Microchip Technology - A3PE3000L-1PQ208I 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	516096
Number of I/O	147
Number of Gates	3000000
Voltage - Supply	1.14V ~ 1.575V
Mounting Type	Surface Mount
Operating Temperature	-40°C ~ 100°C (TJ)
Package / Case	208-BFQFP
Supplier Device Package	208-PQFP (28x28)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/a3pe3000l-1pq208i

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong

1 – FPGA Array Architecture in Low Power Flash Devices

Device Architecture

Advanced Flash Switch

Unlike SRAM FPGAs, the low power flash devices use a live-at-power-up ISP flash switch as their programming element. Flash cells are distributed throughout the device to provide nonvolatile, reconfigurable programming to connect signal lines to the appropriate VersaTile inputs and outputs. In the flash switch, two transistors share the floating gate, which stores the programming information (Figure 1-1). One is the sensing transistor, which is only used for writing and verification of the floating gate voltage. The other is the switching transistor. The latter is used to connect or separate routing nets, or to configure VersaTile logic. It is also used to erase the floating gate. Dedicated high-performance lines are connected as required using the flash switch for fast, low-skew, global signal distribution throughout the device core. Maximum core utilization is possible for virtually any design. The use of the flash switch technology also removes the possibility of firm errors, which are increasingly common in SRAM-based FPGAs.

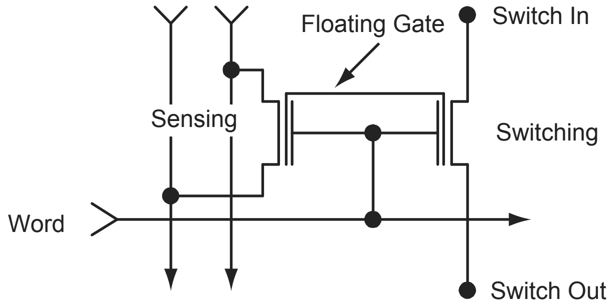

Figure 1-1 **• Flash-Based Switch**

Low Power Modes Overview

Table 2-2 summarizes the low power modes that achieve power consumption reduction when the FPGA or system is idle.

Table 2-2 • **Power Modes Summary**

Mode		VCCI	VCC	Core	Clocks	ULSICC Macro	To Enter Mode	To Resume Operation	Trigger
Active		On	On	On	On	N/A	Initiate clock	None	–
Static	Idle	On	On	On	Off	N/A	Stop clock	Initiate clock	External
	Flash*Freeze type 1	On	On	On	On*	N/A	Assert FF pin	Deassert FF pin	External
	Flash*Freeze type 2	On	On	On	On*	Used to enter Flash*Freeze mode	Assert FF pin and assert LSICC	Deassert FF pin	External
Sleep		On	Off	Off	Off	N/A	Shut down VCC	Turn on VCC supply	External
Shutdown		Off	Off	Off	Off	N/A	Shut down VCC and VCCI supplies	Turn on VCC and VCCI supplies	External

* *External clocks can be left toggling while the device is in Flash*Freeze mode. Clocks generated by the embedded PLL will be turned off automatically.*

Static (Idle) Mode

In Static (Idle) mode, none of the clock inputs is switching, and static power is the only power consumed by the device. This mode can be achieved by switching off the incoming clocks to the FPGA, thus benefitting from reduced power consumption. In addition, I/Os draw only minimal leakage current. In this mode, embedded SRAM, I/Os, and registers retain their values so the device can enter and exit this mode just by switching the clocks on or off.

If the device-embedded PLL is used as the clock source, Static (Idle) mode can easily be entered by pulling the PLL POWERDOWN pin LOW (active Low), which will turn off the PLL.

- There will be added skew and clock insertion delay due to the clock gating circuit. The user should analyze external setup/hold times carefully. The user should also ensure the additional skew across the clock gating filter circuit is accounted for in any paths where the launch register is driven from the filter input clock and captured by a register driven by the gated clock filter output clock.

Power Analysis

SmartPower identifies static and dynamic power consumption problems quickly within a design. It provides a hierarchical view, allowing users to drill down and estimate the power consumption of individual components or events. SmartPower analyzes power consumption for nets, gates, I/Os, memories, clocks, cores, clock domains, power supply rails, peak power during a clock cycle, and switching transitions.

