Spartan-6 FPGA Power Management

User Guide

UG394 (v1.0) May 18, 2010

XILINX®

Xilinx is disclosing this user guide, manual, release note, and/or specification (the “Documentation”) to you solely for use in the development of designs to operate with Xilinx hardware devices. You may not reproduce, distribute, republish, download, display, post, or transmit the Documentation in any form or by any means including, but not limited to, electronic, mechanical, photocopying, recording, or otherwise, without the prior written consent of Xilinx. Xilinx expressly disclaims any liability arising out of your use of the Documentation. Xilinx reserves the right, at its sole discretion, to change the Documentation without notice at any time. Xilinx assumes no obligation to correct any errors contained in the Documentation, or to advise you of any corrections or updates. Xilinx expressly disclaims any liability in connection with technical support or assistance that may be provided to you in connection with the Information.

THE DOCUMENTATION IS DISCLOSED TO YOU “AS-IS” WITH NO WARRANTY OF ANY KIND. XILINX MAKES NO OTHER WARRANTIES, WHETHER EXPRESS, IMPLIED, OR STATUTORY, REGARDING THE DOCUMENTATION, INCLUDING ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT OF THIRD-PARTY RIGHTS. IN NO EVENT WILL XILINX BE LIABLE FOR ANY CONSEQUENTIAL, INDIRECT, EXEMPLARY, SPECIAL, OR INCIDENTAL DAMAGES, INCLUDING ANY LOSS OF DATA OR LOST PROFITS, ARISING FROM YOUR USE OF THE DOCUMENTATION.

© Copyright 2010 Xilinx, Inc. XILINX, the Xilinx logo, Virtex, Spartan, ISE, and other designated brands included herein are trademarks of Xilinx in the United States and other countries. All other trademarks are the property of their respective owners.

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
05/18/10	1.0	Initial Xilinx release.

Table of Contents

Chapter 2: Voltage Supplies

Chapter 3: Lower-Power Spartan-6 LX Devices

Chapter 4: Power-On and Power-Down Behavior Including Hibernate

Chapter 5: Power Estimation

About This Guide

This document provides information on the various hardware methods of power management in Spartan-6 FPGAs, primarily focusing on the suspend mode. Other power management topics include the lower-power Spartan-6 LX devices (-1L) and the programmable V_{CCAUX} level available in all Spartan-6 devices. In addition, more detail is provided on the power rails, including hot swap and hibernate (power-off) options.

Guide Contents

This user guide contains the following chapters:

- Chapter 1, Power Management With Suspend Mode
- Chapter 2, Voltage Supplies
- Chapter 3, Lower-Power Spartan-6 LX Devices
- Chapter 4, Power-On and Power-Down Behavior Including Hibernate
- Chapter 5, Power Estimation

Additional Documentation

The following documents are also available for download at http://www.xilinx.com/support/documentation/spartan-6.htm.

- Spartan-6 Family Overview

  This overview outlines the features and product selection of the Spartan-6 FPGA Configurable Logic Blocks User Guide

  This guide describes the capabilities of the configurable logic blocks (CLBs) available in all Spartan-6 devices. family.

- Spartan-6 FPGA Data Sheet: DC and Switching Characteristics

  This data sheet contains the DC and switching characteristic specifications for the Spartan-6 family.

- Spartan-6 FPGA Packaging and Pinout Specifications

  This specification includes the tables for device/package combinations and maximum I/Os, pin definitions, pinout tables, pinout diagrams, mechanical drawings, and thermal specifications.

- Spartan-6 FPGA Configuration User Guide

  This all-encompassing configuration guide includes chapters on configuration interfaces (serial and parallel), multi-bitstream management, bitstream encryption, boundary-scan and JTAG configuration, and reconfiguration techniques.

- Spartan-6 FPGA SelectIO Resources User Guide

  This guide describes the SelectIO™ resources available in all Spartan-6 devices.

- Spartan-6 FPGA Clocking Resources User Guide

  This guide describes the clocking resources available in all Spartan-6 devices, including the DCMs and PLLs.

- Spartan-6 FPGA Block RAM Resources User Guide

  This guide describes the Spartan-6 device block RAM capabilities.

- Spartan-6 FPGA Configurable Logic Blocks User Guide

  This guide describes the capabilities of the configurable logic blocks (CLBs) available in all Spartan-6 devices.

- Spartan-6 FPGA GTP Transceivers User Guide

  This guide describes the GTP transceivers available in the Spartan-6 LXT FPGAs.

- Spartan-6 FPGA DSP48A1 Slice User Guide

  This guide describes the architecture of the DSP48A1 slice in Spartan-6 FPGAs and provides configuration examples.

- Spartan-6 FPGA Memory Controller User Guide

  This guide describes the Spartan-6 FPGA memory controller block, a dedicated embedded multi-port memory controller that greatly simplifies interfacing Spartan-6 FPGAs to the most popular memory standards.

- Spartan-6 FPGA PCB Design and Pin Planning Guide

  This guide provides information on PCB design for Spartan-6 devices, with a focus on strategies for making design decisions at the PCB and interface level.

Additional Support Resources

To search the database of silicon and software questions and answers or to create a technical support case in WebCase, see the Xilinx website at:

http://www.xilinx.com/support.

XILINX®

Chapter 1

Power Management With Suspend Mode

Introduction

Some applications require the lowest possible system cost or highest performance, and other applications require the lowest possible standby power. Spartan®-6 FPGAs offer low-power options to balance these cost and performance trade-offs.

The Spartan-6 family offers the suspend mode, an advanced static power-management feature, which reduces FPGA power consumption while retaining the FPGA's configuration data and maintaining the design. The device can quickly enter and exit suspend mode as required in an application.

Differences from Extended Spartan-3A Family

The suspend mode in Spartan-6 FPGAs is a superset of the suspend feature in the Extended Spartan-3A FPGAs. Two new enhancements include multi-pin wake-up and suspend synchronization.

Multi-Pin Wake-up

The multi-pin wake-up feature allows the FPGA to monitor for a wake-up signal on up to eight pins. In the Extended Spartan-3A family, monitoring was limited to the SUSPEND pin itself. Multi-pin wake-up also allows a number of independent sources to trigger the FPGA to return to the normal application.

Suspend Synchronization

The Spartan-6 FPGA primitive, SUSPEND_SYNC, enables the synchronization of the suspend action with the application design. In the Extended Spartan-3A family, the suspend mode activation begins immediately upon asserting the SUSPEND pin. The Spartan-6 FPGA SUSPEND_SYNC primitive allows the application design to acknowledge a suspend request, thereby allowing the application to finish necessary functions prior to entering the suspend mode.

Suspend Features

The significant features and benefits of the suspend mode:

- Quickly and easily puts the FPGA into a static condition, eliminating most active current.
- Reduces quiescent current by 40% or more.
- Retains FPGA configuration data and the state of the FPGA application during suspend mode.
- Fast, programmable FPGA wake-up time from suspend mode.
- Individual control on each user-I/O pin to define pin behavior while in suspend mode.
- Activated externally by the system using a single dedicated control pin (SUSPEND).
- Indicates the present suspend mode status using the AWAKE pin.
- Awakens an FPGA in suspend mode using any of eight SUSPEND control pins (SCP).
- SUSPEND_SYNC primitive to acknowledge a ready state prior to entering suspend mode.

Design Steps

To use the suspend feature:

- Enable the Suspend Feature and Glitch Filtering, page 14
- Define the Multi-Pin Wake-Up Feature and Pins, page 15
- Define the I/O Behavior During Suspend Mode, page 15
- Implement steps to maintain application data during suspend mode (SUSPEND_SYNC) (see Design Requirements to Maintain Application Data, page 17)
- Define the Suspend Mode Wake-Up Timing Controls, page 17
- Define the AWAKE Pin Behavior when Suspend Feature is Enabled, page 21

Entering Suspend Mode

Figure 1-1 is a block diagram of the FPGA entering suspend mode. Figure 1-2, page 10 shows example waveforms.

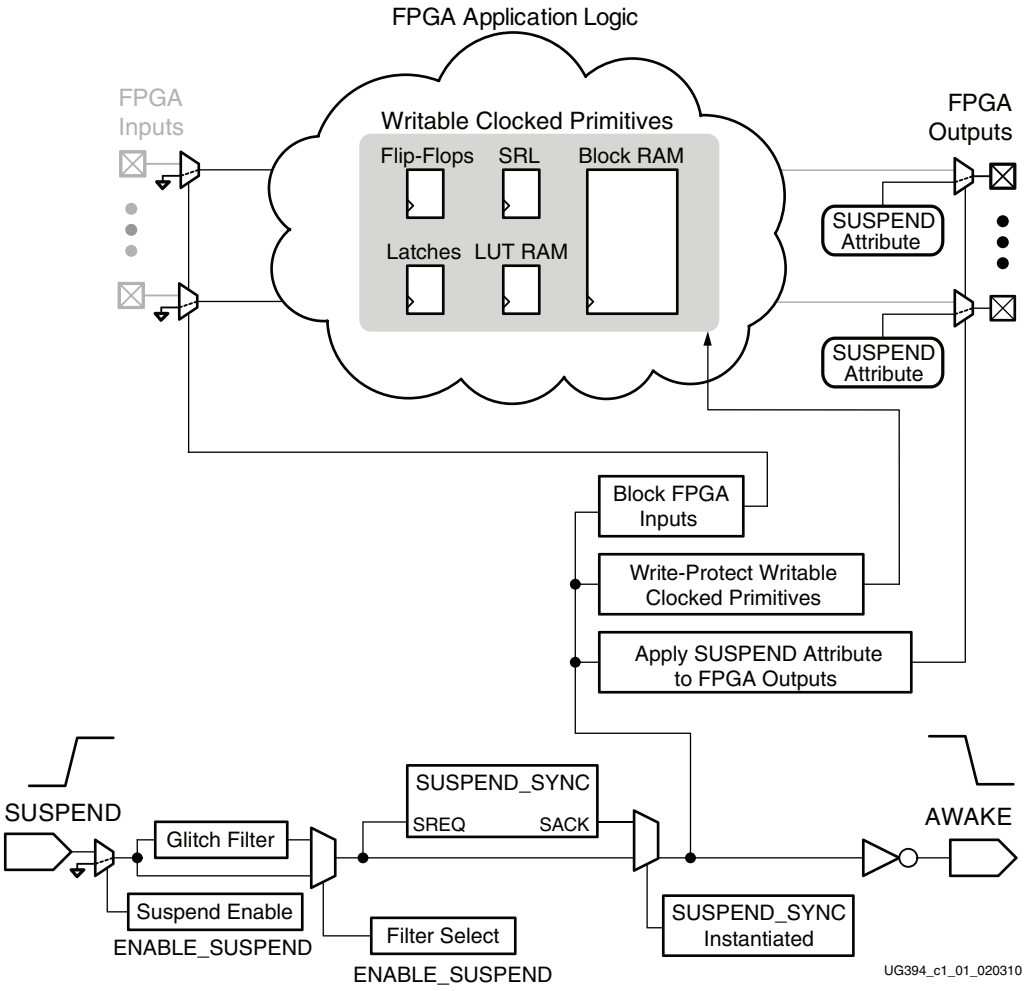

Figure 1-1: **Entering Suspend Mode**

The FPGA can only enter suspend mode if enabled in the configuration bitstream (see Enable the Suspend Feature and Glitch Filtering, page 14). The SUSPEND pin must be Low during power up and configuration. Once enabled through the bitstream, and the SUSPEND_SYNC primitive is not present in the design, when the SUSPEND pin is asserted, the FPGA unconditionally and quickly enters suspend mode.

