• FlashROM Design Flow**

FlashROM Generation and Instantiation in the Design

The SmartGen core generator, available in Libero SoC and Designer, is the only tool that can be used to generate the FlashROM content. SmartGen has several user-friendly features to help generate the FlashROM contents. Instead of selecting each byte and assigning values, you can create a region within a page, modify the region, and assign properties to that region. The FlashROM user interface, shown in Figure 5-10, includes the configuration grid, existing regions list, and properties field. The properties field specifies the region-specific information and defines the data used for that region. You can assign values to the following properties:

1. Static Fixed Data—Enables you to fix the data so it cannot be changed during programming time. This option is useful when you have fixed data stored in this region, which is required for the operation of the design in the FPGA. Key storage is one example.
2. Static Modifiable Data—Select this option when the data in a particular region is expected to be static data (such as a version number, which remains the same for a long duration but could conceivably change in the future). This option enables you to avoid changing the value every time you enter new data.
3. Read from File—This provides the full flexibility of FlashROM usage to the customer. If you have a customized algorithm for generating the FlashROM data, you can specify this setting. You can then generate a text file with data for as many devices as you wish to program, and load that into the FlashPoint programming file generation software to get programming files that include all the data. SmartGen will optionally pass the location of the file where the data is stored if the file is specified in SmartGen. Each text file has only one type of data format (binary, decimal, hex, or ASCII text). The length of each data file must be shorter than or equal to the selected region length. If the data is shorter than the selected region length, the most significant bits will be padded with 0s. For multiple text files for multiple regions, the first lines are for the first device. In SmartGen, **Load Sim. Value From File** allows you to load the first device data in the MEM file for simulation.
4. Auto Increment/Decrement—This scenario is useful when you specify the contents of FlashROM for a large number of devices in a series. You can specify the step value for the serial number and a maximum value for inventory control. During programming file generation, the actual number of devices to be programmed is specified and a start value is fed to the software.

Figure 5-10 **• SmartGen GUI of the FlashROM**

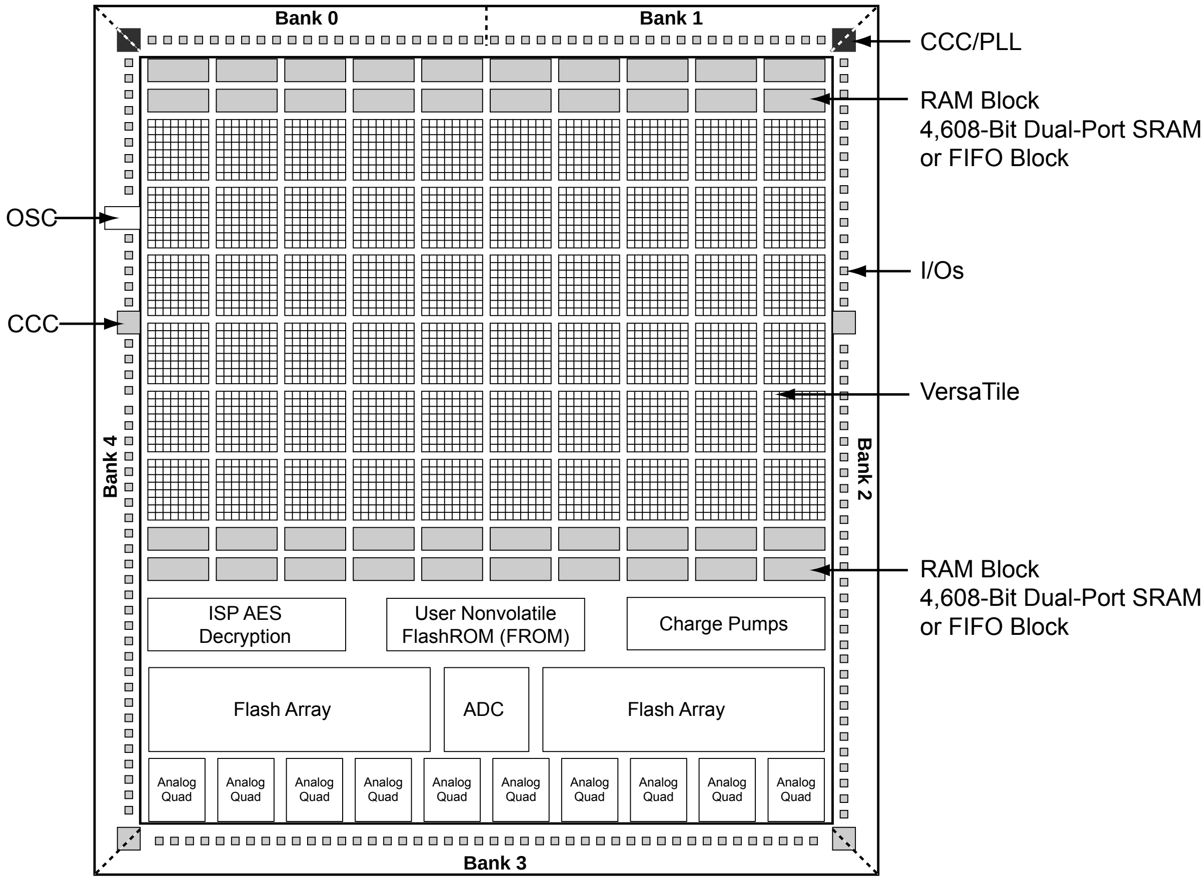
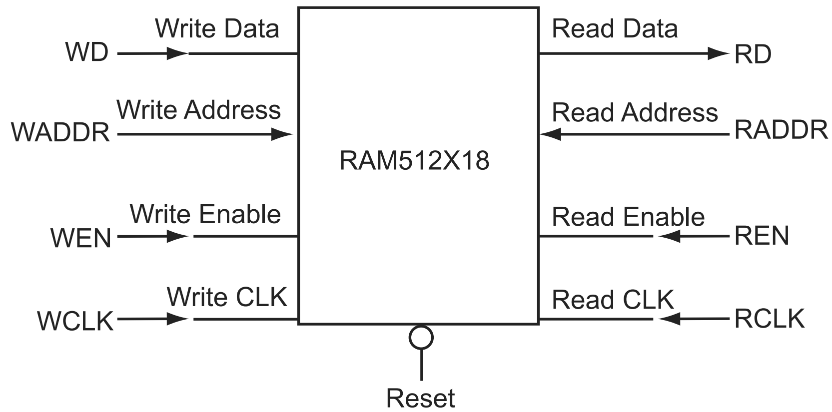


