Microsemi Corporation - M1A3P1000L-PQG208 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	147456
Number of I/O	154
Number of Gates	1000000
Voltage - Supply	1.14V ~ 1.575V
Mounting Type	Surface Mount
Operating Temperature	0°C ~ 85°C (TJ)
Package / Case	208-BFQFP
Supplier Device Package	208-PQFP (28x28)
Purchase URL	https://www.e-xfl.com/product-detail/microsemi/m1a3p1000l-pqg208

Email: info@E-XFL.COM
Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong

Flash*Freeze Type 2: Control by Dedicated Flash*Freeze Pin and Internal Logic

The device can be made to enter Flash*Freeze mode by activating the FF pin together with Microsemi's Flash*Freeze management IP core (refer to the "Flash*Freeze Management IP" section on page 36 for more information) or user-defined control logic (Figure 2-3 on page 27) within the FPGA core. This method enables the design to perform important activities before allowing the device to enter Flash*Freeze mode, such as transitioning into a safe state, completing the processing of a critical event. Designers are encouraged to take advantage of Microsemi's Flash*Freeze Management IP to handle clean entry and exit of Flash*Freeze mode (described later in this document). The device will only enter Flash*Freeze mode when the Flash*Freeze pin is asserted (active Low) and the User Low Static I_{CC} (ULSICC) macro input signal, called the LSICC signal, is asserted (High). One condition is not sufficient to enter Flash*Freeze mode type 2; both the FF pin and LSICC signal must be asserted.

When Flash*Freeze type 2 is implemented in the design, the ULSICC macro needs to be instantiated by the user. There are no functional differences in the device whether the ULSICC macro is instantiated or not, and whether the LSICC signal is asserted or deasserted. The LSICC signal is used only to control entering Flash*Freeze mode. Figure 2-4 on page 27 shows the timing diagram for entering and exiting Flash*Freeze mode type 2.

After exiting Flash*Freeze mode type 2 by deasserting the Flash*Freeze pin, the LSICC signal must be deasserted by the user design. This will prevent entering Flash*Freeze mode by asserting the Flash*Freeze pin only.

Refer to Table 2-3 for Flash*Freeze (FF) pin and LSICC signal assertion and deassertion values.

Table 2-3 • **Flash*Freeze Mode Type 1 and Type 2 – Signal Assertion and Deassertion Values**

Signal	Assertion Value	Deassertion Value
Flash*Freeze (FF) pin	Low	High
LSICC signal	High	Low

Notes:

1. *The Flash*Freeze (FF) pin is an active-Low signal, and LSICC is an active-High signal.*
2. *The LSICC signal is used only in Flash*Freeze mode type 2.*

Flash*Freeze Mode Device Behavior

Entering Flash*Freeze Mode

- IGLOO, IGLOO nano, IGLOO PLUS, ProASCI3L, and RT ProASIC3 devices are designed and optimized to enter Flash*Freeze mode only when power supplies are stable. If the device is being powered up while the FF pin is asserted (Flash*Freeze mode type 1), or while both FF pin and LSICC signal are asserted (Flash*Freeze mode type 2), the device is expected to enter Flash*Freeze mode within 5 µs after the I/Os and FPGA core have reached their activation levels.
- If the device is already powered up when the FF pin is asserted, the device will enter Flash*Freeze mode within 1 µs (type 1). In Flash*Freeze mode type 2 operation, entering Flash*Freeze mode is completed within 1 µs after both FF pin and LSICC signal are asserted. Exiting Flash*Freeze mode is completed within 1 µs after deasserting the FF pin only.

PLLs

- If an embedded PLL is used, entering Flash*Freeze mode will automatically power down the PLL.
- The PLL output clocks will stop toggling within 1 µs after the assertion of the FF pin in type 1, or after both FF pin and LSICC signal are asserted in type 2. At the same time, I/Os will transition into the state specified in Table 2-6 on page 29. The user design must ensure it is safe to enter Flash*Freeze mode.

I/Os and Globals

- While entering Flash*Freeze mode, inputs, globals, and PLLs will enter their Flash*Freeze state asynchronously to each other. As a result, clock and data glitches and narrow pulses may be generated while entering Flash*Freeze mode, as shown in Figure 2-5.