SmartPower generates detailed hierarchical reports of the dynamic power consumption of a design for easy inspection. These reports include design-level power summary, average switching activity, and ambient and junction temperature readings. Enter the target clock and data frequencies for a design, and let SmartPower perform a detailed and accurate power analysis. SmartPower supports importing files in the VCD (Value-Change Dump) format as specified in the IEEE 1364 standard. It also supports the Synopsys® Switching Activity Interchange Format (SAIF) standard. Support for these formats lets designers generate switching activity information in a variety of simulators and then import this information directly into SmartPower.

For portable or battery-operated applications, a power profile feature enables you to measure power and battery life, based on a sequence of operational modes of the design. In most portable and battery-operated applications, the system is seldom fully "on" 100 percent of the time. "On" is a combination of fully active, standby, sleep, or other functional modes. SmartPower allows users to create a power profile for a design by specifying operational modes and the percent of time the device will run in each of the modes. Power is calculated for each of the modes, and total power is calculated based on the weighted average of all modes.

SmartPower also provides an estimated battery life based on the power profile. The current capacity for a given battery is entered and used to estimate the life of the battery. The result is an accurate and realistic indication of battery life.

More information on SmartPower can be found on the Microsemi SoC Products Group website:
http://www.microsemi.com/soc/products/software/libero/smartpower.aspx.

Additional Power Conservation Techniques

IGLOO, IGLOO nano, IGLOO PLUS, ProASIC3L, and RT ProASIC3 FPGAs provide many ways to inherently conserve power; however, there are also several design techniques that can be used to reduce power on the board.

- Microsemi recommends that the designer use the minimum number of I/O banks possible and tie any unused power supplies (such as V_{CCPLL}, V_{CCI}, VMV, and V_{PUMP}) to ground.
- Leave unused I/O ports floating. Unused I/Os are configured by the software as follows:
  - Output buffer is disabled (with tristate value of high impedance)
  - Input buffer is disabled (with tristate value of high impedance)
- Use the lowest available voltage I/O standard, the lowest drive strength, and the slowest slew rate to reduce I/O switching contribution to power consumption.
- Advanced and pro I/O banks may consume slightly higher static current than standard and standard plus banks—avoid using advanced and pro banks whenever practical.
  - The small static power benefit obtained by avoiding advanced or pro I/O banks is usually negligible compared to the benefit of using a low power I/O standard.
- Deselect RAM blocks that are not being used.
- Only enable read and write ports on RAM blocks when they are needed.
- Gating clocks LOW offers improved static power of RAM blocks.
- Drive the FF port of RAM blocks with the Flash_Freeze_Enabled signal from the Flash*Freeze management IP.
- Drive inputs to the full voltage level so that all transistors are turned on or off completely.

Table 3-5 **• Globals/Spines/Rows for IGLOO PLUS Devices**

IGLOO PLUS Devices	Chip Globals	Quadrant Globals (4×3)	Clock Trees	Globals/ Spines per Tree	Total Spines per Device	VersaTiles in Each Tree	Total VersaTiles	Rows in Each Spine
AGLP030	6	0	2	9	18	384*	792	12
AGLP060	6	12	4	9	36	384*	1,584	12
AGLP125	6	12	8	9	72	384*	3,120	12

*Note: *Clock trees that are located at far left and far right will support more VersaTiles.*