If the SUSPEND_SYNC primitive is present in the design, the FPGA does not enter suspend mode until the suspend-acknowledge signal (SACK) is asserted. After the SUSPEND pin is asserted, the SREQ port of the SUSPEND_SYNC primitive transitions High. This can be used in the design to initiate any functions that must be completed prior to the FPGA entering suspend mode. When these functions are complete, drive the SACK port High.

After the FPGA enters suspend mode, all nonessential FPGA functions are shut down to minimize power dissipation. The FPGA retains all configuration data while in suspend mode. After entering suspend mode, all writable clocked primitives are write-protected against spurious write operations, and all FPGA inputs and interconnects are shut down. This allows the design state to be held static during suspend mode. If a specific design state must be maintained, see Design Requirements to Maintain Application Data, page 17.

Each FPGA output pin or bidirectional I/O pin assumes its defined suspend mode behavior, which is described as part of the FPGA design using a SUSPEND attribute.

The AWAKE pin goes Low, indicating that the FPGA is in suspend mode. The DONE pin remains High while the FPGA is in suspend mode because the FPGA configuration data is not lost.

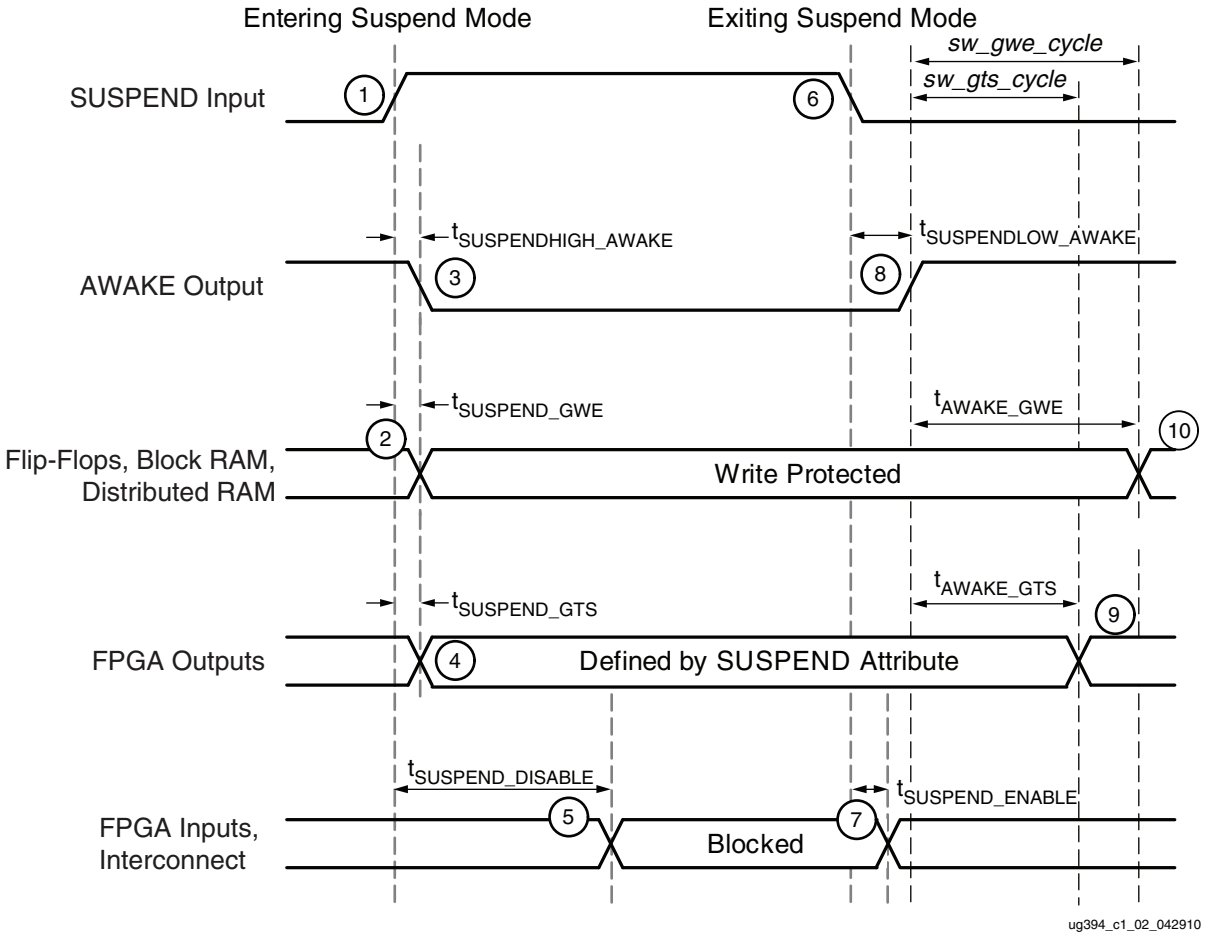

Figure 1-2: **Suspend Mode Waveforms (Entering and Exiting)**

This section details the waveform notes in Figure 1-2.

Entering Suspend in Figure 1-2

1. An external signal drives the FPGA's SUSPEND pin High, unconditionally forcing the FPGA into the power-saving suspend mode (if SUSPEND_SYNC is not used). When SUSPEND_SYNC is used, this phase does not complete until the SACK port of the SUSPEND_SYNC primitive is asserted. Data values are captured for I/O pins with a SUSPEND attribute set to DRIVE_LAST_VALUE; however, this value is not presented until Step 4.
2. In response to the SUSPEND input going High or SACK assertion on the SUSPEND_SYNC primitive, and after a delay of $t_{SUSPEND\_GWE}$, the FPGA write protects and preserves the states of all clocked primitives. The states of all flip-flops, block RAM, distributed RAM (LUT RAM), shift registers (SRL), and I/O latches are preserved during suspend mode.

3. After a delay of $t_{SUSPENDHIGH\_AWAKE}$, the FPGA drives the AWAKE output Low to indicate that it is entering suspend mode.
4. After a delay of $t_{SUSPEND\_GTS}$, the FPGA switches the normal behavior of all outputs over to the suspend mode behavior defined by the SUSPEND attribute assigned to each I/O. See Define the I/O Behavior During Suspend Mode, page 15.
5. After a delay of $t_{SUSPEND\_DISABLE}$, FPGA inputs are blocked and the interconnect shut off (High) to prevent any internal switching activity.

Exiting Suspend in Figure 1-2

6. The system drives the FPGA's SUSPEND input Low, causing the FPGA to exit suspend mode. If using multi-pin wake-up mode, the system first drives the FPGA's SUSPEND input LOW, then drives any of the enabled multi-pin wake-up pins High, causing the FPGA to exit suspend mode.
7. The FPGA releases the inputs and interconnect after a delay of $t_{SUSPEND\_ENABLE}$, allowing signals to propagate internally. There is no danger of corrupting the internal state because all clocked primitives are still write protected.
8. After a delay of $t_{SUSPENDLOW\_AWAKE}$ or t_{SCP_AWAKE}, the FPGA asserts the AWAKE signal with the bitstream option **drive_awake:yes**. If the option is **drive_awake:no**, then the FPGA releases AWAKE to become an open-drain output. In this case, an external pull-up resistor is required or an external signal must drive AWAKE High before the FPGA continues to awaken. All subsequent timing is measured from when the AWAKE output transitions High. If multiple FPGAs are waking up and need to be synchronized, set **drive_awake:no** in each and then use an external pull-up resistor to synchronize the AWAKE pins. If other devices are waking up and the FPGA(s) need to wait, set **drive_awake:no** and use an external signal to control the AWAKE pin and drive it High once the rest of the system is ready.
9. After a delay of t_{AWAKE_GTS}, the FPGA switches output behavior from the specified SUSPEND attribute to the function specified in the FPGA application. The timing of this switch-over is controlled by the suspend/wake **sw_gts_cycle** bitstream generation setting, which defines when the FPGA's internal global three-state (GTS) control is released. After the specified number of clock cycles, the outputs are active according to the normal FPGA application. By default, the outputs are enabled four clock cycles after AWAKE goes High. The outputs are generally released before the clocked primitives to allow signals to propagate out of the FPGA.
10. After a delay of t_{AWAKE_GWE}, the writable, clocked primitives are released according to the suspend/wake **sw_gwe_cycle** bitstream generator setting, which defines when the FPGA's internal global write enable (GWE) control is asserted. After the specified cycle, it is again possible to write to flip-flops, block RAM, distributed RAM (LUT RAM), shift registers (SRL), and I/O latches. By default, the clocked primitives are released five clock cycles after AWAKE transitions High. The write-protect lock should be held until after outputs are enabled.

Exiting Suspend Mode

There are four possible ways to exit suspend mode in a powered system:

- Drive the SUSPEND input Low, exiting suspend mode.
- If multi-pin wake-up mode is enabled, drive the SUSPEND input Low and then assert any one of the user enabled SCP pins.
- Pulse the PROGRAM_B input Low to reset the FPGA and cause the FPGA to reprogram.
- Power cycle the FPGA, causing the FPGA to reprogram.

The block diagram in Figure 1-3 shows how to exit suspend mode using the SUSPEND pin.

When SUSPEND transitions Low, the FPGA automatically re-enables all inputs and interconnects after a delay of $t_{SUSPEND\_ENABLE}$. If using multi-pin wake-up mode, SUSPEND must first transition Low, then when any of the user enabled SCP pins for multi-pin wake up mode transition High, the FPGA re-enables all inputs and interconnects after a delay of $t_{SUSPEND\_ENABLE}$.

When enabled in the FPGA bitstream, all flip-flops are optionally globally set or reset according to the FPGA design description. By default, the flip-flops are not globally set or reset, which preserves the state of the FPGA application from the beginning of suspend mode.

