



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	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-pq208i

FlashROM Security Use Models	311
Generating Programming Files	313
Conclusion	324
Glossary	324
References	324
Related Documents	325
List of Changes	325
13 In-System Programming (ISP) of Microsemi's Low Power Flash Devices Using FlashPro4/3/3X...	327
Introduction	327
ISP Architecture	327
ISP Support in Flash-Based Devices	328
Programming Voltage (VPUMP) and VJTAG	329
Nonvolatile Memory (NVM) Programming Voltage	329
IEEE 1532 (JTAG) Interface	330
Security	330
Security in ARM-Enabled Low Power Flash Devices	331
FlashROM and Programming Files	333
Programming Solution	334
ISP Programming Header Information	335
Board-Level Considerations	337
Conclusion	338
Related Documents	338
List of Changes	339
14 Core Voltage Switching Circuit for IGLOO and ProASIC3L In-System Programming	341
Introduction	341
Microsemi's Flash Families Support Voltage Switching Circuit	342
Circuit Description	343
Circuit Verification	344
DirectC	346
Conclusion	346
List of Changes	347
15 Microprocessor Programming of Microsemi's Low Power Flash Devices	349
Introduction	349
Microprocessor Programming Support in Flash Devices	350
Programming Algorithm	351
Implementation Overview	351
Hardware Requirement	354
Security	354
Conclusion	355
List of Changes	356
16 Boundary Scan in Low Power Flash Devices	357
Boundary Scan	357
TAP Controller State Machine	357
Microsemi's Flash Devices Support the JTAG Feature	358
Boundary Scan Support in Low Power Devices	359
Boundary Scan Opcodes	359

Global Macro and Placement Selections

Low power flash devices provide the flexibility of choosing one of the three global input pad locations available to connect to a global / quadrant global network. For 60K gate devices and above, if the single-ended I/O standard is chosen, there is flexibility to choose one of the global input pads (the first, second, and fourth input). Once chosen, the other I/O locations are used as regular I/Os. If the differential I/O standard is chosen, the first and second inputs are considered as paired, and the third input is paired with a regular I/O. The user then has the choice of selecting one of the two sets to be used as the global input source. There is also the option to allow an internal clock signal to feed the global network. A multiplexer tree selects the appropriate global input for routing to the desired location. Note that the global I/O pads do not need to feed the global network; they can also be used as regular I/O pads.

Hardwired I/O Clock Source

Hardwired I/O refers to global input pins that are hardwired to the multiplexer tree, which directly accesses the global network. These global input pins have designated pin locations and are indicated with the I/O naming convention Gmn (m refers to any one of the positions where the global buffers is available, and n refers to any one of the three global input MUXes and the pin number of the associated global location, m). Choosing this option provides the benefit of directly connecting to the global buffers, which provides less delay. See Figure 3-11 for an example illustration of the connections, shown in red. If a CLKBUF macro is initiated, the clock input can be placed at one of nine dedicated global input pin locations: GmA0, GmA1, GmA2, GmB0, GmB1, GmB2, GmC0, GmC1, or GmC2. Note that the placement of the global will determine whether you are using chip global or quadrant global. For example, if the CLKBIF is placed in one of the GF pin locations, it will use the chip global network; if the CLKBIF is placed in one of the GA pin locations, it will use quadrant global network. This is shown in Figure 3-12 on page 65 and Figure 3-13 on page 65.

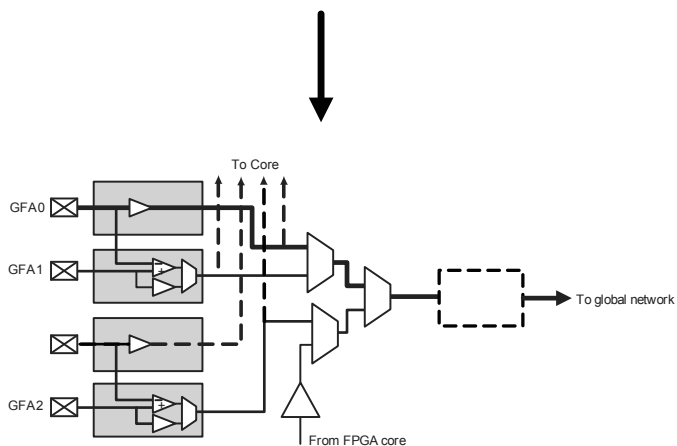


Figure 3-11 • CLKBUF Macro

The following will happen during demotion of a global signal to regular nets:

- CLKBUF_x becomes INBUF_x; CLKINT is removed from the netlist.
- The essential global macro, such as the output of the Clock Conditioning Circuit, cannot be demoted.
- No automatic buffering will happen.

