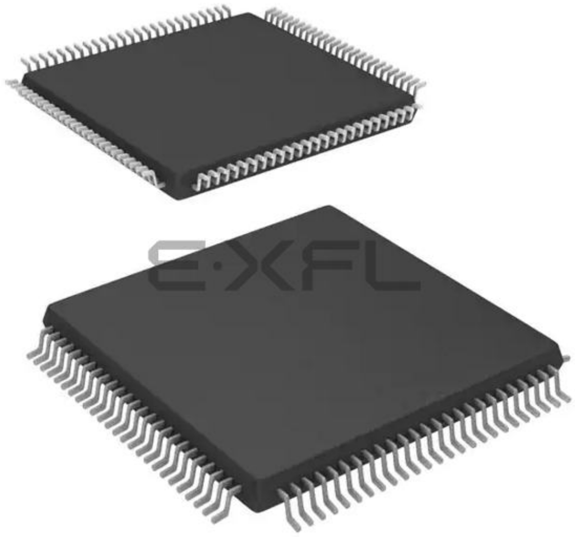
# Microsemi Corporation - A3P250L-1VQ100I Datasheet



**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 | Obsolete |
| Number of LABs/CLBs | - |
| Number of Logic Elements/Cells | - |
| Total RAM Bits | 36864 |
| Number of I/O | 68 |
| Number of Gates | 250000 |
| Voltage - Supply | 1.14V ~ 1.575V |
| Mounting Type | Surface Mount |
| Operating Temperature | -40°C ~ 100°C (TJ) |
| Package / Case | 100-TQFP |
| Supplier Device Package | 100-VQFP (14x14) |
| Purchase URL | https://www.e-xfl.com/product-detail/microsemi/a3p250l-1vq100i |

# Table of Contents

| Date | Changes | Page |
|---|---|---|
| v2.1<br>(October 2008) | The title changed from "Flash*Freeze Technology and Low Power Modes in IGLOO, IGLOO PLUS, and ProASIC3L Devices" to Actel's Flash*Freeze Technology and Low Power Modes." | N/A |
| | The "Flash Families Support the Flash*Freeze Feature" section was updated. | 22 |
| | Significant changes were made to this document to support Libero IDE v8.4 and later functionality. RT ProASIC3 device support information is new. In addition to the other major changes, the following tables and figures were updated or are new:<br>Figure 2-3 • Flash*Freeze Mode Type 2 – Controlled by Flash*Freeze Pin and Internal Logic (LSICC signal) – updated<br>Figure 2-5 • Narrow Clock Pulses During Flash*Freeze Entrance and Exit – new<br>Figure 2-10 • Flash*Freeze Management IP Block Diagram – new<br>Figure 2-11 • FSM State Diagram – new<br>Table 2-6 • IGLOO nano and IGLOO PLUS Flash*Freeze Mode (type 1 and type 2)—I/O Pad State – updated<br>Please review the entire document carefully. | 27<br>30<br>37<br>38<br>29 |
| v1.3<br>(June 2008) | The family description for ProASIC3L in Table 2-1 • Flash-Based FPGAs was updated to include 1.5 V. | 22 |
| v1.2<br>(March 2008) | The part number for this document was changed from 51700094-003-1 to 51700094-004-2. | N/A |
| | The title of the document was changed to "Flash*Freeze Technology and Low Power Modes in IGLOO, IGLOO PLUS, and ProASIC3L Devices." | N/A |
| | The "Flash*Freeze Technology and Low Power Modes" section was updated to remove the parenthetical phrase, "from 25 µW," in the second paragraph. The following sentence was added to the third paragraph: "IGLOO PLUS has an additional feature when operating in Flash*Freeze mode, allowing it to retain I/O states as well as SRAM and register states." | 21 |
| | The "Power Conservation Techniques" section was updated to add $V_{JTAG}$ to the parenthetical list of power supplies that should be tied to the ground plane if unused. Additional information was added regarding how the software configures unused I/Os. | 2-1 |
| | Table 2-1 • Flash-Based FPGAs and the accompanying text was updated to include the IGLOO PLUS family. The "IGLOO Terminology" section and "ProASIC3 Terminology" section are new. | 22 |
| | The "Flash*Freeze Mode" section was revised to include that I/O states are preserved in Flash*Freeze mode for IGLOO PLUS devices. The last sentence in the second paragraph was changed to, "If the FF pin is not used, it can be used as a regular I/O." The following sentence was added for Flash*Freeze mode type 2: "Exiting the mode is controlled by either the FF pin OR the user-defined LSICC signal." | 24 |
| | The "Flash*Freeze Type 1: Control by Dedicated Flash*Freeze Pin" section was revised to change instructions for implementing this mode, including instructions for implementation with Libero IDE v8.3. | 24 |
| | Figure 2-1 • Flash*Freeze Mode Type 1 – Controlled by the Flash*Freeze Pin was updated. | 25 |
| | The "Flash*Freeze Type 2: Control by Dedicated Flash*Freeze Pin and Internal Logic" section was renamed from "Type 2 Software Implementation." | 26 |
| | The "Type 2 Software Implementation for Libero IDE v8.3" section is new. | 2-6 |

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
```

### External I/O or Local signal as Clock Source

External I/O refers to regular I/O pins are labeled with the I/O convention IOuxwByVz. You can allow the external I/O or internal signal to access the global. To allow the external I/O or internal signal to access the global network, you need to instantiate the CLKINT macro. Refer to Figure 3-4 on page 51 for an example illustration of the connections. Instead of using CLKINT, you can also use PDC to promote signals from external I/O or internal signal to the global network. However, it may cause layout issues because of synthesis logic replication. Refer to the "Global Promotion and Demotion Using PDC" section on page 67 for details.