The remaining wake-up process depends on two user-programmable timers which define when FPGA outputs are re-enabled and when the write-protect lock is released from all writable clocked primitives. These timers begin after the AWAKE pin is High. The wake-up timing clock source is also programmable.

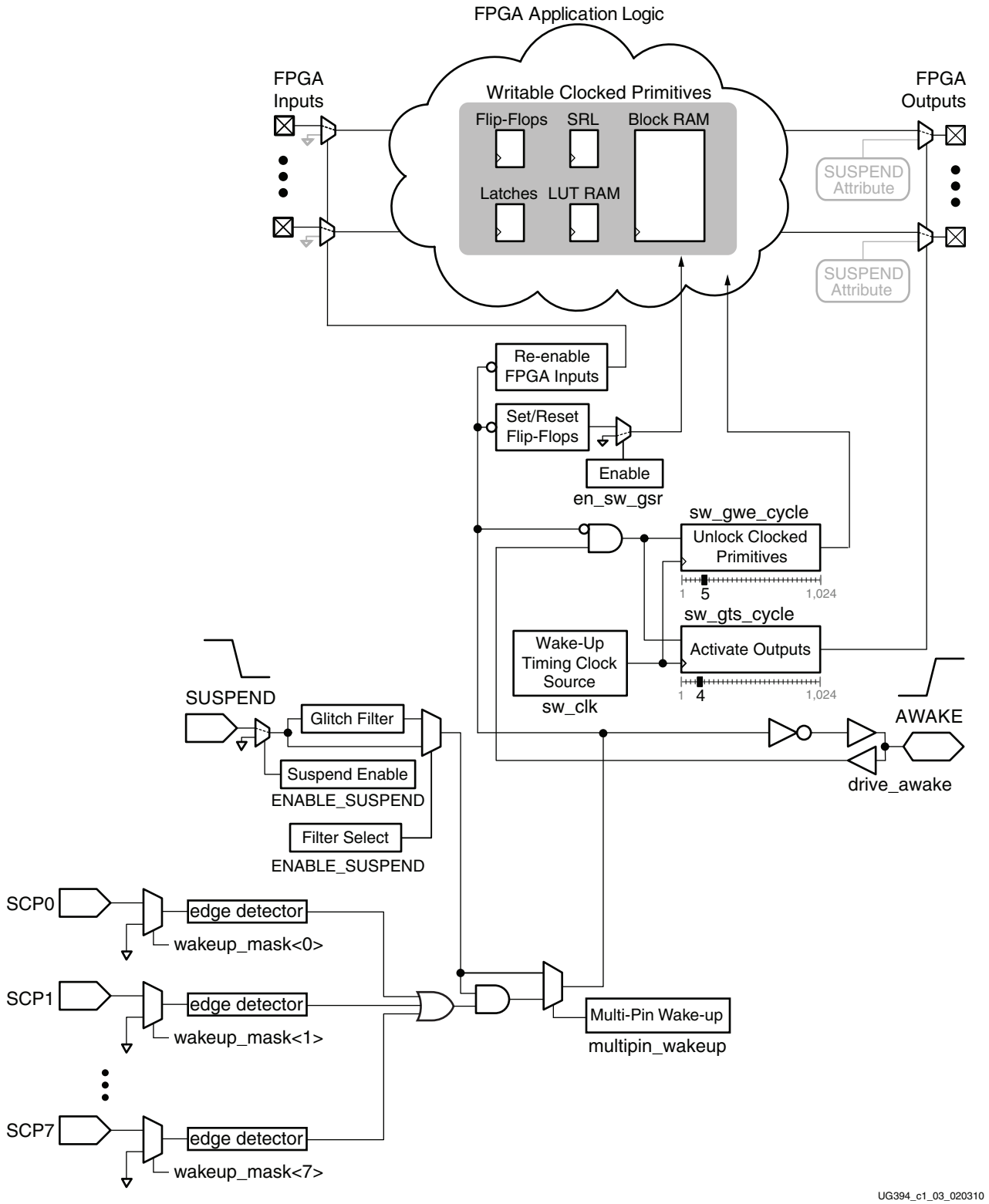

Figure 1-3: **Exiting Suspend Mode**

PROGRAM_B Programming Pin Always Overrides Suspend Mode

Pulsing the PROGRAM_B programming pin Low always overrides suspend mode and forces the FPGA to restart configuration. Power-cycling the FPGA also restarts configuration. If the SUSPEND input remains High, the device re-enters suspend mode after finishing configuration.

Enable the Suspend Feature and Glitch Filtering

Before it can be used, the suspend power-saving feature must first be enabled in the FPGA bitstream. By default, the suspend feature is disabled and driving the SUSPEND pin has no effect. The suspend feature is enabled using the user constraints file (UCF), or through a bitstream generator (BitGen) option.

User Constraints File Enable

Suspend mode is enabled and the SUSPEND input glitch filter option is defined using a CONFIG statement in a UCF. Table 1-1 shows the available options. This is the recommended method for enabling suspend mode as this attribute also automatically reserves the AWAKE pin.

```
Config ENABLE_SUSPEND = "FILTERED" ;
```

Table 1-1: **Available Options for the ENABLE_SUSPEND Attribute**

Option	Suspend Mode	SUSPEND Pin Filter	AWAKE Pin
NO	Suspend mode is disabled	Not applicable. Connect SUSPEND pin to GND.	Available as a user I/O pin in the FPGA application.
FILTERED	Suspend mode is enabled	Glitch filter is enabled.	AWAKE status indicator.
UNFILTERED		Glitch filter is bypassed.	

Bitstream Generator

Setting the `en_suspend` bitstream option is an alternate way to enable the suspend mode. However, this method is not recommended because it does not automatically reserve the AWAKE pin in the application.

```
bitgen -g en_suspend:Yes
```

The following option enables the glitch filter on the SUSPEND pin.

```
bitgen -g suspend_filter:Yes
```

Define the Multi-Pin Wake-Up Feature and Pins

The multi-pin wake-up feature is not required to use the suspend mode feature. If multi-pin wake-up is not enabled, suspend mode is enabled and disabled using just the SUSPEND pin. Multi-pin wake-up is enabled using a BitGen option.

```
bitgen -g multipin_wakeup:Yes
```

If multi-pin wake-up is enabled, select which pins are monitored for a rising edge to bring the FPGA out of suspend mode. Eight SCP pins are used for the multi-pin wake-up feature. Select from one to eight of these pins to monitor. The SCP pins are dual-purpose user I/O pins and can be used as general-purpose I/O independent of the suspend options. Any pins that are not used can be masked out as inputs to the multi-pin wake-up. The option accepts two hex values for the mask. A value of `FF` enables all SCP pins, `0F` enables `SCP<3..0>`.

```
bitgen -g wakeup_mask:FF
```

Define the I/O Behavior During Suspend Mode

Use a SUSPEND attribute to define the behavior of each I/O and output pin during suspend mode.

Single-Ended I/O Standards

Each output, open-drain output, or bidirectional I/O pin in the FPGA application that uses a single-ended I/O standard can be individually programmed for one of the suspend mode behaviors shown in Table 1-2. The default behavior is for a high impedance pin during suspend mode although other options are available.

Table 1-2: **Output Behavior Options during Suspend Mode**

SUSPEND Attribute	Function
DRIVE_LAST_VALUE	The output continues to drive the level that was last stored in the output latch, according to the chosen standard. Requires V_{CCO} to remain at the recommended operating conditions for the bank.
3STATE (default)	The output is in the high-impedance state with no active internal pull-up or pull-down resistor. Results in the lowest possible I/O current draw.
3STATE_PULLUP	The output is in the high-impedance state with an internal pull-up resistor to the associated V_{CCO} supply. Requires V_{CCO} to remain at the recommended operating conditions for the bank.
3STATE_PULLDOWN	The output is in the high-impedance state with an internal pull-down resistor to GND.
3STATE_KEEPER	The output is high impedance. The internal bus keeper circuit is active. Requires V_{CCO} to remain at the recommended operating conditions for the bank.

Differential I/O Standards

The differential output drivers and input receivers consume static power when used in an FPGA application. In suspend mode, differential inputs and outputs are disabled to save power.

The output drivers for the LVDS, RSDS, mini-LVDS, PPDS, and TMDS differential I/O standards are high impedance, using any of the 3STATE attributes described in Table 1-2. The DRIVE_LAST_VALUE attribute is not supported for differential output drivers.

Treat the pseudo-differential I/O standards, such as BLVDS, DIFF_HSTL, and DIFF_SSTL, as two single-ended I/O pins. All the attributes apply as for Single-Ended I/O Standards although for any differential standard the settings must be set appropriately for both pins of the complementary pair.

When in the high-impedance state, the differential driver pair does not conduct current to the power or ground rails, or between adjacent pins.

Differential-input receivers are disabled in suspend mode. Differential input termination (DIFF_TERM) is disabled when in suspend mode.

SUSPEND Attribute

The SUSPEND attribute allows each pin to have an individually defined behavior during suspend mode. The available options are listed in Table 1-2.

UCF Example

This UCF constraint example defines the suspend mode behavior for a specific pin. The SUSPEND attribute can be included on the same UCF line as other constraints for a pin.

```
Net "<net_name>" SUSPEND = "io_type" ;
```

UCF entries for a single-ended pin and a differential pair are shown in the following example:

```
NET "TX<0>" IOSTANDARD = LVCMOS_33 | SUSPEND = "DRIVE_LAST_VALUE" ;
NET "TX_P<0>" IOSTANDARD = LVDS_33 | SUSPEND = "3STATE_PULLUP" ;
NET "TX_N<0>" IOSTANDARD = LVDS_33 | SUSPEND = "3STATE_PULLDOWN" ;
```

Design Maintained during Suspend Mode

After entering suspend mode, all writable clocked primitives are write-protected after a delay of $t_{SUSPEND\_GWE}$. The state of all clocked memory primitives is maintained during suspend mode.

- Logic block flip-flops
- I/O block latches and flip-flops
- Logic block distributed RAM (LUT RAM)
- Logic block shift registers (SRL)
- Block RAM and registers

When exiting suspend mode, all writable clocked primitives are re-enabled, controlled by the `sw_gwe_cycle` setting.

An additional bitstream option, `en_sw_gsr`, controls whether all clocked primitives are globally set or reset when the FPGA awakens from suspend mode. By default, **en_sw_gsr:No** signifies that clocked primitives are not set or reset when the FPGA awakens and all states are preserved.