Table 3-6 **• Globals/Spines/Rows for Fusion Devices**

Fusion Device	Chip Globals	Quadrant Globals (4×3)	Clock Trees	Globals/ Spines per Tree	Total Spines per Device	VersaTiles in Each Tree	Total VersaTiles	Rows in Each Spine
AFS090	6	12	6	9	54	384	2,304	12
AFS250	6	12	8	9	72	768	6,144	24
AFS600	6	12	12	9	108	1,152	13,824	36
AFS1500	6	12	20	9	180	1,920	38,400	60

```
    wire VCC, GND;

    VCC VCC_1_net(.Y(VCC));
    GND GND_1_net(.Y(GND));
    PLL Core(.CLKA(CLKA), .EXTFB(GND), .POWERDOWN(POWERDOWN),
        .GLA(GLA), .LOCK(LOCK), .GLB(), .YB(), .GLC(), .YC(),
        .OADIV0(GND), .OADIV1(GND), .OADIV2(GND), .OADIV3(GND),
        .OADIV4(GND), .OAMUX0(GND), .OAMUX1(GND), .OAMUX2(VCC),
        .DLYGLA0(GND), .DLYGLA1(GND), .DLYGLA2(GND), .DLYGLA3(GND)
        , .DLYGLA4(GND), .OBDIV0(GND), .OBDIV1(GND), .OBDIV2(GND),
        .OBDIV3(GND), .OBDIV4(GND), .OBMUX0(GND), .OBMUX1(GND),
        .OBMUX2(GND), .DLYYB0(GND), .DLYYB1(GND), .DLYYB2(GND),
        .DLYYB3(GND), .DLYYB4(GND), .DLYGLB0(GND), .DLYGLB1(GND),
        .DLYGLB2(GND), .DLYGLB3(GND), .DLYGLB4(GND), .OCDIV0(GND),
        .OCDIV1(GND), .OCDIV2(GND), .OCDIV3(GND), .OCDIV4(GND),
        .OCMUX0(GND), .OCMUX1(GND), .OCMUX2(GND), .DLYYC0(GND),
        .DLYYC1(GND), .DLYYC2(GND), .DLYYC3(GND), .DLYYC4(GND),
        .DLYGLC0(GND), .DLYGLC1(GND), .DLYGLC2(GND), .DLYGLC3(GND)
        , .DLYGLC4(GND), .FINDIV0(VCC), .FINDIV1(GND), .FINDIV2(
        VCC), .FINDIV3(GND), .FINDIV4(GND), .FINDIV5(GND),
        .FINDIV6(GND), .FBDIV0(VCC), .FBDIV1(GND), .FBDIV2(VCC),
        .FBDIV3(GND), .FBDIV4(GND), .FBDIV5(GND), .FBDIV6(GND),
        .FBDLY0(GND), .FBDLY1(GND), .FBDLY2(GND), .FBDLY3(GND),
        .FBDLY4(GND), .FBSEL0(VCC), .FBSEL1(GND), .XDLYSEL(GND),
        .VCOSEL0(GND), .VCOSEL1(GND), .VCOSEL2(GND));
    defparam Core.VCOFREQUENCY = 33.000;
endmodule
```

The "PLL Configuration Bits Description" section on page 106 provides descriptions of the PLL configuration bits for completeness. The configuration bits are shown as busses only for purposes of illustration. They will actually be broken up into individual pins in compilation libraries and all simulation models. For example, the FBSEL[1:0] bus will actually appear as pins FBSEL1 and FBSEL0. The setting of these select lines for the static PLL configuration is performed by the software and is completely transparent to the user.

5 – FlashROM in Microsemi's Low Power Flash Devices

Introduction

The Fusion, IGLOO, and ProASIC3 families of low power flash-based devices have a dedicated nonvolatile FlashROM memory of 1,024 bits, which provides a unique feature in the FPGA market. The FlashROM can be read, modified, and written using the JTAG (or UJTAG) interface. It can be read but not modified from the FPGA core. Only low power flash devices contain on-chip user nonvolatile memory (NVM).

Architecture of User Nonvolatile FlashROM

Low power flash devices have 1 kbit of user-accessible nonvolatile flash memory on-chip that can be read from the FPGA core fabric. The FlashROM is arranged in eight banks of 128 bits (16 bytes) during programming. The 128 bits in each bank are addressable as 16 bytes during the read-back of the FlashROM from the FPGA core. Figure 5-1 shows the FlashROM logical structure.

The FlashROM can only be programmed via the IEEE 1532 JTAG port. It cannot be programmed directly from the FPGA core. When programming, each of the eight 128-bit banks can be selectively reprogrammed. The FlashROM can only be reprogrammed on a bank boundary. Programming involves an automatic, on-chip bank erase prior to reprogramming the bank. The FlashROM supports synchronous read. The address is latched on the rising edge of the clock, and the new output data is stable after the falling edge of the same clock cycle. For more information, refer to the timing diagrams in the DC and Switching Characteristics chapter of the appropriate datasheet. The FlashROM can be read on byte boundaries. The upper three bits of the FlashROM address from the FPGA core define the bank being accessed. The lower four bits of the FlashROM address from the FPGA core define which of the 16 bytes in the bank is being accessed.

