

Welcome to <u>E-XFL.COM</u>

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	5
Details	•

Details	
Product Status	Obsolete
Number of LABs/CLBs	-
Number of Logic Elements/Cells	-
Total RAM Bits	110592
Number of I/O	154
Number of Gates	600000
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/microsemi/a3p600l-pqg208i

Email: info@E-XFL.COM

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



FPGA Array Architecture in Low Power Flash Devices



Note: + Flash*Freeze mode is supported on IGLOO devices.









FPGA Array Architecture in Low Power Flash Devices

Array Coordinates

During many place-and-route operations in the Microsemi Designer software tool, it is possible to set constraints that require array coordinates. Table 1-2 provides array coordinates of core cells and memory blocks for IGLOO and ProASIC3 devices. Table 1-3 provides the information for IGLOO PLUS devices. Table 1-4 on page 17 provides the information for IGLOO nano and ProASIC3 nano devices. The array coordinates are measured from the lower left (0, 0). They can be used in region constraints for specific logic groups/blocks, designated by a wildcard, and can contain core cells, memories, and I/Os.

I/O and cell coordinates are used for placement constraints. Two coordinate systems are needed because there is not a one-to-one correspondence between I/O cells and core cells. In addition, the I/O coordinate system changes depending on the die/package combination. It is not listed in Table 1-2. The Designer ChipPlanner tool provides the array coordinates of all I/O locations. I/O and cell coordinates are used for placement constraints. However, I/O placement is easier by package pin assignment.

Figure 1-9 on page 17 illustrates the array coordinates of a 600 k gate device. For more information on how to use array coordinates for region/placement constraints, see the *Designer User's Guide* or online help (available in the software) for software tools.

		VersaTile		aTiles		Memory Rows		Entire Die	
Device		Mi	n.	Max.		Bottom	Тор	Min.	Max.
IGLOO	ProASIC3/ ProASIC3L	x	у	x	у	(x, y)	(x, y)	(x, y)	(x, y)
AGL015	A3P015	3	2	34	13	None	None	(0, 0)	(37, 15)
AGL030	A3P030	3	3	66	13	None	None	(0, 0)	(69, 15)
AGL060	A3P060	3	2	66	25	None	(3, 26)	(0, 0)	(69, 29)
AGL125	A3P125	3	2	130	25	None	(3, 26)	(0, 0)	(133, 29)
AGL250	A3P250/L	3	2	130	49	None	(3, 50)	(0, 0)	(133, 53)
AGL400	A3P400	3	2	194	49	None	(3, 50)	(0, 0)	(197, 53)
AGL600	A3P600/L	3	4	194	75	(3, 2)	(3, 76)	(0, 0)	(197, 79)
AGL1000	A3P1000/L	3	4	258	99	(3, 2)	(3, 100)	(0, 0)	(261, 103)
AGLE600	A3PE600/L, RT3PE600L	3	4	194	75	(3, 2)	(3, 76)	(0, 0)	(197, 79)
	A3PE1500	3	4	322	123	(3, 2)	(3, 124)	(0, 0)	(325, 127)
AGLE3000	A3PE3000/L, RT3PE3000L	3	6	450	173	(3, 2) or (3, 4)	(3, 174) or (3, 176)	(0, 0)	(453, 179)

Table 1-2 • IGLOO and ProASIC3 Array Coordinates

Table 1-3 • IGLOO PLUS Array Coordinates

	VersaTiles		Memory Rows		Entire Die			
Device	Mi	n.	Ма	ax.	Bottom	Тор	Min.	Max.
IGLOO PLUS	х	У	x	У	(x, y)	(x, y)	(x, y)	(x, y)
AGLP030	2	3	67	13	None	None	(0, 0)	(69, 15)
AGLP060	2	2	67	25	None	(3, 26)	(0, 0)	(69, 29)
AGLP125	2	2	131	25	None	(3, 26)	(0, 0)	(133, 29)

Flash*Freeze Technology and Low Power Modes

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.

1/1

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 Ty	pe 2 – Signai Ass	ertion and Deassertion values	i.
					_

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

Notes:

T.

