



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	-40°C ~ 100°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-z1vqg100i">https://www.e-xfl.com/product-detail/microchip-technology/a3pn030-z1vqg100i</a>

PLL Core Specifications .....	84
Functional Description .....	85
Software Configuration .....	96
Detailed Usage Information .....	104
Recommended Board-Level Considerations .....	112
Conclusion .....	113
Related Documents .....	113
List of Changes .....	113
 5 FlashROM in Microsemi's Low Power Flash Devices .....	117
Introduction .....	117
Architecture of User Nonvolatile FlashROM .....	117
FlashROM Support in Flash-Based Devices .....	118
FlashROM Applications .....	120
FlashROM Security .....	121
Programming and Accessing FlashROM .....	122
FlashROM Design Flow .....	124
Custom Serialization Using FlashROM .....	129
Conclusion .....	130
Related Documents .....	130
List of Changes .....	130
 6 SRAM and FIFO Memories in Microsemi's Low Power Flash Devices .....	131
Introduction .....	131
Device Architecture .....	131
SRAM/FIFO Support in Flash-Based Devices .....	134
SRAM and FIFO Architecture .....	135
Memory Blocks and Macros .....	135
Initializing the RAM/FIFO .....	148
Software Support .....	154
Conclusion .....	157
List of Changes .....	157
 7 I/O Structures in nano Devices .....	159
Introduction .....	159
Low Power Flash Device I/O Support .....	161
nano Standard I/Os .....	162
I/O Architecture .....	164
I/O Standards .....	166
Wide Range I/O Support .....	166
I/O Features .....	167
Simultaneously Switching Outputs (SSOs) and Printed Circuit Board Layout .....	176
I/O Software Support .....	177
User I/O Naming Convention .....	178
I/O Bank Architecture and CCC Naming Conventions .....	179
Board-Level Considerations .....	181
Conclusion .....	182
Related Documents .....	183
List of Changes .....	183



## CCC Support in Microsemi's Flash Devices

The flash FPGAs listed in Table 4-1 support the CCC feature and the functions described in this document.

**Table 4-1 • 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 PLUS	IGLOO FPGAs with enhanced I/O capabilities
	IGLOO nano	The industry's lowest-power, smallest-size solution
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 4-1. 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 4-1. 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*.



```
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
*****
```

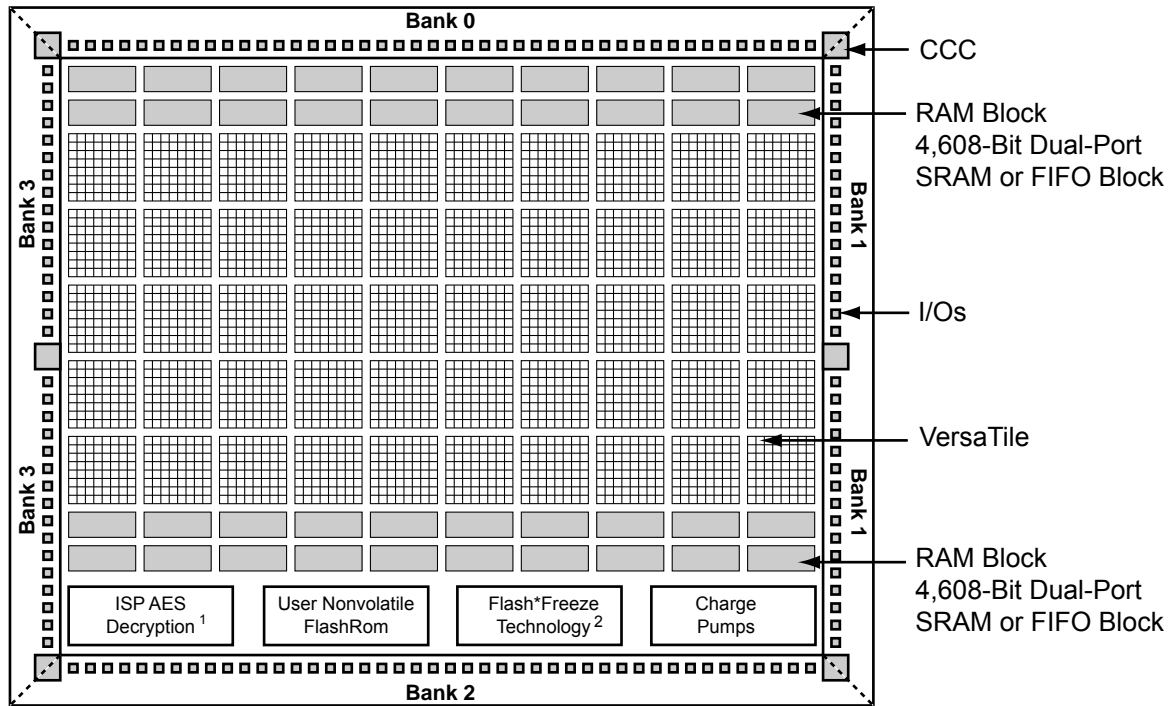
Name	: delay_macro
Family	: ProASIC3
Output Format	: Verilog
Type	: Delayed Clock
Delay Index	: 2
CLKA Source	: Hardwired I/O

Total Clock Delay = 0.935 ns.