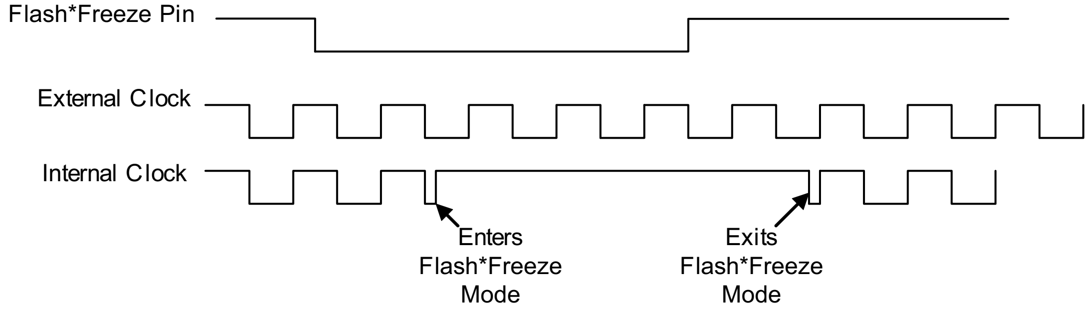


Figure 2-5 • **Narrow Clock Pulses During Flash*Freeze Entrance and Exit**

- I/O banks are not all deactivated simultaneously when entering Flash*Freeze mode. This can cause clocks and inputs to become disabled at different times, resulting in unexpected data being captured.
- Upon entering Flash*Freeze mode, all inputs and globals become tied High internally (except when an input hold state is used on IGLOO nano or IGLOO PLUS devices). If any of these signals are driven Low or tied Low externally, they will experience a Low to High transition internally when entering Flash*Freeze mode.
- Upon entering type 2 Flash*Freeze mode, ensure the LSICC signal (active High) does not de-assert. This can prevent the device from entering Flash*Freeze mode.
- Asynchronous input to output paths may experience output glitches. For example, on a direct in-to-out path, if the current state is '0' and the input bank turns off first, the input and then the output will transition to '1' before the output enters its Flash*Freeze state. This can be prevented by using latches in asynchronous in-to-out paths.
- The above situations can cause glitches or invalid data to be clocked into and preserved in the device. Refer to the "Flash*Freeze Design Guide" section on page 34 for solutions.

3 – Global Resources in Low Power Flash Devices

Introduction

IGLOO, Fusion, and ProASIC3 FPGA devices offer a powerful, low-delay VersaNet global network scheme and have extensive support for multiple clock domains. In addition to the Clock Conditioning Circuits (CCCs) and phase-locked loops (PLLs), there is a comprehensive global clock distribution network called a VersaNet global network. Each logical element (VersaTile) input and output port has access to these global networks. The VersaNet global networks can be used to distribute low-skew clock signals or high-fanout nets. In addition, these highly segmented VersaNet global networks contain spines (the vertical branches of the global network tree) and ribs that can reach all the VersaTiles inside their region. This allows users the flexibility to create low-skew local clock networks using spines. This document describes VersaNet global networks and discusses how to assign signals to these global networks and spines in a design flow. Details concerning low power flash device PLLs are described in the "Clock Conditioning Circuits in Low Power Flash Devices and Mixed Signal FPGAs" section on page 77. This chapter describes the low power flash devices' global architecture and uses of these global networks in designs.

Global Architecture

Low power flash devices offer powerful and flexible control of circuit timing through the use of global circuitry. Each chip has up to six CCCs, some with PLLs.

- In IGLOOe, ProASIC3EL, and ProASIC3E devices, all CCCs have PLLs—hence, 6 PLLs per device (except the PQ208 package, which has only 2 PLLs).
- In IGLOO, IGLOO nano, IGLOO PLUS, ProASIC3, and ProASIC3L devices, the west CCC contains a PLL core (except in 10 k through 30 k devices).
- In Fusion devices, the west CCC also contains a PLL core. In the two larger devices (AFS600 and AFS1500), the west and east CCCs each contain a PLL.

Refer to Table 4-6 on page 100 for details. Each PLL includes delay lines, a phase shifter (0°, 90°, 180°, 270°), and clock multipliers/dividers. Each CCC has all the circuitry needed for the selection and interconnection of inputs to the VersaNet global network. The east and west CCCs each have access to three chip global lines on each side of the chip (six chip global lines total). The CCCs at the four corners each have access to three quadrant global lines in each quadrant of the chip (except in 10 k through 30 k gate devices).

The nano 10 k, 15 k, and 20 k devices support four VersaNet global resources, and 30 k devices support six global resources. The 10 k through 30 k devices have simplified CCCs called CCC-GLs.

The flexible use of the VersaNet global network allows the designer to address several design requirements. User applications that are clock-resource-intensive can easily route external or gated internal clocks using VersaNet global routing networks. Designers can also drastically reduce delay penalties and minimize resource usage by mapping critical, high-fanout nets to the VersaNet global network.

Note: Microsemi recommends that you choose the appropriate global pin and use the appropriate global resource so you can realize these benefits.

The following sections give an overview of the VersaNet global network, the structure of the global network, access point for the global networks, and the clock aggregation feature that enables a design to have very low clock skew using spines.

Figure 3-6 shows all nine global inputs for the location A connected to the top left quadrant global network via CCC.