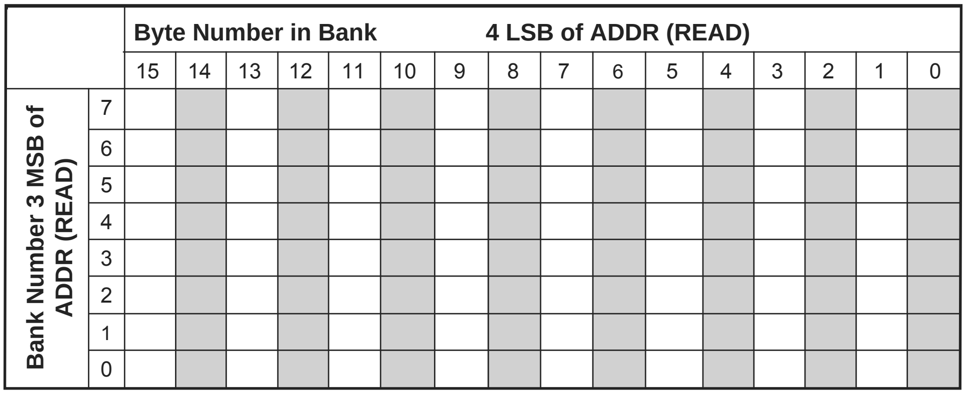

Figure 5-1 • **FlashROM Architecture**

Figure 5-12 shows the programming file generator, which enables different STAPL file generation methods. When you select **Program FlashROM** and choose the UFC file, the FlashROM Settings window appears, as shown in Figure 5-13. In this window, you can select the FlashROM page you want to program and the data value for the configured regions. This enables you to use a different page for different programming files.

Figure 5-12 • **Programming File Generator**

Figure 5-13 • **Setting FlashROM during Programming File Generation**

The programming hardware and software can load the FlashROM with the appropriate STAPL file. Programming software handles the single STAPL file that contains multiple FlashROM contents for multiple devices, and programs the FlashROM in sequential order (e.g., for device serialization). This feature is supported in the programming software. After programming with the STAPL file, you can run DEVICE_INFO to check the FlashROM content.

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

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

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

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

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

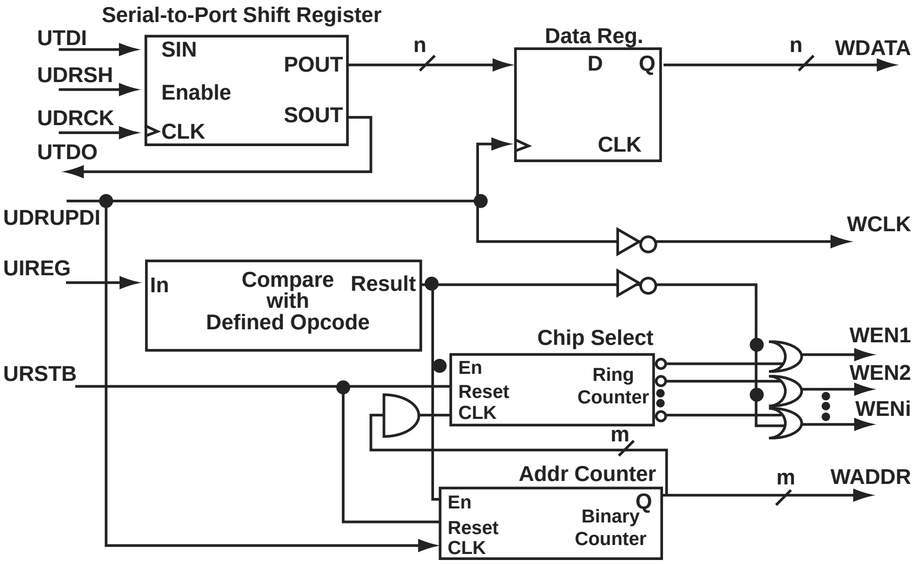

Figure 6-9 • **Block Diagram of a Sample User Interface**

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

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

IGLOO and ProASIC3

For boards and cards with three levels of staging, card power supplies must have time to reach their final values before the I/Os are connected. Pay attention to the sizing of power supply decoupling capacitors on the card to ensure that the power supplies are not overloaded with capacitance.

Cards with three levels of staging should have the following sequence:

- Grounds
- Powers
- I/Os and other pins

For Level 3 and Level 4 compliance with the 30K gate device, cards with two levels of staging should have the following sequence:

- Grounds
- Powers, I/Os, and other pins

Cold-Sparing Support

Cold-sparing refers to the ability of a device to leave system data undisturbed when the system is powered up, while the component itself is powered down, or when power supplies are floating.

The resistor value is calculated based on the decoupling capacitance on a given power supply. The RC constant should be greater than 3 µs.

To remove resistor current during operation, it is suggested that the resistor be disconnected (e.g., with an NMOS switch) from the power supply after the supply has reached its final value. Refer to the "Power-Up/-Down Behavior of Low Power Flash Devices" section on page 373 for details on cold-sparing.

Cold-sparing means that a subsystem with no power applied (usually a circuit board) is electrically connected to the system that is in operation. This means that all input buffers of the subsystem must present very high input impedance with no power applied so as not to disturb the operating portion of the system.