The resultant CLKDLY macro Verilog netlist is as follows:

```
module delay_macro(GL,CLK);

output GL;
input CLK;
```



Notes:

1. AES decryption not supported in 30 k gate devices and smaller.
2. Flash\*Freeze is supported in all IGLOO devices and the ProASIC3L devices.

**Figure 6-1 • IGLOO and ProASIC3 Device Architecture Overview**

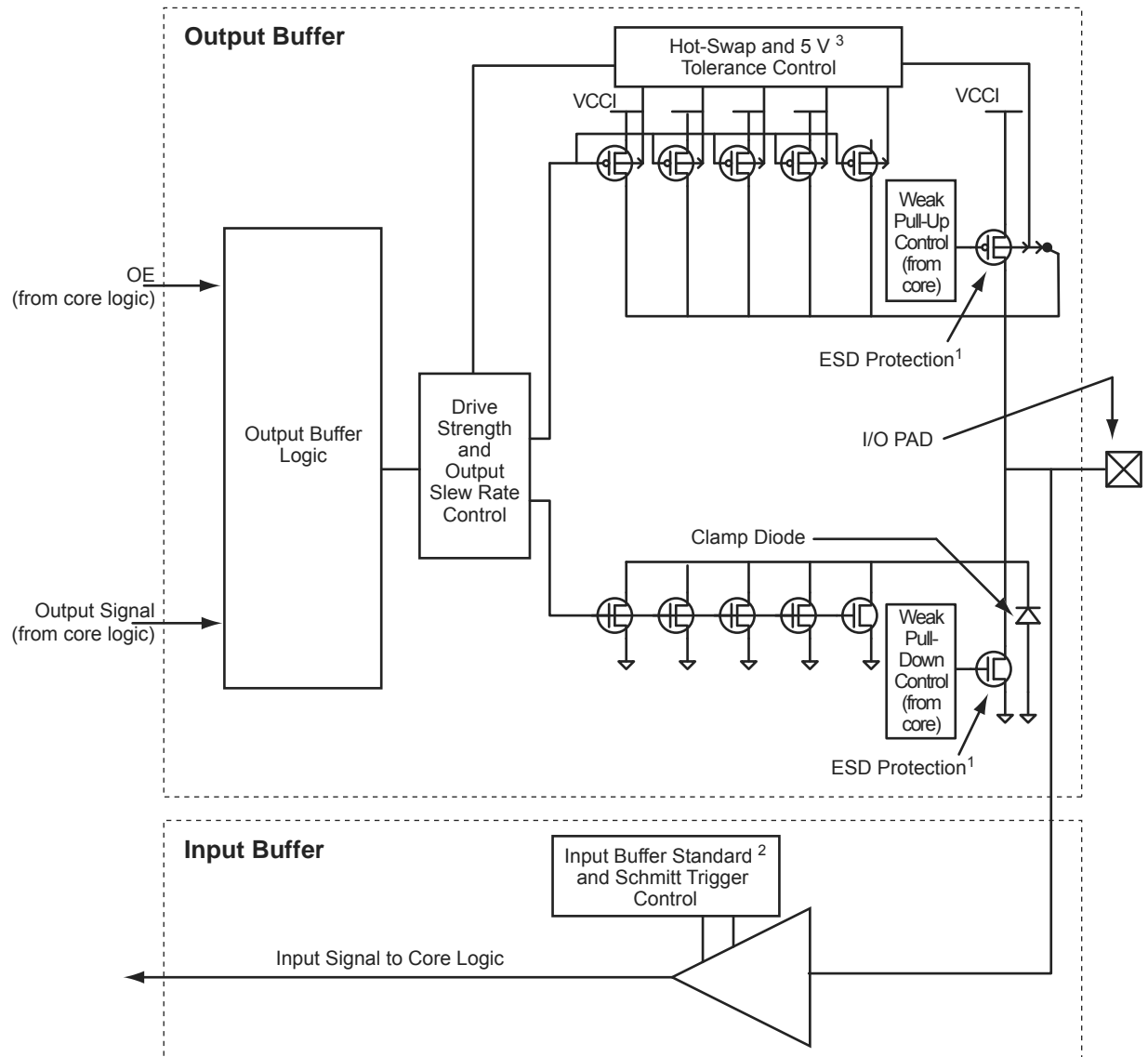
**Table 6-10 • RAM and FIFO Memory Block Consumption**

