



Welcome to [E-XFL.COM](https://www.e-xfl.com)

Understanding [Embedded - FPGAs \(Field Programmable Gate Array\)](#)

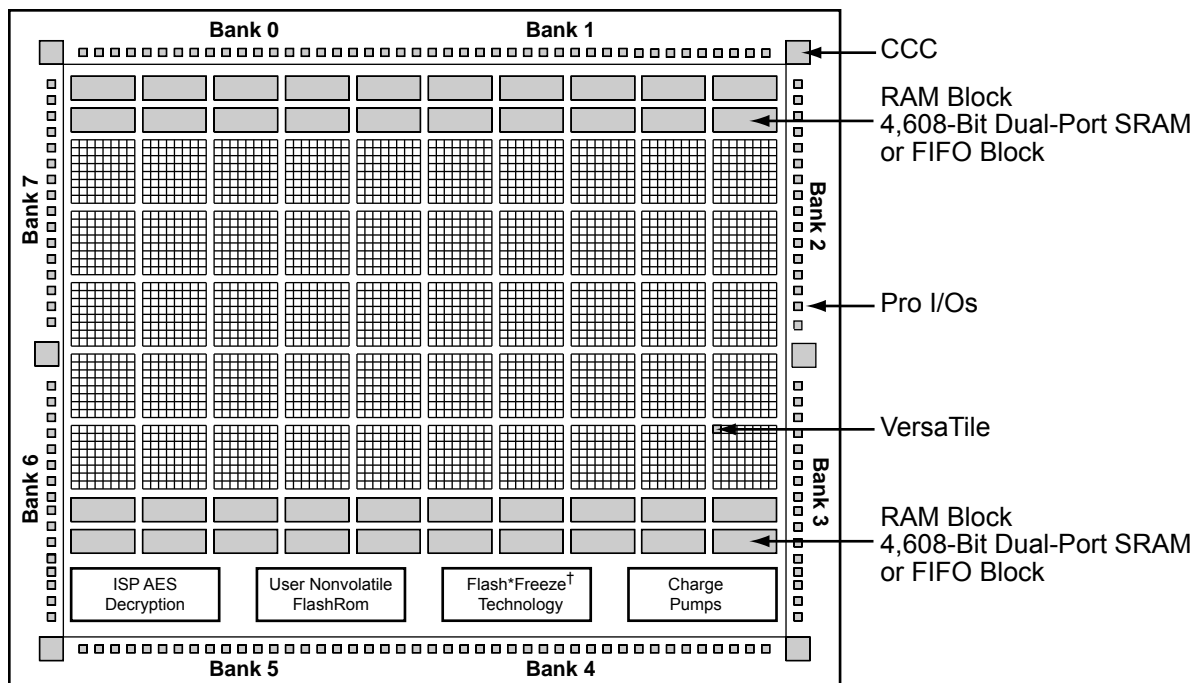
Embedded - FPGAs, or Field Programmable Gate Arrays, are advanced integrated circuits that offer unparalleled flexibility and performance for digital systems. Unlike traditional fixed-function logic devices, FPGAs can be programmed and reprogrammed to execute a wide array of logical operations, enabling 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 | 110592 |
| Number of I/O | 177 |
| Number of Gates | 600000 |
| Voltage - Supply | 1.14V ~ 1.575V |
| Mounting Type | Surface Mount |
| Operating Temperature | 0°C ~ 85°C (TJ) |
| Package / Case | 256-LBGA |
| Supplier Device Package | 256-FPBGA (17x17) |
| Purchase URL | https://www.e-xfl.com/product-detail/microsemi/a3p600l-1fgg256 |



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.