Since no automatic buffering happens when a signal is demoted, this net may have a high delay due to large fanout. This may have a negative effect on the quality of the results. Microsemi recommends that the automatic global demotion only be used on small-fanout nets. Use clock networks for high-fanout nets to improve timing and routability.

Spine Assignment

The low power flash device architecture allows the global networks to be segmented and used as clock spines. These spines, also called local clock networks, enable the use of PDC or MVN to assign a signal to a spine.

PDC syntax to promote a net to a spine/local clock:

```
assign_local_clock -net netname -type [quadrant|chip] Tn|Bn|Tn:Bm
```

If the net is driven by a clock macro, Designer automatically demotes the clock net to a regular net before it is assigned to a spine. Nets driven by a PLL or CLKDLY macro cannot be assigned to a local clock.

When assigning a signal to a spine or quadrant global network using PDC (pre-compile), the Designer software will legalize the shared instances. The number of shared instances to be legalized can be controlled by compile options. If these networks are created in MVN (only quadrant globals can be created), no legalization is done (as it is post-compile). Designer does not do legalization between non-clock nets.

As an example, consider two nets, net_clk and net_reset, driving the same flip-flop. The following PDC constraints are used:

```
assign_local_clock -net net_clk -type chip T3
assign_local_clock -net net_reset -type chip T1:T2
```

During Compile, Designer adds a buffer in the reset net and places it in the T1 or T2 region, and places the flip-flop in the T3 spine region (Figure 3-16).

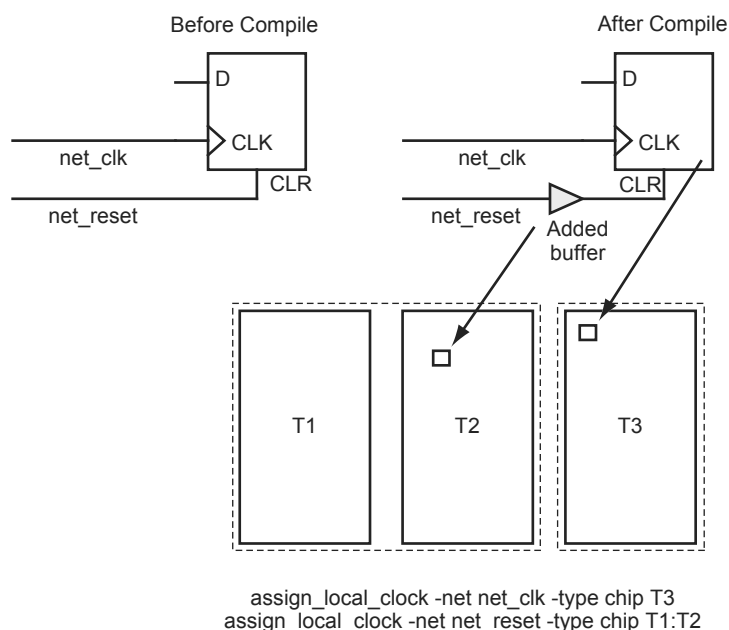


Figure 3-16 • Adding a Buffer for Shared Instances

Figure 3-18 • Globals Management GUI in Designer

3. Occasionally, the synthesis tool assigns a global macro to clock nets, even though the fanout is significantly less than other asynchronous signals. Select **Demote global nets whose fanout is less than** and enter a reasonable value for fanouts. This frees up some global networks from the signals that have very low fanouts. This can also be done using PDC.
4. Use a local clock network for the signals that do not need to go to the whole chip but should have low skew. This local clock network assignment can only be done using PDC.
5. Assign the I/O buffer using MVN if you have fixed I/O assignment. As shown in Figure 3-10 on page 61, there are three sets of global pins that have a hardwired connection to each global network. Do not try to put multiple CLKBUF macros in these three sets of global pins. For example, do not assign two CLKBUFs to GAA0x and GAA2x pins.
6. You must click **Commit** at the end of MVN assignment. This runs the pre-layout checker and checks the validity of global assignment.
7. Always run Compile with the **Keep existing physical constraints** option on. This uses the quadrant clock network assignment in the MVN assignment and checks if you have the desired signals on the global networks.
8. Run Layout and check the timing.