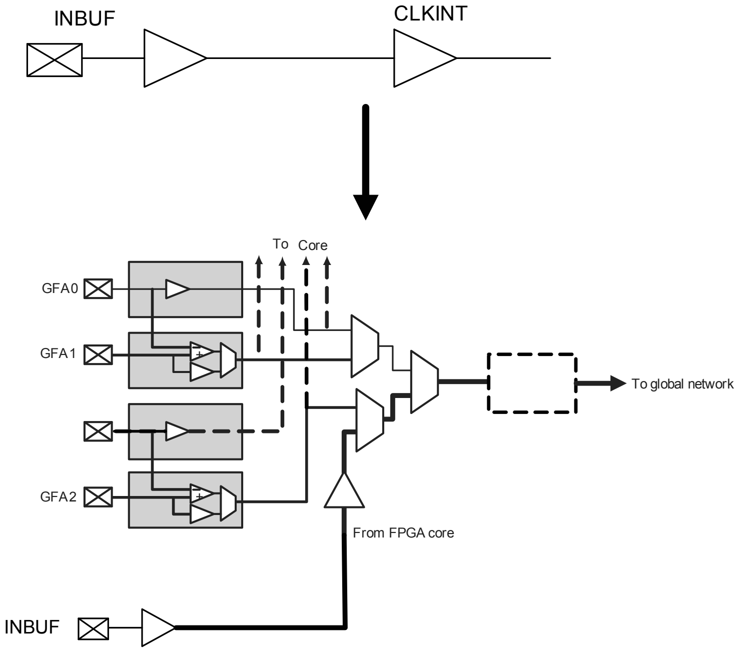


*Figure 3-14* • **CLKINT Macro**

## Using Global Macros in Synplicity

The Synplify® synthesis tool automatically inserts global buffers for nets with high fanout during synthesis. By default, Synplicity® puts six global macros (CLKBUF or CLKINT) in the netlist, including any global instantiation or PLL macro. Synplify always honors your global macro instantiation. If you have a PLL (only primary output is used) in the design, Synplify adds five more global buffers in the netlist. Synplify uses the following global counting rule to add global macros in the netlist:

1. CLKBUF: 1 global buffer
2. CLKINT: 1 global buffer
3. CLKDLY: 1 global buffer
4. PLL: 1 to 3 global buffers
   – GLA, GLB, GLC, YB, and YC are counted as 1 buffer.
   – GLB or YB is used or both are counted as 1 buffer.
   – GLC or YC is used or both are counted as 1 buffer.

## CLKDLY Macro Usage

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

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

The programmable delay elements are shown in the block diagram of the PLL block shown in Figure 4-6 on page 87. Note that any CCC locations with no PLL present contain only the programmable delay blocks going to the global networks (labeled "Programmable Delay Type 2"). Refer to the "Clock Delay Adjustment" section on page 102 for a description of the programmable delay types used for the PLL. Also refer to Table 4-14 on page 110 for Programmable Delay Type 1 step delay values, and Table 4-15 on page 110 for Programmable Delay Type 2 step delay values. CCC locations with a PLL present can be configured to utilize only the programmable delay blocks (Programmable Delay Type 2) going to the global networks A, B, and C.

Global network A can be configured to use only the programmable delay element (bypassing the PLL) if the PLL is not used in the design. Figure 4-6 on page 87 shows a block diagram of the PLL, where the programmable delay elements are used for the global networks (Programmable Delay Type 2).

## Core Logic Clock Source

*Core logic* refers to internal routed nets. Internal routed signals access the CCC via the FPGA Core Fabric. Similar to the External I/O option, whenever the clock source comes internally from the core itself, the routed signal is instantiated with a PLLINT macro before connecting to the CCC clock input (see Figure 4-12 for an example illustration of the connections, shown in red).