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.

 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.



Clock Conditioning Circuits in Low Power Flash Devices and Mixed Signal FPGAs

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).

Microsemi

Clock Conditioning Circuits in Low Power Flash Devices and Mixed Signal FPGAs

Dividers n and m (the input divider and feedback divider, respectively) provide integer frequency division factors from 1 to 128. The output dividers u, v, and w provide integer division factors from 1 to 32. Frequency scaling of the reference clock CLKA is performed according to the following formulas:

$$f_{GLA} = f_{CLKA} \times m / (n \times u) - GLA Primary PLL Output Clock$$

$$EQ 4-1$$

$$f_{GLB} = f_{YB} = f_{CLKA} \times m / (n \times v) - GLB Secondary 1 PLL Output Clock(s)$$

$$EQ 4-2$$

$$f_{GLC} = f_{YC} = f_{CLKA} \times m / (n \times w) - GLC$$
 Secondary 2 PLL Output Clock(s)

EQ 4-3

SmartGen provides a user-friendly method of generating the configured PLL netlist, which includes automatically setting the division factors to achieve the closest possible match to the requested frequencies. Since the five output clocks share the *n* and *m* dividers, the achievable output frequencies are interdependent and related according to the following formula:

$$f_{GLA} = f_{GLB} \times (v / u) = f_{GLC} \times (w / u)$$

EQ 4-4

Clock Delay Adjustment

There are a total of seven configurable delay elements implemented in the PLL architecture.

Two of the delays are located in the feedback path, entitled System Delay and Feedback Delay. System Delay provides a fixed delay of 2 ns (typical), and Feedback Delay provides selectable delay values from 0.6 ns to 5.56 ns in 160 ps increments (typical). For PLLs, delays in the feedback path will effectively advance the output signal from the PLL core with respect to the reference clock. Thus, the System and Feedback delays generate negative delay on the output clock. Additionally, each of these delays can be independently bypassed if necessary.

The remaining five delays perform traditional time delay and are located at each of the outputs of the PLL. Besides the fixed global driver delay of 0.755 ns for each of the global networks, the global multiplexer outputs (GLA, GLB, and GLC) each feature an additional selectable delay value, as given in Table 4-7.

Device	Typical	Starting Values	Increments	Ending Value
ProASIC3	200 ps	0 to 735 ps	200 ps	6.735 ns
IGLOO/ProASIC3L 1.5 V	360 ps	0 to 1.610 ns	360 ps	12.410 ns
IGLOO/ProASIC3L 1.2 V	580 ps	0 to 2.880 ns	580 ps	20.280 ns

Table 4-7 • Delay Values in Libero SoC Software per Device Family

The additional YB and YC signals have access to a selectable delay from 0.6 ns to 5.56 ns in 160 ps increments (typical). This is the same delay value as the CLKDLY macro. It is similar to CLKDLY, which bypasses the PLL core just to take advantage of the phase adjustment option with the delay value.

The following parameters must be taken into consideration to achieve minimum delay at the outputs (GLA, GLB, GLC, YB, and YC) relative to the reference clock: routing delays from the PLL core to CCC outputs, core outputs and global network output delays, and the feedback path delay. The feedback path delay acts as a time advance of the input clock and will offset any delays introduced beyond the PLL core output. The routing delays are determined from back-annotated simulation and are configuration-dependent.

When SmartGen is used to define the configuration that will be shifted in via the serial interface, SmartGen prints out the values of the 81 configuration bits. For ease of use, several configuration bits are automatically inferred by SmartGen when the dynamic PLL core is generated; however, <71:73> (STATASEL, STATBSEL, STATCSEL) and <77:79> (DYNASEL, DYNBSEL, DYNCSEL) depend on the input clock source of the corresponding CCC. Users must first run Layout in Designer to determine the exact setting for these ports. After Layout is complete, generate the "CCC_Configuration" report by choosing **Tools** > **Reports** > **CCC_Configuration** in the Designer software. Refer to "PLL Configuration Bits Description" on page 106 for descriptions of the PLL configuration bits. For simulation purposes, bits <71:73> and <78:80> are "don't care." Therefore, it is strongly suggested that SmartGen be used to generate the correct configuration bit settings for the dynamic PLL core.