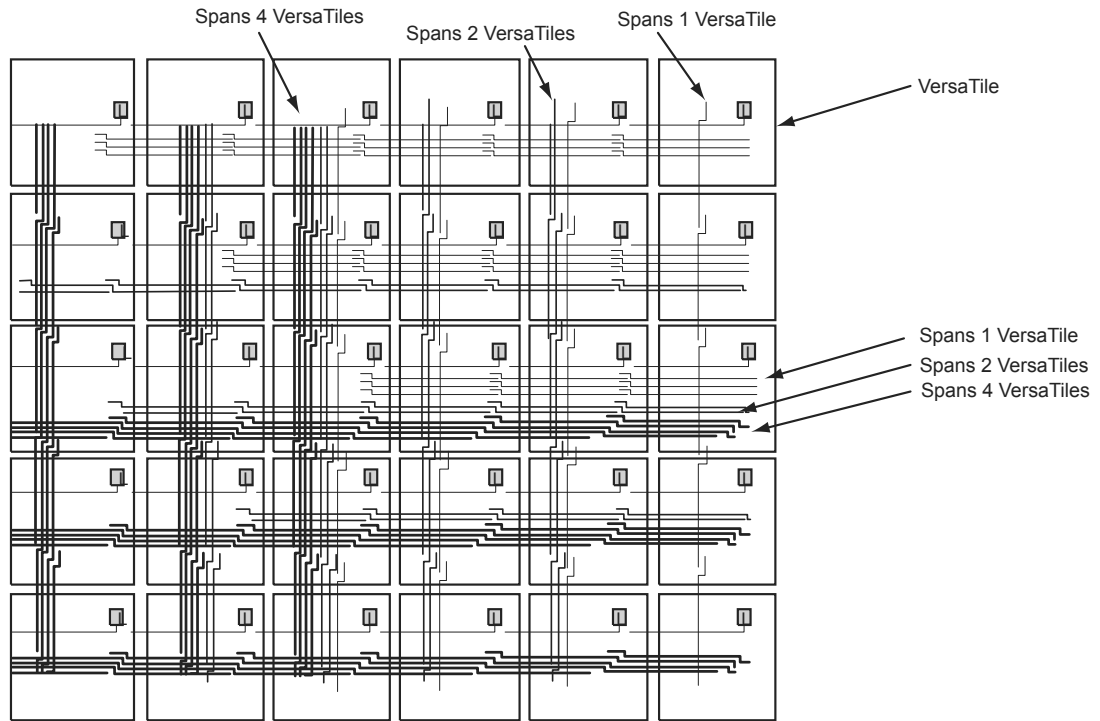


Figure 1-11 • Efficient Long-Line Resources

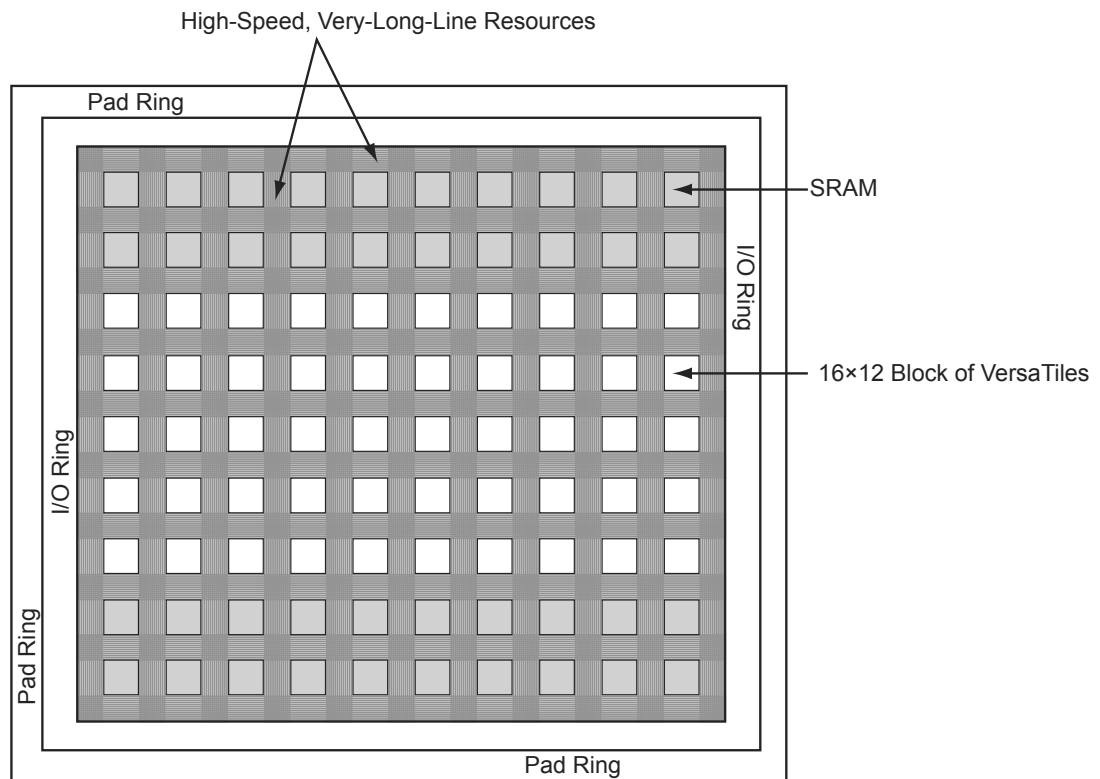


Figure 1-12 • Very-Long-Line Resources

During Flash*Freeze Mode

- PLLs are turned off during Flash*Freeze mode.
- I/O pads are configured according to Table 2-5 on page 28 and Table 2-6 on page 29.
- Inputs and input clocks to the FPGA can toggle without any impact on static power consumption, assuming weak pull-up or pull-down is not selected.
- If weak pull-up or pull-down is selected and the input is driven to the opposite direction, power dissipation will occur.
- Any toggling signals will be charging and discharging the package pin capacitance.
- IGLOO and ProASIC3L outputs will be tristated unless the I/O is configured with weak pull-up or pull-down. The output of the I/O to the FPGA core is logic High regardless of whether the I/O pin is configured with a weak pull-up or pull-down. Refer to Table 2-5 on page 28 for more information.
- IGLOO nano and IGLOO PLUS output behavior will be based on the configuration defined by the user. Refer to Table 2-6 on page 29 for a description of output behavior during Flash*Freeze mode.
- The JTAG circuit is active; however, JTAG operations, such as JTAG commands, JTAG bypass, programming, and authentication, cannot be executed. The device must exit Flash*Freeze mode before JTAG commands can be sent. TCK should be static to avoid extra power consumption from the JTAG state machine.
- The FF pin must be externally asserted for the device to stay in Flash*Freeze mode.
- The FF pin is still active; i.e., the pin is used to exit Flash*Freeze mode when deasserted.

Exiting Flash*Freeze Mode

I/Os and Globals

- While exiting Flash*Freeze mode, inputs and globals will exit their Flash*Freeze state asynchronously to each other. As a result, clock and data glitches and narrow pulses may be generated while exiting Flash*Freeze mode, unless clock gating schemes are used.
- I/O banks are not all activated simultaneously when exiting Flash*Freeze mode. This can cause clocks and inputs to become enabled at different times, resulting in unexpected data being captured.
- Upon exiting Flash*Freeze mode, inputs and globals will no longer be tied High internally (does not apply to input hold state on IGLOO nano and IGLOO PLUS). If any of these signals are driven Low or tied Low externally, they will experience a High-to-Low transition internally when exiting Flash*Freeze mode.
- Applies only to IGLOO nano and IGLOO PLUS: Output hold state is asynchronously controlled by the signal driving the output buffer (output signal). This ensures a clean, glitch-free transition from hold state to output drive. However, any glitches on the output signal during exit from Flash*Freeze mode may result in glitches on the output pad.
- The above situations can cause glitches or invalid data to be clocked into and preserved in the device. Refer to the "Flash*Freeze Design Guide" on page 34 for solutions.

