



Welcome to [E-XFL.COM](https://www.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.

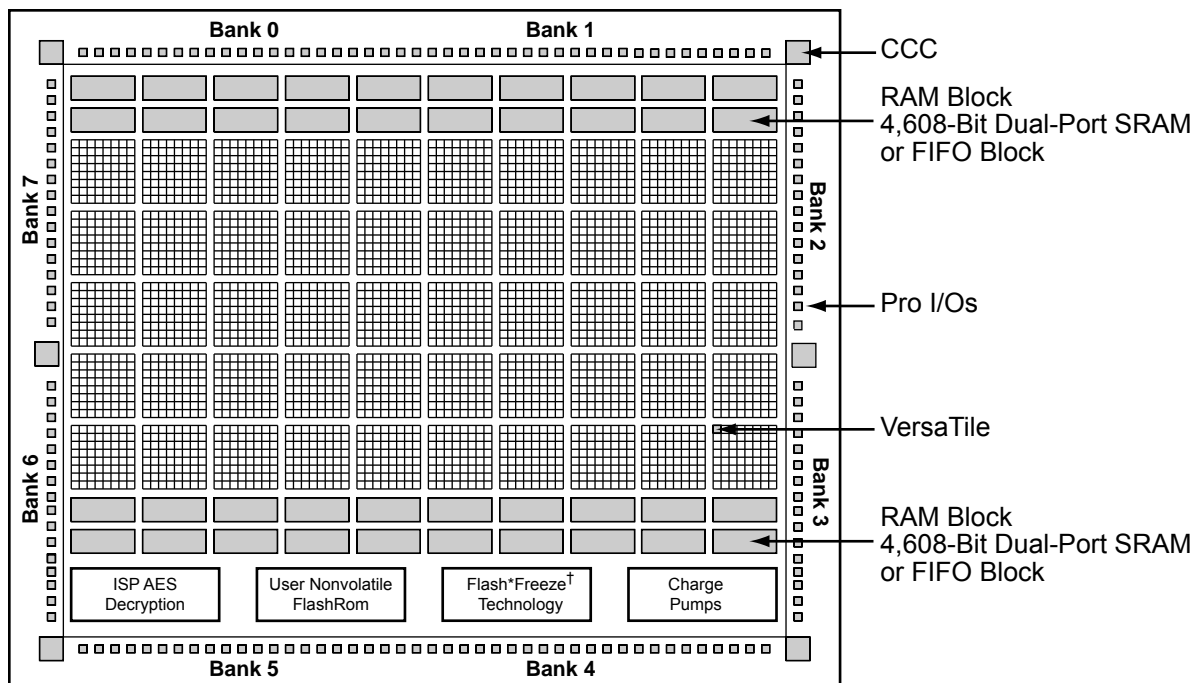
### Applications of Embedded - FPGAs

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	-
Number of I/O	77
Number of Gates	30000
Voltage - Supply	1.425V ~ 1.575V
Mounting Type	Surface Mount
Operating Temperature	-20°C ~ 85°C (TJ)
Package / Case	100-TQFP
Supplier Device Package	100-VQFP (14x14)
Purchase URL	<a href="https://www.e-xfl.com/product-detail/microchip-technology/a3pn030-z1vq100">https://www.e-xfl.com/product-detail/microchip-technology/a3pn030-z1vq100</a>

8	I/O Software Control in Low Power Flash Devices	185
	Flash FPGAs I/O Support	186
	Software-Controlled I/O Attributes	187
	Implementing I/Os in Microsemi Software	188
	Assigning Technologies and VREF to I/O Banks	198
	Conclusion	203
	Related Documents	203
	List of Changes	204
9	DDR for Microsemi's Low Power Flash Devices	205
	Introduction	205
	Double Data Rate (DDR) Architecture	205
	DDR Support in Flash-Based Devices	206
	I/O Cell Architecture	207
	Input Support for DDR	209
	Output Support for DDR	209
	Instantiating DDR Registers	210
	Design Example	216
	Conclusion	218
	List of Changes	219
10	Programming Flash Devices	221
	Introduction	221
	Summary of Programming Support	221
	Programming Support in Flash Devices	222
	General Flash Programming Information	223
	Important Programming Guidelines	229
	Related Documents	231
	List of Changes	232
11	Security in Low Power Flash Devices	235
	Security in Programmable Logic	235
	Security Support in Flash-Based Devices	236
	Security Architecture	237
	Security Features	238
	Security in Action	242
	FlashROM Security Use Models	245
	Generating Programming Files	247
	Conclusion	258
	Glossary	258
	References	258
	Related Documents	259
	List of Changes	259
12	In-System Programming (ISP) of Microsemi's Low Power Flash Devices Using FlashPro4/3/3X	261
	Introduction	261
	ISP Architecture	261
	ISP Support in Flash-Based Devices	262
	Programming Voltage (VPUMP) and VJTAG	263
	Nonvolatile Memory (NVM) Programming Voltage	263



Note: Flash\*Freeze technology only applies to IGL00e devices.

**Figure 1-7 • IGL00e and ProASIC3E Device Architecture Overview (AGLE600 device is shown)**

## I/O State of Newly Shipped Devices

Devices are shipped from the factory with a test design in the device. The power-on switch for VCC is OFF by default in this test design, so I/Os are tristated by default. Tristated means the I/O is not actively driven and floats. The exact value cannot be guaranteed when it is floating. Even in simulation software, a tristate value is marked as unknown. Due to process variations and shifts, tristated I/Os may float toward High or Low, depending on the particular device and leakage level.

If there is concern regarding the exact state of unused I/Os, weak pull-up/pull-down should be added to the floating I/Os so their state is controlled and stabilized.