The 30 k gate devices fully support cold-sparing, since the I/O clamp diode is always off (see Table 7-12 on page 193). If the 30 k gate device is used in applications requiring cold-sparing, a discharge path from the power supply to ground should be provided. This can be done with a discharge resistor or a switched resistor. This is necessary because the 30K gate devices do not have built-in I/O clamp diodes.

For other IGLOO and ProASIC3 devices, since the I/O clamp diode is always active, cold-sparing can be accomplished either by employing a bus switch to isolate the device I/Os from the rest of the system or by driving each I/O pin to 0 V. If the resistor is chosen, the resistor value must be calculated based on decoupling capacitance on a given power supply on the board (this decoupling capacitance is in parallel with the resistor). The RC time constant should ensure full discharge of supplies before cold-sparing functionality is required. The resistor is necessary to ensure that the power pins are discharged to ground every time there is an interruption of power to the device.

IGLOOe and ProASIC3E devices support cold-sparing for all I/O configurations. Standards, such as PCI, that require I/O clamp diodes can also achieve cold-sparing compliance, since clamp diodes get disconnected internally when the supplies are at 0 V.

When targeting low power applications, I/O cold-sparing may add additional current if a pin is configured with either a pull-up or pull-down resistor and driven in the opposite direction. A small static current is induced on each I/O pin when the pin is driven to a voltage opposite to the weak pull resistor. The current is equal to the voltage drop across the input pin divided by the pull resistor. Refer to the "Detailed I/O DC Characteristics" section of the appropriate family datasheet for the specific pull resistor value for the corresponding I/O standard.

For example, assuming an LVTTL 3.3 V input pin is configured with a weak pull-up resistor, a current will flow through the pull-up resistor if the input pin is driven LOW. For LVTTL 3.3 V, the pull-up resistor is ~45 kΩ, and the resulting current is equal to 3.3 V / 45 kΩ = 73 µA for the I/O pin. This is true also when a weak pull-down is chosen and the input pin is driven HIGH. This current can be avoided by driving the input LOW when a weak pull-down resistor is used and driving it HIGH when a weak pull-up resistor is used.

This current draw can occur in the following cases:

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.

Table 8-11 • **Hot-Swap Level 3**

Description	Hot-swap while bus idle
Power Applied to Device	Yes
Bus State	Held idle (no ongoing I/O processes during insertion/removal)
Card Ground Connection	Reset must be maintained for 1 ms before, during, and after insertion/removal.
Device Circuitry Connected to Bus Pins	Must remain glitch-free during power-up or power-down
Example Application	Board bus shared with card bus is "frozen," and there is no toggling activity on the bus. It is critical that the logic states set on the bus signal not be disturbed during card insertion/removal.
Compliance of IGLOO and ProASIC3 Devices	30 k gate devices, all IGLOOe/ProASIC3E devices: Compliant with two levels of staging (first: GND; second: all other pins) Other IGLOO/ProASIC3 devices: Compliant: Option A – Two levels of staging (first: GND; second: all other pins) together with bus switch on the I/Os Option B – Three levels of staging (first: GND; second: supplies; third: all other pins)

Table 8-12 • **Hot-Swap Level 4**

Description	Hot-swap on an active bus
Power Applied to Device	Yes
Bus State	Bus may have active I/O processes ongoing, but device being inserted or removed must be idle.
Card Ground Connection	Reset must be maintained for 1 ms before, during, and after insertion/removal.
Device Circuitry Connected to Bus Pins	Must remain glitch-free during power-up or power-down
Example Application	There is activity on the system bus, and it is critical that the logic states set on the bus signal not be disturbed during card insertion/removal.
Compliance of IGLOO and ProASIC3 Devices	30 k gate devices, all IGLOOe/ProASIC3E devices: Compliant with two levels of staging (first: GND; second: all other pins) Other IGLOO/ProASIC3 devices: Compliant: Option A – Two levels of staging (first: GND; second: all other pins) together with bus switch on the I/Os Option B – Three levels of staging (first: GND; second: supplies; third: all other pins)

IGLOOe and ProASIC3E

For devices requiring Level 3 and/or Level 4 compliance, the board drivers connected to the I/Os must have 10 kΩ (or lower) output drive resistance at hot insertion, and 1 kΩ (or lower) output drive resistance at hot removal. This resistance is the transmitter resistance sending a signal toward the I/O, and no additional resistance is needed on the board. If that cannot be assured, three levels of staging can be used to achieve Level 3 and/or Level 4 compliance. Cards with two levels of staging should have the following sequence:

- Grounds
- Powers, I/Os, and other pins

Cold-Sparing Support

Cold-sparing refers to the ability of a device to leave system data undisturbed when the system is powered up, while the component itself is powered down, or when power supplies are floating.