PLLs

- If the embedded PLL is used, the design must allow maximum acquisition time (per device datasheet) for the PLL to acquire the lock signal.

Flash*Freeze Pin Locations

Refer to the Pin Descriptions and Packaging chapter of specific device datasheets for information regarding Flash*Freeze pin location on the available packages. The Flash*Freeze pin location is independent of the device, allowing migration to larger or smaller devices while maintaining the same pin location on the board.

Implementing EXTFB in ProASIC3/E Devices

When the external feedback (EXTFB) signal of the PLL in the ProASIC3/E devices is implemented, the phase detector of the PLL core receives the reference clock (CLKA) and EXTFB as inputs. EXTFB must be sourced as an INBUF macro and located at the global/chip clock location associated with the target PLL by Designer software. EXTFB cannot be sourced from the FPGA fabric.

The following example shows CLKA and EXTFB signals assigned to two global I/Os in the same global area of ProASIC3E device.

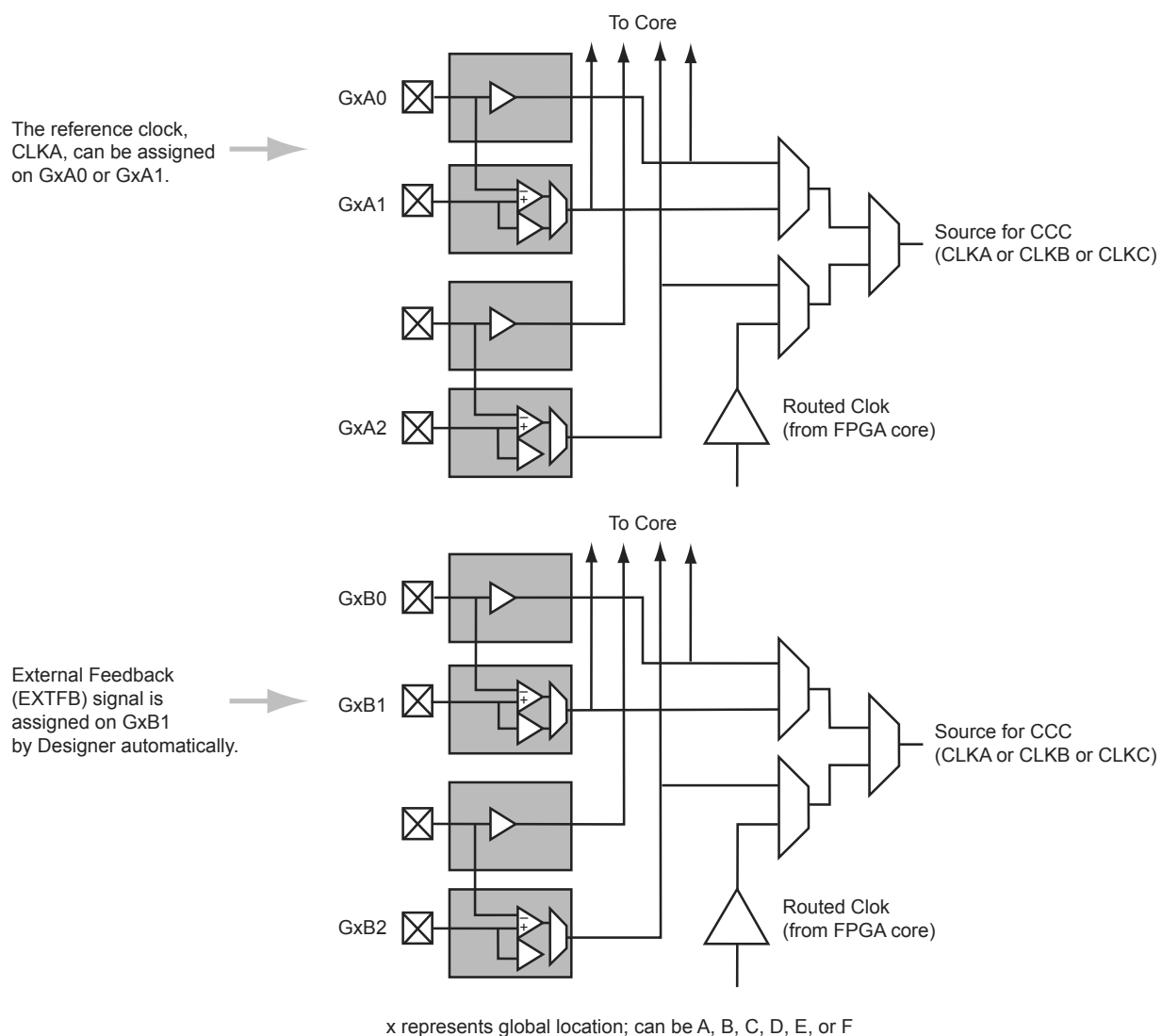


Figure 4-5 • CLKA and EXTFB Assigned to Global I/Os

```
DYNCCC Core(.CLKA(CLKA), .EXTFB(GND), .POWERDOWN(POWERDOWN), .GLA(GLA), .LOCK(LOCK),
.CLKB(CLKB), .GLB(GLB), .YB(), .CLKC(CLKC), .GLC(GLC), .YC(), .SDIN(SDIN),
.SCLK(SCLK), .SShift(SShift), .SUPDATE(SUPDATE), .MODE(MODE), .SDOUT(SDOUT),
.OADIV0(GND), .OADIV1(GND), .OADIV2(VCC), .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(GND), .FBDIV1(GND), .FBDIV2(GND),
.FBDIV3(GND), .FBDIV4(GND), .FBDIV5(VCC), .FBDIV6(GND), .FBDLY0(GND), .FBDLY1(GND),
.FBDLY2(GND), .FBDLY3(GND), .FBDLY4(GND), .FBSEL0(VCC), .FBSEL1(GND),
.XDLYSEL(GND), .VCOSEL0(GND), .VCOSEL1(GND), .VCOSEL2(VCC));
defparam Core.VCOFREQUENCY = 165.000;

endmodule
```

Delayed Clock Configuration

The CLKDLY macro can be generated with the desired delay and input clock source (Hardwired I/O, External I/O, or Core Logic), as in Figure 4-28.

Figure 4-28 • Delayed Clock Configuration Dialog Box

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 | : delay_macro |
| Family | : ProASIC3 |
| Output Format | : Verilog |
| Type | : Delayed Clock |
| Delay Index | : 2 |
| CLKA Source | : Hardwired I/O |

Total Clock Delay = 0.935 ns.

The resultant CLKDLY macro Verilog netlist is as follows:

```
module delay_macro(GL,CLK);

output GL;
input CLK;
```

Table 6-2 • Allowable Aspect Ratio Settings for WIDTHA[1:0]

| WIDTHA[1:0] | WIDTHB[1:0] | D×W |
|-------------|-------------|-------|
| 00 | 00 | 4k×1 |
| 01 | 01 | 2k×2 |
| 10 | 10 | 1k×4 |
| 11 | 11 | 512×9 |

Note: The aspect ratio settings are constant and cannot be changed on the fly.

BLKA and BLKB

These signals are active-low and will enable the respective ports when asserted. When a BLKx signal is deasserted, that port's outputs hold the previous value.