## Related Documents

### User's Guides

*Designer User's Guide*

[http://www.microsemi.com/soc/documents/designer\\_ug.pdf](http://www.microsemi.com/soc/documents/designer_ug.pdf)

## List of Changes

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

Date	Changes	Page
August 2012	The "I/O State of Newly Shipped Devices" section is new (SAR 39542).	14
July 2010	This chapter is no longer published separately with its own part number and version but is now part of several FPGA fabric user's guides.	N/A
v1.4 (December 2008)	IGLOO nano and ProASIC3 nano devices were added to Table 1-1 • Flash-Based FPGAs.	10
	Figure 1-2 • IGLOO and ProASIC3 nano Device Architecture Overview with Two I/O Banks (applies to 10 k and 30 k device densities, excluding IGLOO PLUS devices) through Figure 1-5 • IGLOO, IGLOO nano, ProASIC3 nano, and ProASIC3/L Device Architecture Overview with Four I/O Banks (AGL600 device is shown) are new.	11, 12
	Table 1-4 • IGLOO nano and ProASIC3 nano Array Coordinates is new.	17
v1.3 (October 2008)	The title of this document was changed from "Core Architecture of IGLOO and ProASIC3 Devices" to "FPGA Array Architecture in Low Power Flash Devices."	9
	The "FPGA Array Architecture Support" section was revised to include new families and make the information more concise.	10
	Table 1-2 • IGLOO and ProASIC3 Array Coordinates was updated to include Military ProASIC3/EL and RT ProASIC3 devices.	16
v1.2 (June 2008)	The following changes were made to the family descriptions in Table 1-1 • Flash-Based FPGAs: <ul style="list-style-type: none"> <li>ProASIC3L was updated to include 1.5 V.</li> <li>The number of PLLs for ProASIC3E was changed from five to six.</li> </ul>	10
v1.1 (March 2008)	Table 1-1 • Flash-Based FPGAs and the accompanying text was updated to include the IGLOO PLUS family. The "IGLOO Terminology" section and "Device Overview" section are new.	10
	The "Device Overview" section was updated to note that 15 k devices do not support SRAM or FIFO.	11
	Figure 1-6 • IGLOO PLUS Device Architecture Overview with Four I/O Banks is new.	13
	Table 1-2 • IGLOO and ProASIC3 Array Coordinates was updated to add A3P015 and AGL015.	16
	Table 1-3 • IGLOO PLUS Array Coordinates is new.	16



YB and YC are identical to GLB and GLC, respectively, with the exception of a higher selectable final output delay. The SmartGen PLL Wizard will configure these outputs according to user specifications and can enable these signals with or without the enabling of Global Output Clocks.

The above signals can be enabled in the following output groupings in both internal and external feedback configurations of the static PLL:

- One output – GLA only
- Two outputs – GLA + (GLB and/or YB)
- Three outputs – GLA + (GLB and/or YB) + (GLC and/or YC)

## PLL Macro Block Diagram

As illustrated, the PLL supports three distinct output frequencies from a given input clock. Two of these (GLB and GLC) can be routed to the B and C global network access, respectively, and/or routed to the device core (YB and YC).

There are five delay elements to support phase control on all five outputs (GLA, GLB, GLC, YB, and YC).

There are delay elements in the feedback loop that can be used to advance the clock relative to the reference clock.

The PLL macro reference clock can be driven in the following ways:

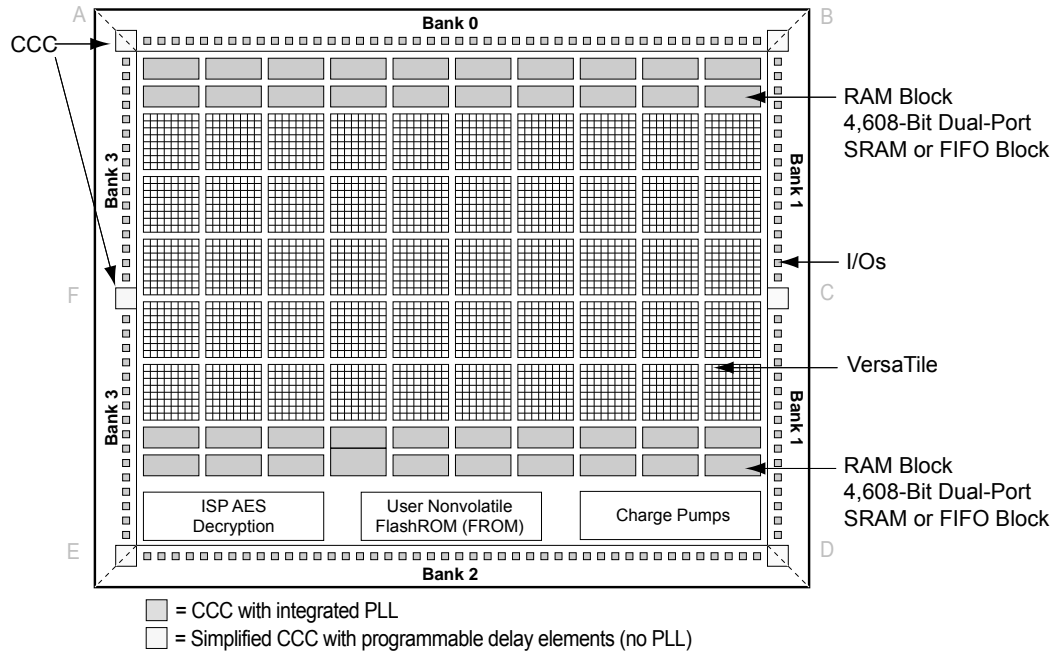
1. By an INBUF\* macro to create a composite macro, where the I/O macro drives the global buffer (with programmable delay) using a hardwired connection. In this case, the I/O must be placed in one of the dedicated global I/O locations.
2. Directly from the FPGA core.
3. From an I/O that is routed through the FPGA regular routing fabric. In this case, users must instantiate a special macro, PLLINT, to differentiate from the hardwired I/O connection described earlier.

During power-up, the PLL outputs will toggle around the maximum frequency of the voltage-controlled oscillator (VCO) gear selected. Toggle frequencies can range from 40 MHz to 250 MHz. This will continue as long as the clock input (CLKA) is constant (HIGH or LOW). This can be prevented by LOW assertion of the POWERDOWN signal.