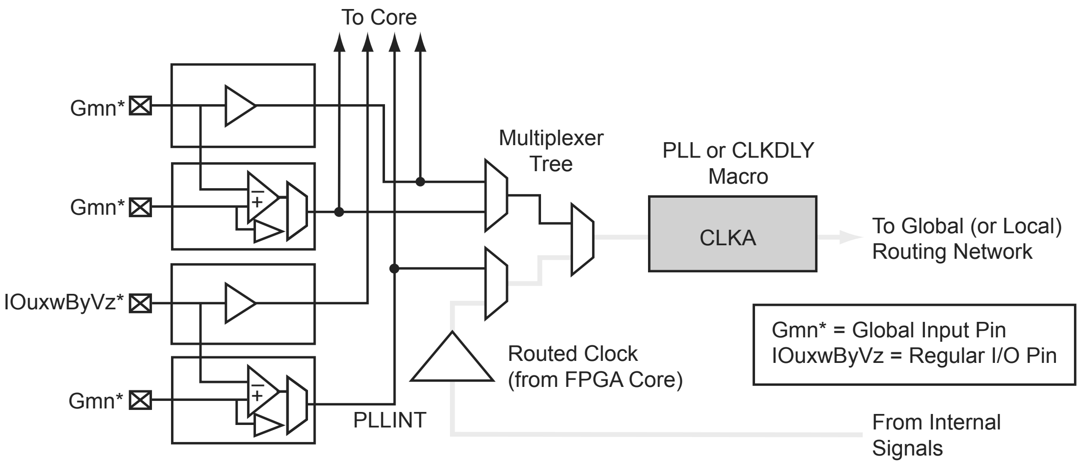


***Figure 4-12*** **• Illustration of Core Logic Usage**

For Fusion devices, the input reference clock can also be from the embedded RC oscillator and crystal oscillator. In this case, the CCC configuration is the same as the hardwired I/O clock source, and users are required to instantiate the RC oscillator or crystal oscillator macro and connect its output to the input reference clock of the CCC block.

### External Feedback Configuration

For certain applications, such as those requiring generation of PCB clocks that must be matched with existing board delays, it is useful to implement an external feedback, EXTFB. The Phase Detector of the PLL core will receive CLKA and EXTFB as inputs. EXTFB may be processed by the fixed System Delay element as well as the *M* divider element. The EXTFB option is currently not supported.

After setting all the required parameters, users can generate one or more PLL configurations with HDL or EDIF descriptions by clicking the **Generate** button. SmartGen gives the option of saving session results and messages in a log file:

```
****************
Macro Parameters
****************

Name                            : test_pll
Family                          : ProASIC3E
Output Format                   : VHDL
Type                            : Static PLL
Input Freq(MHz)                 : 10.000
CLKA Source                     : Hardwired I/O
Feedback Delay Value Index      : 1
Feedback Mux Select             : 2
XDLY Mux Select                 : No
Primary Freq(MHz)               : 33.000
Primary PhaseShift              : 0
Primary Delay Value Index       : 1
Primary Mux Select              : 4
Secondary1 Freq(MHz)            : 66.000
Use GLB                         : YES
Use YB                          : YES
GLB Delay Value Index           : 1
YB Delay Value Index            : 1
Secondary1 PhaseShift           : 0
Secondary1 Mux Select           : 4
Secondary2 Freq(MHz)            : 101.000
Use GLC                         : YES
Use YC                          : NO
GLC Delay Value Index           : 1
YC Delay Value Index            : 1
Secondary2 PhaseShift           : 0
Secondary2 Mux Select           : 4

…
…
…

Primary Clock frequency 33.333
Primary Clock Phase Shift 0.000
Primary Clock Output Delay from CLKA 0.180

Secondary1 Clock frequency 66.667
Secondary1 Clock Phase Shift 0.000
Secondary1 Clock Global Output Delay from CLKA 0.180
Secondary1 Clock Core Output Delay from CLKA 0.625

Secondary2 Clock frequency 100.000
Secondary2 Clock Phase Shift 0.000
Secondary2 Clock Global Output Delay from CLKA 0.180
```

Below is an example Verilog HDL description of a legal PLL core configuration generated by SmartGen:

```
module test_pll(POWERDOWN,CLKA,LOCK,GLA);
input POWERDOWN, CLKA;
output  LOCK, GLA;
```

### RD

This is the output data bus and is 18 bits wide. Not all 18 bits are valid in all configurations. Like the WD bus, high-order bits become unusable if the data width is less than 18. The output data on unused pins is undefined (Table 6-7).

***Table 6-7* • Input Data Signal Usage for Different Aspect Ratios**

| D×W | WD/RD Unused |
|---|---|
| 4k×1 | WD[17:1], RD[17:1] |
| 2k×2 | WD[17:2], RD[17:2] |
| 1k×4 | WD[17:4], RD[17:4] |
| 512×9 | WD[17:9], RD[17:9] |
| 256×18 | – |

### ESTOP, FSTOP

ESTOP is used to stop the FIFO read counter from further counting once the FIFO is empty (i.e., the EMPTY flag goes HIGH). A HIGH on this signal inhibits the counting.

FSTOP is used to stop the FIFO write counter from further counting once the FIFO is full (i.e., the FULL flag goes HIGH). A HIGH on this signal inhibits the counting.

For more information on these signals, refer to the "ESTOP and FSTOP Usage" section.