Cold-sparing is supported on ProASIC3E devices only when the user provides resistors from each power supply to ground. The resistor value is calculated based on the decoupling capacitance on a given power supply. The RC constant should be greater than 3 µs.

To remove resistor current during operation, it is suggested that the resistor be disconnected (e.g., with an NMOS switch) from the power supply after the supply has reached its final value. Refer to the "Power-Up/-Down Behavior of Low Power Flash Devices" section on page 373 for details on cold-sparing.

Cold-sparing means that a subsystem with no power applied (usually a circuit board) is electrically connected to the system that is in operation. This means that all input buffers of the subsystem must present very high input impedance with no power applied so as not to disturb the operating portion of the system.

The 30 k gate devices fully support cold-sparing, since the I/O clamp diode is always off (see Table 8-13 on page 231). If the 30 k gate device is used in applications requiring cold-sparing, a discharge path from the power supply to ground should be provided. This can be done with a discharge resistor or a switched resistor. This is necessary because the 30 k gate devices do not have built-in I/O clamp diodes.

For other IGLOOe and ProASIC3E devices, since the I/O clamp diode is always active, cold-sparing can be accomplished either by employing a bus switch to isolate the device I/Os from the rest of the system or by driving each I/O pin to 0 V. If the resistor is chosen, the resistor value must be calculated based on decoupling capacitance on a given power supply on the board (this decoupling capacitance is in parallel with the resistor). The RC time constant should ensure full discharge of supplies before cold-sparing functionality is required. The resistor is necessary to ensure that the power pins are discharged to ground every time there is an interruption of power to the device.

IGLOOe and ProASIC3E devices support cold-sparing for all I/O configurations. Standards, such as PCI, that require I/O clamp diodes can also achieve cold-sparing compliance, since clamp diodes get disconnected internally when the supplies are at 0 V.

When targeting low power applications, I/O cold-sparing may add additional current if a pin is configured with either a pull-up or pull-down resistor and driven in the opposite direction. A small static current is induced on each I/O pin when the pin is driven to a voltage opposite to the weak pull resistor. The current is equal to the voltage drop across the input pin divided by the pull resistor. Refer to the "Detailed I/O DC Characteristics" section of the appropriate family datasheet for the specific pull resistor value for the corresponding I/O standard.

For example, assuming an LVTTL 3.3 V input pin is configured with a weak pull-up resistor, a current will flow through the pull-up resistor if the input pin is driven LOW. For LVTTL 3.3 V, the pull-up resistor is ~45 kΩ, and the resulting current is equal to 3.3 V / 45 kΩ = 73 µA for the I/O pin. This is true also when a weak pull-down is chosen and the input pin is driven High. This current can be avoided by driving the input Low when a weak pull-down resistor is used and driving it High when a weak pull-up resistor is used.