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).

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).

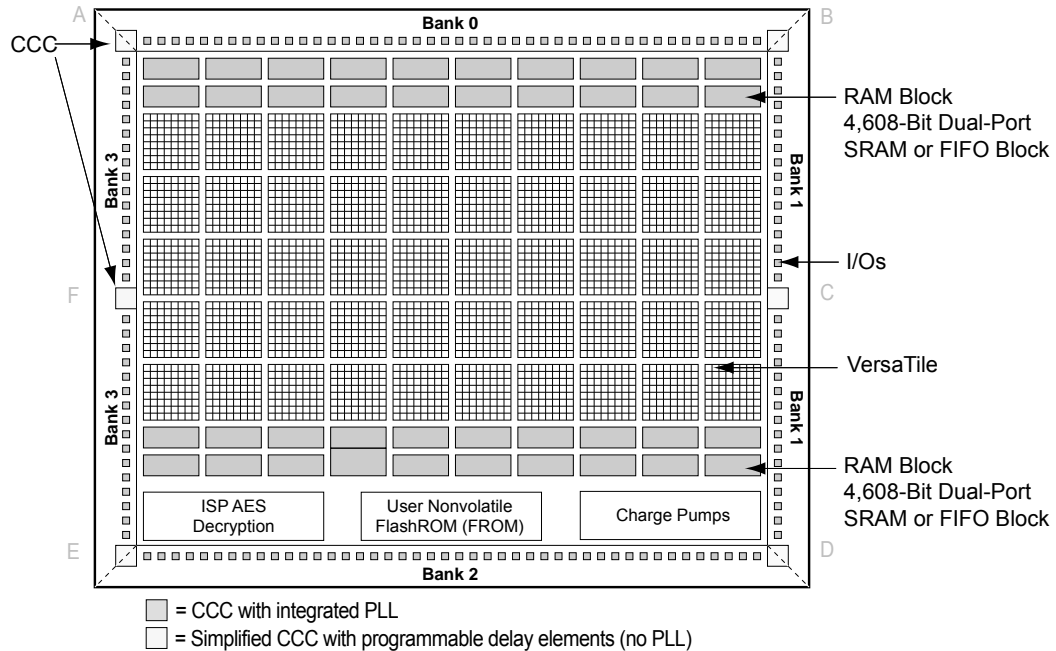


Figure 4-17 • CCC Locations in Fusion Family Devices (AFS090, AFS250, M1AFS250)

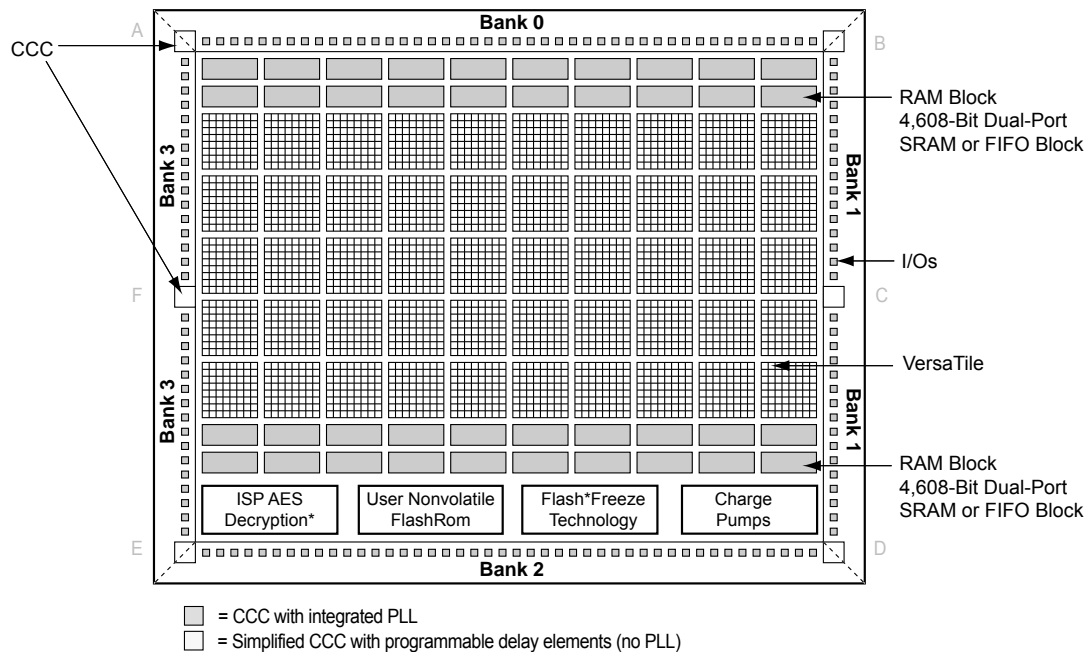


Figure 4-18 • CCC Locations in Fusion Family Devices (except AFS090, AFS250, M1AFS250)