Figure 6-2 **• Fusion Device Architecture Overview (AFS600)**



Note: For timing diagrams of the RAM signals, refer to the appropriate family datasheet.

***Figure 6-5* • 512X18 Two-Port RAM Block Diagram**

Signal Descriptions for RAM512X18

RAM512X18 has slightly different behavior from RAM4K9, as it has dedicated read and write ports.

WW and RW

These signals enable the RAM to be configured in one of the two allowable aspect ratios (Table 6-5).

***Table 6-5* • Aspect Ratio Settings for WW[1:0]**

WW[1:0]	RW[1:0]	D×W
01	01	512×9
10	10	256×18
00, 11	00, 11	Reserved

WD and RD

These are the input and output data signals, and they are 18 bits wide. When a 512×9 aspect ratio is used for write, WD[17:9] are unused and must be grounded. If this aspect ratio is used for read, RD[17:9] are undefined.

WADDR and RADDR

These are read and write addresses, and they are nine bits wide. When the 256×18 aspect ratio is used for write or read, WADDR[8] and RADDR[8] are unused and must be grounded.

WCLK and RCLK

These signals are the write and read clocks, respectively. They can be clocked on the rising or falling edge of WCLK and RCLK.

WEN and REN

These signals are the write and read enables, respectively. They are both active-low by default. These signals can be configured as active-high.

RESET

This active-low signal resets the control logic, forces the output hold state registers to zero, disables reads and writes from the SRAM block, and clears the data hold registers when asserted. It does not reset the contents of the memory array.

While the RESET signal is active, read and write operations are disabled. As with any asynchronous reset signal, care must be taken not to assert it too close to the edges of active read and write clocks.

PIPE

This signal is used to specify pipelined read on the output. A LOW on PIPE indicates a nonpipelined read, and the data appears on the output in the same clock cycle. A HIGH indicates a pipelined read, and data appears on the output in the next clock cycle.

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

Solution	Board Components	Speed	Current Limitations
1	Two resistors	Low to High[1]	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[2,3,4,5] R = 47 Ω at T_J = 70°C R = 150 Ω at T_J = 85°C R = 420 Ω at T_J = 100°C	Medium	Maximum diode current at 100% duty cycle, signal constantly at 1 52.7 mA at T_J = 70°C / 10-year lifetime 16.5 mA at T_J = 85°C / 10-year lifetime 5.9 mA at T_J = 100°C / 10-year lifetime For duty cycles other than 100%, the currents can be increased by a factor of 1 / (duty cycle). Example: 20% duty cycle at 70°C 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 Ω 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 Ω) and the current is higher. Also, the designer can still use 420 Ω and use the solution on every I/O.*
5. *At 100°C, the 5 V solution on every I/O is permitted, since 420 Ω 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 LVTTL 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 LVTTL 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 LVTTL 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 LVTTL, 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.