### FULL, EMPTY

When the FIFO is full and no more data can be written, the FULL flag asserts HIGH. The FULL flag is synchronous to WCLK to inhibit writing immediately upon detection of a full condition and to prevent overflows. Since the write address is compared to a resynchronized (and thus time-delayed) version of the read address, the FULL flag will remain asserted until two WCLK active edges after a read operation eliminates the full condition.

When the FIFO is empty and no more data can be read, the EMPTY flag asserts HIGH. The EMPTY flag is synchronous to RCLK to inhibit reading immediately upon detection of an empty condition and to prevent underflows. Since the read address is compared to a resynchronized (and thus time-delayed) version of the write address, the EMPTY flag will remain asserted until two RCLK active edges after a write operation removes the empty condition.

For more information on these signals, refer to the "FIFO Flag Usage Considerations" section on page 161.

### AFULL, AEMPTY

These are programmable flags and will be asserted on the threshold specified by AFVAL and AEVAL, respectively.

When the number of words stored in the FIFO reaches the amount specified by AEVAL while reading, the AEMPTY output will go HIGH. Likewise, when the number of words stored in the FIFO reaches the amount specified by AFVAL while writing, the AFULL output will go HIGH.

### AFVAL, AEVAL

The AEVAL and AFVAL pins are used to specify the almost-empty and almost-full threshold values. They are 12-bit signals. For more information on these signals, refer to the "FIFO Flag Usage Considerations" section on page 161.

## *FIFO Usage*

### ESTOP and FSTOP Usage

The ESTOP pin is used to stop the read counter from counting any further once the FIFO is empty (i.e., the EMPTY flag goes HIGH). Likewise, the FSTOP pin is used to stop the write counter from counting any further once the FIFO is full (i.e., the FULL flag goes HIGH).

The FIFO counters in the device start the count at zero, reach the maximum depth for the configuration (e.g., 511 for a 512×9 configuration), and then restart at zero. An example application for ESTOP, where the read counter keeps counting, would be writing to the FIFO once and reading the same content over and over without doing another write.

*Table 7-8* • **Hot-Swap Level 1**

| **Description** | Cold-swap |
|---|---|
| **Power Applied to Device** | No |
| **Bus State** | – |
| **Card Ground Connection** | – |
| **Device Circuitry Connected to Bus Pins** | – |
| **Example Application** | System and card with Microsemi FPGA chip are powered down, and the card is plugged into the system. Then the power supplies are turned on for the system but not for the FPGA on the card. |
| **Compliance of IGLOO and ProASIC3 Devices** | 30 k gate devices: Compliant<br>Other IGLOO/ProASIC3 devices: Compliant if bus switch used to isolate FPGA I/Os from rest of system<br>IGLOOe/ProASIC3E devices: Compliant I/Os can but do not have to be set to hot-insertion mode. |

*Table 7-9* • **Hot-Swap Level 2**

| **Description** | Hot-swap while reset |
|---|---|
| **Power Applied to Device** | Yes |
| **Bus State** | Held in reset state |
| **Card Ground Connection** | Reset must be maintained for 1 ms before, during, and after insertion/removal. |
| **Device Circuitry Connected to Bus Pins** | – |
| **Example Application** | In the PCI hot-plug specification, reset control circuitry isolates the card busses until the card supplies are at their nominal operating levels and stable. |
| **Compliance of IGLOO and ProASIC3 Devices** | 30 k gate devices, all IGLOOe/ProASIC3E devices: Compliant I/Os can but do not have to be set to hot-insertion mode.<br>Other IGLOO/ProASIC3 devices: Compliant |

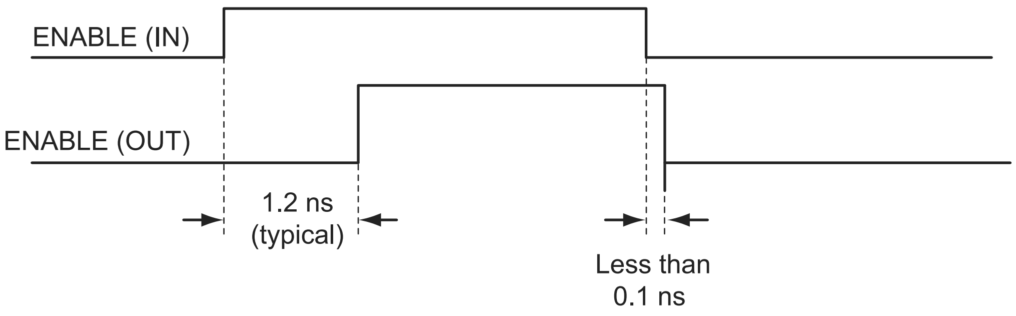


*Figure 7-15* • **Timing Diagram (option 2: enables skew circuit)**

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.