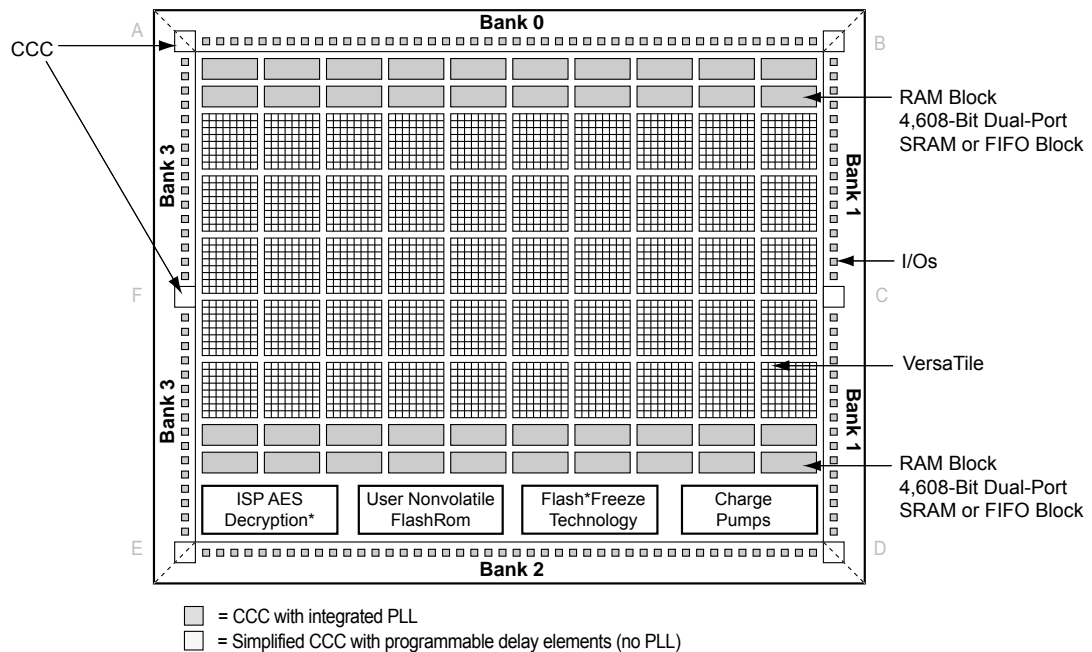
The visual PLL configuration in SmartGen, a component of the Libero SoC and Designer tools, will derive the necessary internal divider ratios based on the input frequency and desired output frequencies selected by the user.

## Fusion CCC Locations

Fusion devices have six CCCs: one in each of the four corners and one each in the middle of the east and west sides of the device (Figure 4-17 and Figure 4-18). The device can have one integrated PLL in the middle of the west side of the device or two integrated PLLs in the middle of the east and west sides of the device (middle right and middle left).



**Figure 4-17 • CCC Locations in Fusion Family Devices (AFS090, AFS250, M1AFS250)**



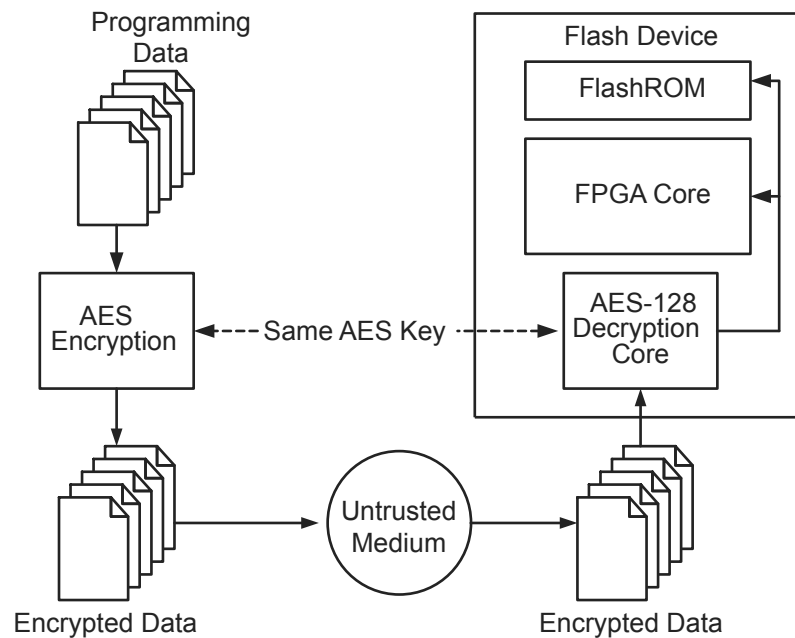
**Figure 4-18 • CCC Locations in Fusion Family Devices (except AFS090, AFS250, M1AFS250)**

## FlashROM Security

Low power flash devices have an on-chip Advanced Encryption Standard (AES) decryption core, combined with an enhanced version of the Microsemi flash-based lock technology (FlashLock®). Together, they provide unmatched levels of security in a programmable logic device. This security applies to both the FPGA core and FlashROM content. These devices use the 128-bit AES (Rijndael) algorithm to encrypt programming files for secure transmission to the on-chip AES decryption core. The same algorithm is then used to decrypt the programming file. This key size provides approximately  $3.4 \times 10^{38}$  possible 128-bit keys. A computing system that could find a DES key in a second would take approximately 149 trillion years to crack a 128-bit AES key. The 128-bit FlashLock feature in low power flash devices works via a FlashLock security Pass Key mechanism, where the user locks or unlocks the device with a user-defined key. Refer to the "Security in Low Power Flash Devices" section on page 235.

If the device is locked with certain security settings, functions such as device read, write, and erase are disabled. This unique feature helps to protect against invasive and noninvasive attacks. Without the correct Pass Key, access to the FPGA is denied. To gain access to the FPGA, the device first must be unlocked using the correct Pass Key. During programming of the FlashROM or the FPGA core, you can generate the security header programming file, which is used to program the AES key and/or FlashLock Pass Key. The security header programming file can also be generated independently of the FlashROM and FPGA core content. The FlashLock Pass Key is not stored in the FlashROM.

Low power flash devices with AES-based security allow for secure remote field updates over public networks such as the Internet, and ensure that valuable intellectual property (IP) remains out of the hands of IP thieves. Figure 5-5 shows this flow diagram.



**Figure 5-5 • Programming FlashROM Using AES**

## Simulation of FlashROM Design

The MEM file has 128 rows of 8 bits, each representing the contents of the FlashROM used for simulation. For example, the first row represents page 0, byte 0; the next row is page 0, byte 1; and so the pattern continues. Note that the three MSBs of the address define the page number, and the four LSBs define the byte number. So, if you send address 0000100 to FlashROM, this corresponds to the page 0 and byte 4 location, which is the fifth row in the MEM file. SmartGen defaults to 0s for any unspecified location of the FlashROM. Besides using the MEM file generated by SmartGen, you can create a binary file with 128 rows of 8 bits each and use this as a MEM file. Microsemi recommends that you use different file names if you plan to generate multiple MEM files. During simulation, Libero SoC passes the MEM file used as the generic file in the netlist, along with the design files and testbench. If you want to use different MEM files during simulation, you need to modify the generic file reference in the netlist.