Design Requirements to Maintain Application Data

When a design requires that application data be preserved when entering suspend mode, the SUSPEND_SYNC primitive should be used. When the FPGA enters suspend mode, the global write enable (GWE) is removed, maintaining the state of all flip-flops and user RAM. The FPGA requires a delay of $t_{SUSPEND\_GWE}$ between recognizing a High on the SUSPEND pin and disabling GWE internally. This is the first event after SUSPEND transitions High, before AWAKE toggles, and before the inputs are disabled if SUSPEND_SYNC is not used. During this delay, additional user clocks to flip-flops or RAM can continue to update their contents. Since the GWE signal can have some skew between locations on the device, some locations can be disabled while others remain enabled on the last clock edge before GWE takes full effect. This situation can be avoided when using the SUSPEND_SYNC feature. After the suspend request is driven out of the SUSPEND_SYNC primitive, disable the clocks and/or clock enables on the logic that must retain its current state. After the disable is complete, drive the SACK port of the SUSPEND_SYNC primitive and the FPGA begins the process to enter suspend mode.

To avoid initializing the flip-flops when exiting suspend mode, choose **en_sw_gsr:No**. Exiting suspend mode should be synchronized to a user clock to avoid race conditions corrupting the application data. Inputs are enabled first, allowing control signals to continue to hold off the toggling of storage primitives. The assertion of GWE can be synchronized to a user clock to align it with a system clock edge.

Suspend Mode Wake-Up Timing Controls

When exiting suspend mode, the wake-up sequence for the FPGA is programmable and controlled by a single clock.

Wake-Up Timing Clock Source

The wake-up timing when exiting suspend mode is controlled by a selectable clock source as shown in Figure 1-4 and described in Table 1-3. The clock source is defined by one or two bitstream generator options, **sw_clk** and sometimes **StartupClk**.

The internal oscillator is disabled during suspend mode to conserve power.

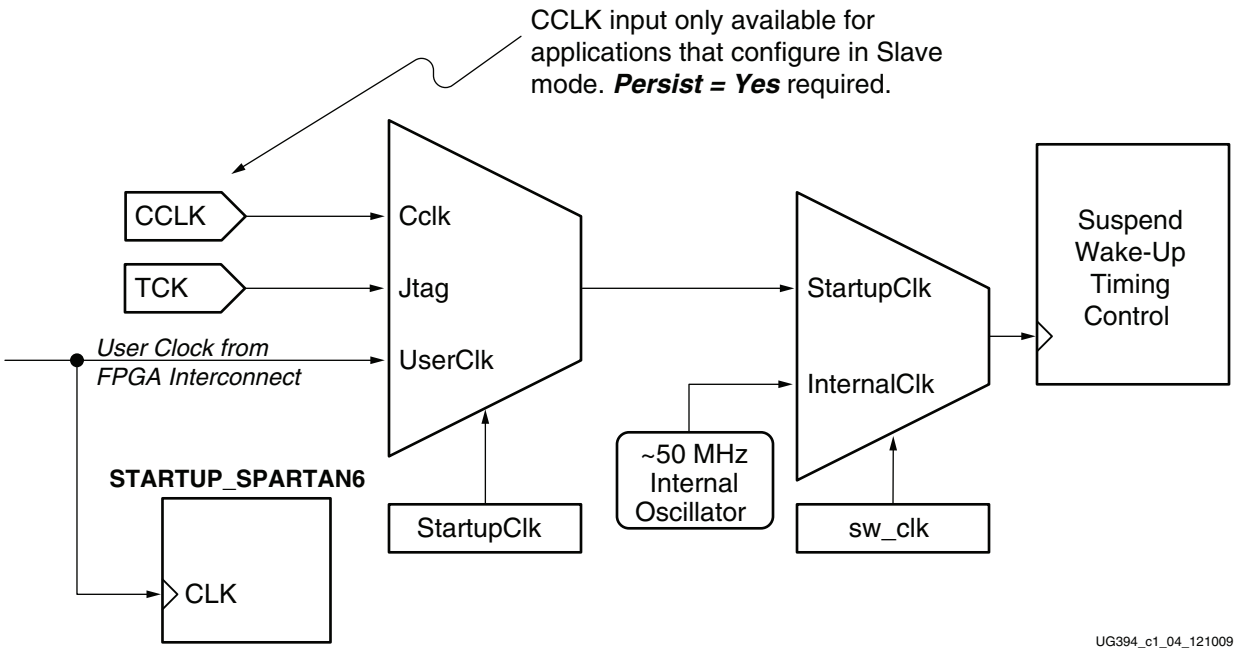

Figure 1-4: **Suspend Mode Wake-Up Timing Control Clock Selection**

- The **sw_clk** option is specific to the suspend feature. By default, `sw_clk:InternalClk`.
- The **StartupClk** option is available on every application. The same option used to clock the start-up process at the end of configuration can be used to clock the wake-up process at the end of suspend. `StartupClk:Cclk` is the default; however, using this for suspend wake-up requires a persisted slave configuration mode. When using `sw_clk:StartupClk` and `StartupClk:Cclk`, and exiting suspend mode, the CCLK pin becomes the clock source. The **Persist** option also retains the dual-purpose configuration pins associated with the configuration logic.

Table 1-3: **Clock Sources to Wake-Up from Suspend Mode**

sw_clk Setting	StartupClk Setting	Clock Source	Restriction
InternalClk	NA	Internal Oscillator	The oscillator has an imprecise frequency of about 50 MHz.
StartupClk	Cclk	CCLK pin on FPGA	This option is only available for FPGAs using Slave configuration mode. The bitstream option **Persist:Yes** must be set. This option is not available for FPGAs using the master configuration mode; use InternalClk instead.
	JtagClk	TCK pin on FPGA	The JTAG interface must be active to exit suspend mode.
	UserClk	CLK input on the STARTUP_SPARTAN6 design primitive	The clock input to the STARTUP design primitive can originate from any non-clocked signal in the FPGA. It cannot originate from a flip-flop source because all clocked primitives are write-protected while in suspend mode.

Switch Outputs from Suspend to Normal Behavior

The suspend/wake **sw_gts_cycle** bitstream option controls when I/O pins are released from their SUSPEND attribute settings and returned to normal operation. The timing is controlled by the Wake-Up Timing Clock Source, page 17. The default **sw_gts_cycle** setting is four cycles, but this control can be set for any value between one and 1,024 clock cycles.

The suspend/wake control becomes active after the AWAKE pin transitions High. After the specified number of clock cycles, all output, open-drain output, and bidirectional I/O pins transition from their suspend behavior, either the default 3STATE or individually specified using the SUSPEND attribute, back to the normal behavior specified in the original FPGA application.

The outputs should be released before releasing the write-protect lock on all clocked primitives.

Release Write Protect on Clocked Primitives

The suspend/wake **sw_gwe_cycle** bitstream option controls when the write-protect lock is released on all clocked primitives.

The timing is controlled by **sw_clk** the Wake-Up Timing Clock Source, page 17. The default **sw_gwe_cycle** setting is five cycles, but the suspend/wake control can be set for any value between one and 1,024 clock cycles.

This suspend/wake control becomes active after the AWAKE pin transitions High. After the specified number of clock cycles, the write-protect lock is released from all writable, clocked primitives such as flip-flops, block RAM, etc.

When the **en_sw_gsr:yes** option is set, the clocked primitives are already globally set or reset to the value specified in the original FPGA design before the write-protect lock is released. The option **en_sw_gsr:no** signifies that the state of the FPGA after entering suspend mode is preserved.

The outputs should be released before releasing the write-protect lock on all clocked primitives.

Dedicated Configuration Pins Unaffected During Suspend Mode

The following dedicated configuration pins are unaffected when the FPGA is in suspend mode:

- JTAG pins: TDI, TMS, TCK, and TDO
- DONE pin
- PROGRAM_B pin

JTAG Operations Allowed During Suspend Mode

Table 1-4 shows the JTAG operations permitted when the FPGA is in suspend mode. Executing these JTAG operations increases the FPGA's power consumption while in suspend mode.

Table 1-4: **JTAG Operations Allowed during Suspend Mode**

Boundary-Scan Command	Description
IDCODE	Read the JTAG ID code that describes the Spartan-6 FPGA array type in the JTAG chain. This value is different from the Device DNA identifier, which is unique to every device.
BYPASS	Enables BYPASS.
USERCODE	Read the user-defined code embedded in the FPGA bitstream.

Do not use any other JTAG instructions when in suspend mode or while transitioning into and out of suspend mode. Furthermore, do not enter suspend mode when performing a readback operation.

SUSPEND Pin

When the suspend feature is enabled (see Enable the Suspend Feature and Glitch Filtering, page 14), the SUSPEND pin controls when the FPGA enters suspend mode. During normal FPGA operation, the SUSPEND pin must be Low. When High, the SUSPEND pin forces the FPGA into the low-power suspend mode. Table 1-5 describes the functionality of the SUSPEND pin.

If the suspend feature is not enabled for an application (the application never enters low-power mode), then connect the SUSPEND pin to GND. Do not leave the pin floating.

Table 1-5: **SUSPEND Pin Functionality**

ENABLE_SUSPEND Settings	SUSPEND Pin	Function
NO (default) Suspend Mode Disabled	X	The suspend feature is disabled. The SUSPEND pin is unused and ignored. Connect the SUSPEND pin to GND.
Filtered, Unfiltered Suspend Mode Enabled	0	The FPGA performs the application described in the bitstream loaded into the FPGA during configuration. When the SUSPEND pin changes from High to Low, wake the FPGA from suspend mode. Return from suspend mode also dependents on the SCP pins, if used.
	1	Force the FPGA to enter power-saving suspend mode pending SACK assertion on SUSPEND_SYNC primitive, if used.

Characteristics

The SUSPEND pin is an LVCMOS/LVTTL receiver, and power to the input buffer is supplied by the V_{CCAUX} power rail. The SUSPEND pin has no pull-up resistors during configuration, and the HSWAPEN control has no effect on the SUSPEND pin.

SUSPEND Input Glitch Filter

The SUSPEND pin has a programmable glitch filter to guard against short pulses, which could cause the FPGA to spuriously enter suspend mode. Turning off the filter allows the FPGA to enter or exit suspend mode more quickly, but the application must guard against spurious pulses. The difference in delay is the $t_{SUSPENDFILTER}$ value in DS162, *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*. See Enable the Suspend Feature and Glitch Filtering, page 14.

SUSPEND_SYNC Primitive

The SUSPEND_SYNC primitive is the application interface to a suspend request. If this primitive is not present in the design, the FPGA begins the suspend sequence solely on the state of the SUSPEND pin.

When the SUSPEND_SYNC primitive is in the design, after the SUSPEND pin is asserted High and the filter delay (when the glitch filter is enabled), the SUSPEND_SYNC primitive drives the SREQ port High on the next rising clock edge on the CLK port. This indicates that a request has been received to enter suspend mode. The FPGA does not enter suspend mode until the SACK port is driven High on a rising edge of CLK.