Note: When using the SRAM in single-port mode for Automotive ProASIC3 devices, BLKB should be tied to ground.

WENA and WENB

These signals switch the RAM between read and write modes for the respective ports. A LOW on these signals indicates a write operation, and a HIGH indicates a read.

Note: When using the SRAM in single-port mode for Automotive ProASIC3 devices, WENB should be tied to ground.

CLKA and CLKB

These are the clock signals for the synchronous read and write operations. These can be driven independently or with the same driver.

Note: For Automotive ProASIC3 devices, dual-port mode is supported if the clocks to the two SRAM ports are the same and 180° out of phase (i.e., the port A clock is the inverse of the port B clock). For use of this macro as a single-port SRAM, the inputs and clock of one port should be tied off (grounded) to prevent errors during design compile.

PIPEA and PIPEB

These signals are used to specify pipelined read on the output. A LOW on PIPEA or PIPEB indicates a nonpipelined read, and the data appears on the corresponding output in the same clock cycle. A HIGH indicates a pipelined read, and data appears on the corresponding output in the next clock cycle.

Note: When using the SRAM in single-port mode for Automotive ProASIC3 devices, PIPEB should be tied to ground. For use in dual-port mode, the same clock with an inversion between the two clock pins of the macro should be used in the design to prevent errors during compile.

WMODEA and WMODEB

These signals are used to configure the behavior of the output when the RAM is in write mode. A LOW on these signals makes the output retain data from the previous read. A HIGH indicates pass-through behavior, wherein the data being written will appear immediately on the output. This signal is overridden when the RAM is being read.

Note: When using the SRAM in single-port mode for Automotive ProASIC3 devices, WMODEB should be tied to ground.

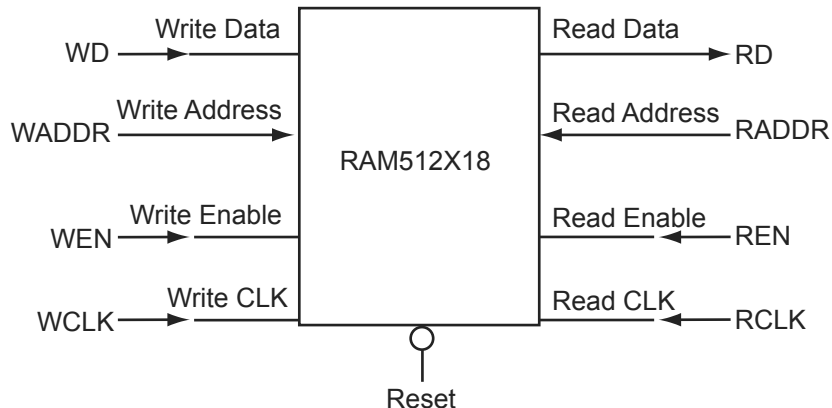
RESET

This active-low signal resets the control logic, forces the output hold state registers to zero, disables reads and writes from the SRAM block, and clears the data hold registers when asserted. It does not reset the contents of the memory array.

While the RESET signal is active, read and write operations are disabled. As with any asynchronous reset signal, care must be taken not to assert it too close to the edges of active read and write clocks.

ADDRA and ADDRb

These are used as read or write addresses, and they are 12 bits wide. When a depth of less than 4 k is specified, the unused high-order bits must be grounded (Table 6-3 on page 155).



Note: For timing diagrams of the RAM signals, refer to the appropriate family datasheet.

Figure 6-5 • 512X18 Two-Port RAM Block Diagram

Signal Descriptions for RAM512X18

RAM512X18 has slightly different behavior from RAM4K9, as it has dedicated read and write ports.