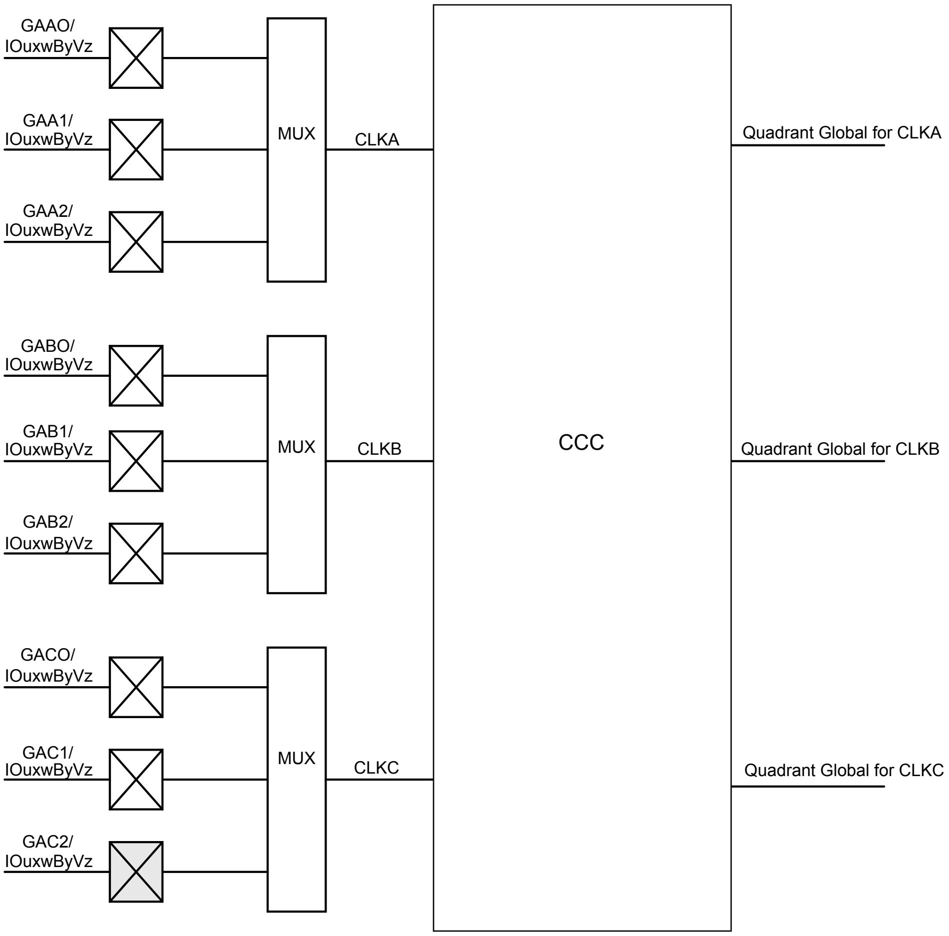


***Figure 3-6* • Global Inputs**

Since each bank can have a different I/O standard, the user should be careful to choose the correct global I/O for the design. There are 54 global pins available to access 18 global networks. For the single-ended and voltage-referenced I/O standards, you can use any of these three available I/Os to access the global network. For differential I/O standards such as LVDS and LVPECL, the I/O macro needs to be placed on (A0, A1), (B0, B1), (C0, C1), or a similar location. The unassigned global I/Os can be used as regular I/Os. Note that pin names starting with GF and GC are associated with the chip global networks, and GA, GB, GD, and GE are used for quadrant global networks. Table 3-2 on page 54 and Table 3-3 on page 55 show the general chip and quadrant global pin names.

Using Spines of Occupied Global Networks

When a signal is assigned to a global network, the flash switches are programmed to set the MUX select lines (explained in the "Clock Aggregation Architecture" section on page 61) to drive the spines of that network with the global net. However, if the global net is restricted from reaching into the scope of a spine, the MUX drivers of that spine are available for other high-fanout or critical signals (Figure 3-20).

For example, if you want to limit the CLK1_c signal to the left half of the chip and want to use the right side of the same global network for CLK2_c, you can add the following PDC commands:

```
define_region -name region1 -type inclusive 0 0 34 29
assign_net_macros region1 CLK1_c
assign_local_clock –net CLK2_c –type chip B2
```

Figure 3-20 **• Design Example Using Spines of Occupied Global Networks**

Conclusion

IGLOO, Fusion, and ProASIC3 devices contain 18 global networks: 6 chip global networks and 12 quadrant global networks. These global networks can be segmented into local low-skew networks called spines. The spines provide low-skew networks for the high-fanout signals of a design. These allow you up to 252 different internal/external clocks in an A3PE3000 device. This document describes the architecture for the global network, plus guidelines and methodologies in assigning signals to globals and spines.

Related Documents

User’s Guides

IGLOO, ProASIC3, SmartFusion, and Fusion Macro Library Guide

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

Available I/O Standards

Table 4-4 **• Available I/O Standards within CLKBUF and CLKBUF_LVDS/LVPECL Macros**

CLKBUF_LVCMOS5
CLKBUF_LVCMOS33 [1]
CLKBUF_LVCMOS25 [2]
CLKBUF_LVCMOS18
CLKBUF_LVCMOS15
CLKBUF_PCI
CLKBUF_PCIX [3]
CLKBUF_GTL25 [2,3]
CLKBUF_GTL33 [2,3]
CLKBUF_GTLP25 [2,3]
CLKBUF_GTLP33 [2,3]
CLKBUF_HSTL_I [2,3]
CLKBUF_HSTL_II [2,3]
CLKBUF_SSTL3_I [2,3]
CLKBUF_SSTL3_II [2,3]
CLKBUF_SSTL2_I [2,3]
CLKBUF_SSTL2_II [2,3]
CLKBUF_LVDS [4,5]
CLKBUF_LVPECL[5]

Notes:

1. *By default, the CLKBUF macro uses 3.3 V LVTTL I/O technology. For more details, refer to the IGLOO, ProASIC3, SmartFusion, and Fusion Macro Library Guide.*
2. *I/O standards only supported in ProASIC3E and IGLOOe families.*
3. *I/O standards only supported in the following Fusion devices: AFS600 and AFS1500.*
4. *B-LVDS and M-LVDS standards are supported by CLKBUF_LVDS.*
5. *Not supported for IGLOO nano and ProASIC3 nano devices.*

Global Synthesis Constraints

The Synplify® synthesis tool, by default, allows six clocks in a design for Fusion, IGLOO, and ProASIC3. When more than six clocks are needed in the design, a user synthesis constraint attribute, syn_global_buffers, can be used to control the maximum number of clocks (up to 18) that can be inferred by the synthesis engine.

High-fanout nets will be inferred with clock buffers and/or internal clock buffers. If the design consists of CCC global buffers, they are included in the count of clocks in the design.