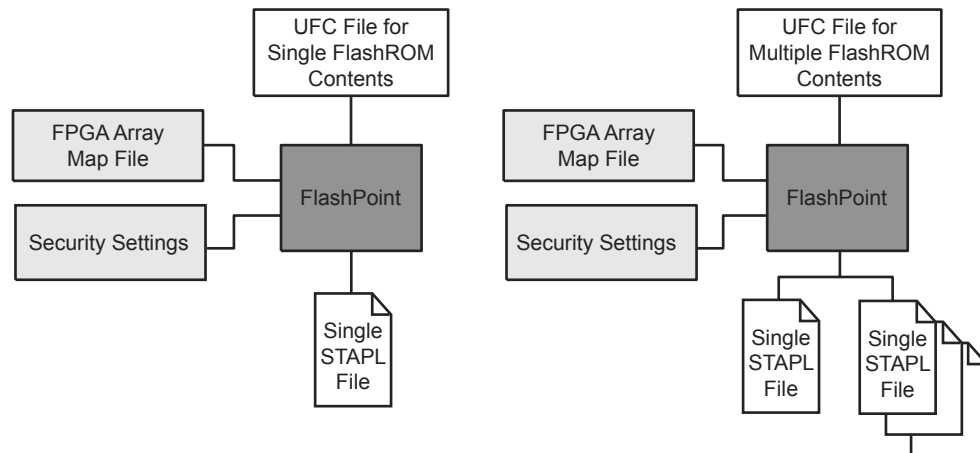
```
.....
UFROM0: UFROM
--generic map(MEMORYFILE => "F:\Appsnotes\FROM\test_designs\testa\smartgen\FROM_a.mem")
--generic map(MEMORYFILE => "F:\Appsnotes\FROM\test_designs\testa\smartgen\FROM_b.mem")
.....
```

The VITAL and Verilog simulation models accept the generics passed by the netlist, read the MEM file, and perform simulation with the data in the file.

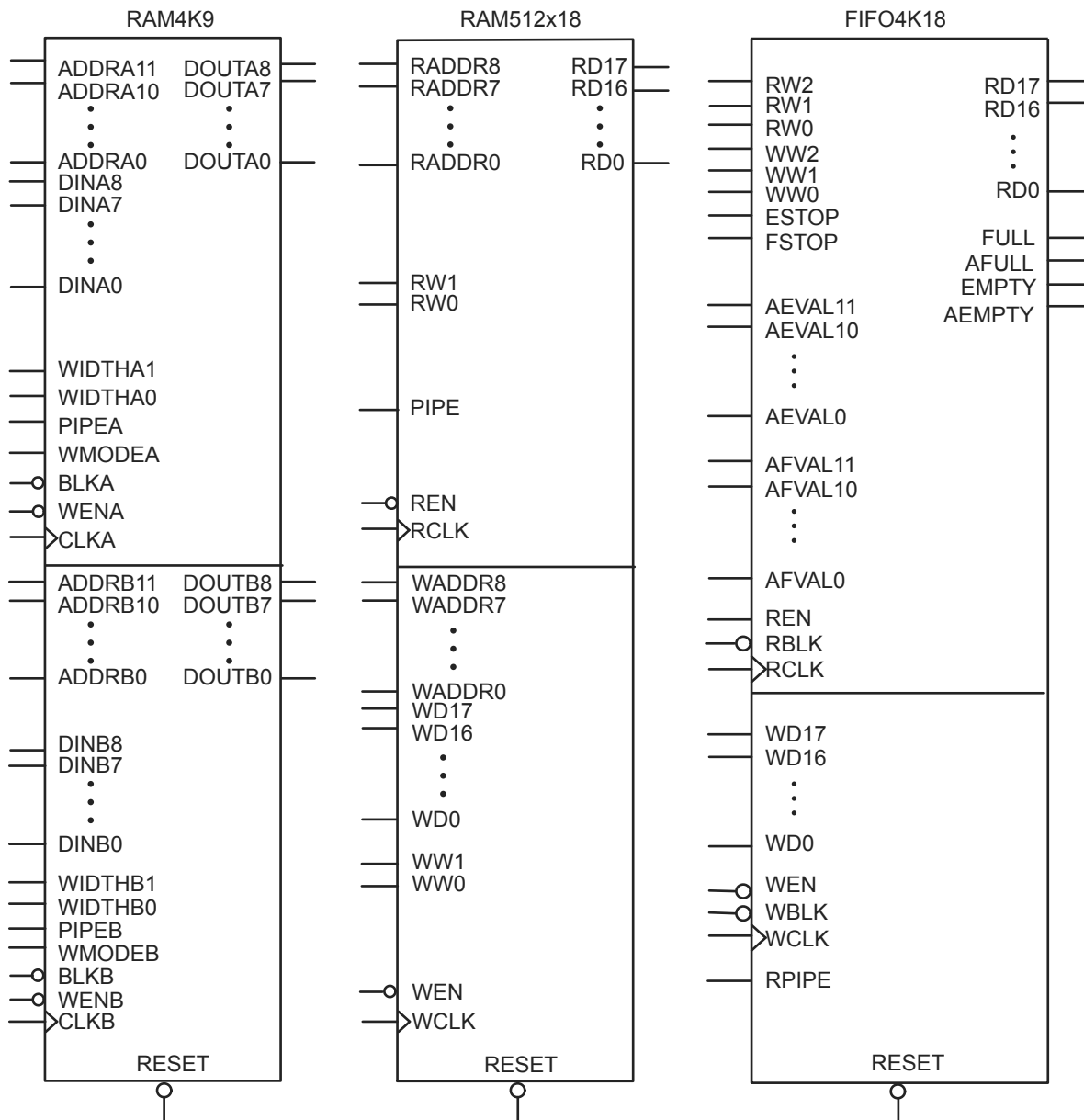
## Programming File Generation for FlashROM Design

FlashPoint is the programming software used to generate the programming files for flash devices. Depending on the applications, you can use the FlashPoint software to generate a STAPL file with different FlashROM contents. In each case, optional AES decryption is available. To generate a STAPL file that contains the same FPGA core content and different FlashROM contents, the FlashPoint software needs an Array Map file for the core and UFC file(s) for the FlashROM. This final STAPL file represents the combination of the logic of the FPGA core and FlashROM content.

FlashPoint generates the STAPL files you can use to program the desired FlashROM page and/or FPGA core of the FPGA device contents. FlashPoint supports the encryption of the FlashROM content and/or FPGA Array configuration data. In the case of using the FlashROM for device serialization, a sequence of unique FlashROM contents will be generated. When generating a programming file with multiple unique FlashROM contents, you can specify in FlashPoint whether to include all FlashROM content in a single STAPL file or generate a different STAPL file for each FlashROM (Figure 5-11). The programming software (FlashPro) handles the single STAPL file that contains the FlashROM content from multiple devices. It enables you to program the FlashROM content into a series of devices sequentially (Figure 5-11). See the *FlashPro User's Guide* for information on serial programming.