WW and RW

These signals enable the RAM to be configured in one of the two allowable aspect ratios (Table 6-5).

Table 6-5 • Aspect Ratio Settings for WW[1:0]

| WW[1:0] | RW[1:0] | D×W |
|---------|---------|----------|
| 01 | 01 | 512×9 |
| 10 | 10 | 256×18 |
| 00, 11 | 00, 11 | Reserved |

WD and RD

These are the input and output data signals, and they are 18 bits wide. When a 512×9 aspect ratio is used for write, WD[17:9] are unused and must be grounded. If this aspect ratio is used for read, RD[17:9] are undefined.

WADDR and RADDR

These are read and write addresses, and they are nine bits wide. When the 256×18 aspect ratio is used for write or read, WADDR[8] and RADDR[8] are unused and must be grounded.

WCLK and RCLK

These signals are the write and read clocks, respectively. They can be clocked on the rising or falling edge of WCLK and RCLK.

WEN and REN

These signals are the write and read enables, respectively. They are both active-low by default. These signals can be configured as active-high.

RESET

This active-low signal resets the control logic, forces the output hold state registers to zero, disables reads and writes from the SRAM block, and clears the data hold registers when asserted. It does not reset the contents of the memory array.

While the RESET signal is active, read and write operations are disabled. As with any asynchronous reset signal, care must be taken not to assert it too close to the edges of active read and write clocks.

PIPE

This signal is used to specify pipelined read on the output. A LOW on PIPE indicates a nonpipelined read, and the data appears on the output in the same clock cycle. A HIGH indicates a pipelined read, and data appears on the output in the next clock cycle.

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.

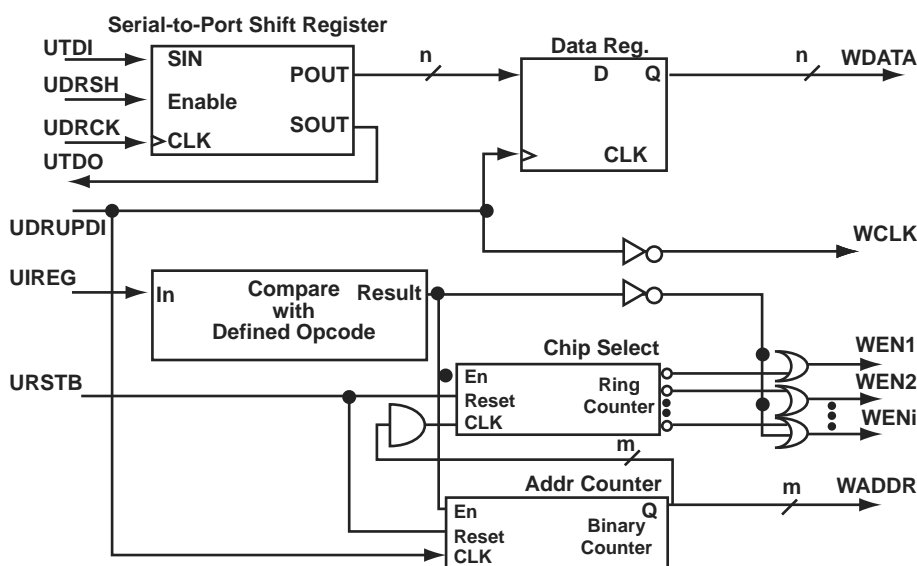


Figure 6-9 • Block Diagram of a Sample User Interface

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 UDRUPDI 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. UDRUPDI 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 UDRUPDI 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 UDRUPDI. This will provide enough time (equal to the UDRUPDI 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.

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

Solution 1

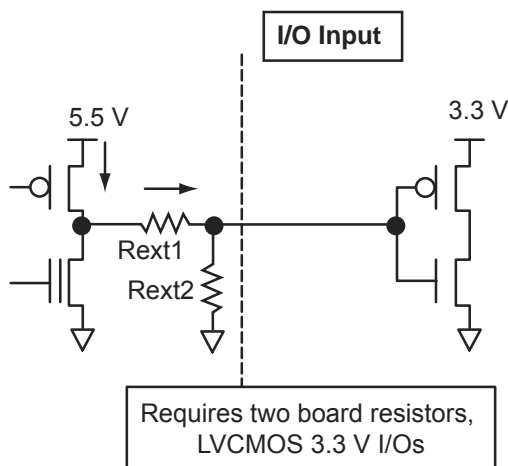


Figure 7-9 • Solution 1

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 7-10. Relying on the diode clamping would create an excessive pad DC voltage of $3.3\text{ V} + 0.7\text{ V} = 4\text{ V}$.