The subsections below discuss the clock input source (global buffers with no programmable delays) and the clock conditioning functional block (global buffers with programmable delays and/or PLL function) in detail.

```
  wire VCC, GND;

  VCC VCC_1_net(.Y(VCC));
  GND GND_1_net(.Y(GND));
  CLKDLY Inst1(.CLK(CLK), .GL(GL), .DLYGL0(VCC), .DLYGL1(GND), .DLYGL2(VCC),
    .DLYGL3(GND), .DLYGL4(GND));

endmodule
```

Detailed Usage Information

Clock Frequency Synthesis

Deriving clocks of various frequencies from a single reference clock is known as frequency synthesis. The PLL has an input frequency range from 1.5 to 350 MHz. This frequency is automatically divided down to a range between 1.5 MHz and 5.5 MHz by input dividers (not shown in Figure 4-19 on page 100) between PLL macro inputs and PLL phase detector inputs. The VCO output is capable of an output range from 24 to 350 MHz. With dividers before the input to the PLL core and following the VCO outputs, the VCO output frequency can be divided to provide the final frequency range from 0.75 to 350 MHz. Using SmartGen, the dividers are automatically set to achieve the closest possible matches to the specified output frequencies.

Users should be cautious when selecting the desired PLL input and output frequencies and the I/O buffer standard used to connect to the PLL input and output clocks. Depending on the I/O standards used for the PLL input and output clocks, the I/O frequencies have different maximum limits. Refer to the family datasheets for specifications of maximum I/O frequencies for supported I/O standards. Desired PLL input or output frequencies will not be achieved if the selected frequencies are higher than the maximum I/O frequencies allowed by the selected I/O standards. Users should be careful when selecting the I/O standards used for PLL input and output clocks. Performing post-layout simulation can help detect this type of error, which will be identified with pulse width violation errors. Users are strongly encouraged to perform post-layout simulation to ensure the I/O standard used can provide the desired PLL input or output frequencies. Users can also choose to cascade PLLs together to achieve the high frequencies needed for their applications. Details of cascading PLLs are discussed in the "Cascading CCCs" section on page 125.

In SmartGen, the actual generated frequency (under typical operating conditions) will be displayed beside the requested output frequency value. This provides the ability to determine the exact frequency that can be generated by SmartGen, in real time. The log file generated by SmartGen is a useful tool in determining how closely the requested clock frequencies match the user specifications. For example, assume a user specifies 101 MHz as one of the secondary output frequencies. If the best output frequency that could be achieved were 100 MHz, the log file generated by SmartGen would indicate the actual generated frequency.

Simulation Verification

The integration of the generated PLL and CLKDLY modules is similar to any VHDL component or Verilog module instantiation in a larger design; i.e., there is no special requirement that users need to take into account to successfully synthesize their designs.

For simulation purposes, users need to refer to the VITAL or Verilog library that includes the functional description and associated timing parameters. Refer to the Software Tools section of the Microsemi SoC Products Group website to obtain the family simulation libraries. If Designer is installed, these libraries are stored in the following locations:

<Designer_Installation_Directory>\lib\vtl\95\proasic3.vhd

<Designer_Installation_Directory>\lib\vtl\95\proasic3e.vhd

<Designer_Installation_Directory>\lib\vlog\proasic3.v

<Designer_Installation_Directory>\lib\vlog\proasic3e.v

For Libero users, there is no need to compile the simulation libraries, as they are conveniently pre-compiled in the Model*Sim*® Microsemi simulation tool.

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).

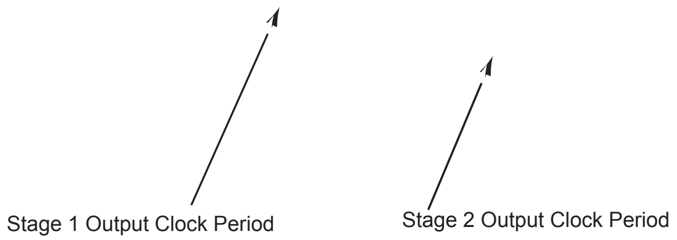


Figure 4-37 • **Model*Sim* Simulation Results**

SRAM Usage

The following descriptions refer to the usage of both RAM4K9 and RAM512X18.

Clocking

The dual-port SRAM blocks are only clocked on the rising edge. SmartGen allows falling-edge-triggered clocks by adding inverters to the netlist, hence achieving dual-port SRAM blocks that are clocked on either edge (rising or falling). For dual-port SRAM, each port can be clocked on either edge and by separate clocks by port. Note that for Automotive ProASIC3, the same clock, with an inversion between the two clock pins of the macro, should be used in design to prevent errors during compile.

Low power flash devices support inversion (bubble-pushing) throughout the FPGA architecture, including the clock input to the SRAM modules. Inversions added to the SRAM clock pin on the design schematic or in the HDL code will be automatically accounted for during design compile without incurring additional delay in the clock path.

The two-port SRAM can be clocked on the rising or falling edge of WCLK and RCLK.

If negative-edge RAM and FIFO clocking is selected for memory macros, clock edge inversion management (bubble-pushing) is automatically used within the development tools, without performance penalty.