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		:	dyn_pll_hardio
Family		:	ProASIC3E
Output Format		:	VERILOG
Туре		:	Dynamic CCC
Input Freq(MH	z)	:	30.000
CLKA Source		:	Hardwired I/O
Feedback Dela	y Value Index	:	1
Feedback Mux	Select	:	1
XDLY Mux Sele	ct	:	No
Primary Freq(MHz)	:	33.000
Primary Phase	Shift	:	0
Primary Delay	Value Index	:	1
Primary Mux S	elect	:	4
Secondary1 Fr	eq(MHz)	:	40.000
Use GLB	-	:	YES
Use YB		:	NO
GLB Delay Val	ue Index	:	1
YB Delay Valu	e Index	:	1
Secondary1 Ph	aseShift	:	0
Secondaryl Mu	x Select	:	0
Secondaryl In	put Freg(MHz)	:	40.000
CLKB Source		:	Hardwired I/O
Secondary2 Fr	eq(MHz)	:	50.000
Use GLC	1. ,	:	YES
Use YC		:	NO
GLC Delay Val	ue Index	:	1
YC Delay Valu	e Index	:	1
Secondarv2 Ph	aseShift	:	0
Secondarv2 Mu	x Select	:	0
Secondary2 In	put Freg(MHz)	:	50.000
CLKC Source	F	:	Hardwired I/O
olito bouloc			narawiroa 1,0
Configuration	Bits:		
FINDIV[6:0]	0000101		
FBDIV[6:0]	0100000		
OADIV[4:0]	00100		
OBDIV[4:0]	00000		
OCDIV[4:0]	00000		
OAMUX[2:0]	100		
OBMUX[2:0]	000		
OCMUX[2:0]	000		
FBSEL[1:0]	01		
FBDLY[4:0]	00000		
XDLYSEL	0		

00000

00000

DLYGLA[4:0] DLYGLB[4:0]



FlashROM in Microsemi's Low Power Flash Devices

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.

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 Other IGLOO/ProASIC3 devices: Compliant if bus switch used to isolate FPGA I/Os from rest of system 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. Other IGLOO/ProASIC3 devices: Compliant

Microsemi

I/O Structures in IGLOO and ProASIC3 Devices

Table 7-10 • 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	30K 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 7-11 • 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	30K 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)

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 μ s.

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 k Ω , and the resulting current is equal to 3.3 V / 45 k Ω = 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:

Power-Up Behavior

Low power flash devices are power-up/-down friendly; i.e., no particular sequencing is required for power-up and power-down. This eliminates extra board components for power-up sequencing, such as a power-up sequencer.

During power-up, all I/Os are tristated, irrespective of I/O macro type (input buffers, output buffers, I/O buffers with weak pull-ups or weak pull-downs, etc.). Once I/Os become activated, they are set to the user-selected I/O macros. Refer to the "Power-Up/-Down Behavior of Low Power Flash Devices" section on page 373 for details.

Drive Strength

Low power flash devices have up to seven programmable output drive strengths. The user can select the drive strength of a particular output in the I/O Attribute Editor or can instantiate a specialized I/O macro, such as OUTBUF_S_12 (slew = low, out_drive = 12 mA).

The maximum available drive strength is 24 mA per I/O. Though no I/O should be forced to source or sink more than 24 mA indefinitely, I/Os may handle a higher amount of current (refer to the device IBIS model for maximum source/sink current) during signal transition (AC current). Every device package has its own power dissipation limit; hence, power calculation must be performed accurately to determine how much current can be tolerated per I/O within that limit.

I/O Interfacing

Low power flash devices are 5 V–input– and 5 V–output–tolerant if certain I/O standards are selected (refer to the "5 V Input and Output Tolerance" section on page 232). Along with other low-voltage I/O macros, this 5 V tolerance makes these devices suitable for many types of board component interfacing.