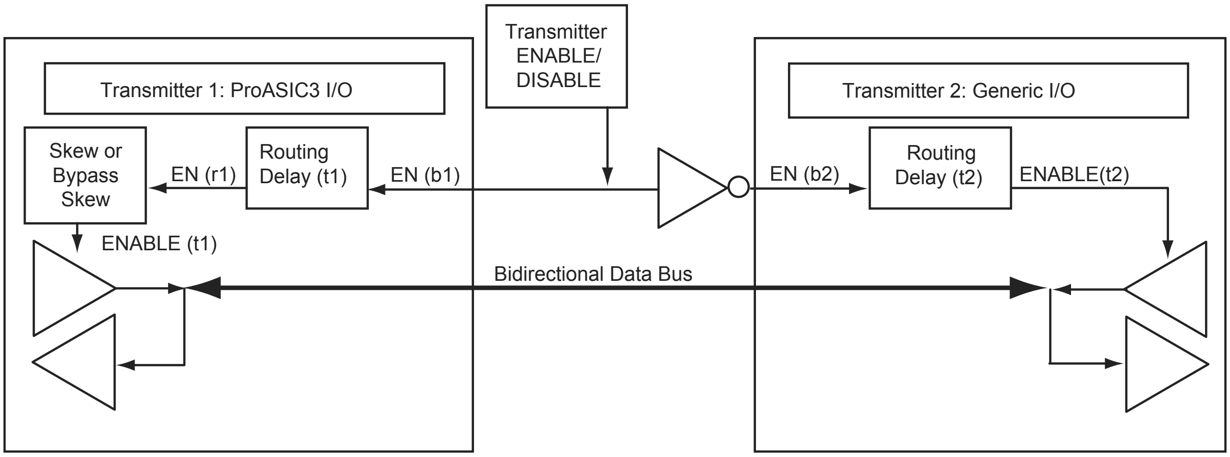


*Figure 7-16* • **Example of Implementation of Skew Circuits in Bidirectional Transmission Systems Using IGLOO or ProASIC3 Devices**

Table 7-19 shows some high-level interfacing examples using low power flash devices.

***Table 7-19* • High-Level Interface Examples**

| | Clock | | I/O | | | |
|---|---|---|---|---|---|---|
| **Interface** | **Type** | **Frequency** | **Type** | **Signals In** | **Signals Out** | **Data I/O** |
| GM | Src Sync | 125 MHz | LVTTL | 8 | 8 | 125 Mbps |
| TBI | Src Sync | 125 MHz | LVTTL | 10 | 10 | 125 Mbps |
| XSBI | Src Sync | 644 MHz | LVDS | 16 | 16 | 644 Mbps |
| XGMI | Src Sync DDR | 156 MHz | HSTL1 | 32 | 32 | 312 Mbps |
| FlexBus 3 | Sys Sync | 104 MHz | LVTTL | ≤ 32 | ≤ 32 | ≤ 104 |
| Pos-PHY3/SPI-3 | Sys Sync | 104 | LVTTL | 8, 16, 32 | 8, 16, 32 | ≤ 104 Mbps |
| FlexBus 4/SPI-4.1 | Src Sync | 200 MHz | HSTL1 | 16,64 | 16,64 | 200 Mbps |
| Pos-PHY4/SPI-4.2 | Src Sync DDR | ≥ 311 MHz | LVDS | 16 | 16 | ≥ 622 Mbps |
| SFI-4.1 | Src Sync | 622 MHz | LVDS | 16 | 16 | 622 Mbps |
| CSIX L1 | Sys Sync | ≤ 250 MHz | HSTL1 | 32,64,96,128 | 32,64,96,128 | ≤ 250 Mbps |
| Hyper Transport | Sys Sync DDR | ≤ 800 MHz | LVDS | 2,4,8,16 | 2,4,8,16 | ≤ 1.6 Gbps |
| Rapid I/O Parallel | Sys Sync DDR | 250 MHz – 1 GHz | LVDS | 8,16 | 8,16 | ≤ 2 Gbps |
| Star Fabric | CDR | | LVDS | 4 | 4 | 622 Mbps |

*Note: Sys Sync = System Synchronous Clocking, Src Sync = Source Synchronous Clocking, and CDR = Clock and Data Recovery.*

# Conclusion

IGLOO and ProASIC3 support for multiple I/O standards minimizes board-level components and makes possible a wide variety of applications. The Microsemi Designer software, integrated with Libero SoC, presents a clear visual display of I/O assignments, allowing users to verify I/O and board-level design requirements before programming the device. The IGLOO and ProASIC3 device I/O features and functionalities ensure board designers can produce low-cost and low power FPGA applications fulfilling the complexities of contemporary design needs.

# Conclusion

IGLOOe and ProASIC3E support for multiple I/O standards minimizes board-level components and makes possible a wide variety of applications. The Microsemi Designer software, integrated with Libero SoC, presents a clear visual display of I/O assignments, allowing users to verify I/O and board-level design requirements before programming the device. The IGLOOe and ProASIC3E device I/O features and functionalities ensure board designers can produce low-cost and low power FPGA applications fulfilling the complexities of contemporary design needs.