		Depth										
			256		512	1,024	2,048	4,096	8,192	16,384	32,768	65,536
			Two-Port	Dual-Port	Dual-Port	Dual-Port	Dual-Port	Dual-Port	Dual-Port	Dual-Port	Dual-Port	Dual-Port
Width	1	Number Block	1	1	1	1	1	1	2	4	8	16 × 1
		Configuration	Any	Any	Any	1,024 × 4	2,048 × 2	4,096 × 1	2 × (4,096 × 1) Cascade Deep	4 × (4,096 × 1) Cascade Deep	8 × (4,096 × 1) Cascade Deep	16 × (4,096 × 1) Cascade Deep
	2	Number Block	1	1	1	1	1	2	4	8	16	32
		Configuration	Any	Any	Any	1,024×4	2,048 × 2	2 × (4,096 × 1) Cascaded Wide	4 × (4,096 × 1) Cascaded 2 Deep and 2 Wide	8 × (4,096 × 1) Cascaded 4 Deep and 2 Wide	16 × (4,096 × 1) Cascaded 8 Deep and 2 Wide	32 × (4,096 × 1) Cascaded 16 Deep and 2 Wide
	4	Number Block	1	1	1	1	2	4	8	16	32	64
		Configuration	Any	Any	Any	1,024 × 4	2 × (2,048 × 2) Cascaded Wide	4 × (4,096 × 1) Cascaded Wide	4 × (4,096 × 1) Cascaded 2 Deep and 4 Wide	16 × (4,096 × 1) Cascaded 4 Deep and 4 Wide	32 × (4,096 × 1) Cascaded 8 Deep and 4 Wide	64 × (4,096 × 1) Cascaded 16 Deep and 4 Wide
	8	Number Block	1	1	1	2	4	8	16	32	64	
		Configuration	Any	Any	Any	2 × (1,024 × 4) Cascaded Wide	4 × (2,048 × 2) Cascaded Wide	8 × (4,096 × 1) Cascaded Wide	16 × (4,096 × 1) Cascaded 2 Deep and 8 Wide	32 × (4,096 × 1) Cascaded 4 Deep and 8 Wide	64 × (4,096 × 1) Cascaded 8 Deep and 8 Wide	
	9	Number Block	1	1	1	2	4	8	16	32		
		Configuration	Any	Any	Any	2 × (512 × 9) Cascaded Deep	4 × (512 × 9) Cascaded Deep	8 × (512 × 9) Cascaded Deep	16 × (512 × 9) Cascaded Deep	32 × (512 × 9) Cascaded Deep		
	16	Number Block	1	1	1	4	8	16	32	64		
		Configuration	256 × 18	256 × 18	256 × 18	4 × (1,024 × 4) Cascaded Wide	8 × (2,048 × 2) Cascaded Wide	16 × (4,096 × 1) Cascaded Wide	32 × (4,096 × 1) Cascaded 2 Deep and 16 Wide	32 × (4,096 × 1) Cascaded 4 Deep and 16 Wide		
	18	Number Block	1	2	2	4	8	18	32			
		Configuration	256 × 8	2 × (512 × 9) Cascaded Wide	2 × (512 × 9) Cascaded Wide	4 × (512 × 9) Cascaded 2 Deep and 2 Wide	8 × (512 × 9) Cascaded 4 Deep and 2 Wide	16 × (512 × 9) Cascaded 8 Deep and 2 Wide	16 × (512 × 9) Cascaded 16 Deep and 2 Wide			
	32	Number Block	2	4	4	8	16	32	64			
		Configuration	2 × (256 × 18) Cascaded Wide	4 × (512 × 9) Cascaded Wide	4 × (512 × 9) Cascaded Wide	8 × (1,024 × 4) Cascaded Wide	16 × (2,048 × 2) Cascaded Wide	32 × (4,096 × 1) Cascaded Wide	64 × (4,096 × 1) Cascaded 2 Deep and 32 Wide			
	36	Number Block	2	4	4	8	16	32				
		Configuration	2 × (256 × 18) Cascaded Wide	4 × (512 × 9) Cascaded Wide	4 × (512 × 9) Cascaded Wide	4 × (512 × 9) Cascaded 2 Deep and 4 Wide	16 × (512 × 9) Cascaded 4 Deep and 4 Wide	16 × (512 × 9) Cascaded 8 Deep and 4 Wide				
	64	Number Block	4	8	8	16	32	64				
		Configuration	4 × (256 × 18) Cascaded Wide	8 × (512 × 9) Cascaded Wide	8 × (512 × 9) Cascaded Wide	16 × (1,024 × 4) Cascaded Wide	32 × (2,048 × 2) Cascaded Wide	64 × (4,096 × 1) Cascaded Wide				
	72	Number Block	4	8	8	16	32					
		Configuration	4 × (256 × 18) Cascaded Wide	8 × (512 × 9) Cascaded Wide	8 × (512 × 9) Cascaded Wide	16 × (512 × 9) Cascaded Wide	16 × (512 × 9) Cascaded 4 Deep and 8 Wide					

Note: Memory configurations represented by grayed cells are not supported.

## I/O Bank Structure

Low power flash device I/Os are divided into multiple technology banks. The number of banks is device-dependent, supporting two, three, or four banks. Each bank has its own  $V_{CCI}$  power supply pin. Refer to Figure 7-2 on page 160 for more information.



Notes:

1. All NMOS transistors connected to the I/O pad serve as ESD protection.
2. See Table 7-2 on page 162 for available I/O standards.
3. 5 V tolerance requires external resistor.