Solution 2

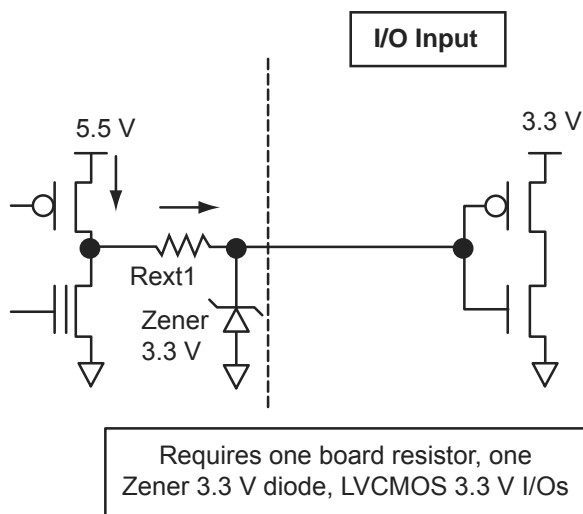


Figure 7-10 • Solution 2

Table 8-3 • VCCI Voltages and Compatible IGLOOe and ProASIC3E Standards

| VCCI and VMV (typical) | Compatible Standards |
|------------------------|---|
| 3.3 V | LVTTL/LVCMOS 3.3, PCI 3.3, SSTL3 (Class I and II), GTL+ 3.3, GTL 3.3, LVPECL |
| 2.5 V | LVC MOS 2.5, LVC MOS 2.5/5.0, SSTL2 (Class I and II), GTL+ 2.5, GTL 2.5, LVDS, DDR LVDS, B-LVDS, and M-LVDS |
| 1.8 V | LVC MOS 1.8 |
| 1.5 V | LVC MOS 1.5, HSTL (Class I and II) |
| 1.2 V | LVC MOS 1.2 |

Table 8-4 • VREF Voltages and Compatible IGLOOe and ProASIC3E Standards

| VREF (typical) | Compatible Standards |
|----------------|------------------------|
| 1.5 V | SSTL3 (Class I and II) |
| 1.25 V | SSTL2 (Class I and II) |
| 1.0 V | GTL+ 2.5, GTL+ 3.3 |
| 0.8 V | GTL 2.5, GTL 3.3 |
| 0.75 V | HSTL (Class I and II) |

Table 8-5 • Legal IGLOOe and ProASIC3E I/O Usage Matrix within the Same Bank

| I/O Bank Voltage (typical) | Minibank Voltage (typical) | LVTTL/LVCMOS 3.3 V | LVC MOS 2.5 V | LVC MOS 1.8 V | LVC MOS 1.5 V | 3.3 V PCI/PCI-X | GTL+ (3.3 V) | GTL+ (2.5 V) | GTL (3.3 V) | GTL (2.5 V) | HSTL Class I and II (1.5 V) | SSTL2 Class I and II (2.5 V) | SSTL3 Class I and II (3.3 V) | LVDS, B-LVDS, and M-LVDS, DDR (2.5 V ± 5%) | LVPECL (3.3 V) |
|----------------------------|----------------------------|--------------------|---------------|---------------|---------------|-----------------|--------------|--------------|-------------|-------------|-----------------------------|------------------------------|------------------------------|--|----------------|
| 3.3 V | – | | | | | | | | | | | | | | |
| | 0.80 V | | | | | | | | | | | | | | |
| | 1.00 V | | | | | | | | | | | | | | |
| | 1.50 V | | | | | | | | | | | | | | |
| 2.5 V | – | | | | | | | | | | | | | | |
| | 0.80 V | | | | | | | | | | | | | | |
| | 1.00 V | | | | | | | | | | | | | | |
| | 1.25 V | | | | | | | | | | | | | | |
| 1.8 V | – | | | | | | | | | | | | | | |
| 1.5 V | – | | | | | | | | | | | | | | |
| | 0.75 V | | | | | | | | | | | | | | |

Note: White box: Allowable I/O standard combination
Gray box: Illegal I/O standard combination

Rules for the DDR I/O Function

- The fanout between an I/O pin (D or Y) and a DDR (DDR_REG or DDR_OUT) macro must be equal to one for the combining to happen on that pin.
- If a DDR_REG macro and a DDR_OUT macro are combined on the same bidirectional I/O, they must share the same clear signal.
- Registers will not be combined in an I/O in the presence of DDR combining on the same I/O.

Using the I/O Buffer Schematic Cell

Libero SoC software includes the ViewDraw schematic entry tool. Using ViewDraw, the user can insert any supported I/O buffer cell in the top-level schematic. Figure 9-5 shows a top-level schematic with different I/O buffer cells. When synthesized, the netlist will contain the same I/O macro.

Figure 9-5 • I/O Buffer Schematic Cell Usage

I/O Bank Resource Usage

This is an important portion of the report. The user must meet the requirements stated in this table. Figure 9-10 shows the I/O Bank Resource Usage table included in the I/O bank report:

Figure 9-10 • I/O Bank Resource Usage Table

The example in Figure 9-10 shows that none of the I/O macros is assigned to the bank because more than one VCCI is detected.

I/O Voltage Usage

The I/O Voltage Usage table provides the number of VREF (E devices only) and V_{CCI} assignments required in the design. If the user decides to make I/O assignments manually (PDC or MVN), the issues listed in this table must be resolved before proceeding to Layout. As stated earlier, VREF assignments must be made if there are any voltage-referenced I/Os.

Figure 9-11 shows the I/O Voltage Usage table included in the I/O bank report.

Figure 9-11 • I/O Voltage Usage Table

The table in Figure 9-11 indicates that there are two voltage-referenced I/Os used in the design. Even though both of the voltage-referenced I/O technologies have the same VCCI voltage, their VREF voltages are different. As a result, two I/O banks are needed to assign the VCCI and VREF voltages.

In addition, there are six single-ended I/Os used that have the **same VCCI voltage**. **Since two banks are already assigned with the same VCCI voltage and there are enough unused bonded I/Os in**

| 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 · 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 |

12 – Security in Low Power Flash Devices

Security in Programmable Logic

The need for security on FPGA programmable logic devices (PLDs) has never been greater than today. If the contents of the FPGA can be read by an external source, the intellectual property (IP) of the system is vulnerable to unauthorized copying. Fusion, IGLOO, and ProASIC3 devices contain state-of-the-art circuitry to make the flash-based devices secure during and after programming. Low power flash devices have a built-in 128-bit Advanced Encryption Standard (AES) decryption core (except for 30 k gate devices and smaller). The decryption core facilitates secure in-system programming (ISP) of the FPGA core array fabric, the FlashROM, and the Flash Memory Blocks (FBs) in Fusion devices. The FlashROM, Flash Blocks, and FPGA core fabric can be programmed independently of each other, allowing the FlashROM or Flash Blocks to be updated without the need for change to the FPGA core fabric.