This primitive provides an ideal interface for the application to complete any functions prior to entering suspend mode. Any I/O interface ports can be closed, buffers flushed, and clocks disabled to ensure the application is in a ready state prior to being suspended. For more details, see Design Maintained during Suspend Mode, page 16.

AWAKE Pin

The AWAKE pin (optionally) provides status on the suspend power-savings mode.

General Behavior (Suspend Feature Disabled)

Unless the suspend feature is enabled, the AWAKE pin is a general-purpose user-I/O pin.

AWAKE Pin Behavior when Suspend Feature is Enabled

If the suspend feature is enabled, then the AWAKE pin indicates the present state of the FPGA, as summarized in Table 1-6. The AWAKE pin cannot be used by the FPGA application as a general-purpose I/O pin.

Table 1-6: **AWAKE Pin Status**

AWAKE Pin	Indication
0	The FPGA is presently in the low-power suspend mode.
1	The FPGA is active.

The AWAKE pin can further be configured as an open-drain output (the default) or a full-swing output driver, as shown in Figure 1-5. This behavior is controlled by a bitstream generator (BitGen) option:

```
bitgen -g drive_awake:no
```

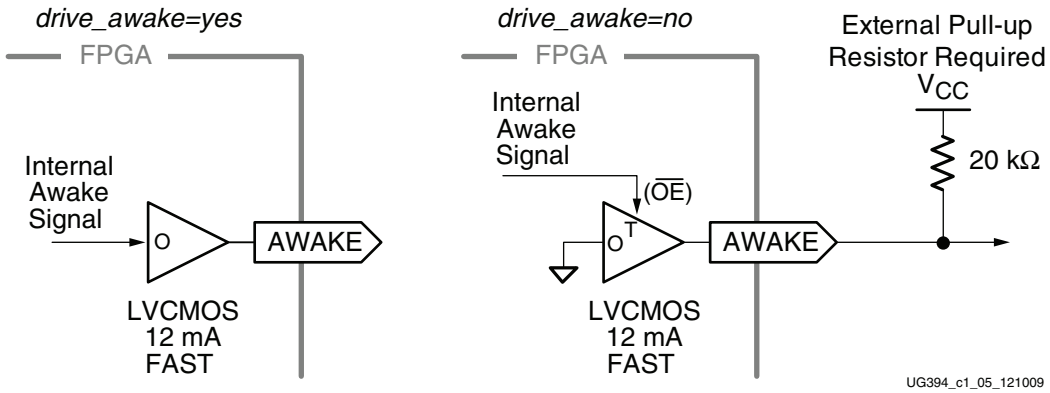

Figure 1-5: **AWAKE Output Drive Options if Suspend Mode Enabled**

The AWAKE output pin is supplied by the V_{CCAUX} power rail.

When the option **drive_awake:yes** is set, the AWAKE pin is an active output driver, equivalent to a user I/O configured as LVCMOS, with a 12 mA output drive and a fast slew rate.

Controlling Wake-Up from an External Source

The default option is **drive_awake:no**. The **drive_awake:no** option signifies that the AWAKE pin is an open-drain output capable of sinking 12 mA. In this case, an external pull-up resistor is required to exit suspend mode. To minimize the amount of current flow during suspend mode, the resistor value should be high. The resistor needs to be strong enough to overcome the I/O pin leakage. A large resistor value also equates to a longer AWAKE rise time. The FPGA does not exit suspend mode and begin the wake-up process until AWAKE transitions High.

Synchronizing Wake-Up

The wake-up process can be synchronized across multiple FPGAs or between the FPGAs and the system by using one SUSPEND signal to control multiple devices. The AWAKE pin can also synchronize multiple devices. To start the wake-up process at the same time, the AWAKE pins of multiple FPGAs can be tied to a single pull-up resistor. The wake-up counters can also be synchronized if **sw_clk:StartupClk** and **StartupClk:UserClk**.

Holding the AWAKE pin Low delays the transition from suspend mode to active mode by holding off the **sw_gwe_cycle** and **sw_gts_cycle** counters, and allows an external controller to decide when to begin the wake-up process in the FPGA.

Post-Configuration CRC Limitations When Using Suspend Mode

To minimize power, post-configuration CRC checking stops during suspend mode.

If an active application uses the post-configuration CRC feature and an error occurs, do not enter suspend mode. If there is a CRC error, the FPGA does not wake from suspend mode without reprogramming, such as asserting PROGRAM_B or power-cycling the FPGA.

Several design options are possible:

1. Do not use the post-configuration CRC feature when the suspend mode feature is enabled and vice versa.
2. Always reprogram the device when a CRC error occurs.

Table 1-7 summarizes the various bitstream options associated with suspend mode.

Table 1-7: **Suspend Mode Bitstream Generator Options**

Suspend Mode Bitstream Options	Options (Default)	Description
drive_awake	No (Default)	If suspend mode is enabled, indicates the present status on AWAKE using an open-drain output. An external pull-up resistor or High signal is required to exit SUSPEND mode.
	Yes	If suspend mode is enabled, indicates the current status by actively driving the AWAKE output.
en_sw_gsr	No (Default)	The state of all clocked primitives in the FPGA is preserved.
	Yes	Pulses the GSR signal during wake-up, setting or resetting all clocked primitives, as originally specified in the FPGA application. The GSR pulse occurs before the AWAKE pin transitions High and before the **sw_gwe_cycle** and **sw_gts_cycle** settings are active.
sw_clk	StartupClk (Default)	Uses the clock defined by the `StartupClk` bitstream generator setting to control the suspend wake-up timing.
	InternalClk	Uses the internally generated 50 MHz oscillator to control the suspend wake-up timing.
sw_gwe_cycle	1,..,5,...,1024 Default is 5	After the AWAKE pin is High, indicates the number of clock cycles as defined by the **sw_clk** setting, when the global write-protect lock is released for writable clocked primitives (flip-flops, block RAM, etc.). The default value is five clock cycles after the AWAKE pin transitions High. Generally, this value is equal to or greater than the **sw_gts_cycle** setting.
sw_gts_cycle	1,..,4,...,1024 Default is 4	After the AWAKE pin is High, indicates the number of clock cycles as defined by the **sw_clk** setting, when the I/O pins switch from their SUSPEND Attribute, page 16 settings back to their normal functions. The default value is four clock cycles after the AWAKE pin goes High. Generally, this value is equal to or less than the **sw_gwe_cycle** setting.
multipin_wakeup	No (Default)	Disables multi-pin wake-up.
	Yes	Enables multi-pin wake-up.
wakeup_mask	`0x00`	Masks out SCP<7:0>
	`<hex string>` (Default)	`FF` enables SCP<7:0>, `0F` enables SCP<3:0>.

FPGA Voltage Requirements During Suspend Mode

During suspend mode, the V_{CCINT} and V_{CCAUX} rails must remain powered at the data sheet levels. However, the V_{CCO} supply to each of the I/O banks can be (potentially) turned off to conserve additional power, depending on system requirements. Optionally, V_{CCO} can be reduced during suspend mode, but this also affects the voltage levels for any output pin with a SUSPEND attribute set to DRIVE_LAST_VALUE.

The FPGA's power-on reset (POR) circuit continues to monitor the V_{CCINT} and V_{CCAUX} supplies. Although V_{CCO2} is an input to the POR circuit at initial power-on, the POR circuit does not monitor the V_{CCO} supplies after configuration. By default, if the V_{CCINT} or V_{CCAUX} supply dips below the minimum specified data sheet voltage limit, then the FPGA restarts configuration.

Memory Controller Block

Recommendations and methods for using the memory controller block interface with the Spartan-6 FPGA suspend mode are found in UG388, *Spartan-6 FPGA Memory Controller Block User Guide*.

XILINX®

Chapter 2

Voltage Supplies

Introduction

Spartan-6 FPGAs have multiple voltage supply inputs, as shown in Table 2-1. There are two supply inputs for internal logic functions, V_{CCINT} and V_{CCAUX}. Each of the I/O banks has a separate V_{CCO} supply input that powers the output buffers within the associated I/O bank. V_{CCO} is also used for input buffers for some I/O standards. A V_{REF} reference voltage is needed for HSTL/SSTL standards. The GTP transceivers have dedicated analog power rails (see UG386, *Spartan-6 FPGA GTP Transceivers User Guide* for more details). The AES circuitry has its own power supplies for the encryption key, depending on how it is stored (see UG380, *Spartan-6 FPGA Configuration User Guide* for more details).

Table 2-1: **Spartan-6 FPGA Voltage Supplies**

Supply Input	Description	Devices	Nominal Supply Voltage
V_{CCINT}	Internal core supply voltage. Supplies all internal logic functions, such as CLBs, block RAM, and DSP blocks. Input to the power-on reset (POR) circuit. Powers input signals for most standards at 1.2V, 1.5V, and 1.8V.	All	1.2V; 1.0V (-1L) in lower-power Spartan-6 LX devices
V_{CCAUX}	Auxiliary supply voltage. Supplies clock management tiles (CMTs), some I/O resources, dedicated configuration pins, and JTAG interface. Powers input signals for most standards at 2.5V and 3.3V. Input to the POR circuit.	All	2.5V; 3.3V optional
V_{CCO_0}	Supplies the output buffers in I/O bank 0, the bank along the top edge of the FPGA.	All	Selectable: 3.3V, 2.5V, 1.8V, 1.5V, or 1.2V
V_{CCO_1}	Supplies the output buffers in I/O bank 1, the bank along the right edge of the FPGA. During configuration in byte-wide peripheral interface (BPI) Parallel Flash Mode, connects to the same voltage as the Flash PROM.	All	Selectable: 3.3V, 2.5V, 1.8V, 1.5V, or 1.2V
V_{CCO_2}	Supplies the output buffers in I/O bank 2, the bank along the bottom edge of the FPGA. Connects to the same voltage as the FPGA configuration source. Input to the POR circuit.	All	Selectable: 3.3V, 2.5V, 1.8V, 1.5V, or 1.2V
V_{CCO_3}	Supplies the output buffers in I/O bank 3, the bank along the left edge of the FPGA.	All	Selectable: 3.3V, 2.5V, 1.8V, 1.5V, or 1.2V