# Related Documents

## Application Notes

*Board-Level Considerations*

http://www.microsemi.com/soc/documents/ALL_AC276_AN.pdf

## User’s Guides

*ProASIC3 FPGA Fabric User’s Guide*

http://www.microsemi.com/soc/documents/PA3_UG.pdf

*ProASIC3E FPGA Fabric User’s Guide*

http://www.microsemi.com/soc/documents/PA3E_UG.pdf

*IGLOOe FPGA Fabric User’s Guide*

http://www.microsemi.com/soc/documents/IGLOOe_UG.pdf

*Libero SoC User’s Guide*

http://www.microsemi.com/soc/documents/libero_ug.pdf

*IGLOO, Fusion, and ProASIC3 Macro Library Guide*

http://www.microsemi.com/soc/documents/pa3_libguide_ug.pdf

*SmartGen Core Reference Guide*

http://www.microsemi.com/soc/documents/genguide_ug.pdf those banks, the user does not need to assign the same VCCI voltage to another bank. The user needs to assign the other three VCCI voltages to three more banks.

# Assigning Technologies and VREF to I/O Banks

Low power flash devices offer a wide variety of I/O standards, including voltage-referenced standards. Before proceeding to Layout, each bank must have the required VCCI voltage assigned for the corresponding I/O technologies used for that bank. The voltage-referenced standards require the use of a reference voltage (VREF). This assignment can be done manually or automatically. The following sections describe this in detail.

## Manually Assigning Technologies to I/O Banks

The user can import the PDC at this point and resolve this requirement. The PDC command is

```
set_iobank [bank name] –vcci [vcci value]
```

Another method is to use the I/O Bank Settings dialog box (**MVN** > **Edit** > **I/O Bank Settings**) to set up the $V_{CCI}$ voltage for the bank (Figure 9-12).

***Figure 9-12*** **• Setting VCCI for a Bank**

If the assignment is not successful, an error message appears in the Output window.

To undo the I/O bank assignments, choose **Undo** from the **Edit** menu. Undo removes the I/O technologies assigned by the IOBA. It does not remove the I/O technologies previously assigned.

To redo the changes undone by the Undo command, choose **Redo** from the **Edit** menu.

To clear I/O bank assignments made before using the Undo command, manually unassign or reassign I/O technologies to banks. To do so, choose **I/O Bank Settings** from the **Edit** menu to display the I/O Bank Settings dialog box.

# Conclusion

Fusion, IGLOO, and ProASIC3 support for multiple I/O standards minimizes board-level components and makes possible a wide variety of applications. The Microsemi Designer software, integrated with Libero SoC, presents a clear visual display of I/O assignments, allowing users to verify I/O and board-level design requirements before programming the device. The device I/O features and functionalities ensure board designers can produce low-cost and low power FPGA applications fulfilling the complexities of contemporary design needs.

# Related Documents

## User's Guides

*Libero SoC User's Guide*

http://www.microsemi.com/soc/documents/libero_ug.pdf

*IGLOO, ProASIC3, SmartFusion, and Fusion Macro Library Guide*

http://www.microsemi.com/soc/documents/pa3_libguide_ug.pdf

*SmartGen Core Reference Guide*

http://www.microsemi.com/soc/documents/genguide_ug.pdf

## Design Example

Figure 10-9 shows a simple example of a design using both DDR input and DDR output registers. The user can copy the HDL code in Libero SoC software and go through the design flow. Figure 10-10 and Figure 10-11 on page 283 show the netlist and ChipPlanner views of the ddr_test design. Diagrams may vary slightly for different families.

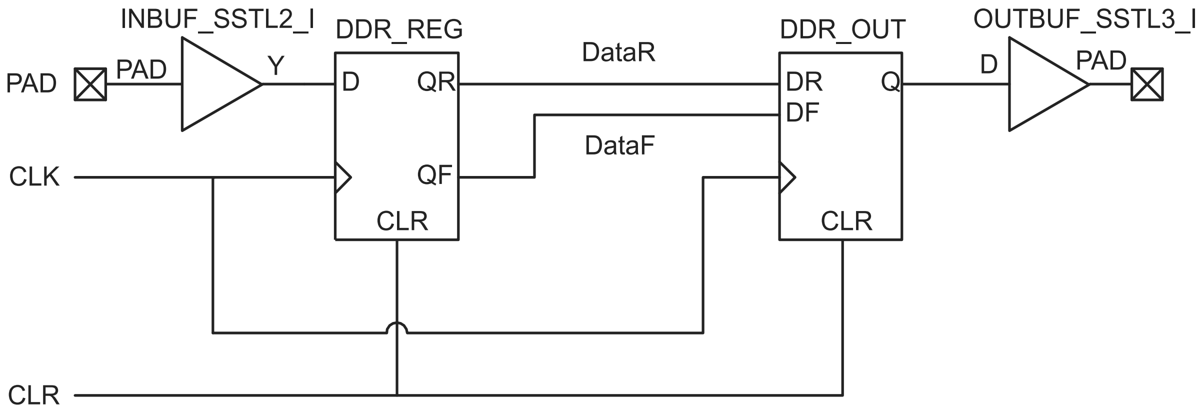


*Figure 10-9* • **Design Example**

*Figure 10-10* • **DDR Test Design as Seen by NetlistViewer for IGLOO/e Devices**

## Programming Solutions

Details for the available programmers can be found in the programmer user's guides listed in the "Related Documents" section on page 297.