**Figure 7-3 • Simplified I/O Buffer Circuitry**

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

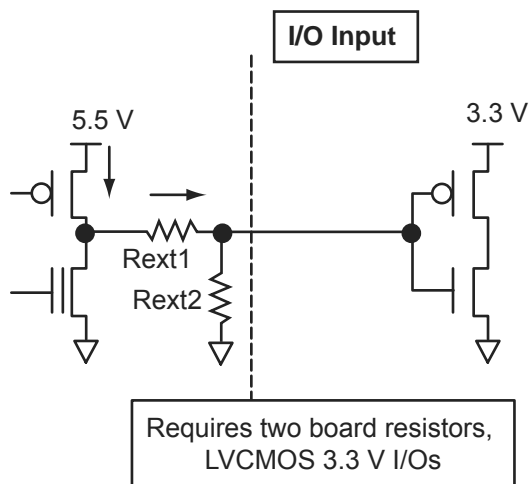


Figure 7-5 • Solution 1

## 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.

4. Right-click and then choose **Highlight VREF range**. All the pins covered by that VREF pin will be highlighted (Figure 8-14).
- 

---

**Figure 8-14 • VREF Range**

Using PinEditor or ChipPlanner, VREF pins can also be assigned (Figure 8-15).

---

---

**Figure 8-15 • Assigning VREF from PinEditor**

To unassign a VREF pin:

1. Select the pin to unassign.
2. Right-click and choose **Use Pin for VREF**. The check mark next to the command disappears. The VREF pin is now a regular pin.

Resetting the pin may result in unassigning I/O cores, even if they are locked. In this case, a warning message appears so you can cancel the operation.

After you assign the VREF pins, right-click a VREF pin and choose **Highlight VREF Range** to see how many I/Os are covered by that pin. To unhighlight the range, choose **Unhighlight All** from the **Edit** menu.

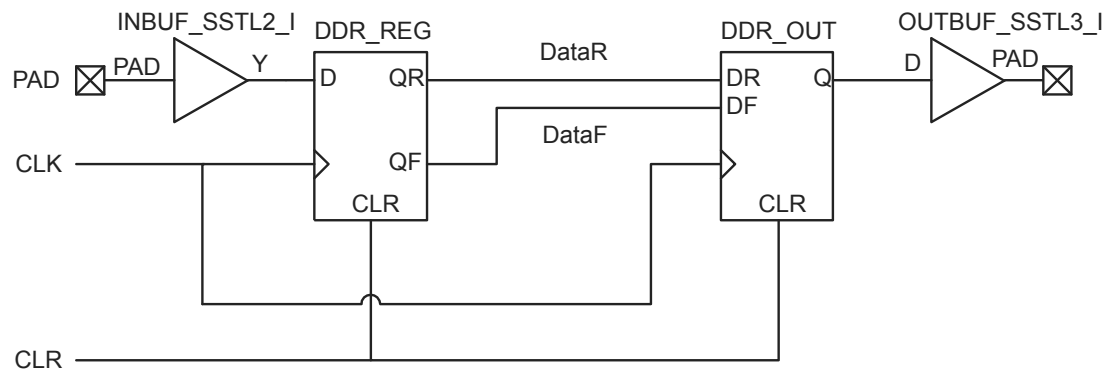
## 9 – DDR for Microsemi’s Low Power Flash Devices

### Introduction

The I/Os in Fusion, IGLOO, and ProASIC3 devices support Double Data Rate (DDR) mode. In this mode, new data is present on every transition (or clock edge) of the clock signal. This mode doubles the data transfer rate compared with Single Data Rate (SDR) mode, where new data is present on one transition (or clock edge) of the clock signal. Low power flash devices have DDR circuitry built into the I/O tiles. I/Os are configured to be DDR receivers or transmitters by instantiating the appropriate special macros (examples shown in Figure 9-4 on page 210 and Figure 9-5 on page 211) and buffers (DDR\_OUT or DDR\_REG) in the RTL design. This document discusses the options the user can choose to configure the I/Os in this mode and how to instantiate them in the design.

### Double Data Rate (DDR) Architecture

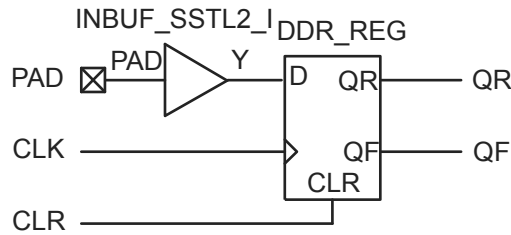
Low power flash devices support 350 MHz DDR inputs and outputs. In DDR mode, new data is present on every transition of the clock signal. Clock and data lines have identical bandwidths and signal integrity requirements, making them very efficient for implementing very high-speed systems. High-speed DDR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



**Figure 9-1 • DDR Support in Low Power Flash Devices**



## DDR Input Register



**Figure 9-5 • DDR Input Register (SSTL2 Class I)**

The corresponding structural representations, as generated by SmartGen, are shown below:

### Verilog

```

module DDR_InBuf_SSTL2_I (PAD, CLR, CLK, QR, QF);

input  PAD, CLR, CLK;
output QR, QF;

wire Y;

    INBUF_SSTL2_I INBUF_SSTL2_I_0_inst(.PAD(PAD),.Y(Y));
    DDR_REG DDR_REG_0_inst(.D(Y),.CLK(CLK),.CLR(CLR),.QR(QR),.QF(QF));

endmodule
  