Modes of Operation

There are two read modes and one write mode:

- Read Nonpipelined (synchronous—1 clock edge): In the standard read mode, new data is driven onto the RD bus in the same clock cycle following RA and REN valid. The read address is registered on the read port clock active edge, and data appears at RD after the RAM access time. Setting PIPE to OFF enables this mode.
- Read Pipelined (synchronous—2 clock edges): The pipelined mode incurs an additional clock delay from address to data but enables operation at a much higher frequency. The read address is registered on the read port active clock edge, and the read data is registered and appears at RD after the second read clock edge. Setting PIPE to ON enables this mode.
- Write (synchronous—1 clock edge): On the write clock active edge, the write data is written into the SRAM at the write address when WEN is HIGH. The setup times of the write address, write enables, and write data are minimal with respect to the write clock.

RAM Initialization

Each SRAM block can be individually initialized on power-up by means of the JTAG port using the UJTAG mechanism. The shift register for a target block can be selected and loaded with the proper bit configuration to enable serial loading. The 4,608 bits of data can be loaded in a single operation.

FIFO Features

The FIFO4KX18 macro is created by merging the RAM block with dedicated FIFO logic (Figure 6-6 on page 158). Since the FIFO logic can only be used in conjunction with the memory block, there is no separate FIFO controller macro. As with the RAM blocks, the FIFO4KX18 nomenclature does not refer to a possible aspect ratio, but rather to the deepest possible data depth and the widest possible data width. FIFO4KX18 can be configured into the following aspect ratios: 4,096×1, 2,048×2, 1,024×4, 512×9, and 256×18. In addition to being fully synchronous, the FIFO4KX18 also has the following features:

- Four FIFO flags: Empty, Full, Almost-Empty, and Almost-Full
- Empty flag is synchronized to the read clock
- Full flag is synchronized to the write clock
- Both Almost-Empty and Almost-Full flags have programmable thresholds
- Active-low asynchronous reset
- Active-low block enable
- Active-low write enable
- Active-high read enable
- Ability to configure the FIFO to either stop counting after the empty or full states are reached or to allow the FIFO counters to continue

I/O Architecture

I/O Tile

The I/O tile provides a flexible, programmable structure for implementing a large number of I/O standards. In addition, the registers available in the I/O tile can be used to support high-performance register inputs and outputs, with register enable if desired (Figure 7-2). The registers can also be used to support the JESD-79C Double Data Rate (DDR) standard within the I/O structure (see the "DDR for Microsemi’s Low Power Flash Devices" section on page 271 for more information). In addition, the registers available in the I/O tile can be used to support high-performance register inputs and outputs, with register enable if desired (Figure 7-2).

As depicted in Figure 7-2, all I/O registers share one CLR port. The output register and output enable register share one CLK port.

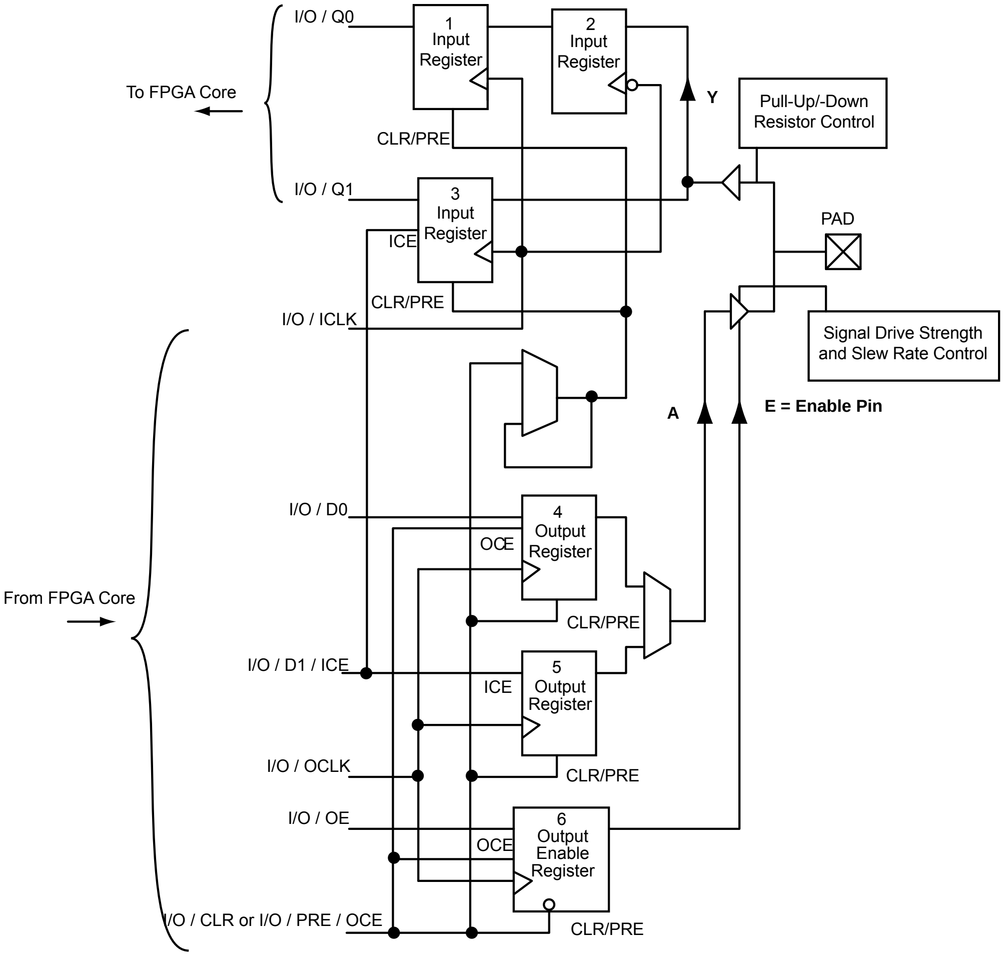


Figure 7-2 • **DDR Configured I/O Block Logical Representation**

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

	Clamp Diode [1]		Hot Insertion		5 V Input Tolerance [2]		
I/O Assignment	**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**	**Input and Output Buffer**
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.*

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	–

User I/O Naming Convention

IGLOO and ProASIC3

Due to the comprehensive and flexible nature of IGLOO and ProASIC3 device user I/Os, a naming scheme is used to show the details of each I/O (Figure 7-19 on page 207 and Figure 7-20 on page 207). The name identifies to which I/O bank it belongs, as well as pairing and pin polarity for differential I/Os.