	(Clock	I/O			
Interface	Туре	Frequency	Туре	Signals In	Signals Out	Data I/O
GM	Src Sync	125 MHz	LVTTL	8	8	125 Mbps
ТВІ	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	\leq 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	\geq 622 Mbps
SFI-4.1	Src Sync	622 MHz	LVDS	16	16	622 Mbps
CSIX L1	Sys Sync	\leq 250 MHz	HSTL1	32,64,96,128	32,64,96,128	\leq 250 Mbps
Hyper Transport	Sys Sync DDR	\leq 800 MHz	LVDS	2,4,8,16	2,4,8,16	\leq 1.6 Gbps
Rapid I/O Parallel	Sys Sync DDR	250 MHz – 1 GHz	LVDS	8,16	8,16	\leq 2 Gbps
Star Fabric	CDR		LVDS	4	4	622 Mbps

Table 8-19 • High-Level Interface Examples

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

Compiling the Design

During Compile, a PDC I/O constraint file can be imported along with the netlist file. If only the netlist file is compiled, certain I/O assignments need to be completed before proceeding to Layout. All constraints that can be entered in PDC can also be entered using ChipPlanner, I/O Attribute Editor, and PinEditor.

There are certain rules that must be followed in implementing I/O register combining and the I/O DDR macro (refer to the I/O Registers section of the handbook for the device that you are using and the "DDR" section on page 256 for details). Provided these rules are met, the user can enable or disable I/O register combining by using the PDC command set_io portname -register yes |no in the I/O Attribute Editor or selecting a check box in the Compile Options dialog box (see Figure 9-7). The Compile Options dialog box appears when the design is compiled for the first time. It can also be accessed by choosing **Options** > **Compile** during successive runs. I/O register combining is off by default. The PDC command overrides the setting in the Compile Options dialog box.

Figure 9-7 • Setting Register Combining During Compile

Understanding the Compile Report

The I/O bank report is generated during Compile and displayed in the log window. This report lists the I/O assignments necessary before Layout can proceed.

When Designer is started, the I/O Bank Assigner tool is run automatically if the Layout command is executed. The I/O Bank Assigner takes care of the necessary I/O assignments. However, these assignments can also be made manually with MVN or by importing the PDC file. Refer to the "Assigning Technologies and VREF to I/O Banks" section on page 264 for further description.

The I/O bank report can also be extracted from Designer by choosing **Tools** > **Report** and setting the Report Type to **IOBank**.

This report has the following tables: I/O Function, I/O Technology, I/O Bank Resource Usage, and I/O Voltage Usage. This report is useful if the user wants to do I/O assignments manually.



DDR for Microsemi's Low Power Flash Devices

Table 10-2 • DDR I/O Options (continued)

DDR Register	I/O Type	I/O Standard	Sub-Ontions	Comments	
Transmit Register	Tristate	Normal	Enable Polarity	Low/bigb (low default)	
(continued)	Buffer	LVTTL	Output Drive	2, 4, 6, 8, 12,16, 24, 36 mA (8 m/ default)	
			Slew Rate	Low/high (high default)	
			Enable Polarity	Low/high (low default)	
			Pull-Up/-Down	None (default)	
		LVCMOS	Voltage	1.5 V, 1.8 V, 2.5 V, 5 V (1.5 V default)	
			Output Drive	2, 4, 6, 8, 12, 16, 24, 36 mA (8 mA default)	
			Slew Rate	Low/high (high default)	
			Enable Polarity	Low/high (low default)	
			Pull-Up/-Down	None (default)	
		PCI/PCI-X	Enable Polarity	Low/high (low default)	
		GTL/GTL+	Voltage	1.8 V, 2.5 V, 3.3 V (3.3 V default)	
			Enable Polarity	Low/high (low default)	
		HSTL	Class	I / II (I default)	
			Enable Polarity	Low/high (low default)	
		SSTL2/SSTL3	Class	I / II (I default)	
			Enable Polarity	Low/high (low default)	
	Bidirectional Buffer	Normal	Enable Polarity	Low/high (low default)	
		LVTTL	Output Drive	2, 4, 6, 8, 12, 16, 24, 36 mA (8 mA default)	
			Slew Rate	Low/high (high default)	
			Enable Polarity	Low/high (low default)	
			Pull-Up/-Down	None (default)	
		LVCMOS	Voltage	1.5 V, 1.8 V, 2.5 V, 5 V (1.5 V default)	
			Enable Polarity	Low/high (low default)	
			Pull-Up	None (default)	
		PCI/PCI-X	None		
			Enable Polarity	Low/high (low default)	
		GTL/GTL+	Voltage	1.8 V, 2.5 V, 3.3 V (3.3 V default)	
			Enable Polarity	Low/high (low default)	
		HSTL	Class	I / II (I default)	
			Enable Polarity	Low/high (low default)	
		SSTL2/SSTL3	Class	I / II (I default)	
			Enable Polarity	Low/high (low default)	