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.

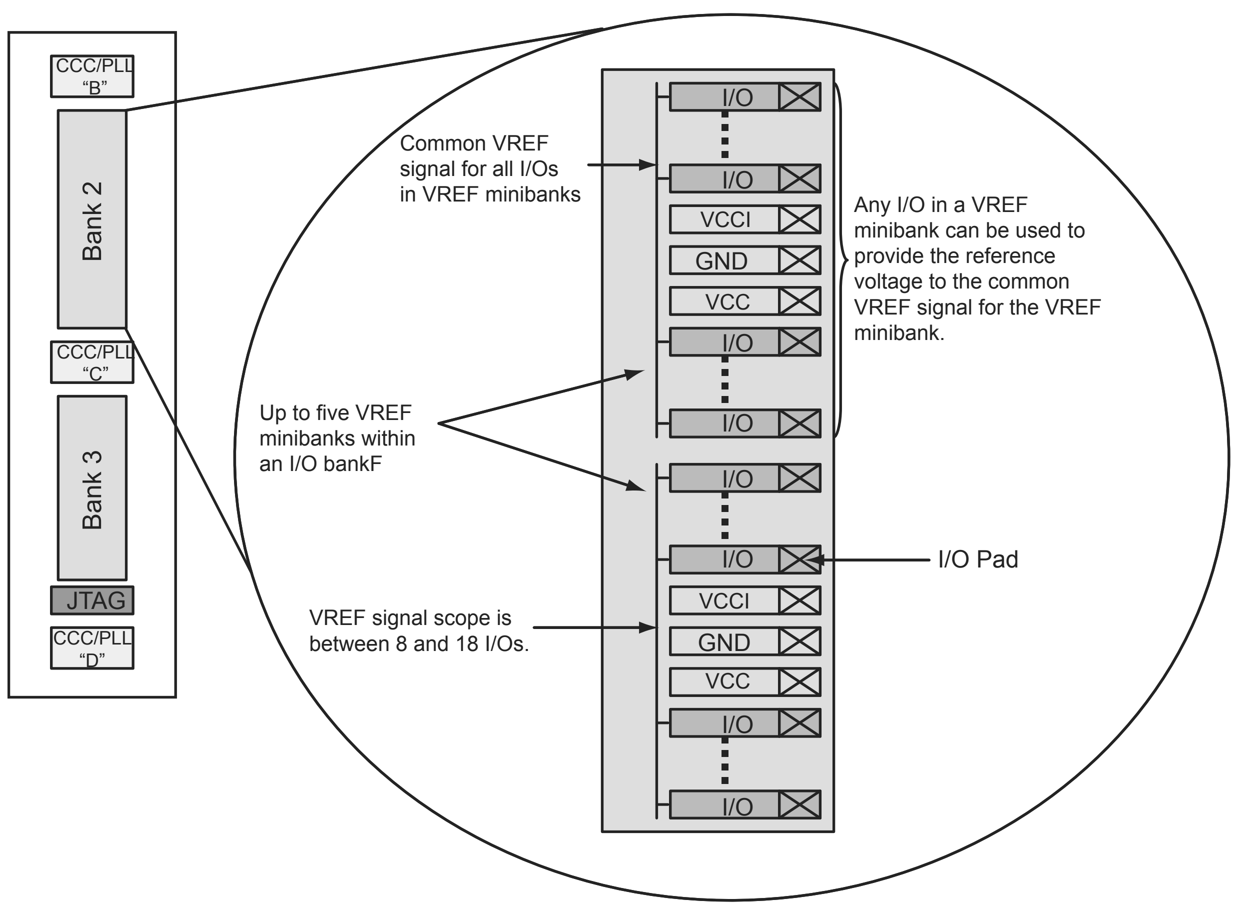


Figure 8-2 • **Typical IGLOOe and ProASIC3E I/O Bank Detail Showing V_{REF} Minibanks**

I/O Features

Low power flash devices support multiple I/O features that make board design easier. For example, an I/O feature like Schmitt Trigger in the ProASIC3E input buffer saves the board space that would be used by an external Schmitt trigger for a slow or noisy input signal. These features are also programmable for each I/O, which in turn gives flexibility in interfacing with other components. The following is a detailed description of all available features in low power flash devices.