```

### VHDL

```

library ieee;
use ieee.std_logic_1164.all;
--The correct library will be inserted automatically by SmartGen
library proasic3; use proasic3.all;
--library fusion; use fusion.all;
--library igloo; use igloo.all;

entity DDR_InBuf_SSTL2_I is
    port(PAD, CLR, CLK : in std_logic;  QR, QF : out std_logic) ;
end DDR_InBuf_SSTL2_I;

architecture DEF_ARCH of  DDR_InBuf_SSTL2_I is

    component INBUF_SSTL2_I
        port(PAD : in std_logic := 'U'; Y : out std_logic) ;
    end component;

    component DDR_REG
        port(D, CLK, CLR : in std_logic := 'U'; QR, QF : out std_logic) ;
    end component;

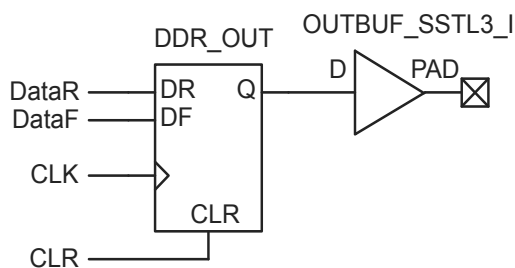
    signal Y : std_logic ;

begin

    INBUF_SSTL2_I_0_inst : INBUF_SSTL2_I
    port map(PAD => PAD, Y => Y);
    DDR_REG_0_inst : DDR_REG
    port map(D => Y, CLK => CLK, CLR => CLR, QR => QR, QF => QF);

end DEF_ARCH;
  
```

## DDR Output Register



**Figure 9-6 • DDR Output Register (SSTL3 Class I)**

### Verilog

```
module DDR_OutBuf_SSTL3_I(DataR,DataF,CLR,CLK,PAD);

input  DataR, DataF, CLR, CLK;
output PAD;

wire Q, VCC;

    VCC VCC_1_net(.Y(VCC));
    DDR_OUT DDR_OUT_0_inst(.DR(DataR),.DF(DataF),.CLK(CLK),.CLR(CLR),.Q(Q));
    OUTBUF_SSTL3_I OUTBUF_SSTL3_I_0_inst(.D(Q),.PAD(PAD));

endmodule
```

### VHDL

```
library ieee;
use ieee.std_logic_1164.all;
library proasic3; use proasic3.all;

entity DDR_OutBuf_SSTL3_I is
    port(DataR, DataF, CLR, CLK : in std_logic;  PAD : out std_logic) ;
end DDR_OutBuf_SSTL3_I;

architecture DEF_ARCH of  DDR_OutBuf_SSTL3_I is

    component DDR_OUT
        port(DR, DF, CLK, CLR : in std_logic := 'U'; Q : out std_logic) ;
    end component;

    component OUTBUF_SSTL3_I
        port(D : in std_logic := 'U'; PAD : out std_logic) ;
    end component;

    component VCC
        port( Y : out std_logic);
    end component;

    signal Q, VCC_1_net : std_logic ;

begin

    VCC_2_net : VCC port map(Y => VCC_1_net);
    DDR_OUT_0_inst : DDR_OUT
        port map(DR => DataR, DF => DataF, CLK => CLK, CLR => CLR, Q => Q);
    OUTBUF_SSTL3_I_0_inst : OUTBUF_SSTL3_I
        port map(D => Q, PAD => PAD);

end DEF_ARCH;
```

---

**Figure 11-19 • FlashLock Pass Key, Previously Programmed Devices**

It is important to note that when the security settings need to be updated, the user also needs to select the **Security settings** check box in Step 1, as shown in Figure 11-10 on page 248 and Figure 11-11 on page 248, to modify the security settings. The user must consider the following:

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

## Advanced Options

As mentioned, there may be applications where more complicated security settings are required. The “Custom Security Levels” section in the *FlashPro User's Guide* describes different advanced options available to aid the user in obtaining the best available security settings.