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:

$$f_{GLA} = f_{CLKA} \times m / (n \times u) - \text{GLA Primary PLL Output Clock} \quad \text{EQ 4-1}$$

$$f_{GLB} = f_{YB} = f_{CLKA} \times m / (n \times v) - \text{GLB Secondary 1 PLL Output Clock(s)} \quad \text{EQ 4-2}$$

$$f_{GLC} = f_{YC} = f_{CLKA} \times m / (n \times w) - \text{GLC Secondary 2 PLL Output Clock(s)} \quad \text{EQ 4-3}$$

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:

$$f_{GLA} = f_{GLB} \times (v / u) = f_{GLC} \times (w / u) \quad \text{EQ 4-4}$$

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.

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

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

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
Type                : Dynamic CCC
Input Freq(MHz)     : 30.000
CLKA Source         : Hardwired I/O
Feedback Delay Value Index : 1
Feedback Mux Select : 1
XDLY Mux Select     : No
Primary Freq(MHz)   : 33.000
Primary PhaseShift  : 0
Primary Delay Value Index : 1
Primary Mux Select  : 4
Secondary1 Freq(MHz) : 40.000
Use GLB             : YES
Use YB              : NO
GLB Delay Value Index : 1
YB Delay Value Index : 1
Secondary1 PhaseShift : 0
Secondary1 Mux Select : 0
Secondary1 Input Freq(MHz) : 40.000
CLKB Source         : Hardwired I/O
Secondary2 Freq(MHz) : 50.000
Use GLC             : YES
Use YC              : NO
GLC Delay Value Index : 1
YC Delay Value Index : 1
Secondary2 PhaseShift : 0
Secondary2 Mux Select : 0
Secondary2 Input Freq(MHz) : 50.000
CLKC Source         : Hardwired I/O