I/O Programmable Features

Low power flash devices offer many flexible I/O features to support a wide variety of board designs. Some of the features are programmable, with a range for selection. Table 8-8 lists programmable I/O features and their ranges.

***Table 8-8* • Programmable I/O Features (user control via I/O Attribute Editor)**

Feature[1]	Description	Range
Slew Control	Output slew rate	HIGH, LOW
Output Drive (mA)	Output drive strength	2, 4, 6, 8, 12, 16, 24
Skew Control	Output tristate enable delay option	ON, OFF
Resistor Pull	Resistor pull circuit	Up, Down, None
Input Delay[2]	Input delay	OFF, 0–7
Schmitt Trigger	Schmitt trigger for input only	ON, OFF

Notes:

1. *Limitations of these features with respect to different devices are discussed in later sections.*
2. *Programmable input delay is applicable only to ProASIC3E, IGLOOe, ProASIC3EL, and RT ProASIC3 devices.*

Hot-Swap Support

A pull-up clamp diode must not be present in the I/O circuitry if the hot-swap feature is used. The 3.3 V PCI standard requires a pull-up clamp diode on the I/O, so it cannot be selected if hot-swap capability is required. The A3P030 device does not support 3.3 V PCI, so it is the only device in the ProASIC3 family that supports the hot-swap feature. All devices in the ProASIC3E family are hot-swappable. All standards except LVCMOS 2.5/5.0 V and 3.3 V PCI/PCI-X support the hot-swap feature.

The hot-swap feature appears as a read-only check box in the I/O Attribute Editor that shows whether an I/O is hot-swappable or not. Refer to the *"Power-Up/-Down Behavior of Low Power Flash Devices" section on page 373* for details on hot-swapping.

Hot-swapping (also called hot-plugging) is the operation of hot insertion or hot removal of a card in a powered-up system. The levels of hot-swap support and examples of related applications are described in Table 8-9 on page 228 to Table 8-12 on page 229. The I/Os also need to be configured in hot-insertion mode if hot-plugging compliance is required. The AGL030 and A3P030 devices have an I/O structure that allows the support of Level 3 and Level 4 hot-swap with only two levels of staging.

List of Changes

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

Date	Changes	Page
August 2012	Figure 8-1 • DDR Configured I/O Block Logical Representation and Figure 8-3 • DDR Configured I/O Block Logical Representation were revised to indicate that resets on registers 1, 3, 4, and 5 are active high rather than active low. The title of the figures was revised from "I/O Block Logical Representation" (SAR 40685).	213, 220
	AGLE1500 was removed from Table 8-2 • Supported I/O Standards because it is not a valid offering. LVCMOS 1.2 was added to the single-ended standards. LVCMOS 1.2 was added to Table 8-3 • VCCI Voltages and Compatible IGLOOe and ProASIC3E Standards (SAR 33207).	215, 217
	Lack of a heading for the "User I/O Naming Convention" section made the information difficult to locate. A heading now introduces the user I/O naming conventions (SAR 38059).	245
	Figure 8-5 • Simplified I/O Buffer Circuitry and Table 8-8 • Programmable I/O Features (user control via I/O Attribute Editor) were modified to indicate that programmable input delay control is applicable only to ProASIC3E, IGLOOe, ProASIC3EL, and RT ProASIC3 devices (SAR 39666).	222, 227
	The hyperlink for the *Board-Level Considerations* application note was corrected (SAR 36663).	246, 248
June 2011	Figure 8-1 • DDR Configured I/O Block Logical Representation and Figure 8-3 • DDR Configured I/O Block Logical Representation were revised so that the I/O_CLR and I/O_OCLK nets are no longer joined in front of Input Register 3 but instead on the branch of the CLR/PRE signal (SAR 26052).	213, 220
	The "Pro I/Os—IGLOOe, ProASIC3EL, and ProASIC3E" section was revised. Formerly it stated, "3.3 V PCI and 3.3 V PCI-X are 5 V–tolerant." This sentence now reads, "3.3 V PCI and 3.3 V PCI-X can be configured to be 5 V–tolerant" (SAR 20983).	215
	Table 8-5 • Legal IGLOOe and ProASIC3E I/O Usage Matrix within the Same Bank was revised as follows (SAR 22467): The combination of 3.3 V I/O bank voltage with 1.50 V minibank voltage and LVDS, B-LVDS, M-LVDS, and DDR was made an illegal combination (now gray instead of white). The combination of 2.5 V I/O bank voltage with no minibank voltage and LVDS, B-LVDS, M-LVDS, and DDR was made a valid combination (now white instead of gray).	217
	The following sentence was removed from the "LVCMOS (Low-Voltage CMOS)" section (SAR 22634): "All these versions use a 3.3 V–tolerant CMOS input buffer and a push-pull output buffer."	223
	The "Electrostatic Discharge Protection" section was revised to remove references to tolerances (refer to the *Reliability Report* for tolerances). The Machine Model (MM) is not supported and was deleted from this section (SAR 24385).	231
	The "I/O Interfacing" section was revised to state that low power flash devices are 5 V–input– and 5 V–output–tolerant if certain I/O standards are selected, removing "without adding any extra circuitry," which was incorrect (SAR 21404).	247
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
v1.4 (December 2008)	The terminology in the "Low Power Flash Device I/O Support" section was revised.	214