**Figure 5-11 • Single or Multiple Programming File Generation**



Notes:

1. Automotive ProASIC3 devices restrict RAM4K9 to a single port or to dual ports with the same clock 180° out of phase (inverted) between clock pins. In single-port mode, inputs to port B should be tied to ground to prevent errors during compile. This warning applies only to automotive ProASIC3 parts of certain revisions and earlier. Contact Technical Support at [soc\\_tech@microsemi.com](mailto:soc_tech@microsemi.com) for information on the revision number for a particular lot and date code.
2. For FIFO4K18, the same clock 180° out of phase (inverted) between clock pins should be used.

**Figure 6-3 • Supported Basic RAM Macros**

Table 6-8 and Table 6-9 show the maximum potential width and depth configuration for each device. Note that 15 k and 30 k gate devices do not support RAM or FIFO.

**Table 6-8 • Memory Availability per IGLOO and ProASIC3 Device**

Device		RAM Blocks	Maximum Potential Width <sup>1</sup>		Maximum Potential Depth <sup>2</sup>	
IGLOO IGLOO nano IGLOO PLUS	ProASIC3 ProASIC3 nano ProASIC3L		Depth	Width	Depth	Width
AGL060 AGLN060 AGLP060	A3P060 A3PN060	4	256	72 (4×18)	16,384 (4,096×4)	1
AGL125 AGLN125 AGLP125	A3P125 A3PN125	8	256	144 (8×18)	32,768 (4,096×8)	1
AGL250 AGLN250	A3P250/L A3PN250	8	256	144 (8×18)	32,768 (4,096×8)	1
AGL400	A3P400	12	256	216 (12×18)	49,152 (4,096×12)	1
AGL600	A3P600/L	24	256	432 (24×18)	98,304 (4,096×24)	1
AGL1000	A3P1000/L	32	256	576 (32×18)	131,072 (4,096×32)	1
AGLE600	A3PE600	24	256	432 (24×18)	98,304 (4,096×24)	1
	A3PE1500	60	256	1,080 (60×18)	245,760 (4,096×60)	1
AGLE3000	A3PE3000/L	112	256	2,016 (112×18)	458,752 (4,096×112)	1

Notes:

1. Maximum potential width uses the two-port configuration.
2. Maximum potential depth uses the dual-port configuration.

**Table 6-9 • Memory Availability per Fusion Device**

Device	RAM Blocks	Maximum Potential Width <sup>1</sup>		Maximum Potential Depth <sup>2</sup>	
		Depth	Width	Depth	Width
AFS090	6	256	108 (6×18)	24,576 (4,096×6)	1
AFS250	8	256	144 (8×18)	32,768 (4,096×8)	1
AFS600	24	256	432 (24×18)	98,304 (4,096×24)	1
AFS1500	60	256	1,080 (60×18)	245,760 (4,096×60)	1

Notes:

1. Maximum potential width uses the two-port configuration.
2. Maximum potential depth uses the dual-port configuration.

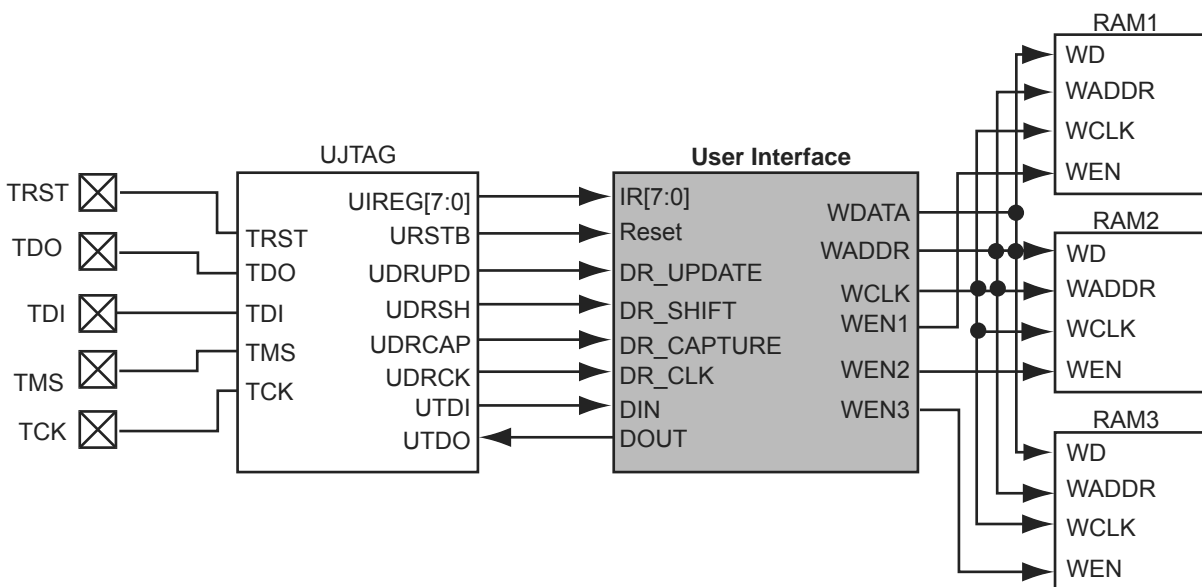
## Initializing the RAM/FIFO

The SRAM blocks can be initialized with data to use as a lookup table (LUT). Data initialization can be accomplished either by loading the data through the design logic or through the UJTAG interface. The UJTAG macro is used to allow access from the JTAG port to the internal logic in the device. By sending the appropriate initialization string to the JTAG Test Access Port (TAP) Controller, the designer can put the JTAG circuitry into a mode that allows the user to shift data into the array logic through the JTAG port using the UJTAG macro. For a more detailed explanation of the UJTAG macro, refer to the "FlashROM in Microsemi's Low Power Flash Devices" section on page 117.

A user interface is required to receive the user command, initialization data, and clock from the UJTAG macro. The interface must synchronize and load the data into the correct RAM block of the design. The main outputs of the user interface block are the following:

- Memory block chip select: Selects a memory block for initialization. The chip selects signals for each memory block that can be generated from different user-defined pockets or simple logic, such as a ring counter (see below).
- Memory block write address: Identifies the address of the memory cell that needs to be initialized.
- Memory block write data: The interface block receives the data serially from the UTDI port of the UJTAG macro and loads it in parallel into the write data ports of the memory blocks.
- Memory block write clock: Drives the WCLK of the memory block and synchronizes the write data, write address, and chip select signals.