I/O Nomenclature = FF/Gmn/IOuxwBy

Gmn is only used for I/Os that also have CCC access—i.e., global pins.

FF = Indicates the I/O dedicated for the Flash*Freeze mode activation pin in IGLOO and ProASIC3L devices only

G = Global

m = Global pin location associated with each CCC on the device: A (northwest corner), B (northeast corner), C (east middle), D (southeast corner), E (southwest corner), and F (west middle)

n = Global input MUX and pin number of the associated Global location m—either A0, A1, A2, B0, B1, B2, C0, C1, or C2. Refer to the "Global Resources in Low Power Flash Devices" section on page 47 for information about the three input pins per clock source MUX at CCC location m.

u = I/O pair number in the bank, starting at 00 from the northwest I/O bank and proceeding in a clockwise direction

x = P or U (Positive), N or V (Negative) for differential pairs, or R (Regular—single-ended) for the I/Os that support single-ended and voltage-referenced I/O standards only. U (Positive) or V (Negative)—for LVDS, DDR LVDS, B-LVDS, and M-LVDS only—restricts the I/O differential pair from being selected as an LVPECL pair.

w = D (Differential Pair), P (Pair), or S (Single-Ended). D (Differential Pair) if both members of the pair are bonded out to adjacent pins or are separated only by one GND or NC pin; P (Pair) if both members of the pair are bonded out but do not meet the adjacency requirement; or S (Single-Ended) if the I/O pair is not bonded out. For Differential Pairs (D), adjacency for ball grid packages means only vertical or horizontal. Diagonal adjacency does not meet the requirements for a true differential pair.

B = Bank

y = Bank number (0–3). The Bank number starts at 0 from the northwest I/O bank and proceeds in a clockwise direction.

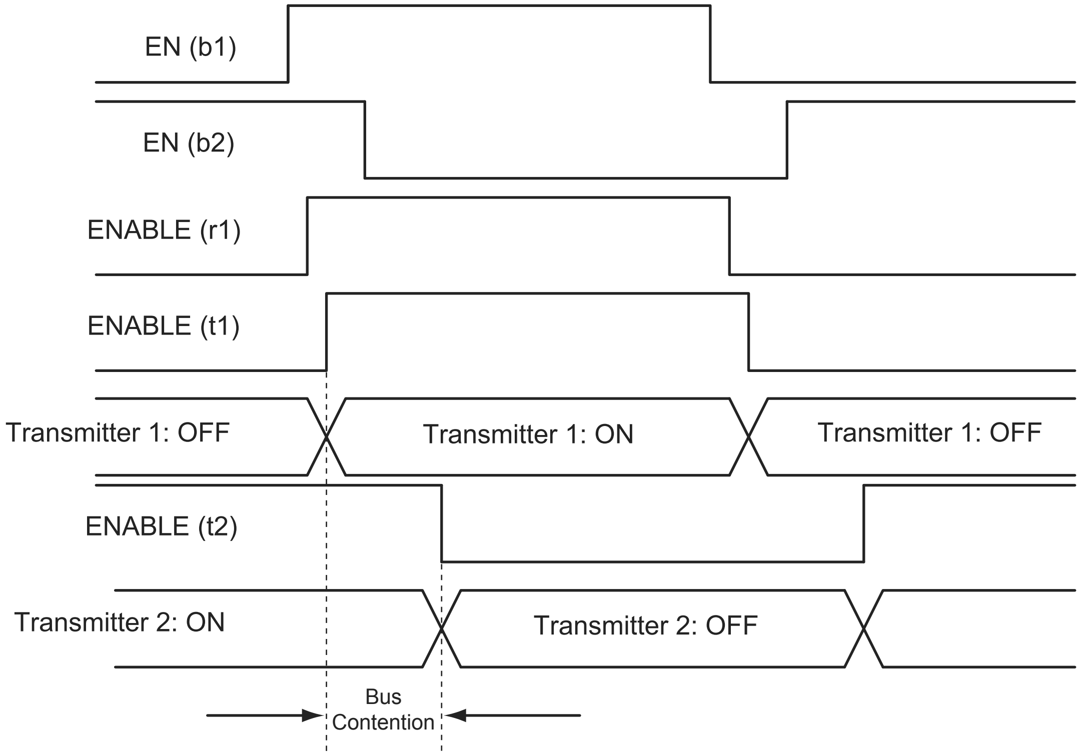
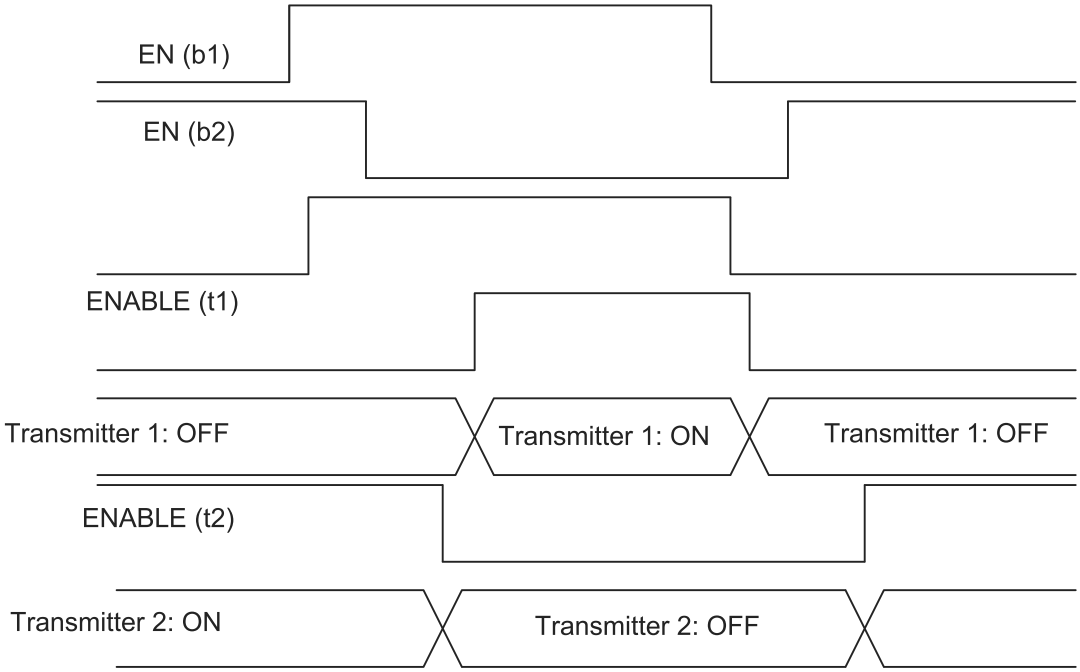