The procedure is as follows:

1. Select the bank to which you want VCCI to be assigned from the **Choose Bank** list.
2. Select the I/O standards for that bank. If you select any standard, the tool will automatically show all compatible standards that have a common VCCI voltage requirement.
3. Click **Apply**.
4. Repeat steps 1–3 to assign VCCI voltages to other banks. Refer to Figure 9-11 on page 263 to find out how many I/O banks are needed for VCCI bank assignment.

Manually Assigning VREF Pins

Voltage-referenced inputs require an input reference voltage (VREF). The user must assign VREF pins before running Layout. Before assigning a VREF pin, the user must set a VREF technology for the bank to which the pin belongs.

VREF Rules for the Implementation of Voltage-Referenced I/O Standards

The VREF rules are as follows:

1. Any I/O (except JTAG I/Os) can be used as a V_{REF} pin.
2. One V_{REF} pin can support up to 15 I/Os. It is recommended, but not required, that eight of them be on one side and seven on the other side (in other words, all 15 can still be on one side of VREF).
3. SSTL3 (I) and (II): Up to 40 I/Os per north or south bank in any position
4. LVPECL / GTL+ 3.3 V / GTL 3.3 V: Up to 48 I/Os per north or south bank in any position (not applicable for IGLOO nano and ProASIC3 nano devices)
5. SSTL2 (I) and (II) / GTL+ 2.5 V / GTL 2.5 V: Up to 72 I/Os per north or south bank in any position
6. VREF minibanks partition rule: Each I/O bank is physically partitioned into VREF minibanks. The VREF pins within a VREF minibank are interconnected internally, and consequently, only one VREF voltage can be used within each VREF minibank. If a bank does not require a VREF signal, the VREF pins of that bank are available as user I/Os.
7. The first VREF minibank includes all I/Os starting from one end of the bank to the first power triple and eight more I/Os after the power triple. Therefore, the first VREF minibank may contain (0 + 8), (2 + 8), (4 + 8), (6 + 8), or (8 + 8) I/Os.

   The second VREF minibank is adjacent to the first VREF minibank and contains eight I/Os, a power triple, and eight more I/Os after the triple. An analogous rule applies to all other VREF minibanks but the last.

   The last VREF minibank is adjacent to the previous one but contains eight I/Os, a power triple, and all I/Os left at the end of the bank. This bank may also contain (8 + 0), (8 + 2), (8 + 4), (8 + 6), or (8 + 8) available I/Os.

   Example:

   4 I/Os → Triple → 8 I/Os, 8 I/Os → Triple → 8 I/Os, 8 I/Os → Triple → 2 I/Os

   That is, minibank A = (4 + 8) I/Os, minibank B = (8 + 8) I/Os, minibank C = (8 + 2) I/Os.
8. 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.

Assigning the VREF Voltage to a Bank

When importing the PDC file, the VREF voltage can be assigned to the I/O bank. The PDC command is as follows:

```
set_iobank –vref [value]
```

Another method for assigning VREF is by using **MVN** > **Edit** > **I/O Bank Settings** (Figure 9-13 on page 266).

Application 3: Nontrusted Environment—Field Updates/Upgrades

Programming or reprogramming of devices may occur at remote locations. Reconfiguration of devices in consumer products/equipment through public networks is one example. Typically, the remote system is already programmed with particular design contents. When design update (FPGA array contents update) and/or data upgrade (FlashROM and/or FB contents upgrade) is necessary, an updated programming file with AES encryption can be generated, sent across public networks, and transmitted to the remote system. Reprogramming can then be done using this AES-encrypted programming file, providing easy and secure field upgrades. Low power flash devices support this secure ISP using AES. The detailed flow for this application is shown in Figure 12-8. Refer to the "Microprocessor Programming of Microsemi's Low Power Flash Devices" chapter of an appropriate FPGA fabric user's guide for more information.