All the programmers except FlashPro4, FlashPro3, FlashPro Lite, and FlashPro require adapter modules, which are designed to support device packages. All modules are listed on the Microsemi SoC Products Group website at http://www.microsemi.com/soc/products/hardware/program_debug/ss/modules.aspx. They are not listed in this document, since this list is updated frequently with new package options and any upgrades required to improve programming yield or support new families.

***Table 11-3* • Programming Solutions**

| Programmer | Vendor | ISP | Single Device | Multi-Device | Availability |
|---|---|---|---|---|---|
| FlashPro4 | Microsemi | Only | Yes | Yes[1] | Available |
| FlashPro3 | Microsemi | Only | Yes | Yes[1] | Available |
| FlashPro Lite[2] | Microsemi | Only | Yes | Yes[1] | Available |
| FlashPro | Microsemi | Only | Yes | Yes[1] | Discontinued |
| Silicon Sculptor 3 | Microsemi | Yes[3] | Yes | Cascade option (up to two) | Available |
| Silicon Sculptor II | Microsemi | Yes[3] | Yes | Cascade option (up to two) | Available |
| Silicon Sculptor | Microsemi | Yes | Yes | Cascade option (up to four) | Discontinued |
| Sculptor 6X | Microsemi | No | Yes | Yes | Discontinued |
| BP MicroProgrammers | BP Microsystems | No | Yes | Yes | Contact BP Microsystems at www.bpmicro.com |

*Notes:*

1. *Multiple devices can be connected in the same JTAG chain for programming.*
2. *If FlashPro Lite is used for programming, the programmer derives all of its power from the target pc board's VDD supply. The FlashPro Lite's VPP and VPN power supplies use the target pc board's VDD as a power source. The target pc board must supply power to both the VDDP and VDD power pins of the ProASIC$^{PLUS}$ device in addition to supplying VDD to the FlashPro Lite. The target pc board needs to provide at least 500 mA of current to the FlashPro Lite VDD connection for programming.*
3. *Silicon Sculptor II and Silicon Sculptor 3 can only provide ISP for ProASIC and ProASIC$^{PLUS}$ families, not for Fusion, IGLOO, or ProASIC3 devices.*

# 16 – Boundary Scan in Low Power Flash Devices

## Boundary Scan

Low power flash devices are compatible with IEEE Standard 1149.1, which defines a hardware architecture and the set of mechanisms for boundary scan testing. JTAG operations are used during boundary scan testing.

The basic boundary scan logic circuit is composed of the TAP controller, test data registers, and instruction register (Figure 16-2 on page 360).

Low power flash devices support three types of test data registers: bypass, device identification, and boundary scan. The bypass register is selected when no other register needs to be accessed in a device. This speeds up test data transfer to other devices in a test data path. The 32-bit device identification register is a shift register with four fields (LSB, ID number, part number, and version). The boundary scan register observes and controls the state of each I/O pin. Each I/O cell has three boundary scan register cells, each with serial-in, serial-out, parallel-in, and parallel-out pins.

## TAP Controller State Machine

The TAP controller is a 4-bit state machine (16 states) that operates as shown in Figure 16-1.

The 1s and 0s represent the values that must be present on TMS at a rising edge of TCK for the given state transition to occur. IR and DR indicate that the instruction register or the data register is operating in that state.

The TAP controller receives two control inputs (TMS and TCK) and generates control and clock signals for the rest of the test logic architecture. On power-up, the TAP controller enters the Test-Logic-Reset state. To guarantee a reset of the controller from any of the possible states, TMS must remain HIGH for five TCK cycles. The TRST pin can also be used to asynchronously place the TAP controller in the Test-Logic-Reset state.

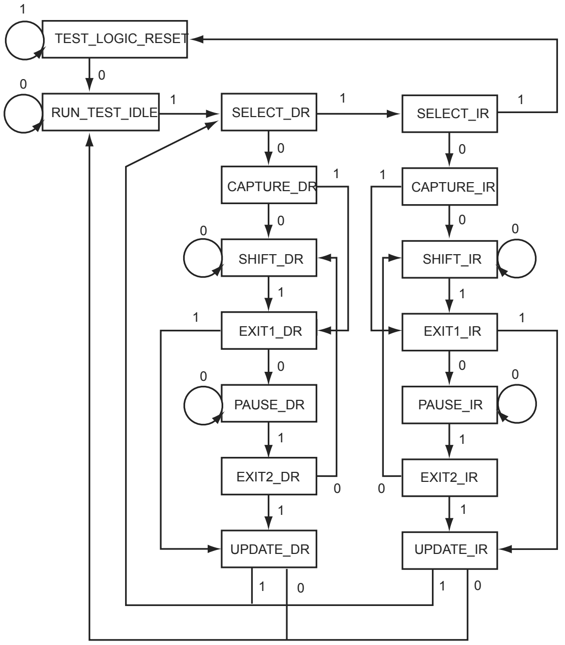


*Figure 16-1* • **TAP Controller State Machine**

### Internal Pull-Up and Pull-Down

Low power flash device I/Os are equipped with internal weak pull-up/-down resistors that can be used by designers. If used, these internal pull-up/-down resistors will be activated during power-up, once both VCC and VCCI are above their functional activation level. Similarly, during power-down, these internal pull-up/-down resistors will turn off once the first supply voltage falls below its brownout deactivation level.