Table 2-1: **Spartan-6 FPGA Voltage Supplies *(Cont'd)***

Supply Input	Description	Devices	Nominal Supply Voltage
V_{CCO_4}	Supplies the output buffers in I/O bank 4, the bank along the top of the left edge of the FPGA in 6-bank devices.	LX75/T, LX100/T, and the LX150/T in FG(G)676 and FG(G)900	Selectable: 3.3V, 2.5V, 1.8V, 1.5V, or 1.2V
V_{CCO_5}	Supplies the output buffers in I/O bank 5, the bank along the top of the right edge of the FPGA in 6-bank devices.	LX75/T, LX100/T, and the LX150/T in FG(G)676 and FG(G)900	Selectable: 3.3V, 2.5V, 1.8V, 1.5V, or 1.2V
V_{REF}	Input threshold voltage pins when HSTL/SSTL standards are used in the bank, otherwise user I/Os. When used as a reference voltage within a bank, all V_{REF} pins within that bank must be connected.	All	Varies
MGTAVCC	Power-supply pin for the transceiver mixed-signal circuitry.	LXT	1.2V
MGTAVCCPLL0/1	Power-supply pin for the transceiver PLL	LXT	1.2V
MGTAVTTTX/RX	Power-supply pin for the transceiver TX and RX circuitry.	LXT	1.2V
MGTAVTTRCAL	Power-supply pin for the transceiver resistor calibration circuit.	LXT	1.2V
V_{BATT}	Decryptor key memory backup supply. When key is not used, tie this pin to V_{CC} or GND, or it can be left floating.	LX75/T, LX100/T, LX150/T	3.3V
V_{FS}	Decryptor key EFUSE power supply pin for programming. When key is not used, tie this pin to V_{CC} or GND, or it can be left floating.	LX75/T, LX100/T, LX150/T	3.3V

V_{CCINT}

V_{CCINT} is the primary power supply for the FPGA. In the Spartan-6 LXT family and the standard devices in the Spartan-6 LX family (-2 and -3 speed grades), V_{CCINT} has a nominal value of 1.2V. The lower-power Spartan-6 LX devices (-1L speed grade) uses a nominal V_{CCINT} of 1.0V. See Chapter 3, Lower-Power Spartan-6 LX Devices.

V_{CCAUX}

V_{CCAUX} powers the auxiliary logic, including configuration logic and some internal and I/O resources. The Spartan-6 FPGA's V_{CCAUX} is either 2.5V or 3.3V. These two voltages provide greater flexibility and allow V_{CCAUX} to be set to the same level as an existing V_{CCO} rail, to minimize the number of power rails. Reducing V_{CCAUX} to 2.5V can reduce the power consumption on the V_{CCAUX} rail by 40%.

Setting the V_{CCAUX} Level

The user must set the CONFIG V_{CCAUX} attribute according to the voltage being provided to the V_{CCAUX} rails. The valid values for this attribute are 2.5 (default) or 3.3. This attribute affects the banking rules for I/O placement within the automated placer, as well as in the pin assignments tool. It also affects the end-generated bitstream for the device. The V_{CCAUX} attribute is a global attribute for the Spartan-6 device and is not attached to any particular primitive.

```
CONFIG VCCAUX=3.3;
```

V_{CCAUX} Specifications

Both the 2.5V and 3.3V settings for V_{CCAUX} allow a ±5% variation (2.375V to 2.625V, or 3.15V to 3.45V). The data retention voltage is the same for both at 2.0V, so more care must be taken with a 2.5V rail to not let it drop more than 0.5V. See DS162, *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics* for complete specifications.

The CONFIG V_{CCAUX} attribute is used by the ISE® Design Suite software to determine if LVCMOS25 inputs can be powered by V_{CCAUX}. If CONFIG V_{CCAUX} = 2.5, V_{CCAUX} is used to power LVCMOS25 inputs. If CONFIG V_{CCAUX} = 3.3, V_{CCO} must be 2.5V for any banks with LVCMOS25 inputs. Setting V_{CCAUX} to match whichever is more common between LVCMOS25 and LVCMOS33 can help optimize placement.

There are some slight changes to resistor values depending on whether V_{CCAUX} is set to 2.5V or 3.3V. The I/O pull-down resistor values are lower for a V_{CCAUX} of 3.3V. The differential termination resistor (DIFF_TERM) can be more tightly controlled around 100Ω when V_{CCAUX} is 3.3V. See DS162, *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics* and the Spartan-6 FPGA IBIS models at: http://www.xilinx.com/support/download/index.htm

V_{CCO}

V_{CCO} powers the I/O resources, and has separate rails for each bank of I/O for maximum flexibility. All of the V_{CCO} connections to a specific I/O bank must be connected to the same voltage. The V_{CCO} voltage can be 1.2V to 3.3V, depending on the output standard specified for a given bank. Most devices have four I/O banks, while the XC6SLX75/T and larger in the FG(G)676 and FG(G)900 packages offer six I/O banks. The V_{CCO} pins for a bank should all be tied to a supply rail, even if the bank is completely unused.

In a 3.3V-only application, all V_{CCO} supplies and V_{CCAUX} connect to 3.3V. Spartan-6 FPGAs allow bridging between different I/O voltages and standards by applying different voltages to the V_{CCO} inputs of different banks. Refer to the I/O banking rules section in to UG381, *Spartan-6 FPGA SelectIO Resources User Guide* for the I/O standards that can be mixed within a single I/O bank.

The Spartan-6 FPGA V_{CCO} ranges support ±5% variation around the nominal supply voltage. Refer to DS162, *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics* for specific voltage levels.

V_{REF}

Each I/O bank also has a separate, optional input voltage reference supply, called V_{REF}. If the I/O bank includes an I/O standard that requires a voltage reference such as HSTL or SSTL, then all V_{REF} pins within the I/O bank must be connected to the same voltage. The V_{REF} pins are available as I/O pins if no standards within a bank require them.

Xilinx recommends always separating V_{REF} from V_{TT} as the V_{TT} supply can be very noisy. A stable V_{REF} using a small LDO is the desirable implementation. A voltage divider implementation is also possible. Knowledge of the PCB environment, such as frequency of coupled noise, is required to correctly calculate the resistance and capacitance values of the divider circuit. As a result, an isolated reference supply is usually a more robust and simpler approach. Refer to UG381, *Spartan-6 FPGA SelectIO Resources User Guide* for more details on V_{REF}.

Board Design and Signal Integrity

Building a working system today requires knowledge of the many options available. The advantages of feature size reduction and reduced power consumption have reduced core voltages down to the 1.0V range. This change in voltage and signal frequency content requires the use of advanced design practices to manage electrical effects. The documents and links on the Xilinx Signal Integrity website at http://www.xilinx.com/products/design_resources/signal_integrity/index.htm provide everything needed to achieve reliable PCB designs the first time.

Simultaneously Switching Outputs

Ground or power bounce occurs when a large number of outputs simultaneously switch in the same direction. Each FPGA family provides guidelines for the recommended maximum allowable number of simultaneously switching outputs (SSOs). For more information on SSO, see the Simultaneously Switching Outputs section of UG381, *Spartan-6 FPGA SelectIO Resources User Guide* and DS162, *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

Power Distribution System Design and Decoupling/Bypass Capacitors

Good power distribution system (PDS) design is important for all FPGA designs, especially for high-performance applications greater than 100 MHz. Proper design results in better overall performance, lower clock jitter, and a generally more robust system. Before designing the printed circuit board (PCB) for the FPGA design, review UG393, *Spartan-6 FPGA PCB Design Guide*.

XILINX®

Chapter 3

Lower-Power Spartan-6 LX Devices

Introduction

The lower-power Spartan-6 LX devices (-1L) meet lower quiescent and dynamic current levels than the standard Spartan-6 LX devices. They also operate at a reduced V_{CCINT} of 1.0V, versus the 1.2V of the standard Spartan-6 family, thus reducing core power. The lower-power Spartan-6 LX devices are supported by the -1L speed grade. The -1L is approximately one speed grade slower (~15%) than the standard Spartan-6 LX family's slowest speed grade (-2) but provides an additional 30–40% power savings.

Use L1 as the speed grade when ordering the lower-power Spartan-6 LX devices, which is also the way the speed grade is marked on the device. For example, the FPGA ordered as the XC6SLX16-L1CSG324 is marked as either L1C for commercial temperature range or L1I for industrial temperature range.

Designing Using the Lower-Power Spartan-6 LX Devices

To design for the lower-power Spartan-6 LX devices, select the appropriate device during implementation. A design targeted to the lower-power Spartan-6 LX devices can be defined using all the same methods and options as available to the standard Spartan-6 LX devices. All primitives that support the Spartan-6 LX family also support the lower-power Spartan-6 LX devices. The lower-power Spartan-6 LX devices can be selected using the ISE Design Suite in the Project Navigator. Different from the ordering code or actual device marking, the Xilinx tools display the part number with an appended L (for example, XC6SLX16L). The only speed grade supported for these devices is the -1L. The same speed grade supports both the commercial and industrial temperature ranges.

The resulting bitstream is identical in format between the standard Spartan-6 LX devices and the lower-power Spartan-6 LX devices. The JTAG device IDCODEs are identical and the iMPACT software identifies the device under the same standard Spartan-6 FPGA part number. However, there are small implementation differences between the two families, including a reset circuit used for the DCM CLKFX output, and differences in block RAM configuration. Therefore, the designer must target the correct device both to generate the correct timing information, and to account for implementation differences between the two families. The standard and lower-power Spartan-6 LX device bitstreams can not be used in both families. A standard Spartan-6 LX device should not be powered at 1.0V, and a lower-power Spartan-6 LX device should not be powered at 1.2V.

Table 3-1 summarizes the designations for a member of the lower-power Spartan-6 LX devices.

Table 3-1: **Lower-Power Spartan-6 LX Device Designation Examples**

Designation	Example
Ordering Code	XC6SLX16-L1CSG324C
Mark	XC6SLX16 CSG324 DxxxxxxxA (lot code) L1C
Software Family	In software choose: **Spartan6 Lower Power**
Software Device	XC6SLX16L-1LCSG324
Speed Specification	-1L

Lower-Power Spartan-6 LX Device Specifications

Several specifications are different for the lower-power Spartan-6 LX devices than in the standard Spartan-6 LX family. All of the differences are listed in DS162, *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

The lower-power Spartan-6 LX devices require a V_{CCINT} of 1.0V ±5%, or 0.95V to 1.05V. It is not tested or guaranteed at 1.2V, and therefore the V_{CCINT} cannot be scaled up and down between 1.2V and 1.0V. In the same way, a standard Spartan-6 LX device cannot be operated at 1.0V.

The lower-power Spartan-6 LX devices have lower-power specifications, as seen in the data sheet for quiescent current, and in the Power Estimators for both quiescent and dynamic current and power. See Chapter 5, Power Estimation.

Because of the reduction in maximum V_{CCINT}, the maximum time for ramping up is shorter. See the V_{CCINTR} ramp time specification in DS162, *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

Because the I/O thresholds for LVCMOS12, LVCMOS15, and LVCMOS18 are based on the V_{CCINT} level, they are slightly lower for the lower-power Spartan-6 LX devices than the standard Spartan-6 family. See the *SelectIO Interface DC Input and Output Levels* table in DS162, *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*.