Microsemi has incorporated the AES decryption core into the low power flash devices and has also included the Microsemi flash-based lock technology, FlashLock.[®] Together, they provide leading-edge security in a programmable logic device. Configuration data loaded into a device can be decrypted prior to being written to the FPGA core using the AES 128-bit block cipher standard. The AES encryption key is stored in on-chip, nonvolatile flash memory.

This document outlines the security features offered in low power flash devices, some applications and uses, as well as the different software settings for each application.

Figure 12-1 • Overview on Security

Table 13-4 • Programming Header Pin Numbers and Description

| Pin | Signal | Source | Description |
|-----|--------------------|-------------------------|--|
| 1 | TCK | Programmer | JTAG Clock |
| 2 | GND ¹ | – | Signal Reference |
| 3 | TDO | Target Board | Test Data Output |
| 4 | NC | – | No Connect (FlashPro3/3X); Prog_Mode (FlashPro4). See note associated with Figure 13-5 on page 335 regarding Prog_Mode on FlashPro4. |
| 5 | TMS | Programmer | Test Mode Select |
| 6 | VJTAG | Target Board | JTAG Supply Voltage |
| 7 | VPUMP ² | Programmer/Target Board | Programming Supply Voltage |
| 8 | nTRST | Programmer | JTAG Test Reset (Hi-Z with 10 k Ω pull-down, HIGH, LOW, or toggling) |
| 9 | TDI | Programmer | Test Data Input |
| 10 | GND ¹ | – | Signal Reference |

Notes:

1. Both GND pins must be connected.
2. FlashPro4/3/3X can provide VPUMP if there is only one device on the target board.

List of Changes

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

| Date | Changes | Page |
|-------------|---|----------|
| August 2012 | This chapter will now be published standalone as an application note in addition to being part of the IGLOO/ProASIC3/Fusion FPGA fabric user's guides (SAR 38769). | N/A |
| | The "ISP Programming Header Information" section was revised to update the description of FP3-10PIN-ADAPTER-KIT in Table 13-3 • Programming Header Ordering Codes, clarifying that it is the adapter kit used for ProASIC ^{PLUS} based boards, and also for ProASIC3 based boards where a compact programming header is being used (SAR 36779). | 335 |
| June 2011 | The VPUMP programming mode voltage was corrected in Table 13-2 • Power Supplies. The correct value is 3.15 V to 3.45 V (SAR 30668). | 329 |
| | The notes associated with Figure 13-5 • Programming Header (top view) and Figure 13-6 • Board Layout and Programming Header Top View were revised to make clear the fact that IGLOO nano V2 devices can be programmed at 1.2 V (SAR 30787). | 335, 337 |
| | Figure 13-6 • Board Layout and Programming Header Top View was revised to include resistors tying TCK and TRST to GND. Microsemi recommends tying off TCK and TRST to GND if JTAG is not used (SAR 22921). RT ProASIC3 was added to the list of device families. | 337 |
| | In the "ISP Programming Header Information" section, the kit for adapting ProASIC ^{PLUS} devices was changed from FP3-10PIN-ADAPTER-KIT to FP3-26PIN-ADAPTER-KIT (SAR 20878). | 335 |
| 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 |
| | References to FlashPro4 and FlashPro3X were added to this chapter, giving distinctions between them. References to SmartGen were deleted and replaced with Libero IDE Catalog. | N/A |
| | The "ISP Architecture" section was revised to indicate that V2 devices can be programmed at 1.2 V VCC with FlashPro4. | 327 |
| | SmartFusion was added to Table 13-1 • Flash-Based FPGAs Supporting ISP. | 328 |
| | The "Programming Voltage (VPUMP) and VJTAG" section was revised and 1.2 V was added to Table 13-2 • Power Supplies. | 329 |
| | The "Nonvolatile Memory (NVM) Programming Voltage" section is new. | 329 |
| | Cortex-M3 was added to the "Cortex-M1 and Cortex-M3 Device Security" section. | 331 |
| | In the "ISP Programming Header Information" section, the additional header adapter ordering number was changed from FP3-26PIN-ADAPTER to FP3-10PIN-ADAPTER-KIT, which contains 26-pin migration capability. | 335 |
| | The description of NC was updated in Figure 13-5 • Programming Header (top view), Table 13-4 • Programming Header Pin Numbers and Description and Figure 13-6 • Board Layout and Programming Header Top View. | 335, 336 |
| | The "Symptoms of a Signal Integrity Problem" section was revised to add that customers are expected to troubleshoot board-level signal integrity issues by measuring voltages and taking scope plots. "FlashPro4/3/3X allows TCK to be lowered from 6 MHz down to 1 MHz to allow you to address some signal integrity problems" formerly read, "from 24 MHz down to 1 MHz." "The Scan Chain command expects to see 0x2" was changed to 0x1. | 337 |

Microsemi's Flash Families Support Voltage Switching Circuit

The flash FPGAs listed in Table 14-1 support the voltage switching circuit feature and the functions described in this document.

Table 14-1 • Flash-Based FPGAs Supporting Voltage Switching Circuit

| 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 | 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 |

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 14-1. 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 14-1. 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*.

Microprocessor Programming Support in Flash Devices

The flash-based FPGAs listed in Table 15-1 support programming with a microprocessor and the functions described in this document.