Note: *IGLOO nano and ProASIC3 nano devices do not support differential inputs.



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.

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 ^{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.*

Related Documents

Below is a list of related documents, their location on the Microsemi SoC Products Group website, and a brief summary of each document.

Application Notes

Programming Antifuse Devices http://www.microsemi.com/soc/documents/AntifuseProgram_AN.pdf Implementation of Security in Actel's ProASIC and ProASIC^{PLUS} Flash-Based FPGAs http://www.microsemi.com/soc/documents/Flash_Security_AN.pdf

User's Guides

FlashPro Programmers

FlashPro4,¹ FlashPro3, FlashPro Lite, and FlashPro² http://www.microsemi.com/soc/products/hardware/program_debug/flashpro/default.aspx *FlashPro User's Guide* http://www.microsemi.com/soc/documents/FlashPro_UG.pdf The FlashPro User's Guide includes hardware and software setup, self-test instructions, use instructions, and a troubleshooting / error message guide.

Silicon Sculptor 3 and Silicon Sculptor II

http://www.microsemi.com/soc/products/hardware/program_debug/ss/default.aspx

Other Documents

http://www.microsemi.com/soc/products/solutions/security/default.aspx#flashlock The security resource center describes security in Microsemi Flash FPGAs. *Quality and Reliability Guide* http://www.microsemi.com/soc/documents/RelGuide.pdf *Programming and Functional Failure Guidelines* http://www.microsemi.com/soc/documents/FA_Policies_Guidelines_5-06-00002.pdf

^{1.} FlashPro4 replaced FlashPro3 in Q1 2010.

^{2.} FlashPro is no longer available.

Microsemi

ProASIC3L FPGA Fabric User's Guide

Date	Changes	Page
v1.3 (December 2008)	The "Programming Support in Flash Devices" section was updated to include IGLOO nano and ProASIC3 nano devices.	288
	The "Flash Devices" section was updated to include information for IGLOO nano devices. The following sentence was added: IGLOO PLUS devices can also be operated at any voltage between 1.2 V and 1.5 V; the Designer software allows 50 mV increments in the voltage.	289
	Table 11-4 · Programming Ordering Codes was updated to replace FP3-26PIN- ADAPTER with FP3-10PIN-ADAPTER-KIT.	294
	Table 14-6 · Programmer Device Support was updated to add IGLOO nano and ProASIC3 nano devices. AGL400 was added to the IGLOO portion of the table.	317
v1.2 (October 2008)	The "Programming Support in Flash Devices" section was revised to include new families and make the information more concise.	288
	Figure 11-1 \cdot FlashPro Programming Setup and the "Programming Support in Flash Devices" section are new.	287, 288
	Table 14-6 · Programmer Device Support was updated to include A3PE600L with the other ProASIC3L devices, and the RT ProASIC3 family was added.	317
v1.1 (March 2008)	The "Flash Devices" section was updated to include the IGLOO PLUS family. The text, "Voltage switching is required in-system to switch from a 1.2 V core to 1.5 V core for programming," was revised to state, "Although the device can operate at 1.2 V core voltage, the device can only be reprogrammed when the core voltage is 1.5 V. Voltage switching is required in-system to switch from a 1.2 V supply (V_{CC} , V_{CCI} , and V_{JTAG}) to 1.5 V for programming."	289
	The ProASIC3L family was added to Table 14-6 · Programmer Device Support as a separate set of rows rather than combined with ProASIC3 and ProASIC3E devices. The IGLOO PLUS family was included, and AGL015 and A3P015 were added.	317

