Chapter 19

Power Management Solutions

Overview

While some applications require the lowest possible system cost or highest performance, still other applications require the lowest possible standby power. Spartan®-3 generation FPGAs offer low-power options that balance cost and performance trade-offs.

All Spartan-3 generation FPGAs offer a power management option called Hibernate, which essentially allows all or most of the FPGA logic to be turned off to save power. This option also requires that the FPGA be reconfigured before returning to normal operation, and it does not preserve the state of the FPGA application.

The Extended Spartan-3A family (including the Spartan-3A, Spartan-3AN, and Spartan-3A DSP platforms) offers an advanced power management feature called Suspend mode, which reduces FPGA power consumption while retaining the FPGA's configuration data and application state. While Hibernate provides the most power savings, Suspend retains all data and offers fast wake-up times.

Table 19-1 summarizes the difference between Suspend and Hibernate.

Table 19-1: **Spartan-3 Generation Power-Saving Options**

	Extended Spartan-3A Family Suspend	Extended Spartan-3A Family Hibernate	Spartan-3/3E Family Hibernate
Configuration data retained	Yes	No	
Application state retained (flip-flops, RAM, SRL16)	Yes	No	
Time to exit from power-saving mode	Approximately 500 µs + DCM lock time + programmable timing	FPGA configuration time (tens of milliseconds)	
Power consumption while in power-saving mode	Low	Lowest	
Power supplies	Maintained or scaled down to save additional power	Removed	Removed except for V_{CCO}

Extended Spartan-3A Family Suspend Mode

The Extended Spartan-3A family introduces an advanced power management option called Suspend mode. This section describes the system advantages of Suspend mode. More details on Suspend mode can be found in XAPP480, *Using Suspend Mode in Spartan-3 Generation FPGAs*.

Figure 19-1 graphically demonstrates the effect that Suspend mode has on some example, representative designs measured on a typical XC3S1400A FPGA. The results for other array sizes roughly scale with device density. The Suspend mode primarily affects current consumption on the V_{CCINT} and V_{CCAUX} power rails, there are also power savings for the V_{CCO} rail, depending on the how the user-programmable SUSPEND constraints are defined in the application (see "Define the I/O Behavior During Suspend Mode," page 495).

Figure 19-1 includes three example designs that highlight the Suspend mode behavior:

- **Blank:** A blank FPGA design. No logic is used in this application. A blank design provides the lowest quiescent current and establishes the baseline power consumption.
- **32LVDS:** A design that includes 32 LVDS differential input channels (64 pins) connected to 32 LVDS differential output channels (64 pins). On Extended Spartan-3A family FPGAs, the differential I/O buffers are powered by the V_{CCAUX} voltage supply.
- **32LVDS+8DCM:** A design that includes the circuitry described for 32LVDS plus eight Digital Clock Managers (DCMs). On Extended Spartan-3A family FPGAs, the differential I/O buffers and DCMs are powered by the V_{CCAUX} voltage supply.