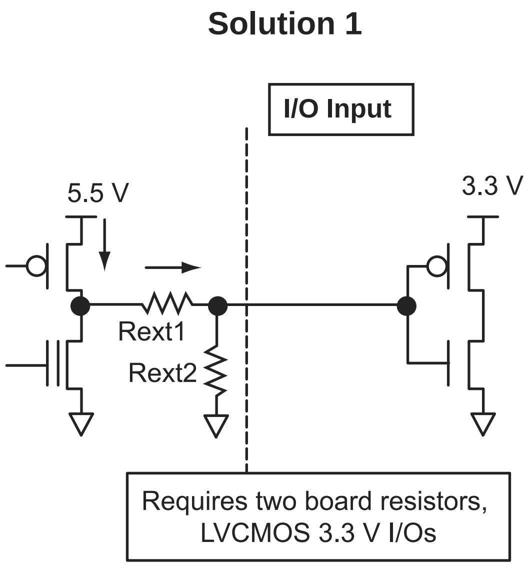

Figure 8-10 • **Solution 1**

Solution 2

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

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

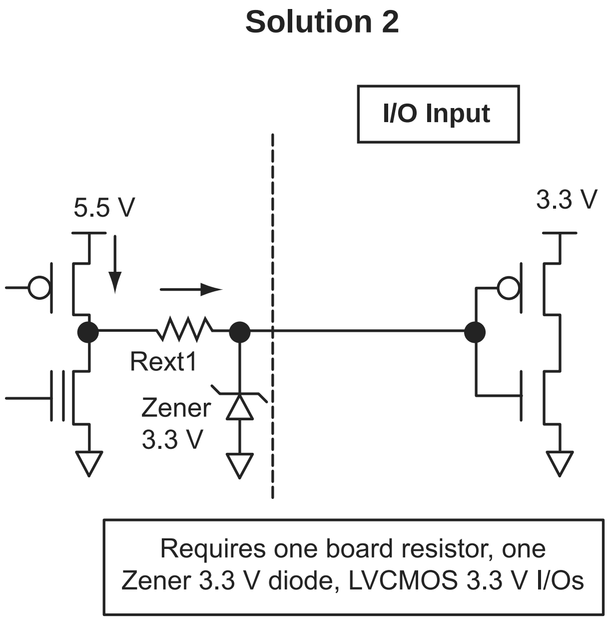

Figure 8-11 • **Solution 2**

10 – DDR for Microsemi's Low Power Flash Devices

Introduction

The I/Os in Fusion, IGLOO, and ProASIC3 devices support Double Data Rate (DDR) mode. In this mode, new data is present on every transition (or clock edge) of the clock signal. This mode doubles the data transfer rate compared with Single Data Rate (SDR) mode, where new data is present on one transition (or clock edge) of the clock signal. Low power flash devices have DDR circuitry built into the I/O tiles. I/Os are configured to be DDR receivers or transmitters by instantiating the appropriate special macros (examples shown in Figure 10-4 on page 276 and Figure 10-5 on page 277) and buffers (DDR_OUT or DDR_REG) in the RTL design. This document discusses the options the user can choose to configure the I/Os in this mode and how to instantiate them in the design.

Double Data Rate (DDR) Architecture

Low power flash devices support 350 MHz DDR inputs and outputs. In DDR mode, new data is present on every transition of the clock signal. Clock and data lines have identical bandwidths and signal integrity requirements, making them very efficient for implementing very high-speed systems. High-speed DDR interfaces can be implemented using LVDS (not applicable for IGLOO nano and ProASIC3 nano devices). In IGLOOe, ProASIC3E, AFS600, and AFS1500 devices, DDR interfaces can also be implemented using the HSTL, SSTL, and LVPECL I/O standards. The DDR feature is primarily implemented in the FPGA core periphery and is not tied to a specific I/O technology or limited to any I/O standard.

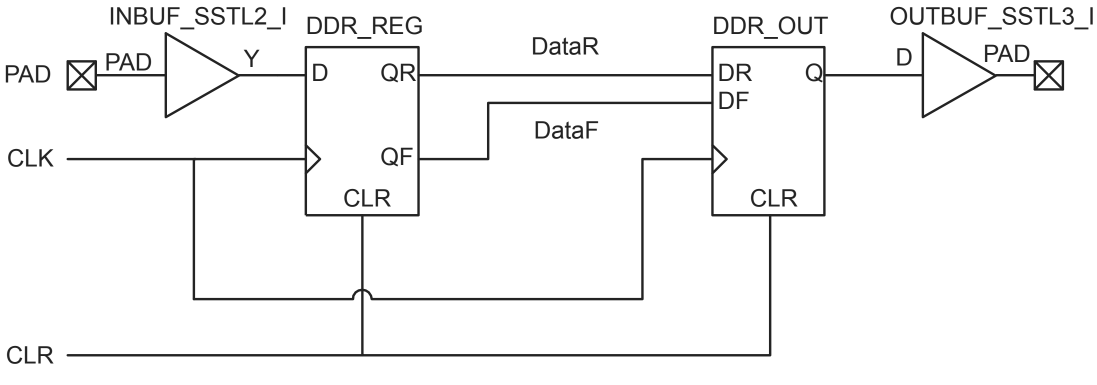

Figure 10-1 • **DDR Support in Low Power Flash Devices**

Programming Support in Flash Devices

The flash FPGAs listed in Table 11-1 support flash in-system programming and the functions described in this document.