## Cold-Sparing

In cold-sparing applications, voltage can be applied to device I/Os before and during power-up. Cold-sparing applications rely on three important characteristics of the device:

1. I/Os must be tristated before and during power-up.
2. Voltage applied to the I/Os must not power up any part of the device.
3. VCCI should not exceed 3.6 V, per datasheet specifications.

As described in the "Power-Up to Functional Time" section on page 378, Microsemi's low power flash I/Os are tristated before and during power-up until the last voltage supply (VCC or VCCI) is powered up past its functional level. Furthermore, applying voltage to the FPGA I/Os does not pull up VCC or VCCI and, therefore, does not partially power up the device. Table 18-4 includes the cold-sparing test results on A3PE600-PQ208 devices. In this test, leakage current on the device I/O and residual voltage on the power supply rails were measured while voltage was applied to the I/O before power-up.

***Table 18-4* • Cold-Sparing Test Results for A3PE600 Devices**

| | Residual Voltage (V) | | |
|---|---|---|---|
| **Device I/O** | **VCC** | **VCCI** | **Leakage Current** |
| Input | 0 | 0.003 | <1 µA |
| Output | 0 | 0.003 | <1 µA |

VCCI must not exceed 3.6 V, as stated in the datasheet specification. Therefore, ProASIC3E devices meet all three requirements stated earlier in this section and are suitable for cold-sparing applications.

The following devices and families support cold-sparing:

- IGLOO: AGL015 and AGL030
- All IGLOO nano
- All IGLOO PLUS
- All IGLOOe
- ProASIC3L: A3PE3000L
- ProASIC3: A3P015 and A3P030
- All ProASIC3 nano
- All ProASIC3E
- Military ProASIC3EL: A3PE600L and A3PE3000L
- RT ProASIC3: RT3PE600L and RT3PE3000L

# A – Summary of Changes

## History of Revision to Chapters

The following table lists chapters that were affected in each revision of this document. Each chapter includes its own change history because it may appear in other device family user's guides. Refer to the individual chapter for a list of specific changes.

| Revision (month/year) | Chapter Affected | List of Changes (page number) |
|---|---|---|
| Revision 4 (September 2012) | "Microprocessor Programming of Microsemi's Low Power Flash Devices" was revised. | 356 |
| Revision 3 (August 2012) | "FPGA Array Architecture in Low Power Flash Devices" was revised. | 20 |
| | "Clock Conditioning Circuits in Low Power Flash Devices and Mixed Signal FPGAs" was revised. | 129 |
| | "SRAM and FIFO Memories in Microsemi's Low Power Flash Devices" was revised. | 173 |
| | "I/O Structures in IGLOO and ProASIC3 Devices" was revised. | 210 |
| | "I/O Structures in IGLOOe and ProASIC3E Devices" was revised. | 249 |
| | The "Pin Descriptions" and "Packaging" chapters were removed. This information is now published in the datasheet for each product line (SAR 34773). | |
| | "In-System Programming (ISP) of Microsemi's Low Power Flash Devices Using FlashPro4/3/3X" was revised. | 339 |
| | "Boundary Scan in Low Power Flash Devices" was revised. | 362 |
| Revision 2 (December 2011) | "Clock Conditioning Circuits in Low Power Flash Devices and Mixed Signal FPGAs" was revised. | 129 |
| | "UJTAG Applications in Microsemi's Low Power Flash Devices" was revised. | 372 |
| Revision 1 (June 2011) | "Clock Conditioning Circuits in Low Power Flash Devices and Mixed Signal FPGAs" was revised. | 129 |
| | "I/O Structures in IGLOO and ProASIC3 Devices" was revised. | 210 |
| | "I/O Structures in IGLOOe and ProASIC3E Devices" was revised. | 249 |
| | "I/O Software Control in Low Power Flash Devices" was revised. | 270 |
| | "In-System Programming (ISP) of Microsemi's Low Power Flash Devices Using FlashPro4/3/3X" was revised. | 339 |
| Revision 0 (July 2010) | The ProASIC3L Flash Family FPGAs Handbook was divided into two parts to create the ProASIC3L Low Power Flash FPGAs Datasheet and the ProASIC3L FPGA Fabric User's Guide. | N/A |
| | "Global Resources in Low Power Flash Devices" was revised. | 75 |
| | "Clock Conditioning Circuits in Low Power Flash Devices and Mixed Signal FPGAs" was revised. | 129 |
| | "I/O Software Control in Low Power Flash Devices" was revised. | 270 |