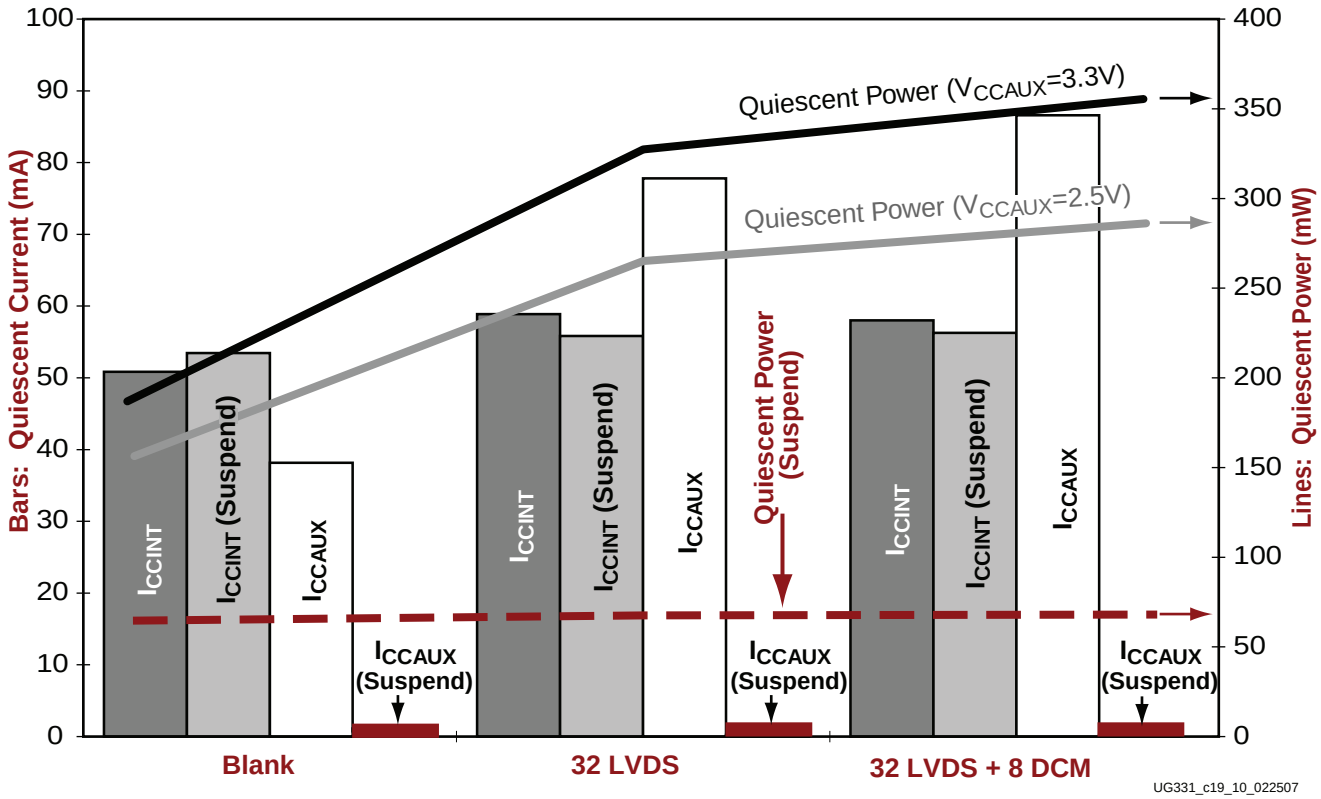

Figure 19-1: **Effects of Suspend Mode on Example Designs Measured on Typical XC3S1400A**

Figure 19-1 also shows four bars, indicating the typical quiescent current on the V_{CCINT} and V_{CCAUX} supplies under normal quiescent conditions with all clocks stopped and during Suspend mode. The associated current measurement, in mA, appears along the left-side vertical axis. Note that the current on V_{CCAUX} during Suspend mode is near the base of the chart, highlighted in burgundy.

Furthermore, Figure 19-1 shows the quiescent power (current multiplied by the voltage applied to each power rail). The associated resulting power measurement, in mW, appears on the right-side vertical axis. On Spartan-3A/3A DSP FPGAs, V_{CCAUX} can be either 2.5V or 3.3V nominally. By physics, the quiescent power is lower when V_{CCAUX} = 2.5V. Note the significant reduction in total power consumption when the Spartan-3A FPGA is in Suspend mode. Although the total power savings is design dependent, Suspend mode typically reduces power consumption by 40% or more, with a minimum power savings of about 20%.

During Suspend mode, some of the circuitry powered by the V_{CCAUX} supply is switched over to the V_{CCINT} supply. Note the **Blank** design example in Figure 19-1. The current on the V_{CCINT} supply actually *increases* while the current on the V_{CCAUX} supply drops significantly! Fortunately, the total V_{CCINT} current during Suspend remains below that used in an active FPGA application. Furthermore, despite the increased V_{CCINT} current, the overall system power is reduced because current is being switched from the 2.5V or 3.3V V_{CCAUX} supply to the 1.2V V_{CCINT} supply.

The power savings are more pronounced in the **32LVDS** and **32LVDS+8DCM** examples and both designs use circuitry that consumes current on the V_{CCAUX} supply.

Suspend Features and Benefits

- Quiescent current is reduced by 40% or more and active current is significantly reduced.
- FPGA configuration data and the present state of the FPGA application during Suspend mode is retained.
- Fast, programmable FPGA wake-up time from Suspend mode, in as little as 500 µs.
- Each user-I/O pin has an individual control that defines how the pin behaves during Suspend mode.
- When enabled in the FPGA bitstream, Suspend mode is externally activated by the system using a single dedicated control pin called SUSPEND. Note that the SUSPEND pin, and therefore Suspend mode, is not available in the VQ100 package.
- The FPGA's AWAKE pin indicates the present Suspend mode status. AWAKE is automatically dedicated when SUSPEND is enabled in the FPGA bitstream.