Chapter 4

Power-On and Power-Down Behavior Including Hibernate

Introduction

Spartan-6 FPGAs are designed for maximum system flexibility and reliability when powering up and powering down. During power-on, the device ensures reliable configuration by waiting until a fixed time after the supply rails are valid. During power-down or hibernate, the device disables the outputs and awaits re-application of power to reconfigure. Both power-up and power-down are enhanced by the hot swap compliance of the I/O, allowing the FPGA to be moved in or out of a powered system without damage.

Power-On Reset

Spartan-6 FPGAs have a built-in power-on reset (POR) circuit that monitors the three power rails required to successfully configure the FPGA (see Figure 4-1). At power-up, the POR circuit holds the FPGA in a reset state until the V_{CCINT}, V_{CCAUX}, and V_{CCO_2} supplies reach their respective input threshold levels. A time t_{POR} after all three supplies reach their respective thresholds, the POR reset is released and the FPGA begins its configuration process.

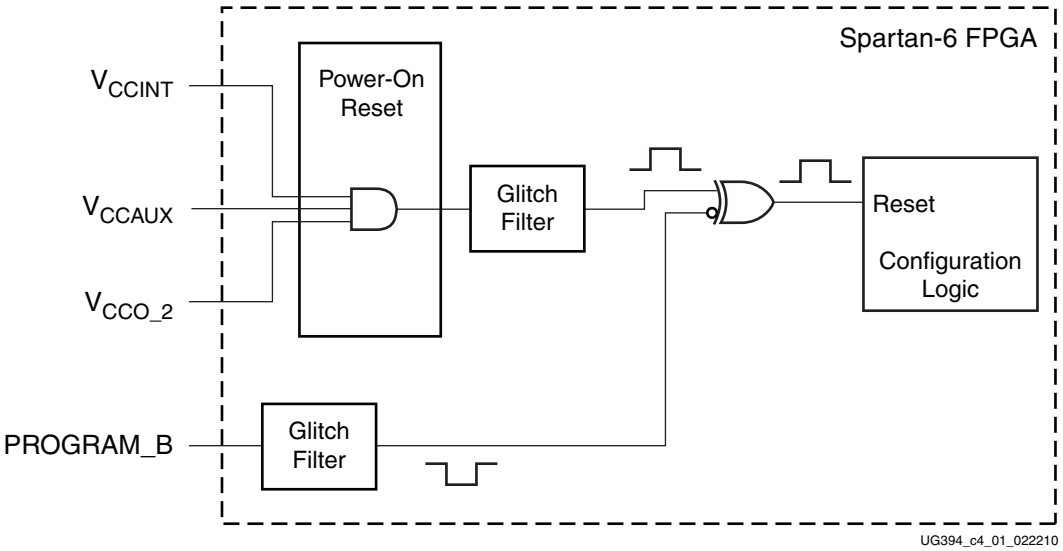

Figure 4-1: **Simplified POR Circuit Diagram**

The V_{CCO_2} supplies are part of the POR circuit, because the primary configuration pins are in bank 2. V_{CCO_2} is typically at 2.5V or 3.3V for the configuration interface. Make sure V_{CCO_2} reaches the proper level for POR and for configuration, especially if it is reduced after configuration for a lower-voltage I/O standard such as LVCMOS15 or LVCMOS12.

For information on the power-on reset step as part of the configuration process, see *Device Power-Up* in UG380, *Spartan-6 FPGA Configuration User Guide*.

Supply Sequencing

The Spartan-6 FPGA can be powered up and powered down in any sequence. Because the three FPGA supply inputs must be valid to release the POR and can be supplied in any order, there is no FPGA-specific voltage sequencing requirement. Although the FPGA has no specific voltage sequence requirements, any potential sequencing requirement of the configuration device attached to the FPGA, such as an SPI serial Flash PROM, a parallel NOR Flash PROM, or a microcontroller should be considered. For example, Flash PROMs have a minimum time requirement before the PROM can be selected, and this time must be considered if the 3.3V supply is the last in the sequence. See *Power-On Sequence Precautions* in UG380, *Spartan-6 FPGA Configuration User Guide*.

Ramp Rate

The power supplies should ramp monotonically within the power supply ramp time range specified in DS162, *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*. To ensure successful power-on, V_{CCINT}, V_{CCO} bank 2, and V_{CCAUX} supplies must rise through their respective threshold-voltage ranges with no dips. The maximum ramp time for V_{CCINT} in the lower-power Spartan-6 LX devices is slightly lower than for the standard family because of the reduced voltage.

Hot Swap

Hot swap, also known as hot plug or hot insertion, refers to plugging an unpowered board into a powered system. To support hot swap, an unpowered board or device must be able to be plugged directly into a powered system or backplane without affecting or damaging the system or the board/device. Spartan-6 FPGAs are fully hot swap compliant and include the following I/O features:

- Signals can be applied to I/O pins before powering the device
- I/O pins are high-impedance (that is, three-stated) before and throughout the power-up and configuration processes
- There is no current path from the I/O pin back to the voltage supplies

Power rails can be disabled in any order without damage to the FPGA. It is recommended that all I/O pins be ignored after any of the power rails (V_{CCINT}, V_{CCO}, or V_{CCAUX}) drops below its minimum operating voltage. To transition cleanly from valid signals to the disabled state, either first disable the outputs in the design, or shut down V_{CCO} followed by V_{CCAUX} and then V_{CCINT}.

Configuration Data Retention and Brown Out

The FPGA's configuration data is stored in robust CMOS configuration latches. The data in these latches is retained even when the voltages drop to the minimum levels necessary to preserve RAM contents, as specified in DS162, *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics* (V_{DRINT} and V_{DRAUX}).

After configuration, if the V_{CCAUX} or V_{CCINT} supply drops below its minimum data retention voltage, the integrity of the CMOS configuration latches is no longer guaranteed, and the current device configuration must be cleared using one of the following methods:

- Force the V_{CCAUX} or V_{CCINT} supply voltage to GND, then raise the voltages back to the recommended operating range (as shown in DS162, *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*)
- Assert PROGRAM_B Low, then raise it back High

The POR circuit does not monitor the V_{CCO_2} supply after configuration. Consequently, dropping the V_{CCO_2} voltage does not reset the device by triggering a POR event. The PROGRAM_B input bypasses the POR circuit (see Figure 4-1, page 31) and therefore can be used as an independent means to initialize the FPGA.

After the INIT_B signal goes High to indicate successful clearing of the FPGA, reconfigure the FPGA.

GTP Transceiver Power-Up and Power-Down

All GTP_DUAL tiles are reset automatically after configuration. The supplies for the calibration resistor and calibration resistor reference must be powered up before configuration to ensure correct calibration of the termination impedance of all transceivers.

The reference clock and the power to the GTP_DUAL tile must be available before configuring the FPGA. If the reference clock or GTP_DUAL tile is powered up after configuration, apply GTPRESET to allow the PMA PLL to lock.

The GTP transceiver supports a range of power-down modes. These modes support both generic power management capabilities as well as those defined in the PCI Express and SATA standards.

Each channel in each direction can be powered down separately using TXPOWERDOWN and RXPOWERDOWN. Each PLLPOWERDOWN port directly affects the associated PLL that is selected by the PLL_SOURCE attribute.

For more details on the GTP Transceivers, see UG386, *Spartan-6 FPGA GTP Transceivers User Guide*.

Hibernate Power Down

Hibernate is effectively powering down the FPGA (Figure 4-2). Due to the hot swap compliance of the Spartan-6 family, this is allowed even if external devices are still providing active signals to the FPGA. The FPGA loses its configuration data and must be re-programmed after power-on. Hibernate provides the maximum possible power savings for applications that can be turned off for long periods of time and that can afford to lose the present design state.

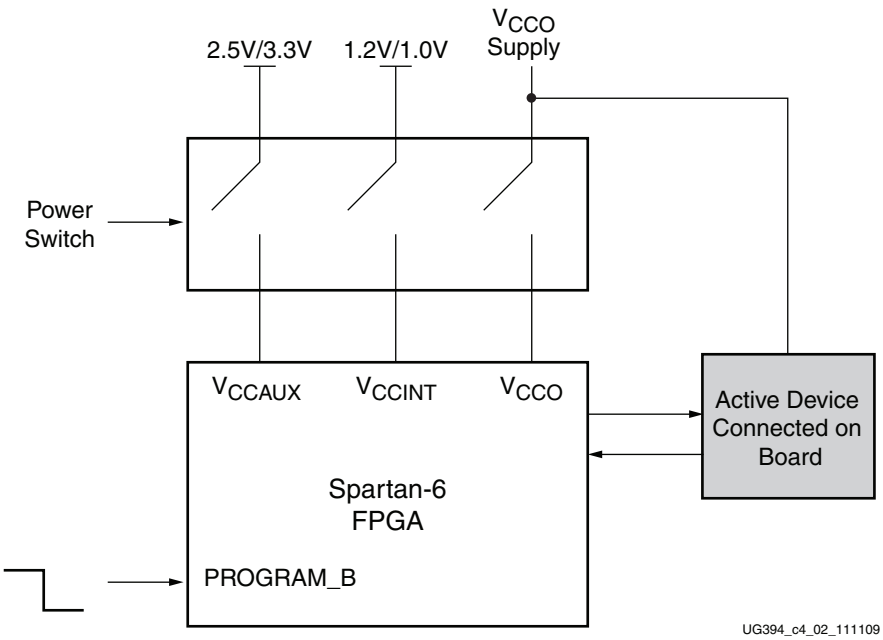

Figure 4-2: **Spartan-6 FPGA Power-Off Diagram**

Forcing FPGA to Quiescent Current Levels

Before removing the power supplies, it is recommended to first put the device into the quiescent state. Pulse PROGRAM_B Low to achieve the quiescent current levels. Driving PROGRAM_B Low forces all I/Os into a high-impedance state, ceases all internal switching, and converts the bitstream held in internal memory to all zeros. During and after the Low pulse on PROGRAM_B, disable the internal pull-up resistors on all I/Os by driving the HSWAPEN input High. Holding PROGRAM_B Low continues clearing the configuration memory. To minimize quiescent current, release PROGRAM_B High but hold off configuration by holding INIT_B Low, or by setting the Mode pins to a slave or JTAG configuration mode and disabling the external configuration clock (CCLK or TCK).

To restart the application, release PROGRAM_B High and in slave or JTAG modes, enable the external configuration source. The FPGA must reconfigure before the application restarts. No state information is preserved. If the application must retain the FPGA configuration bitstream, then use the suspend mode.