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
DLYGLA[4:0]         00000
DLYGLB[4:0]         00000
```

5 – FlashROM in Microsemi’s Low Power Flash Devices

Introduction

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

Architecture of User Nonvolatile FlashROM

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

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

		Byte Number in Bank								4 LSB of ADDR (READ)							
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bank Number 3 MSB of ADDR (READ)	7																
	6																
	5																
	4																
	3																
	2																
	1																
	0																

Figure 5-1 • FlashROM Architecture

6 – SRAM and FIFO Memories in Microsemi's Low Power Flash Devices

Introduction

As design complexity grows, greater demands are placed upon an FPGA's embedded memory. Fusion, IGLOO, and ProASIC3 devices provide the flexibility of true dual-port and two-port SRAM blocks. The embedded memory, along with built-in, dedicated FIFO control logic, can be used to create cascading RAM blocks and FIFOs without using additional logic gates.

IGLOO, IGLOO PLUS, and ProASIC3L FPGAs contain an additional feature that allows the device to be put in a low power mode called Flash*Freeze. In this mode, the core draws minimal power (on the order of 2 to 127 μ W) and still retains values on the embedded SRAM/FIFO and registers. Flash*Freeze technology allows the user to switch to Active mode on demand, thus simplifying power management and the use of SRAM/FIFOs.

Device Architecture

The low power flash devices feature up to 504 kbits of RAM in 4,608-bit blocks (Figure 6-1 on page 148 and Figure 6-2 on page 149). The total embedded SRAM for each device can be found in the datasheets. These memory blocks are arranged along the top and bottom of the device to allow better access from the core and I/O (in some devices, they are only available on the north side of the device). Every RAM block has a flexible, hardwired, embedded FIFO controller, enabling the user to implement efficient FIFOs without sacrificing user gates.

In the IGLOO and ProASIC3 families of devices, the following memories are supported:

- 30 k gate devices and smaller do not support SRAM and FIFO.
- 60 k and 125 k gate devices support memories on the north side of the device only.
- 250 k devices and larger support memories on the north and south sides of the device.

In Fusion devices, the following memories are supported:

- AFS090 and AFS250 support memories on the north side of the device only.
- AFS600 and AFS1500 support memories on the north and south sides of the device.

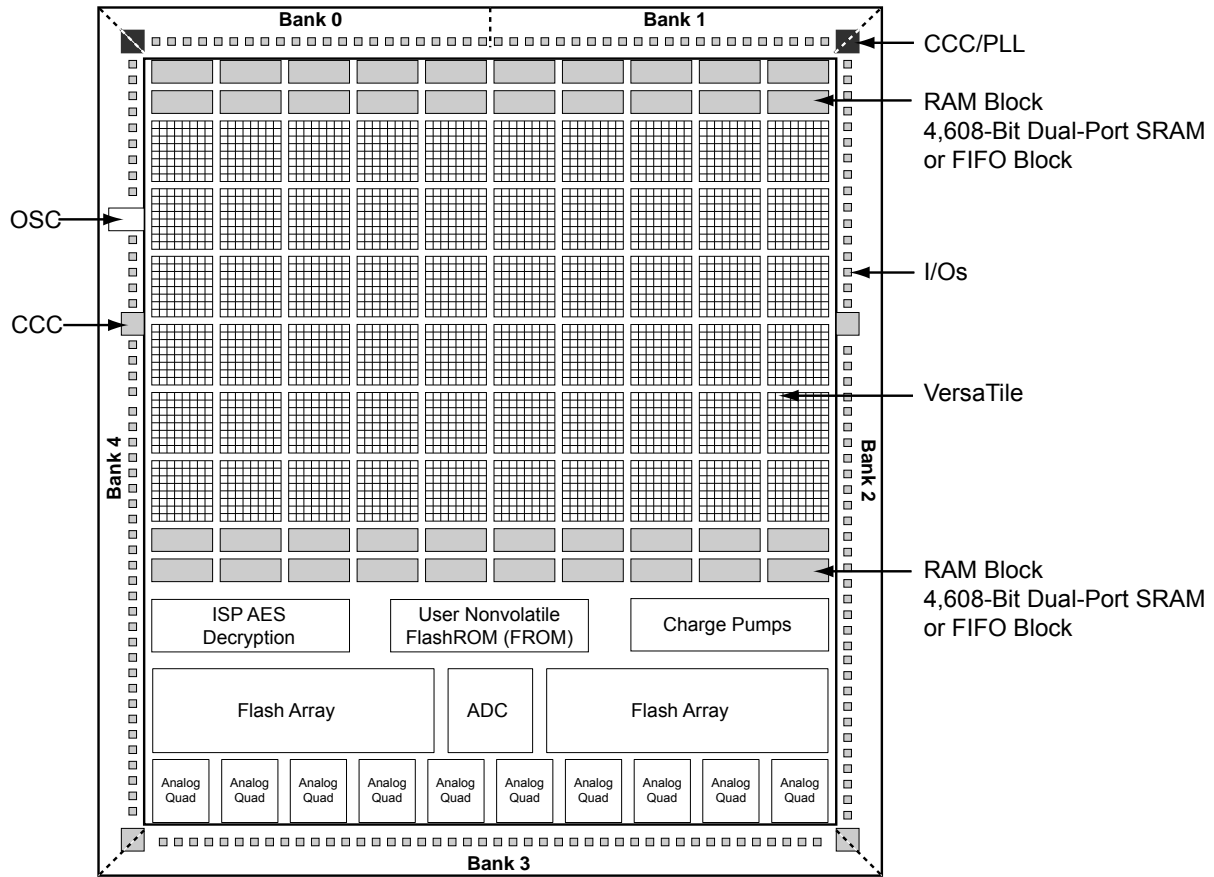


Figure 6-2 • Fusion Device Architecture Overview (AFS600)

```
//
addr_counter counter_1 (.Clock(data_update), .Q(wr_addr), .Aset(rst_n),
    .Enable(enable));
addr_counter counter_2 (.Clock(test_clk), .Q(rd_addr), .Aset(rst_n),
    .Enable( test_active));

endmodule
```

Interface Block / UJTAG Wrapper

This example is a sample wrapper, which connects the interface block to the UJTAG and the memory blocks.