***Table 11-1* • Flash-Based FPGAs**

Series	Family*	Description
IGLOO	IGLOO	Ultra-low power 1.2 V to 1.5 V FPGAs with Flash*Freeze technology
	IGLOOe	Higher density IGLOO FPGAs with six PLLs and additional I/O standards
	IGLOO nano	The industry's lowest-power, smallest-size solution, supporting 1.2 V to 1.5 V core voltage with Flash*Freeze technology
	IGLOO PLUS	IGLOO FPGAs with enhanced I/O capabilities
ProASIC3	ProASIC3	Low power, high-performance 1.5 V FPGAs
	ProASIC3E	Higher density ProASIC3 FPGAs with six PLLs and additional I/O standards
	ProASIC3 nano	Lowest-cost solution with enhanced I/O capabilities
	ProASIC3L	ProASIC3 FPGAs supporting 1.2 V to 1.5 V core voltage with Flash*Freeze technology
	RT ProASIC3	Radiation-tolerant RT3PE600L and RT3PE3000L
	Military ProASIC3/EL	Military temperature A3PE600L, A3P1000, and A3PE3000L
	Automotive ProASIC3	ProASIC3 FPGAs qualified for automotive applications
SmartFusion	SmartFusion	Mixed-signal FPGA integrating FPGA fabric, programmable microcontroller subsystem (MSS), including programmable analog and ARM® Cortex™-M3 hard processor and flash memory in a monolithic device
Fusion	Fusion	Mixed signal FPGA integrating ProASIC3 FPGA fabric, programmable analog block, support for ARM® Cortex™-M1 soft processors, and flash memory into a monolithic device
ProASIC	ProASIC	First generation ProASIC devices
	ProASIC$^{\underline{PLUS}}$	Second generation ProASIC devices

*Note: *The device names link to the appropriate datasheet, including product brief, DC and switching characteristics, and packaging information.*

IGLOO Terminology

In documentation, the terms IGLOO series and IGLOO devices refer to all of the IGLOO devices as listed in Table 11-1. Where the information applies to only one product line or limited devices, these exclusions will be explicitly stated.

ProASIC3 Terminology

In documentation, the terms ProASIC3 series and ProASIC3 devices refer to all of the ProASIC3 devices as listed in Table 11-1. Where the information applies to only one product line or limited devices, these exclusions will be explicitly stated.

To further understand the differences between the IGLOO and ProASIC3 devices, refer to the *Industry's Lowest Power FPGAs Portfolio*.

IEEE 1532 (JTAG) Interface

The supported industry-standard IEEE 1532 programming interface builds on the IEEE 1149.1 (JTAG) standard. IEEE 1532 defines the standardized process and methodology for ISP. Both silicon and software issues are addressed in IEEE 1532 to create a simplified ISP environment. Any IEEE 1532 compliant programmer can be used to program low power flash devices. Device serialization is not supported when using the IEEE1532 standard. Refer to the standard for detailed information about IEEE 1532.

Security

Unlike SRAM-based FPGAs that require loading at power-up from an external source such as a microcontroller or boot PROM, Microsemi nonvolatile devices are live at power-up, and there is no bitstream required to load the device when power is applied. The unique flash-based architecture prevents reverse engineering of the programmed code on the device, because the programmed data is stored in nonvolatile memory cells. Each nonvolatile memory cell is made up of small capacitors and any physical deconstruction of the device will disrupt stored electrical charges.

Each low power flash device has a built-in 128-bit Advanced Encryption Standard (AES) decryption core, except for the 30 k gate devices and smaller. Any FPGA core or FlashROM content loaded into the device can optionally be sent as encrypted bitstream and decrypted as it is loaded. This is particularly suitable for applications where device updates must be transmitted over an unsecured network such as the Internet. The embedded AES decryption core can prevent sensitive data from being intercepted (Figure 13-1 on page 331). A single 128-bit AES Key (32 hex characters) is used to encrypt FPGA core programming data and/or FlashROM programming data in the Microsemi tools. The low power flash devices also decrypt with a single 128-bit AES Key. In addition, low power flash devices support a Message Authentication Code (MAC) for authentication of the encrypted bitstream on-chip. This allows the encrypted bitstream to be authenticated and prevents erroneous data from being programmed into the device. The FPGA core, FlashROM, and Flash Memory Blocks (FBs), in Fusion only, can be updated independently using a programming file that is AES-encrypted (cipher text) or uses plain text.

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.

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

Microsemi Corporate Headquarters
One Enterprise, Aliso Viejo CA 92656 USA
Within the USA: +1 (949) 380-6100
Sales: +1 (949) 380-6136
Fax: +1 (949) 215-4996

Microsemi Corporation (NASDAQ: MSCC) offers a comprehensive portfolio of semiconductor solutions for: aerospace, defense and security; enterprise and communications; and industrial and alternative energy markets. Products include high-performance, high-reliability analog and RF devices, mixed signal and RF integrated circuits, customizable SoCs, FPGAs, and complete subsystems. Microsemi is headquartered in Aliso Viejo, Calif. Learn more at **www.microsemi.com**.