---

## 14 – Microprocessor Programming of Microsemi's Low Power Flash Devices

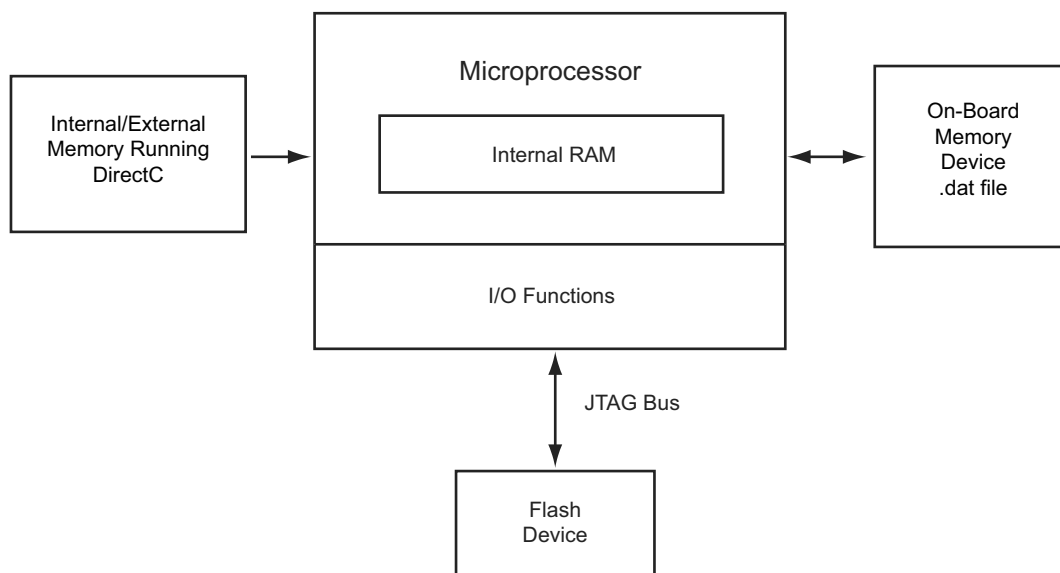
---

### Introduction

The Fusion, IGLOO, and ProASIC3 families of flash FPGAs support in-system programming (ISP) with the use of a microprocessor. Flash-based FPGAs store their configuration information in the actual cells within the FPGA fabric. SRAM-based devices need an external configuration memory, and hybrid nonvolatile devices store the configuration in a flash memory inside the same package as the SRAM FPGA. Since the programming of a true flash FPGA is simpler, requiring only one stage, it makes sense that programming with a microprocessor in-system should be simpler than with other SRAM FPGAs. This reduces bill-of-materials costs and printed circuit board (PCB) area, and increases system reliability.

Nonvolatile flash technology also gives the low power flash devices the advantage of a secure, low power, live-at-power-up, and single-chip solution. Low power flash devices are reprogrammable and offer time-to-market benefits at an ASIC-level unit cost. These features enable engineers to create high-density systems using existing ASIC or FPGA design flows and tools.

This document is an introduction to microprocessor programming only. To explain the difference between the options available, user's guides for DirectC and STAPL provide more detail on implementing each style.