```
// WRAPPER
module top_init (TDI, TRSTB, TMS, TCK, TDO, test, test_clk, test_out);

input TDI, TRSTB, TMS, TCK;
output TDO;
input test, test_clk;
output [3:0] test_out;

wire [7:0] IR;
wire reset, DR_shift, DR_cap, init_clk, DR_update, data_in, data_out;
wire clk_out, wen, ren;
wire [3:0] word_in, word_out;
wire [1:0] write_addr, read_addr;

UJTAG UJTAG_U1 (.UIREG0(IR[0]), .UIREG1(IR[1]), .UIREG2(IR[2]), .UIREG3(IR[3]),
    .UIREG4(IR[4]), .UIREG5(IR[5]), .UIREG6(IR[6]), .UIREG7(IR[7]), .URSTB(reset),
    .UDRSH(DR_shift), .UDRCAP(DR_cap), .UDRCK(init_clk), .UDRUPD(DR_update),
    .UT-DI(data_in), .TDI(TDI), .TMS(TMS), .TCK(TCK), .TRSTB(TRSTB), .TDO(TDO),
    .UT-DO(data_out));
mem_block RAM_block (.DO(word_out), .RCLOCK(clk_out), .WCLOCK(clk_out), .DI(word_in),
    .WRB(wen), .RDB(ren), .WAD-DR(write_addr), .RADDR(read_addr));
interface init_block (.IR(IR), .rst_n(reset), .data_shift(DR_shift), .clk_in(init_clk),
    .data_update(DR_update), .din_ser(data_in), .dout_ser(data_out), .test(test),
    .test_out(test_out), .test_clk(test_clk), .clk_out(clk_out), .wr_en(wen),
    .rd_en(ren), .write_word(word_in), .read_word(word_out), .rd_addr(read_addr),
    .wr_addr(write_addr));

endmodule
```

Address Counter

```
module addr_counter (Clock, Q, Aset, Enable);

input Clock;
output [1:0] Q;
input Aset;
input Enable;

reg [1:0] Qaux;

always @(posedge Clock or negedge Aset)
begin
    if (!Aset) Qaux <= 2'b11;
    else if (Enable) Qaux <= Qaux + 1;
end

assign Q = Qaux;