To prepare devices for this scenario, the user can initially generate a programming file with the available security setting options. This programming file is programmed into the devices before shipment. During the programming file generation step, the user has the option of making the security settings permanent or not. In situations where no changes to the security settings are necessary, the user can select this feature in the software to generate the programming file with permanent security settings. Microsemi recommends that the programming file use encryption with an AES key, especially when ISP is done via public domain.

For example, if the designer wants to use an AES key for the FPGA array and the FlashROM, **Permanent** needs to be chosen for this setting. At first, the user chooses the options to use an AES key for the FPGA array and the FlashROM, and then chooses **Permanently lock the security settings**. A unique AES key is chosen. Once this programming file is generated and programmed to the devices, the AES key is permanently stored in the on-chip memory, where it is secured safely. The devices are sent to distant locations for the intended application. When an update is needed, a new programming file must be generated. The programming file must use the same AES key for encryption; otherwise, the authentication will fail and the file will not be programmed in the device.

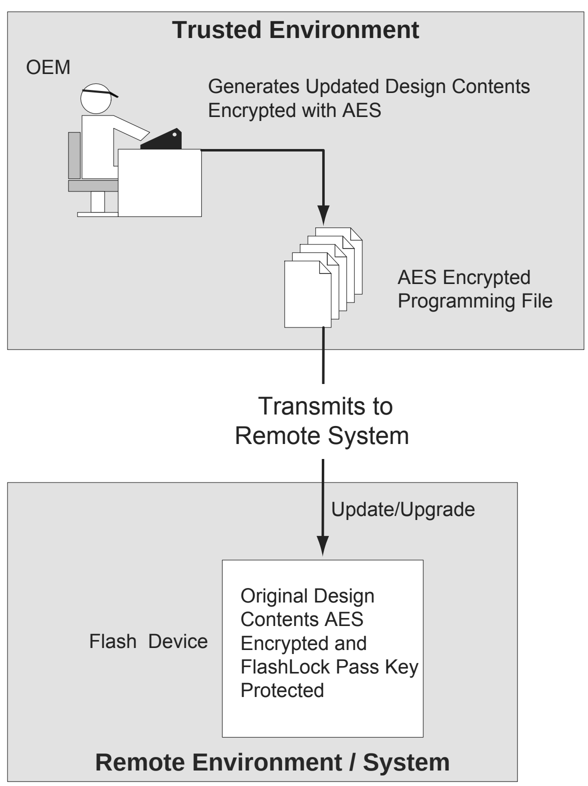
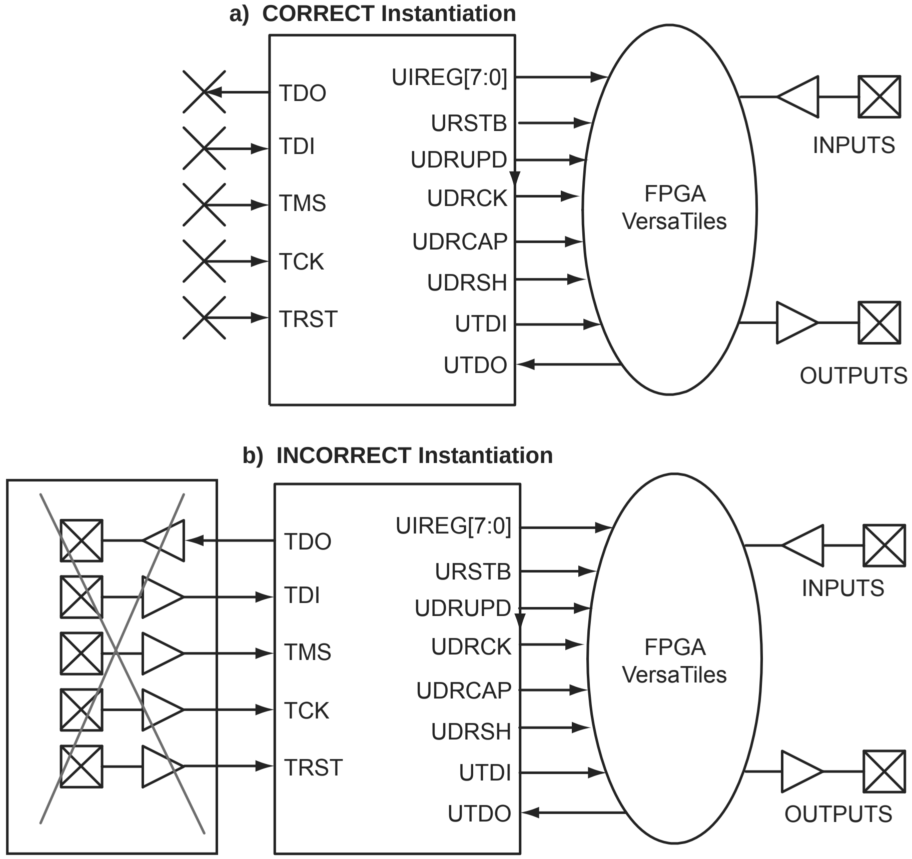