Figure 12-19 • FlashLock Pass Key, Previously Programmed Devices

It is important to note that when the security settings need to be updated, the user also needs to select the **Security settings** check box in Step 1, as shown in Figure 12-10 on page 314 and Figure 12-11 on page 314, to modify the security settings. The user must consider the following:

- If only a new AES key is necessary, the user must re-enter the same Pass Key previously
 programmed into the device in Designer and then generate a programming file with the same
 Pass Key and a different AES key. This ensures the programming file can be used to access and
 program the device and the new AES key.
- If a new Pass Key is necessary, the user can generate a new programming file with a new Pass Key (with the same or a new AES key if desired). However, for programming, the user must first load the original programming file with the Pass Key that was previously used to unlock the device. Then the new programming file can be used to program the new security settings.

Advanced Options

As mentioned, there may be applications where more complicated security settings are required. The "Custom Security Levels" section in the *FlashPro User's Guide* describes different advanced options available to aid the user in obtaining the best available security settings. Microprocessor Programming of Microsemi's Low Power Flash Devices

Remote Upgrade via TCP/IP

Transmission Control Protocol (TCP) provides a reliable bitstream transfer service between two endpoints on a network. TCP depends on Internet Protocol (IP) to move packets around the network on its behalf. TCP protects against data loss, data corruption, packet reordering, and data duplication by adding checksums and sequence numbers to transmitted data and, on the receiving side, sending back packets and acknowledging the receipt of data.

The system containing the low power flash device can be assigned an IP address when deployed in the field. When the device requires an update (core or FlashROM), the programming instructions along with the new programming data (AES-encrypted cipher text) can be sent over the Internet to the target system via the TCP/IP protocol. Once the MCU receives the instruction and data, it can proceed with the FPGA update. Low power flash devices support Message Authentication Code (MAC), which can be used to validate data for the target device. More details are given in the "Message Authentication Code (MAC) Validation/Authentication" section.

Hardware Requirement

To facilitate the programming of the low power flash families, the system must have a microprocessor (with access to the device JTAG pins) to process the programming algorithm, memory to store the programming algorithm, programming data, and the necessary programming voltage. Refer to the relevant datasheet for programming voltages.

Security

Encrypted Programming

As an additional security measure, the devices are equipped with AES decryption. AES works in two steps. The first step is to program a key into the devices in a secure or trusted programming center (such as Microsemi SoC Products Group In-House Programming (IHP) center). The second step is to encrypt any programming files with the same encryption key. The encrypted programming file will only work with the devices that have the same key. The AES used in the low power flash families is the 128-bit AES decryption engine (Rijndael algorithm).

Message Authentication Code (MAC) Validation/Authentication

As part of the AES decryption flow, the devices are equipped with a MAC validation/authentication system. MAC is an authentication tag, also called a checksum, derived by applying an on-chip authentication scheme to a STAPL file as it is loaded into the FPGA. MACs are computed and verified with the same key so they can only be verified by the intended recipient. When the MCU system receives the AES-encrypted programming data (cipher text), it can validate the data by loading it into the FPGA and performing a MAC verification prior to loading the data, via a second programming pass, into the FPGA core cells. This prevents erroneous or corrupt data from getting into the FPGA.

Low power flash devices with AES and MAC are superior to devices with only DES or 3DES encryption. Because the MAC verifies the correctness of the data, the FPGA is protected from erroneous loading of invalid programming data that could damage a device (Figure 15-5 on page 355).

The AES with MAC enables field updates over public networks without fear of having the design stolen. An encrypted programming file can only work on devices with the correct key, rendering any stolen files