Design Preparation for Suspend Mode

To use the Suspend feature in an Extended Spartan-3A family FPGA application, follow these steps:

1. Define the I/O Behavior During Suspend Mode in the source design or in a user constraints file (UCF).
2. Define the AWAKE Pin Behavior when Suspend Feature Enabled.
3. Define the SUSPEND Input Glitch Filter setting.
4. Define the Suspend Mode Wake-Up Timing Controls.
5. Enable the Suspend Feature.
6. Generate the FPGA bitstream.

Entering Suspend Mode

Figure 19-2 provides a block diagram of how an Extended Spartan-3A family FPGA enters Suspend mode. Figure 19-3 provides example waveforms.

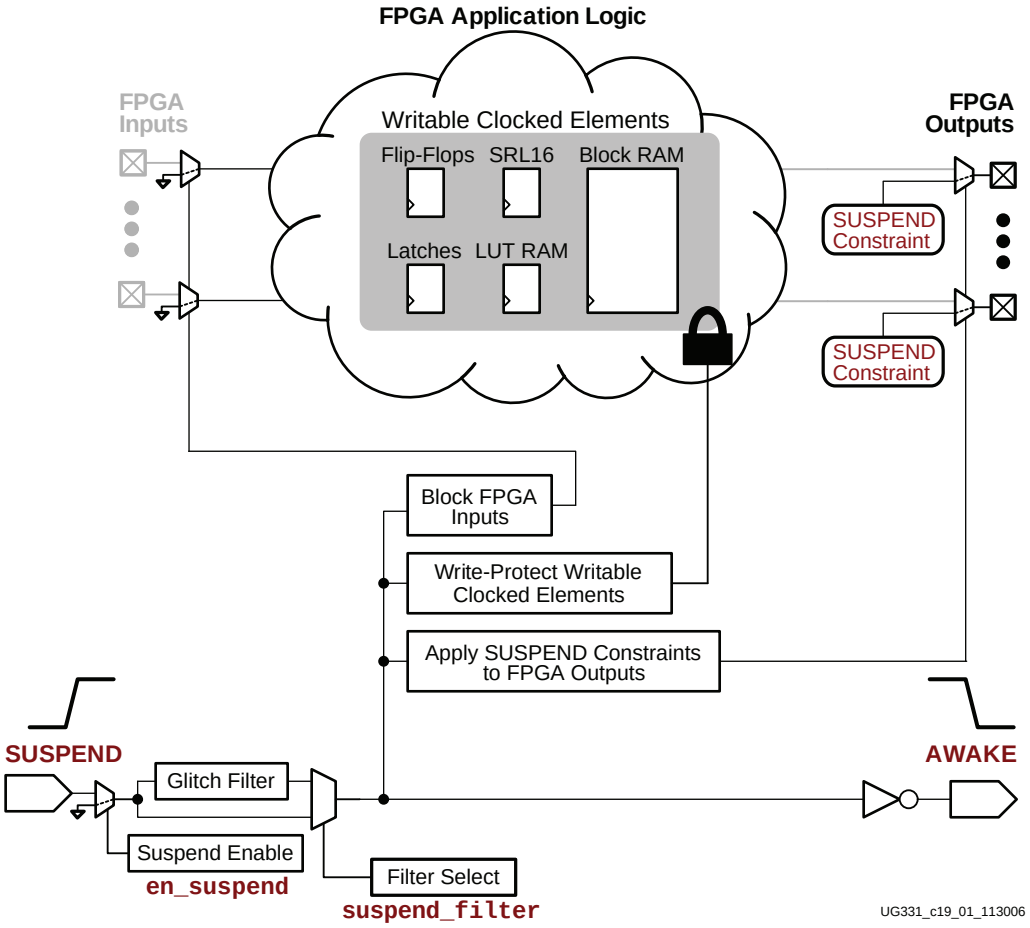

Figure 19-2: **Entering Suspend Mode**

The FPGA can only enter Suspend mode if enabled in the configuration bitstream (see "Enable the Suspend Feature"). Once power is applied to the system, the FPGA always powers up and configures regardless of the value applied to the SUSPEND pin. Once enabled via the bitstream, the FPGA unconditionally and quickly enters Suspend mode if the SUSPEND pin is asserted. If Suspend is not enabled in the bitstream, the SUSPEND input will have no effect and the AWAKE pin will be usable as a general-purpose I/O.