Figure 12-8 • **Application 3: Nontrusted Environment—Field Updates/Upgrades**

List of Changes

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

Date	Changes	Page
September 2012	The "Security" section was modified to clarify that Microsemi does not support read-back of FPGA core-programmed data (SAR 41235).	354
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
v1.4 (December 2008)	IGLOO nano and ProASIC3 nano devices were added to Table 15-1 • Flash-Based FPGAs.	350
v1.3 (October 2008)	The "Microprocessor Programming Support in Flash Devices" section was revised to include new families and make the information more concise.	350
v1.2 (June 2008)	The following changes were made to the family descriptions in Table 15-1 • Flash-Based FPGAs: • ProASIC3L was updated to include 1.5 V. • The number of PLLs for ProASIC3E was changed from five to six.	350
v1.1 (March 2008)	The "Microprocessor Programming Support in Flash Devices" section was updated to include information on the IGLOO PLUS family. The "IGLOO Terminology" section and "ProASIC3 Terminology" section are new.	350



Note: Do not connect JTAG pins (TDO, TDI, TMS, TCK, or TRST) to I/Os in the design.

Figure 17-3 **• Connectivity Method of UJTAG Macro**

UJTAG Operation

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

TAP Controller State Machine

The 16 states of the TAP Controller state machine are shown in Figure 17-4 on page 367. The 1s and 0s, shown adjacent to the state transitions, represent the TMS values that must be present at the time of a rising TCK edge for a state transition to occur. In the states that include the letters "IR," the instruction register operates; in the states that contain the letters "DR," the test data register operates. The TAP Controller receives two control inputs, TMS and TCK, and generates control and clock signals for the rest of the test logic.

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

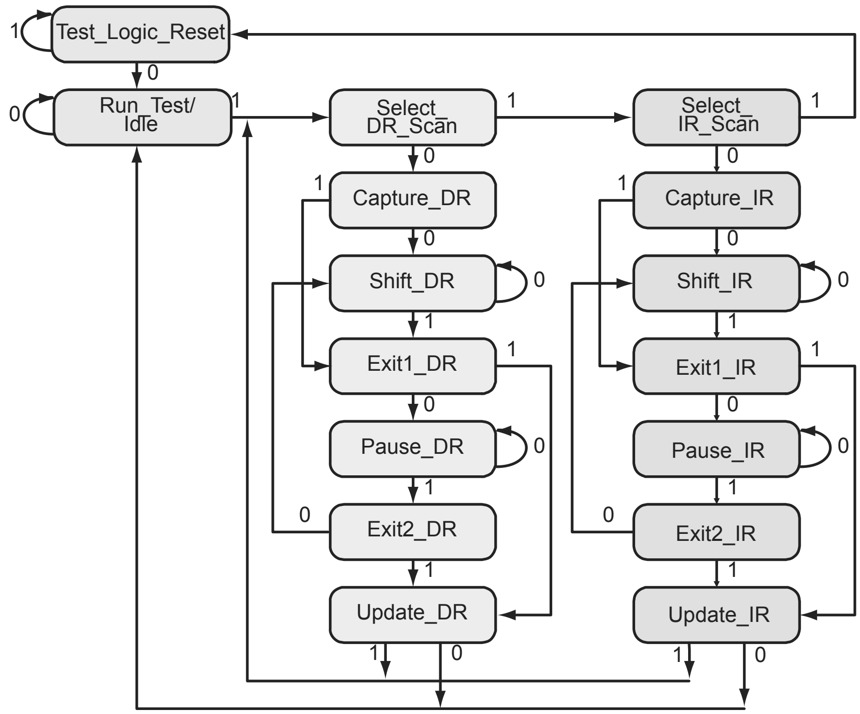


***Figure 17-4* • TAP Controller State Diagram**

UJTAG Port Usage

UIREG[7:0] hold the contents of the JTAG instruction register. The UIREG vector value is updated when the TAP Controller state machine enters the Update_IR state. Instructions 16 to 127 are user-defined and can be employed to encode multiple applications and commands within an application. Loading new instructions into the UIREG vector requires users to send appropriate logic to TMS to put the TAP Controller in a full IR cycle starting from the Select IR_Scan state and ending with the Update_IR state.