Entering Hibernate State

Hibernate starts with the approach described in Forcing FPGA to Quiescent Current Levels, page 34. Hibernate provides further power savings by switching off power rails. This state reduces quiescent power consumption to the lowest possible level. The FPGA enters Hibernate by switching off the V_{CCINT} (core), V_{CCAUX} (auxiliary), and V_{CCO} (output) power supplies. Power FETs with low on resistance are recommended to perform the switching action. Configuration data is lost upon entering Hibernate; therefore, the device reconfigures after exiting the state.

Holding the PROGRAM_B input Low during the transition into Hibernate keeps all FPGA output drivers in a high-impedance state. Release PROGRAM_B after re-applying power. See Design Considerations, page 36 for recommended levels on Dedicated and Dual-Purpose pins.

Turn Off V_{CCO}

Spartan-6 FPGA I/O pins have a floating-well structure, providing full hot-swap/hot-insertion capability. When a Spartan-6 FPGA is in the Hibernate state, the V_{CCO} supply can be safely turned off without adversely affecting either the FPGA or the external application. When entering the Hibernate state, V_{CCO} should be turned off first to disable the outputs without any unwanted transitions.

Figure 4-3 shows the waveforms for entering and exiting Hibernate. The steps for entering Hibernate are as follows:

1. Pull the PROGRAM_B pin Low to force all user-I/O pins into a high-impedance state.
2. The FPGA drives the INIT_B and DONE pins Low.
3. External switches turn off the V_{CCINT}, V_{CCAUX}, and V_{CCO} supply rails to the FPGA.
4. The FPGA is now in Hibernate. While the FPGA is kept in this state, power consumption rests at the lowest possible level.

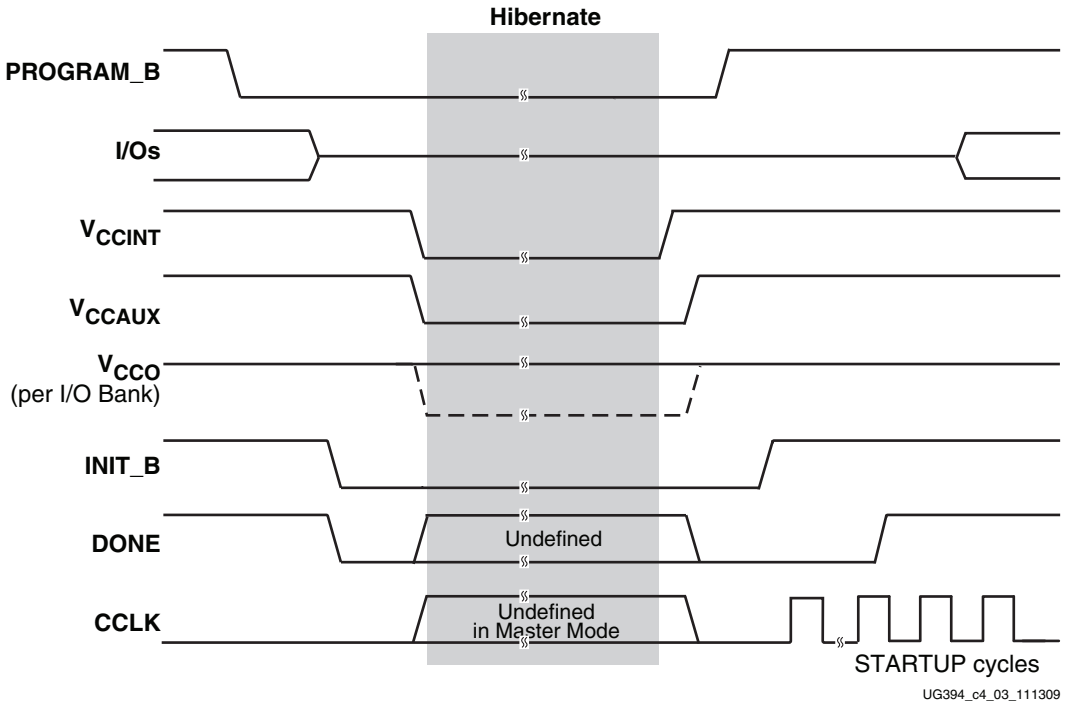

Figure 4-3: **Hibernate Waveform**

Exiting Hibernate

The steps for exiting Hibernate are as follows.

1. Reapply power to all rails that were switched off. Apply power in any sequence.
2. Before FPGA initialization can begin, deassert PROGRAM_B to a High logic level. The rising transition on PROGRAM_B must occur after turning all three power supplies back on.
3. After logic initialization, the FPGA releases the open-drain INIT_B signal. With INIT_B High, the FPGA starts its configuration process.
4. When configuration is complete, the FPGA enters the Start-up phase, asserts DONE, and enables the I/Os, according to how the BitGen options are set.
5. The FPGA is now ready for user operation.

Design Considerations

Be aware of how various pins are powered in the application. Most user-I/O pins, including the dual-purpose configuration pins and the dedicated PROGRAM_B and DONE pins, are powered by a specific V_{CCO} supply input. Dedicated configuration pins such as SUSPEND and the JTAG pins are powered by the V_{CCAUX} supply. If disconnecting power to any of these supplies, consider how that affects FPGA configuration when power is re-applied.

For specific information on configuration pins and their associated power rails, refer to UG380, *Spartan-6 FPGA Configuration User Guide*.

XILINX®

Chapter 5

Power Estimation

Introduction

Xilinx provides a complete power estimation solution using power estimators and analyzers, power-driven implementation tool algorithms, and a variety of power-related documentation. The Power Solutions page provides access to these tools, documentation, news, and supply solutions: http://www.xilinx.com/products/design_resources/power_central/index.htm.

To estimate the total power consumption (quiescent plus dynamic) for a specific design use one of the following tools:

- The XPower Power Estimator spreadsheet provides quick, approximate estimates, and does not require the design's netlist.
- The XPower Analyzer is delivered with the ISE Design Suite software and uses a netlist as input to provide more accurate estimates.

The XPower Power Estimator (XPE) spreadsheet is a power estimation tool typically used in the pre-design and pre-implementation phases of a project. XPE assists with architecture evaluation and device selection, and helps in selecting the appropriate power supply and thermal management components. For comparison and analysis, the XPE spreadsheet for the Spartan-6 family also includes the Extended Spartan-3A family.

XPE considers the resource usage, toggle rates, I/O loading, and many other design factors. Combined with device models, the estimated power distribution can be calculated in XPE. The device models are extracted from measurements, simulation, and/or extrapolation.

After implementation, the XPower Analyzer (XPA) tool (available in the ISE Design Suite software) is used for more accurate estimates and power analysis. For more information about XPA, see the XPower Analyzer Help.

Voltage Regulators

The choice of a voltage regulator depends on system requirements and the estimated power consumption requirements for the FPGA. Use the XPower tools to calculate the requirements for a specific device and design. Then choose a regulator from a pin-compatible family so the current capability can be adjusted up or down. External power FETs are easy to upgrade. A soft-start feature that controls output ramp time is useful.

With care, use of overcurrent protection is possible, such as foldback or fuses. Be aware that capacitors are charging at power-on and might draw a significant amount of current for a short time.

If necessary, slow the supply voltage ramp to control the charge current. If foldback is not a design requirement, it is best to avoid it, keeping the power-supply design simple.

Various power-supply manufacturers offer complete power solutions for Xilinx FPGAs including some with integrated three-rail regulators specifically designed for Xilinx FPGAs. The Xilinx Power Solutions website provides links to vendor solution guides: http://www.xilinx.com/products/design_resources/power_central/index.htm

Saving Power

Lower-power consumption not only reduces power supply requirements but also reduces heat, which increases reliability, allows for smaller form factor packaging, and helps eliminate heat sinks and fans. Spartan-6 FPGAs are designed to minimize power consumption without sacrificing high performance and low cost.

The lowest power state is the quiescent state with no inputs toggling, all outputs disabled, and no pull-up or pull-down resistors in use. This static power state is often dominated by transistor leakage current, which can increase at smaller process geometries. Xilinx has made major advances in the design of the 45 nm Spartan-6 devices. Comparing Spartan-6 FPGAs to Spartan-3A FPGAs, the average static power in Spartan-6 devices is 50% lower. Quiescent current levels are specified in DS162, *Spartan-6 FPGA Data Sheet: DC and Switching Characteristics*. The lower-power Spartan-6 LX devices offer the lowest quiescent current.

Dynamic (active) power is a function of capacitance, voltage, and frequency (CV^2f). In general, capacitance decreases as transistors shrink. But smaller transistors allow users to take advantage of more of them per device, while using faster switching rates, which can lead to increases in dynamic power. Dynamic power consumption can be reduced by reducing the number or frequency of nodes and I/O toggling in a design. Consider the following techniques to eliminate any unnecessary switching in a design and reduce dynamic power:

- Bring all incoming signals to a static state
- Apply rail-to-rail levels to inputs wherever possible
- Turn off as many outputs as possible
- Assign signal standards with small swings to outputs
- Use lower output drive and slower slew rates
- Tie all unused inputs to V_{CCO} or GND outside the device
- Avoid instantiating pull-up and pull-down resistors on I/Os
- Reduce the total length of heavy loaded signals to reduce capacitance
- Disable as many internal oscillating circuits as possible
- Have block RAMs operate in NOCHANGE mode to reduce toggling of the outputs of the block RAM

Saving Clock Routing Power

Clocks are a significant aspect of power consumption because of their high fanout nets and also because controlling them limits the number of logic primitives toggling in a design. If possible, stop the clock where it enters the FPGA, so that it does not consume any FPGA power. If the clock can not be gated externally, then disable it inside the FPGA using the BUFGCE primitive. Avoid using logic to gate clocks, since CLB logic introduces route-dependent skew and makes the design sensitive to the timing hazards of lot-to-lot variations. Minimizing the amount of routing a clock net uses is helpful, since the Xilinx software automatically disables clock nets where possible for unused areas of CLBs.

If possible, minimize the number of DCMs or PLLs required in the design. A single PLL can be shared with both halves of a GTP_DUAL transceiver. A DCM_CLKGEN can be used to dynamically scale clock frequency as needed for the application, helping to minimize power.

ISE Design Suite Power Optimization

Power can also be reduced automatically in the Xilinx design tools. With goal-based implementation, the ISE Design Suite offers a simple, one-step process to specify power optimization. Design Goals and Strategies control the implementation tools by using preset process properties designed to achieve a particular design goal. Power optimization attempts to meet timing constraints as well as reduce power consumption.

For guidelines on design techniques to reduce power consumption, see the Power Solutions page at http://www.xilinx.com/products/design_resources/power_central/index.htm. Many of the techniques described for past FPGA families are also effective for the Spartan-6 FPGAs.