When the FPGA enters Suspend mode, all nonessential FPGA functions are shut down to minimize power dissipation. The FPGA retains all application state and configuration data while in Suspend mode. All writable clock elements are write-protected against spurious write operations. All FPGA inputs and interconnects are shut down.

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 Constraint".

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 does not lose its configuration data.

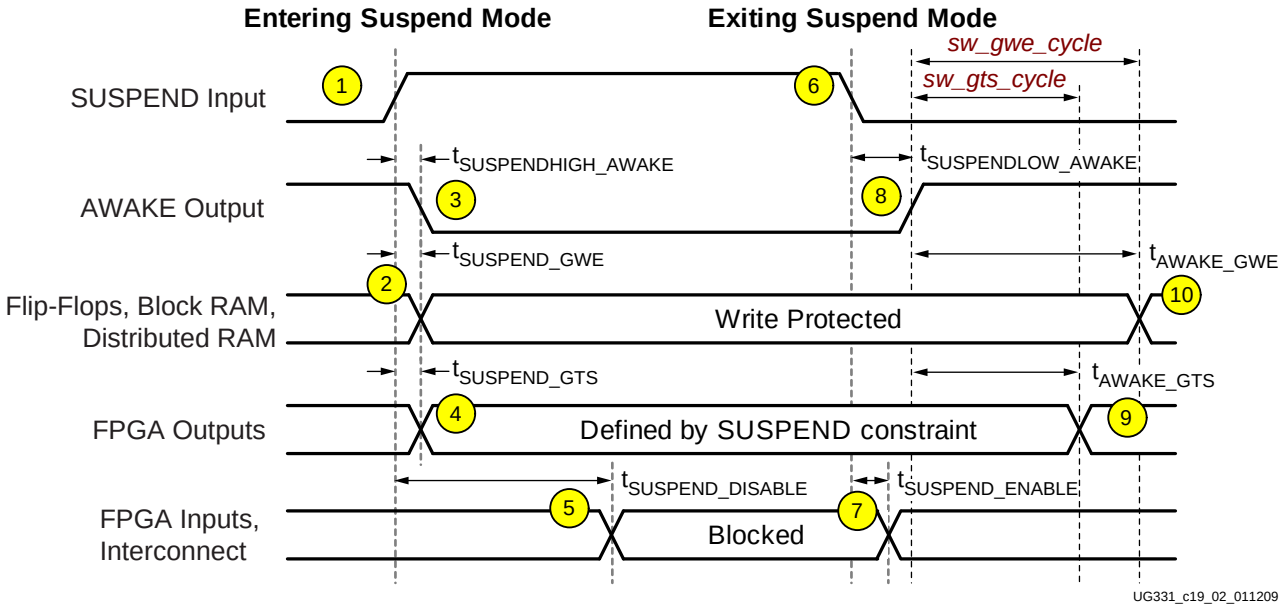

Figure 19-3: **Suspend Mode Waveforms (Entering and Exiting)**

Items 1 through 5 in Figure 19-3 are described below:

1. An external signal drives the FPGA's SUSPEND pin High, unconditionally forcing the FPGA into the power-saving Suspend mode. Data values are captured for I/O pins with a SUSPEND constraint set to DRIVE_LAST_VALUE; however, this value is not presented until Step 4.
2. In response to the SUSPEND input going High, the FPGA immediately write protects and preserves the states of all clocked elements. The states of all flip-flops, block RAM, distributed RAM (LUT RAM), shift registers (SRL16), and I/O latches are preserved during Suspend mode.
3. The FPGA drives the AWAKE output Low to indicate that it is entering SUSPEND mode.
4. The FPGA switches the normal behavior of all outputs over to the Suspend mode behavior defined by the SUSPEND constraint assigned to each I/O. See "Define the I/O Behavior During Suspend Mode," page 495.
5. FPGA inputs are blocked and the interconnects shut off to prevent any internal switching activity.

Exiting Suspend Mode

There are two possible ways to exit Suspend mode in a powered system:

1. Drive the SUSPEND input Low, exiting Suspend mode normally.
2. Pulse the PROG_B input Low, resetting the FPGA and causing the FPGA to reprogram.