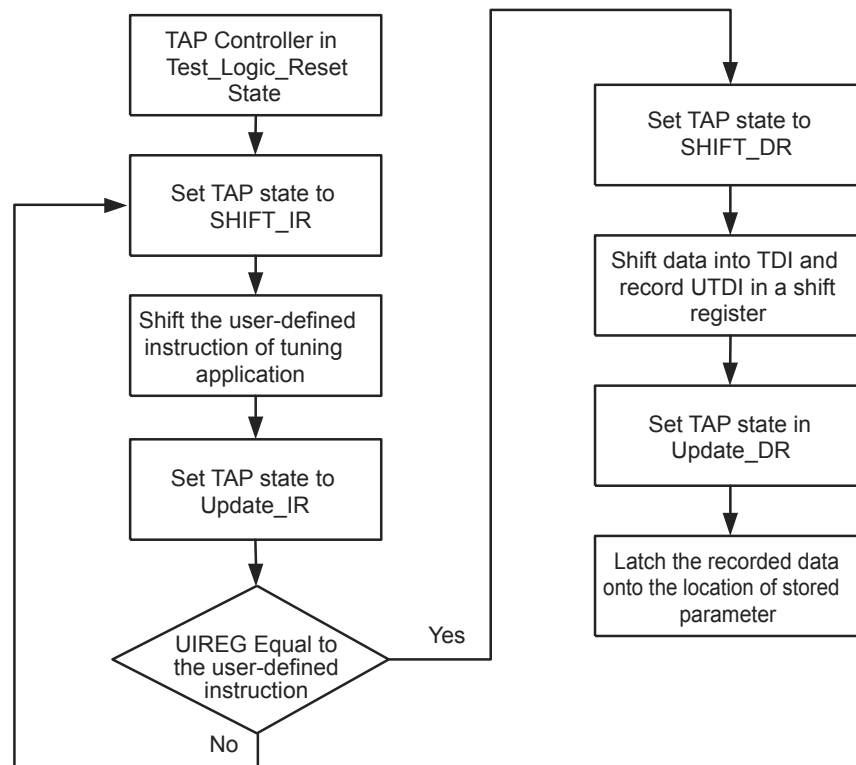
---

**Figure 14-1 • ISP Using Microprocessor**

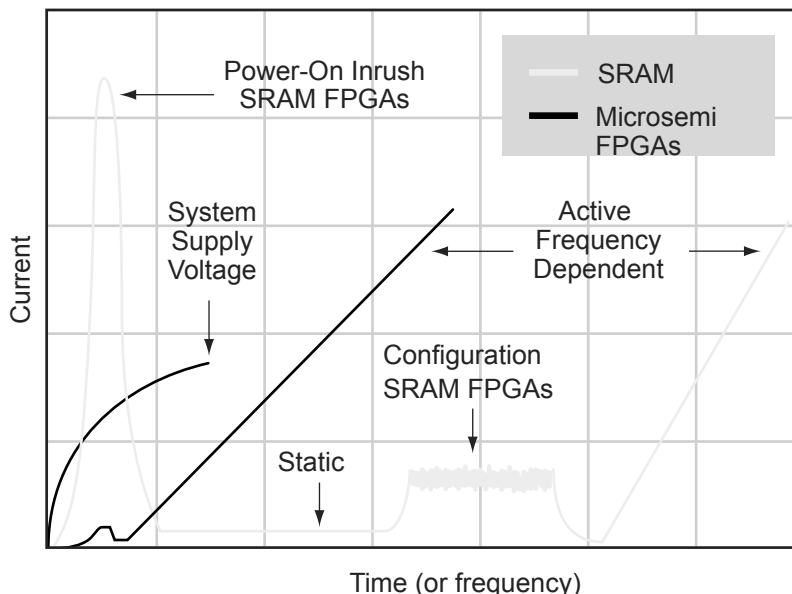
## Fine Tuning

In some applications, design constants or parameters need to be modified after programming the original design. The tuning process can be done using the UJTAG tile without reprogramming the device with new values. If the parameters or constants of a design are stored in distributed registers or embedded SRAM blocks, the new values can be shifted onto the JTAG TAP Controller pins, replacing the old values. The UJTAG tile is used as the “bridge” for data transfer between the JTAG pins and the FPGA VersaTiles or SRAM logic. Figure 16-5 shows a flow chart example for fine-tuning application steps using the UJTAG tile.

In Figure 16-5, the TMS signal sets the TAP Controller state machine to the appropriate states. The flow mainly consists of two steps: a) shifting the defined instruction and b) shifting the new data. If the target parameter is constantly used in the design, the new data can be shifted into a temporary shift register from UTDI. The UDRSH output of UJTAG can be used as a shift-enable signal, and UDRCK is the shift clock to the shift register. Once the shift process is completed and the TAP Controller state is moved to the Update\_DR state, the UDRUPD output of the UJTAG can latch the new parameter value from the temporary register into a permanent location. This avoids any interruption or malfunctioning during the serial shift of the new value.



**Figure 16-5 • Flow Chart Example of Fine-Tuning an Application Using UJTAG**



**Figure 17-1 • Types of Power Consumption in SRAM FPGAs and Microsemi Nonvolatile FPGAs**

## Transient Current on VCC

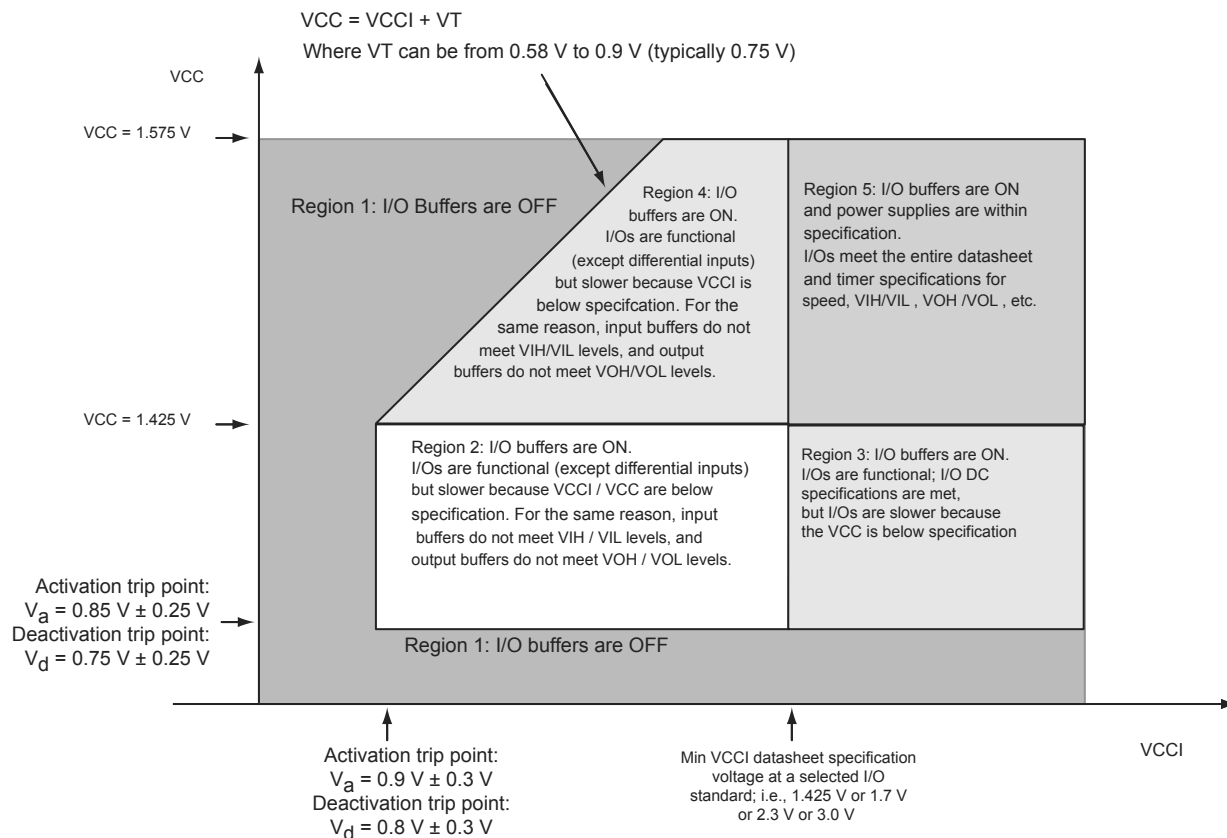
The characterization of the transient current on VCC is performed on nearly all devices within the IGLOO, ProASIC3L, and ProASIC3 families. A sample size of five units is used from each device family member. All the device I/Os are internally pulled down while the transient current measurements are performed. For ProASIC3 devices, the measurements at typical conditions show that the maximum transient current on VCC, when the power supply is powered at ramp-rates ranging from 15 V/ms to 0.15 V/ms, does not exceed the maximum standby current specified in the device datasheets. Refer to the DC and Switching Characteristics chapters of the *ProASIC3 Flash Family FPGAS* datasheet and *ProASIC3E Flash Family FPGAs* datasheet for more information.