Figure 8-18 • **Timing Diagram (bypasses skew circuit)**



Result: No Bus Contention

Figure 8-19 • **Timing Diagram (with skew circuit selected)**

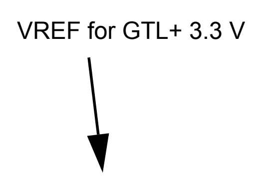


Figure 9-13 **• Selecting VREF Voltage for the I/O Bank**

Assigning VREF Pins for a Bank

The user can use default pins for VREF. In this case, select the **Use default pins for VREFs** check box (Figure 9-13). This option guarantees full VREF coverage of the bank. The equivalent PDC command is as follows:

```
set_vref_default [bank name]
```

To be able to choose VREF pins, adequate VREF pins must be created to allow legal placement of the compatible voltage-referenced I/Os.

To assign VREF pins manually, the PDC command is as follows:

```
set_vref –bank [bank name] [package pin numbers]
```

For ChipPlanner/PinEditor to show the range of a VREF pin, perform the following steps:

1. Assign VCCI to a bank using **MVN** > **Edit** > **I/O Bank Settings**.
2. Open **ChipPlanner**. Zoom in on an I/O package pin in that bank.
3. Highlight the pin and then right-click. Choose **Use Pin for VREF**.

Automatically Assigning Technologies to I/O Banks

The I/O Bank Assigner (IOBA) tool runs automatically when you run Layout. You can also use this tool from within the MultiView Navigator (Figure 9-17). The IOBA tool automatically assigns technologies and VREF pins (if required) to every I/O bank that does not currently have any technologies assigned to it. This tool is available when at least one I/O bank is unassigned.

To automatically assign technologies to I/O banks, choose I/O Bank Assigner from the **Tools** menu (or click the I/O Bank Assigner's toolbar button, shown in Figure 9-16).

Figure 9-16 **• I/O Bank Assigner's Toolbar Button**

Messages will appear in the Output window informing you when the automatic I/O bank assignment begins and ends. If the assignment is successful, the message "I/O Bank Assigner completed successfully" appears in the Output window, as shown in Figure 9-17.

Figure 9-17 **• I/O Bank Assigner Displays Messages in Output Window**

Instantiating DDR Registers

Using SmartGen is the simplest way to generate the appropriate RTL files for use in the design. Figure 10-4 shows an example of using SmartGen to generate a DDR SSTL2 Class I input register. SmartGen provides the capability to generate all of the DDR I/O cells as described. The user, through the graphical user interface, can select from among the many supported I/O standards. The output formats supported are Verilog, VHDL, and EDIF.

Figure 10-5 on page 277 through Figure 10-8 on page 280 show the I/O cell configured for DDR using SSTL2 Class I technology. For each I/O standard, the I/O pad is buffered by a special primitive that indicates the I/O standard type.

***Figure 10-4* • Example of Using SmartGen to Generate a DDR SSTL2 Class I Input Register**

11 – Programming Flash Devices

Introduction

This document provides an overview of the various programming options available for the Microsemi flash families. The electronic version of this document includes active links to all programming resources, which are available at http://www.microsemi.com/soc/products/hardware/default.aspx. For Microsemi antifuse devices, refer to the *Programming Antifuse Devices* document.

Summary of Programming Support

FlashPro4 and FlashPro3 are high-performance in-system programming (ISP) tools targeted at the latest generation of low power flash devices offered by the SmartFusion,® Fusion, IGLOO,® and ProASIC®3 families, including ARM-enabled devices. FlashPro4 and FlashPro3 offer extremely high performance through the use of USB 2.0, are high-speed compliant for full use of the 480 Mbps bandwidth, and can program ProASIC3 devices in under 30 seconds. Powered exclusively via USB, FlashPro4 and FlashPro3 provide a VPUMP voltage of 3.3 V for programming these devices.