UTDI, UTDO, and UDRCK are directly connected to the JTAG TDI, TDO, and TCK ports, respectively. The TDI input can be used to provide either data (TAP Controller in the Shift_DR state) or the new contents of the instruction register (TAP Controller in the Shift_IR state).

UDRSH, UDRUPD, and UDRCAP are HIGH when the TAP Controller state machine is in the Shift_DR, Update_DR, and Capture_DR states, respectively. Therefore, they act as flags to indicate the stages of the data shift process. These flags are useful for applications in which blocks of data are shifted into the design from JTAG pins. For example, an active UDRSH can indicate that UTDI contains the data bitstream, and UDRUPD is a candidate for the end-of-data-stream flag.

As mentioned earlier, users should not connect the TDI, TDO, TCK, TMS, and TRST ports of the UJTAG macro to any port or net of the design netlist. The Designer software will automatically handle the port connection.

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.

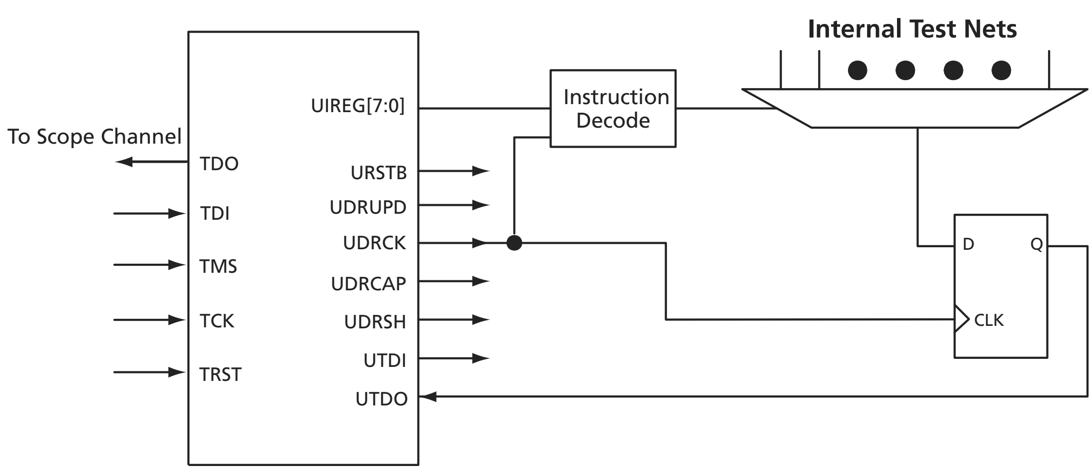


Figure 17-6 **• UJTAG Usage Example in Test and Debug Applications**

Figure 18-3 **• I/O State when VCCI Is Powered before VCC**

Power-Up to Functional Time

At power-up, device I/Os exit the tristate mode and become functional once the last voltage supply in the power-up sequence (VCCI or VCC) reaches its functional activation level. The power-up–to–functional time is the time it takes for the last supply to power up from zero to its functional level. Note that the functional level of the power supply during power-up may vary slightly within the specification at different ramp-rates. Refer to Table 18-2 for the functional level of the voltage supplies at power-up.

Typical I/O behavior during power-up–to–functional time is illustrated in Figure 18-2 on page 377 and Figure 18-3.

Table 18-2 **• Power-Up Functional Activation Levels for VCC and VCCI**

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

Note: **V5 devices will require a 1.5 V* VCC *supply, whereas V2 devices can utilize either a 1.2 V or 1.5 V* VCC.

Microsemi's low power flash devices meet Level 0 LAPU; that is, they can be functional prior to V_{CC} reaching the regulated voltage required. This important advantage distinguishes low power flash devices from their SRAM-based counterparts. SRAM-based FPGAs, due to their volatile technology, require hundreds of milliseconds after power-up to configure the design bitstream before they become functional. Refer to Figure 18-4 on page 379 and Figure 18-5 on page 380 for more information.