endmodule
```

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

IGLOO and ProASIC3 devices support output slew rate control: high and low. Microsemi recommends the high slew rate option to minimize the propagation delay. This high-speed option may introduce noise into the system if appropriate signal integrity measures are not adopted. Selecting a low slew rate reduces this kind of noise but adds some delays in the system. Low slew rate is recommended when bus transients are expected.

Output Drive

The output buffers of IGLOO and ProASIC3 devices can provide multiple drive strengths to meet signal integrity requirements. The LVTTTL and LVCMOS (except 1.2 V LVCMOS) standards have selectable drive strengths. Other standards have a preset value.

Drive strength should also be selected according to the design requirements and noise immunity of the system.

The output slew rate and multiple drive strength controls are available in LVTTTL/LVCMOS 3.3 V, LVCMOS 2.5 V, LVCMOS 2.5 V / 5.0 V input, LVCMOS 1.8 V, and LVCMOS 1.5 V. All other I/O standards have a high output slew rate by default.

For 30 k gate devices, refer to Table 7-14. For other ProASIC3 and IGLOO devices, refer to Table 7-15 through Table 7-16 on page 203 for more information about the slew rate and drive strength specification. Refer to Table 7-4 on page 178 for I/O bank type definitions.

There will be a difference in timing between the Standard Plus I/O banks and the Advanced I/O banks. Refer to the I/O timing tables in the datasheet for the standards supported by each device.

Table 7-14 • IGLOO and ProASIC3 Output Drive and Slew for Standard I/O Bank Type (for 30 k gate devices)

I/O Standards	2 mA	4 mA	6 mA	8 mA	Slew	
LVTTTL/LVCMOS 3.3 V	✓	✓	✓	✓	High	Low
LVCMOS 2.5 V	✓	✓	✓	✓	High	Low
LVCMOS 1.8 V	✓	✓	–	–	High	Low
LVCMOS 1.5 V	✓	–	–	–	High	Low

Table 7-15 • IGLOO and ProASIC3 Output Drive and Slew for Standard Plus I/O Bank Type

I/O Standards	2 mA	4 mA	6 mA	8 mA	12 mA	16 mA	Slew	
LVTTTL	✓	✓	✓	✓	✓	✓	High	Low
LVCMOS 3.3 V	✓	✓	✓	✓	✓	✓	High	Low
LVCMOS 2.5 V	✓	✓*	✓	✓*	✓	–	High	Low
LVCMOS 1.8 V	✓	✓	✓	✓	–	–	High	Low
LVCMOS 1.5 V	✓	✓	–	–	–	–	High	Low

Note: *Not available in Automotive devices.

Table 7-16 • IGLOO and ProASIC3 Output Drive and Slew for Advanced I/O Bank Type

I/O Standards	2 mA	4 mA	6 mA	8 mA	12 mA	16 mA	24 mA	Slew	
LVTTTL	✓	✓	✓	✓	✓	✓	✓	High	Low
LVCMOS 3.3 V	✓	✓	✓	✓	✓	✓	✓	High	Low
LVCMOS 2.5 V	✓	✓*	✓	✓*	✓	✓	✓	High	Low
LVCMOS 2.5/5.0 V	✓	✓*	✓	✓*	✓	✓	✓	High	Low
LVCMOS 1.8 V	✓	✓	✓	✓	✓	✓	–	High	Low
LVCMOS 1.5 V	✓	✓	✓	✓	✓	–	–	High	Low

Note: Not available in Automotive devices.

Low Power Flash Device I/O Support

The low power flash FPGAs listed in Table 8-1 support I/Os and the functions described in this document.

Table 8-1 • Flash-Based FPGAs

Series	Family*	Description
IGLOO	IGLOOe	Higher density IGLOO FPGAs with six PLLs and additional I/O standards
ProASIC3	ProASIC3E	Higher density ProASIC3 FPGAs with six PLLs and additional I/O standards
	ProASIC3L	ProASIC3 FPGAs supporting 1.2 V to 1.5 V with Flash*Freeze technology
	Military ProASIC3/EL	Military temperature A3PE600L, A3P1000, and A3PE3000L
	RT ProASIC3	Radiation-tolerant RT3PE600L and RT3PE3000L

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 8-1. 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 8-1. 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*.

I/O Standards

Single-Ended Standards

These I/O standards use a push-pull CMOS output stage with a voltage referenced to system ground to designate logical states. The input buffer configuration, output drive, and I/O supply voltage (V_{CCI}) vary among the I/O standards (Figure 8-6).

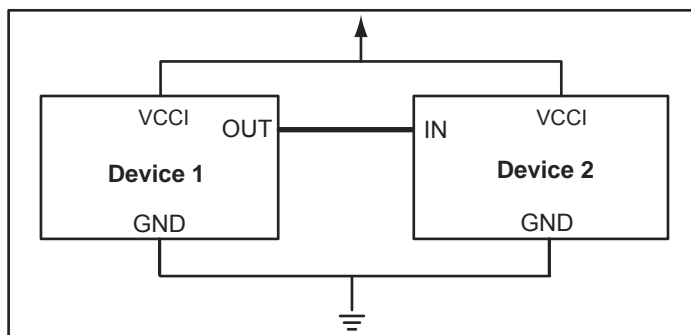


Figure 8-6 • Single-Ended I/O Standard Topology

The advantage of these standards is that a common ground can be used for multiple I/Os. This simplifies board layout and reduces system cost. Their low-edge-rate (dv/dt) data transmission causes less electromagnetic interference (EMI) on the board. However, they are not suitable for high-frequency (>200 MHz) switching due to noise impact and higher power consumption.

LVTTL (Low-Voltage TTL)

This is a general-purpose standard (EIA/JESD8-B) for 3.3 V applications. It uses an LVTTL input buffer and a push-pull output buffer. The LVTTL output buffer can have up to six different programmable drive strengths. The default drive strength is 12 mA. V_{CCI} is 3.3 V. Refer to "I/O Programmable Features" on page 227 for details.

LVC MOS (Low-Voltage CMOS)

The low power flash devices provide four different kinds of LVC MOS: LVC MOS 3.3 V, LVC MOS 2.5 V, LVC MOS 1.8 V, and LVC MOS 1.5 V. LVC MOS 3.3 V is an extension of the LVC MOS standard (JESD8-B-compliant) used for general-purpose 3.3 V applications. LVC MOS 2.5 V is an extension of the LVC MOS standard (JESD8-5-compliant) used for general-purpose 2.5 V applications. LVC MOS 2.5 V for the 30 k gate devices has a clamp diode to V_{CCI} , but for all other devices there is no clamp diode.

There is yet another standard supported by IGLOO and ProASIC3 devices (except A3P030): LVC MOS 2.5/5.0 V. This standard is similar to LVC MOS 2.5 V, with the exception that it can support up to 3.3 V on the input side (2.5 V output drive).

LVC MOS 1.8 V is an extension of the LVC MOS standard (JESD8-7-compliant) used for general-purpose 1.8 V applications. LVC MOS 1.5 V is an extension of the LVC MOS standard (JESD8-11-compliant) used for general-purpose 1.5 V applications.

The V_{CCI} values for these standards are 3.3 V, 2.5 V, 1.8 V, and 1.5 V, respectively. Like LVTTL, the output buffer has up to seven different programmable drive strengths (2, 4, 6, 8, 12, 16, and 24 mA). Refer to "I/O Programmable Features" on page 227 for details.

3.3 V PCI (Peripheral Component Interface)

This standard specifies support for both 33 MHz and 66 MHz PCI bus applications. It uses an LVTTL input buffer and a push-pull output buffer. With the aid of an external resistor, this I/O standard can be 5 V-compliant for low power flash devices. It does not have programmable drive strength.

3.3 V PCI-X (Peripheral Component Interface Extended)

An enhanced version of the PCI specification, 3.3 V PCI-X can support higher average bandwidths; it increases the speed that data can move within a computer from 66 MHz to 133 MHz. It is backward-

those banks, the user does not need to assign the same VCCI voltage to another bank. The user needs to assign the other three VCCI voltages to three more banks.

Assigning Technologies and VREF to I/O Banks

Low power flash devices offer a wide variety of I/O standards, including voltage-referenced standards. Before proceeding to Layout, each bank must have the required VCCI voltage assigned for the corresponding I/O technologies used for that bank. The voltage-referenced standards require the use of a reference voltage (VREF). This assignment can be done manually or automatically. The following sections describe this in detail.

Manually Assigning Technologies to I/O Banks

The user can import the PDC at this point and resolve this requirement. The PDC command is