FlashPro4 replaced FlashPro3 in 2010. FlashPro4 supports SmartFusion, Fusion, ProASIC3,and IGLOO devices as well as future generation flash devices. FlashPro4 also adds 1.2 V programming for IGLOO nano V2 devices. FlashPro4 is compatible with FlashPro3; however it adds a programming mode (PROG_MODE) signal to the previously unused pin 4 of the JTAG connector. The PROG_MODE goes high during programming and can be used to turn on a 1.5 V external supply for those devices that require 1.5 V for programming. If both FlashPro3 and FlashPro4 programmers are used for programming the same boards, pin 4 of the JTAG connector must not be connected to anything on the board because FlashPro4 uses pin 4 for PROG_MODE.

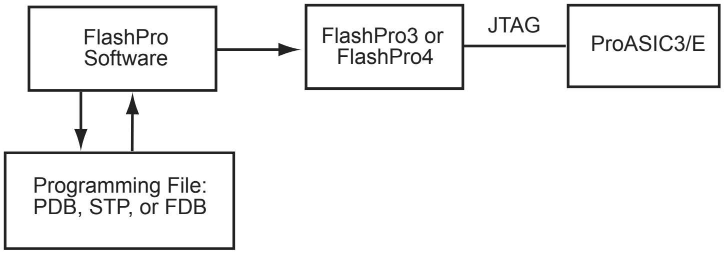


Figure 11-1 • **FlashPro Programming Setup**

Fine Tuning

In some applications, design constants or parameters need to be modified after programming the original design. The tuning process can be done using the UJTAG tile without reprogramming the device with new values. If the parameters or constants of a design are stored in distributed registers or embedded SRAM blocks, the new values can be shifted onto the JTAG TAP Controller pins, replacing the old values. The UJTAG tile is used as the "bridge" for data transfer between the JTAG pins and the FPGA VersaTiles or SRAM logic. Figure 17-5 shows a flow chart example for fine-tuning application steps using the UJTAG tile.

In Figure 17-5, the TMS signal sets the TAP Controller state machine to the appropriate states. The flow mainly consists of two steps: a) shifting the defined instruction and b) shifting the new data. If the target parameter is constantly used in the design, the new data can be shifted into a temporary shift register from UTDI. The UDRSH output of UJTAG can be used as a shift-enable signal, and UDRCK is the shift clock to the shift register. Once the shift process is completed and the TAP Controller state is moved to the Update_DR state, the UDRUPD output of the UJTAG can latch the new parameter value from the temporary register into a permanent location. This avoids any interruption or malfunctioning during the serial shift of the new value.

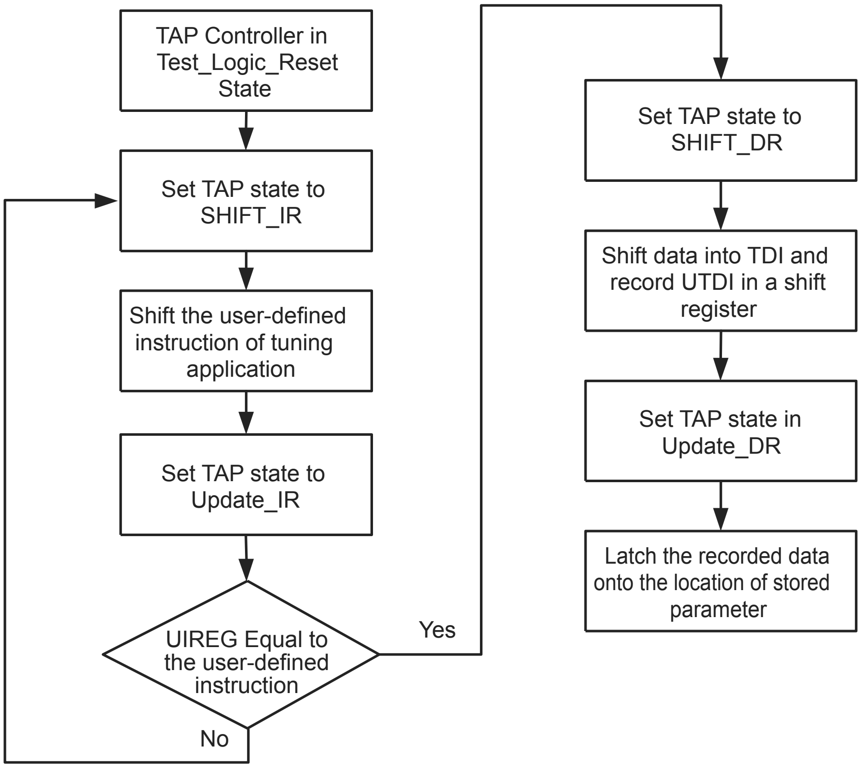


Figure 17-5 • **Flow Chart Example of Fine-Tuning an Application Using UJTAG**

B – Product Support

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

Customer Service

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

From North America, call 800.262.1060
From the rest of the world, call 650.318.4460
Fax, from anywhere in the world, 650.318.8044

Customer Technical Support Center

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

Technical Support

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

Website

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

Contacting the Customer Technical Support Center

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

Email

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

The technical support email address is soc_tech@microsemi.com.