Table 15-1 • 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 15-1. Where the information applies to only one device 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 15-1. Where the information applies to only one device 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*.

Programming Algorithm

JTAG Interface

The low power flash families are fully compliant with the IEEE 1149.1 (JTAG) standard. They support all the mandatory boundary scan instructions (EXTEST, SAMPLE/PRELOAD, and BYPASS) as well as six optional public instructions (USERCODE, IDCODE, HIGHZ, and CLAMP).

IEEE 1532

The low power flash families are also fully compliant with the IEEE 1532 programming standard. The IEEE 1532 standard adds programming instructions and associated data registers to devices that comply with the IEEE 1149.1 standard (JTAG). These instructions and registers extend the capabilities of the IEEE 1149.1 standard such that the Test Access Port (TAP) can be used for configuration activities. The IEEE 1532 standard greatly simplifies the programming algorithm, reducing the amount of time needed to implement microprocessor ISP.

Implementation Overview

To implement device programming with a microprocessor, the user should first download the C-based STAPL player or DirectC code from the Microsemi SoC Products Group website. Refer to the website for future updates regarding the STAPL player and DirectC code.

http://www.microsemi.com/soc/download/program_debug/stapl/default.aspx

http://www.microsemi.com/soc/download/program_debug/directc/default.aspx

Using the easy-to-follow user's guide, create the low-level application programming interface (API) to provide the necessary basic functions. These API functions act as the interface between the programming software and the actual hardware (Figure 15-2).

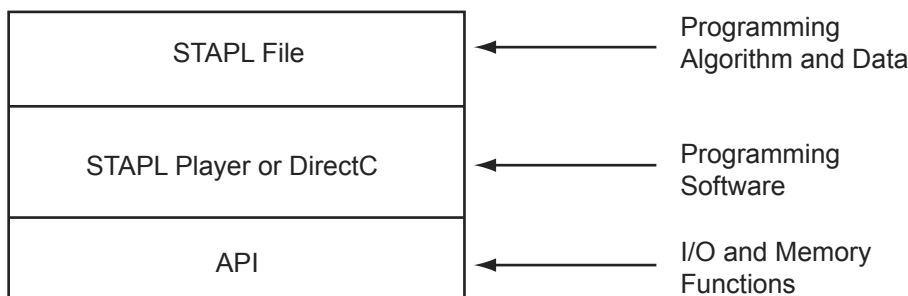


Figure 15-2 • Device Programming Code Relationship

The API is then linked with the STAPL player or DirectC and compiled using the microprocessor's compiler. Once the entire code is compiled, the user must download the resulting binary into the MCU system's program memory (such as ROM, EEPROM, or flash). The system is now ready for programming.

To program a design into the FPGA, the user creates a bitstream or STAPL file using the Microsemi Designer software, downloads it into the MCU system's volatile memory, and activates the stored programming binary file (Figure 15-3 on page 352). Once the programming is completed, the bitstream or STAPL file can be removed from the system, as the configuration profile is stored in the flash FPGA fabric and does not need to be reloaded at every system power-on.

Typical UJTAG Applications

Bidirectional access to the JTAG port from VersaTiles—without putting the device into test mode—creates flexibility to implement many different applications. This section describes a few of these. All are based on importing/exporting data through the UJTAG tiles.

Clock Conditioning Circuitry—Dynamic Reconfiguration

In low power flash devices, CCCs, which include PLLs, can be configured dynamically through either an 81-bit embedded shift register or static flash programming switches. These 81 bits control all the characteristics of the CCC: routing MUX architectures, delay values, divider values, etc. Table 17-3 lists the 81 configuration bits in the CCC.

Table 17-3 • Configuration Bits of Fusion, IGLOO, and ProASIC3 CCC Blocks

| Bit Number(s) | Control Function |
|---------------|------------------|
| 80 | RESET ENABLE |
| 79 | DYNCSSEL |
| 78 | DYNBSEL |
| 77 | DYNASEL |
| <76:74> | VCOSSEL [2:0] |
| 73 | STATCSSEL |
| 72 | STATBSEL |
| 71 | STATASEL |
| <70:66> | DLYC [4:0] |
| <65:61> | DLYB [4:0] |
| <60:56> | DLYGLC [4:0] |
| <55:51> | DLYGLB [4:0] |
| <50:46> | DLYGLA [4:0] |
| 45 | XDLYSEL |
| <44:40> | FBDLY [4:0] |
| <39:38> | FBSEL |
| <37:35> | OCMUX [2:0] |
| <34:32> | OBMUX [2:0] |
| <31:29> | OAMUX [2:0] |
| <28:24> | OCDIV [4:0] |
| <23:19> | OBDIV [4:0] |
| <18:14> | OADIV [4:0] |
| <13:7> | FBDIV [6:0] |
| <6:0> | FINDIV [6:0] |

The embedded 81-bit shift register (for the dynamic configuration of the CCC) is accessible to the VersaTiles, which, in turn, have access to the UJTAG tiles. Therefore, the CCC configuration shift register can receive and load the new configuration data stream from JTAG.

Dynamic reconfiguration eliminates the need to reprogram the device when reconfiguration of the CCC functional blocks is needed. The CCC configuration can be modified while the device continues to operate. Employing the UJTAG core requires the user to design a module to provide the configuration data and control the CCC configuration shift register. In essence, this is a user-designed TAP Controller requiring chip resources.

Similar reconfiguration capability exists in the ProASIC^{PLUS}® family. The only difference is the number of shift register bits controlling the CCC (27 in ProASIC^{PLUS} and 81 in IGLOO, ProASIC3, and Fusion).