Similarly, IGLOO, IGLOO nano, IGLOO PLUS, and ProASIC3L devices exhibit very low transient current on VCC. The transient current does not exceed the typical operating current of the device while in active mode. For example, the characterization of AGL600-FG256 V2 and V5 devices has shown that the transient current on VCC is typically in the range of 1–5 mA.

## Transient Current on VCCI

The characterization of the transient current on VCCI is performed on devices within the IGLOO, IGLOO nano, IGLOO PLUS, ProASIC3, ProASIC3 nano, and ProASIC3L groups of devices, similarly to VCC transient current measurements. For ProASIC3 devices, the measurements at typical conditions show that the maximum transient current on VCCI, when the power supply is powered at ramp-rates ranging from 33 V/ms to 0.33 V/ms, does not exceed the maximum standby current specified in the device datasheet. Refer to the DC and Switching Characteristics chapters of the *ProASIC3 Flash Family FPGAS* datasheet and *ProASIC3E Flash Family FPGAs* datasheet for more information.

Similarly, IGLOO, IGLOO PLUS, and ProASIC3L devices exhibit very low transient current on VCCI. The transient current does not exceed the typical operating current of the device while in active mode. For example, the characterization of AGL600-FG256 V2 and V5 devices has shown that the transient current on VCCI is typically in the range of 1–2 mA.



**Figure 17-4 • I/O State as a Function of  $VCCI$  and  $VCC$  Voltage Levels for IGLOO V5, IGLOO nano V5, IGLOO PLUS V5, ProASIC3L, and ProASIC3 Devices Running at  $VCC = 1.5 \text{ V} \pm 0.075 \text{ V}$**



- architecture of user nonvolatile 117
- configuration 120
- custom serialization 129
- design flow 124
- generation 125
- programming and accessing 122
- programming file 127
- programming files 267
- SmartGen 126

FlashROM read-back 305

## G

- global architecture 31
- global buffers
  - no programmable delays 64
  - with PLL function 67
  - with programmable delays 64
- global macros
  - Synplicity 50
- globals
  - designer flow 53
  - networks 58
  - spines and rows 41

## H

- HLD code
  - instantiating 192
- hot-swap 167
- hot-swapping 317

## I

- I/O banks
  - standards 40
  - standards compatibility 162
- I/O standards 77
  - global macros 46
  - single-ended 166
- I/Os
  - assigning technologies 198
  - assignments defined in PDC file 193
  - automatically assigning 202
  - behavior at power-up/-down 311
  - board-level considerations 181
  - buffer schematic cell 191
  - cell architecture 207
  - configuration with SmartGen 188
  - features 163, 164, 167
  - global, naming 35
  - manually assigning technologies 198
  - nano standard 162
  - register combining 174
  - simplified buffer circuitry 165
  - software support 177
  - software-controlled attributes 187
  - user I/O assignment flow chart 185
  - user naming convention 178
  - wide range support 166

- ISP 223, 224
  - architecture 261
  - board-level considerations 271
  - circuit 277
  - microprocessor 283

## J

- JTAG 1532 261
- JTAG interface 285

## L

- layout
  - device-specific 78
- LTC3025 linear voltage regulator 277

## M

- MAC validation/authentication 288
- macros
  - CLKBUF 77
  - CLKBUF\_LVDS/LVPECL 77
  - CLKDLY 65, 73
  - FIFO4KX18 141
  - PLL 73
  - PLL macro signal descriptions 68
  - RAM4K9 137
  - RAM512X18 139
  - supported basic RAM macros 136
  - UJTAG 299
- MCU FPGA programming model 286
- memory availability 146
- memory blocks 135
- microprocessor programming 283
- Microsemi SoC Products Group
  - email 321
  - web-based technical support 321
  - website 321

## O

- OTP 223
- output slew rate 175

## P

- PDC
  - global promotion and demotion 51
- place-and-route 193
- PLL
  - behavior at brownout condition 315
  - configuration bits 90
  - core specifications 84
  - dynamic PLL configuration 87
  - functional description 85
  - power supply decoupling scheme 112
- PLL block signals 68
- PLL macro block diagram 69
- product support
  - customer service 321