Figure 6-8 shows the user interface between UJTAG and the memory blocks.



**Figure 6-8 • Interfacing TAP Ports and SRAM Blocks**

An important component of the interface between the UJTAG macro and the RAM blocks is a serial-in/parallel-out shift register. The width of the shift register should equal the data width of the RAM blocks. The RAM data arrives serially from the UTDI output of the UJTAG macro. The data must be shifted into a shift register clocked by the JTAG clock (provided at the UDRCK output of the UJTAG macro).

Then, after the shift register is fully loaded, the data must be transferred to the write data port of the RAM block. To synchronize the loading of the write data with the write address and write clock, the output of the shift register can be pipelined before driving the RAM block.

The write address can be generated in different ways. It can be imported through the TAP using a different instruction opcode and another shift register, or generated internally using a simple counter. Using a counter to generate the address bits and sweep through the address range of the RAM blocks is

## nano Standard I/Os

Table 7-2 and Table 7-3 show the voltages and compatible I/O standards for ProASIC3 nano and IGLOO nano devices.

I/Os provide programmable slew rates, drive strengths, and weak pull-up and pull-down circuits. Selectable Schmitt trigger and 5 V tolerant receivers are offered. See the "5 V Input Tolerance" section on page 171 for possible implementations of 5 V tolerance.

All I/Os are in a known state during power-up, and any power-up sequence is allowed without current impact. Refer to the "I/O Power-Up and Supply Voltage Thresholds for Power-On Reset (Commercial and Industrial)" section in the datasheet for more information. During power-up, before reaching activation levels, the I/O input and output buffers are disabled while the weak pull-up is enabled. Activation levels are described in the datasheet.

**Table 7-2 • Supported I/O Standards for IGLOO nano Devices**

IGLOO nano	AGLN010	AGLN015	AGLN020	AGLN060	AGLN125	AGLN250
<b>Single-Ended</b>						
LVTTL/LVCMOS 3.3 V, LVCMOS 2.5 V / 1.8 V / 1.5 V / 1.2 V	✓	✓	✓	✓	✓	✓

**Table 7-3 • Supported I/O Standards for ProASIC3 Devices**

ProASIC3 nano	A3PN010	A3PN015	A3PN020	A3PN060	A3PN125	A3PN250
<b>Single-Ended</b>						
LVTTL/LVCMOS 3.3 V, LVCMOS 2.5 V / 1.8 V / 1.5 V	✓	✓	✓	✓	✓	✓

## I/O Banks and I/O Standards Compatibility

I/Os are grouped into I/O voltage banks. nano devices have two, three, or four I/O banks.

Each I/O voltage bank has dedicated I/O supply and ground voltages. This isolation is necessary to minimize simultaneous switching noise from the input and output (SSI and SSO). The switching noise (ground bounce and power bounce) is generated by the output buffers and transferred into input buffer circuits, and vice versa. Because of these dedicated supplies, only I/Os with compatible standards can be assigned to the same I/O voltage bank. Table 7-4 shows the required voltage compatibility values for each of these voltages.

There are four I/O banks on the 250 k gate device.

There are three I/O banks on the 15 k and 20 k gate devices.

There are two I/O banks on the 10 k, 60 k, and 125 k gate devices.

I/O standards are compatible if their VCCI values are identical. For more information about I/O and global assignments to I/O banks in a device, refer to the specific pin table for the device in the packaging section of the datasheet and the "User I/O Naming Convention" section on page 178.

**Table 7-4 • VCCI Voltages and Compatible Standards**

VCCI (typical)	Compatible Standards
3.3 V	LVTTL/LVCMOS 3.3
2.5 V	LVCMOS 2.5
1.8 V	LVCMOS 1.8
1.5 V	LVCMOS 1.5
1.2 V	LVCMOS 1.2



Example 2 (low–medium speed, medium current):

$$R_{tx\_out\_high} = R_{tx\_out\_low} = 10 \, \Omega$$

$$R1 = 220 \, \Omega (\pm 5\%), P(r1)_{min} = 0.018 \, \Omega$$

$$R2 = 390 \, \Omega (\pm 5\%), P(r2)_{min} = 0.032 \, \Omega$$

$$I_{max\_tx} = 5.5 \, V / (220 \times 0.95 + 390 \times 0.95 + 10) = 9.17 \, mA$$

$$t_{RISE} = t_{FALL} = 4 \, ns \text{ at } C_{pad\_load} = 10 \, pF \text{ (includes up to 25\% safety margin)}$$

$$t_{RISE} = t_{FALL} = 20 \, ns \text{ at } C_{pad\_load} = 50 \, pF \text{ (includes up to 25\% safety margin)}$$

Other values of resistors are also allowed as long as the resistors are sized appropriately to limit the voltage at the receiving end to  $2.5 \, V < V_{in} (rx) < 3.6 \, V$  when the transmitter sends a logic 1. This range of  $V_{in\_dc}(rx)$  must be assured for any combination of transmitter supply ( $5 \, V \pm 0.5 \, V$ ), transmitter output resistance, and board resistor tolerances.

Temporary overshoots are allowed according to the overshoot and undershoot table in the datasheet.

### Solution 1

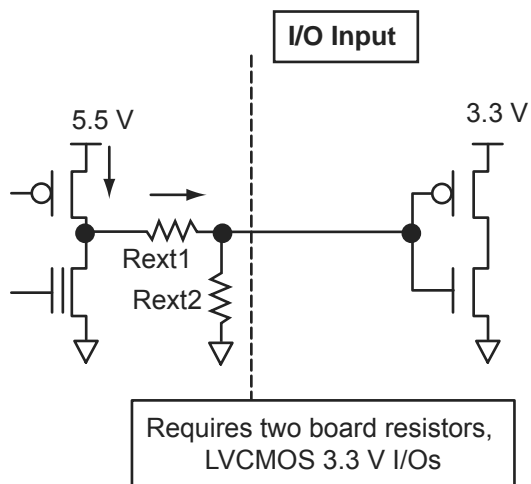


Figure 7-5 • Solution 1

## 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 307 for details.

## Drive Strength

Low power flash devices have up to four 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\_8 (slew = low, out\_drive = 8 mA).

The maximum available drive strength is 8 mA per I/O. Though no I/O should be forced to source or sink more than 8 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 without adding any extra circuitry. Along with other low-voltage I/O macros, this 5 V tolerance makes these devices suitable for many types of board component interfacing.