```
set_iobank [bank name] -vcci [vcci value]
```

Another method is to use the I/O Bank Settings dialog box (**MVN > Edit > I/O Bank Settings**) to set up the V_{CCI} voltage for the bank (Figure 9-12).

Figure 9-12 • Setting VCCI for a Bank

Automatically Assigning Technologies to I/O Banks

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

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

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

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

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

A – Summary of Changes

History of Revision to Chapters

The following table lists chapters that were affected in each revision of this document. Each chapter includes its own change history because it may appear in other device family user's guides. Refer to the individual chapter for a list of specific changes.

Revision (month/year)	Chapter Affected	List of Changes (page number)
Revision 4 (September 2012)	"Microprocessor Programming of Microsemi's Low Power Flash Devices" was revised.	356
Revision 3 (August 2012)	"FPGA Array Architecture in Low Power Flash Devices" was revised.	20
	"Clock Conditioning Circuits in Low Power Flash Devices and Mixed Signal FPGAs" was revised.	129
	"SRAM and FIFO Memories in Microsemi's Low Power Flash Devices" was revised.	173
	"I/O Structures in IGLOO and ProASIC3 Devices" was revised.	210
	"I/O Structures in IGLOOe and ProASIC3E Devices" was revised.	249
	The "Pin Descriptions" and "Packaging" chapters were removed. This information is now published in the datasheet for each product line (SAR 34773).	
	"In-System Programming (ISP) of Microsemi's Low Power Flash Devices Using FlashPro4/3/3X" was revised.	339
Revision 2 (December 2011)	"Boundary Scan in Low Power Flash Devices" was revised.	362
	"Clock Conditioning Circuits in Low Power Flash Devices and Mixed Signal FPGAs" was revised.	129
Revision 1 (June 2011)	"UJTAG Applications in Microsemi's Low Power Flash Devices" was revised.	372
	"Clock Conditioning Circuits in Low Power Flash Devices and Mixed Signal FPGAs" was revised.	129
	"I/O Structures in IGLOO and ProASIC3 Devices" was revised.	210
	"I/O Structures in IGLOOe and ProASIC3E Devices" was revised.	249
	"I/O Software Control in Low Power Flash Devices" was revised.	270
Revision 0 (July 2010)	"In-System Programming (ISP) of Microsemi's Low Power Flash Devices Using FlashPro4/3/3X" was revised.	339
	The ProASIC3L Flash Family FPGAs Handbook was divided into two parts to create the ProASIC3L Low Power Flash FPGAs Datasheet and the ProASIC3L FPGA Fabric User's Guide.	N/A
	"Global Resources in Low Power Flash Devices" was revised.	75
	"Clock Conditioning Circuits in Low Power Flash Devices and Mixed Signal FPGAs" was revised.	129
	"I/O Software Control in Low Power Flash Devices" was revised.	270