Table 7-17 shows some high-level interfacing examples using low power flash devices.

**Table 7-17 • nano High-Level Interface**

Interface	Clock		I/O			
	Type	Frequency	Type	Signals In	Signals Out	Data I/O
GM	Src Sync	125 MHz	LVTTL	8	8	125 Mbps
TBI	Src Sync	125 MHz	LVTTL	10	10	125 Mbps

## Conclusion

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

- The I/O standard of technology-specific I/O macros cannot be changed in the I/O Attribute Editor (see Figure 8-6).
- The user **MUST** instantiate differential I/O macros (LVDS/LVPECL) in the design. This is the only way to use these standards in the design (IGLOO nano and ProASIC3 nano devices do not support differential inputs).
- To implement the DDR I/O function, the user must instantiate a DDR\_REG or DDR\_OUT macro. This is the only way to use a DDR macro in the design.

**Figure 8-6 • Assigning a Different I/O Standard to the Generic I/O Macro**

## Performing Place-and-Route on the Design

The netlist created by the synthesis tool should now be imported into Designer and compiled. During Compile, the user can specify the I/O placement and attributes by importing the PDC file. The user can also specify the I/O placement and attributes using ChipPlanner and the I/O Attribute Editor under MVN.

### Defining I/O Assignments in the PDC File

A PDC file is a Tcl script file specifying physical constraints. This file can be imported to and exported from Designer.

Table 8-3 shows I/O assignment constraints supported in the PDC file.

**Table 8-3 • PDC I/O Constraints**

Command	Action	Example	Comment
<b>I/O Banks Setting Constraints</b>			
set_iobank	Sets the I/O supply voltage, $V_{CCI}$ , and the input reference voltage, $V_{REF}$ , for the specified I/O bank.	<pre>set_iobank bankname [-vcci vcci_voltage] [-vref vref_voltage]  set_iobank Bank7 -vcci 1.50 -vref 0.75</pre>	Must use in case of mixed I/O voltage ( $V_{CCI}$ ) design
set_vref	Assigns a $V_{REF}$ pin to a bank.	<pre>set_vref -bank [bankname] [pinnum]  set_vref -bank Bank0 685 704 723 742 761</pre>	Must use if voltage-referenced I/Os are used
set_vref_defaults	Sets the default $V_{REF}$ pins for the specified bank. This command is ignored if the bank does not need a $V_{REF}$ pin.	<pre>set_vref_defaults bankname  set_vref_defaults bank2</pre>	

*Note:* Refer to the Libero SoC User's Guide for detailed rules on PDC naming and syntax conventions.

The procedure is as follows:

1. Select the bank to which you want VCCI to be assigned from the **Choose Bank** list.
2. Select the I/O standards for that bank. If you select any standard, the tool will automatically show all compatible standards that have a common VCCI voltage requirement.
3. Click **Apply**.
4. Repeat steps 1–3 to assign VCCI voltages to other banks. Refer to Figure 8-11 on page 197 to find out how many I/O banks are needed for VCCI bank assignment.

## Manually Assigning VREF Pins

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

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

The VREF rules are as follows:

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

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

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

### Example:

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

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

8. Only minibanks that contain input or bidirectional I/Os require a VREF. A VREF is not needed for minibanks composed of output or tristated I/Os.

## Assigning the VREF Voltage to a Bank

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

```
set_iobank -vref [value]
```

Another method for assigning VREF is by using **MVN > Edit > I/O Bank Settings** (Figure 8-13 on page 200).

# General Flash Programming Information

## Programming Basics

When choosing a programming solution, there are a number of options available. This section provides a brief overview of those options. The next sections provide more detail on those options as they apply to Microsemi FPGAs.

### ***Reprogrammable or One-Time-Programmable (OTP)***

Depending on the technology chosen, devices may be reprogrammable or one-time-programmable. As the name implies, a reprogrammable device can be programmed many times. Generally, the contents of such a device will be completely overwritten when it is reprogrammed. All Microsemi flash devices are reprogrammable.

An OTP device is programmable one time only. Once programmed, no more changes can be made to the contents. Microsemi flash devices provide the option of disabling the reprogrammability for security purposes. This combines the convenience of reprogrammability during design verification with the security of an OTP technology for highly sensitive designs.

### ***Device Programmer or In-System Programming***

There are two fundamental ways to program an FPGA: using a device programmer or, if the technology permits, using in-system programming. A device programmer is a piece of equipment in a lab or on the production floor that is used for programming FPGA devices. The devices are placed into a socket mounted in a programming adapter module, and the appropriate electrical interface is applied. The programmed device can then be placed on the board. A typical programmer, used during development, programs a single device at a time and is referred to as a single-site engineering programmer.

With ISP, the device is already mounted onto the system printed circuit board when programming occurs. Typically, ISP programming is performed via a JTAG interface on the FPGA. The JTAG pins can be controlled either by an on-board resource, such as a microprocessor, or by an off-board programmer through a header connection. Once mounted, it can be programmed repeatedly and erased. If the application requires it, the system can be designed to reprogram itself using a microprocessor, without the use of any external programmer.

If multiple devices need to be programmed with the same program, various multi-site programming hardware is available in order to program many devices in parallel. Microsemi In House Programming is also available for this purpose.

## Programming Features for Microsemi Devices

### ***Flash Devices***

The flash devices supplied by Microsemi are reprogrammable by either a generic device programmer or ISP. Microsemi supports ISP using JTAG, which is supported by the FlashPro4 and FlashPro3, FlashPro Lite, Silicon Sculptor 3, and Silicon Sculptor II programmers.

Levels of ISP support vary depending on the device chosen:

- All SmartFusion, Fusion, IGLOO, and ProASIC3 devices support ISP.
- IGLOO, IGLOOe, IGLOO nano V5, and IGLOO PLUS devices can be programmed in-system when the device is using a 1.5 V supply voltage to the FPGA core.
- IGLOO nano V2 devices can be programmed at 1.2 V core voltage (when using FlashPro4 only) or 1.5 V. IGLOO nano V5 devices are programmed with a VCC core voltage of 1.5 V.

## Programmer Ordering Codes

The products shown in Table 10-4 can be ordered through Microsemi sales and will be shipped directly from Microsemi. Products can also be ordered from Microsemi distributors, but will still be shipped directly from Microsemi. Table 10-4 includes ordering codes for the full kit, as well as codes for replacement items and any related hardware. Some additional products can be purchased from external suppliers for use with the programmers. Ordering codes for adapter modules used with Silicon Sculptor are available at [http://www.microsemi.com/soc/products/hardware/program\\_debug/ss/modules.aspx](http://www.microsemi.com/soc/products/hardware/program_debug/ss/modules.aspx).

**Table 10-4 • Programming Ordering Codes**

Description	Vendor	Ordering Code	Comment
FlashPro4 ISP programmer	Microsemi	FLASHPRO 4	Uses a 2×5, RA male header connector
FlashPro Lite ISP programmer	Microsemi	FLASHPRO LITE	Supports small programming header or large header through header converter (not included)
Silicon Sculptor 3	Microsemi	SILICON-SCULPTOR 3	USB 2.0 high-speed production programmer
Silicon Sculptor II	Microsemi	SILICON-SCULPTOR II	Requires add-on adapter modules to support devices
Silicon Sculptor ISP module	Microsemi	SMPA-ISP-ACTEL-3-KIT	Ships with both large and small header support
ISP cable for small header	Microsemi	ISP-CABLE-S	Supplied with SMPA-ISP-ACTEL-3-KIT
ISP cable for large header	Microsemi	PA-ISP-CABLE	Supplied with SMPA-ISP-ACTEL-3-KIT

## Programmer Device Support

Refer to [www.microsemi.com/soc](http://www.microsemi.com/soc) for the current information on programmer and device support.

## Certified Programming Solutions

The Microsemi-certified programmers for flash devices are FlashPro4, FlashPro3, FlashPro Lite, FlashPro, Silicon Sculptor II, Silicon Sculptor 3, and any programmer that is built by BP Microsystems. All other programmers are considered noncertified programmers.

- FlashPro4, FlashPro3, FlashPro Lite, FlashPro  
The Microsemi family of FlashPro device programmers provides in-system programming in an easy-to-use, compact system that supports all flash families. Whether programming a board containing a single device or multiple devices connected in a chain, the Microsemi line of FlashPro programmers enables fast programming and reprogramming. Programming with the FlashPro series of programmers saves board space and money as it eliminates the need for sockets on the board. There are no built-in algorithms, so there is no delay between product release and programming support. The FlashPro programmer is no longer available.
- Silicon Sculptor 3, Silicon Sculptor II  
Silicon Sculptor 3 and Silicon Sculptor II are robust, compact, single-device programmers with standalone software for the PC. They are designed to enable concurrent programming of multiple units from the same PC with speeds equivalent to or faster than previous Microsemi programmers.
- Noncertified Programmers  
Microsemi does not test programming solutions from other vendors, and DOES NOT guarantee programming yield. Also, Microsemi will not perform any failure analysis on devices programmed on non-certified programmers. Please refer to the *Programming and Functional Failure Guidelines* document for more information.

## 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](http://www.microsemi.com/soc/documents/AntifuseProgram_AN.pdf)

*Implementation of Security in Actel's ProASIC and ProASIC<sup>PLUS</sup> Flash-Based FPGAs*

[http://www.microsemi.com/soc/documents/Flash\\_Security\\_AN.pdf](http://www.microsemi.com/soc/documents/Flash_Security_AN.pdf)

### User's Guides

#### **FlashPro Programmers**

FlashPro4,<sup>1</sup> FlashPro3, FlashPro Lite, and FlashPro<sup>2</sup>

[http://www.microsemi.com/soc/products/hardware/program\\_debug/flashpro/default.aspx](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](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](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](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.

## Security Support in Flash-Based Devices

The flash FPGAs listed in Table 11-1 support the security feature 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
	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 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
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

*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*.

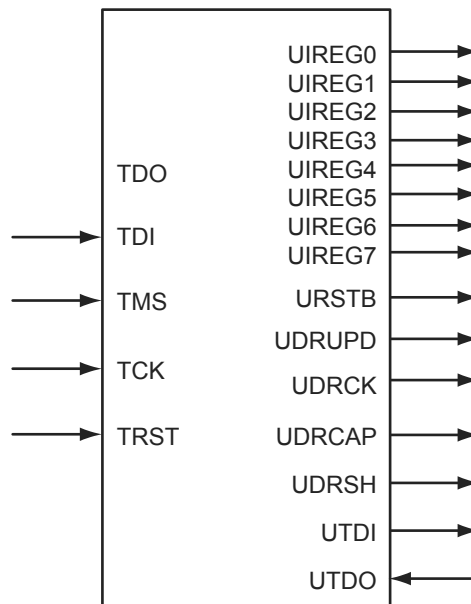


## UJTAG Macro

The UJTAG tiles can be instantiated in a design using the UJTAG macro from the Fusion, IGLOO, or ProASIC3 macro library. Note that "UJTAG" is a reserved name and cannot be used for any other user-defined blocks. A block symbol of the UJTAG tile macro is presented in Figure 16-2. In this figure, the ports on the left side of the block are connected to the JTAG TAP Controller, and the right-side ports are accessible by the FPGA core VersaTiles. The TDI, TMS, TDO, TCK, and TRST ports of UJTAG are only provided for design simulation purposes and should be treated as external signals in the design netlist. However, these ports must NOT be connected to any I/O buffer in the netlist. Figure 16-3 on page 300 illustrates the correct connection of the UJTAG macro to the user design netlist. Microsemi Designer software will automatically connect these ports to the TAP during place-and-route. Table 16-2 gives the port descriptions for the rest of the UJTAG ports:

**Table 16-2 • UJTAG Port Descriptions**

Port	Description
UIREG [7:0]	This 8-bit bus carries the contents of the JTAG Instruction Register of each device. Instruction Register values 16 to 127 are not reserved and can be employed as user-defined instructions.
URSTB	URSTB is an active-low signal and will be asserted when the TAP Controller is in Test-Logic-Reset mode. URSTB is asserted at power-up, and a power-on reset signal resets the TAP Controller. URSTB will stay asserted until an external TAP access changes the TAP Controller state.
UTDI	This port is directly connected to the TAP's TDI signal.
UTDO	This port is the user TDO output. Inputs to the UTDO port are sent to the TAP TDO output MUX when the IR address is in user range.
UDRSH	Active-high signal enabled in the ShiftDR TAP state
UDRCAP	Active-high signal enabled in the CaptureDR TAP state
UDRCK	This port is directly connected to the TAP's TCK signal.
UDRUPD	Active-high signal enabled in the UpdateDR TAP state



**Figure 16-2 • UJTAG Tile Block Symbol**