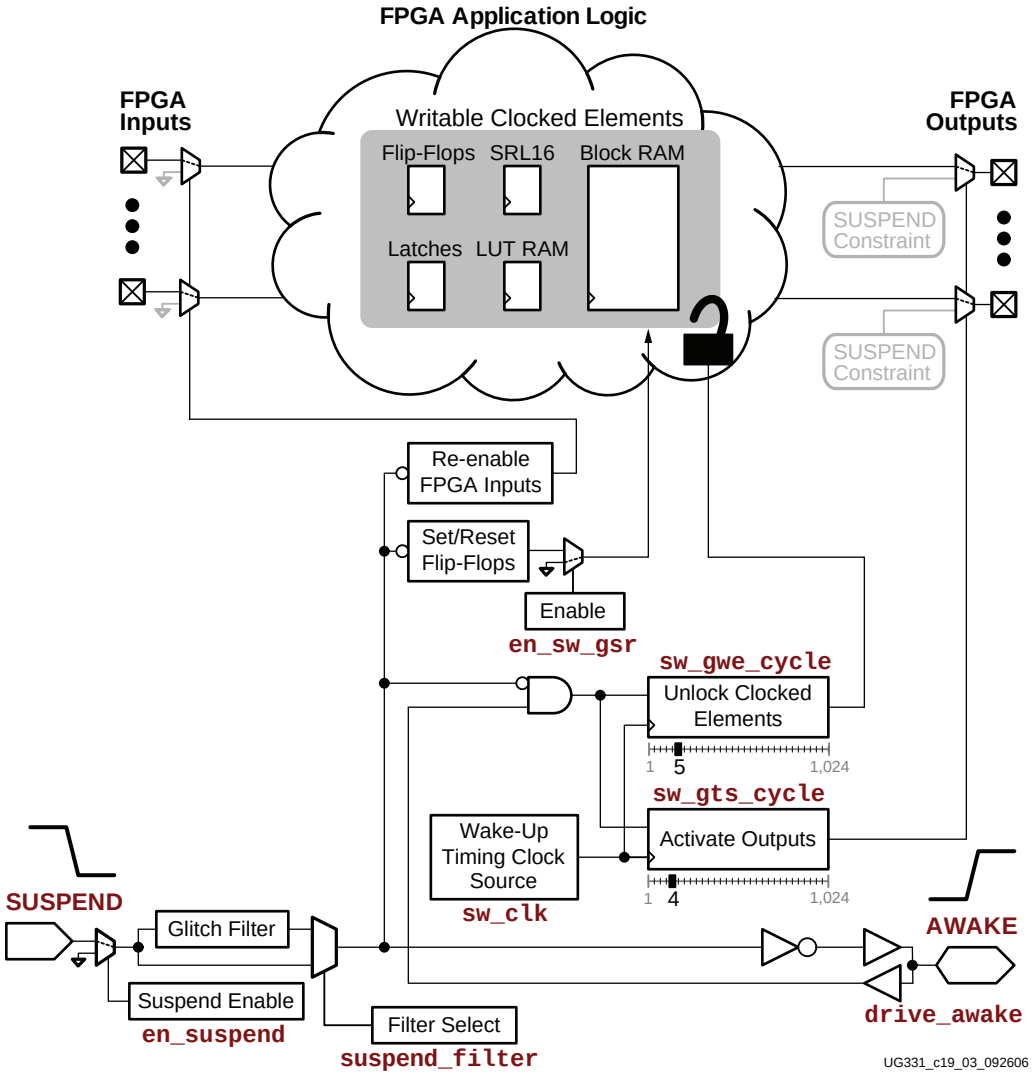

Figure 19-4: **Exiting Spartan-3A/3AN/3A DSP Suspend Mode**

Figure 19-4 is a block diagram showing how to exit Suspend mode using the SUSPEND pin.

When SUSPEND goes Low, the FPGA automatically re-enables all inputs and interconnects.

If enabled in the FPGA bitstream, all flip-flops are optionally, globally set or reset according the FPGA design description. By default, the flip-flops are not globally set or reset, which preserves the state of the FPGA application before entering Suspend mode.

The remaining wake-up process depends on the logic value applied to the AWAKE Pin. Once AWAKE goes High, two user-programmable timers define when FPGA outputs are

re-enabled and when the write-protect lock is released from all writable clocked elements. The wake-up timing clock source is also programmable.

Items 6 through 10 correspond to the markers in Figure 19-3, page 491:

6. The system drives the FPGA's SUSPEND input Low, causing the FPGA to exit Suspend mode.
7. The FPGA releases the inputs and interconnect, allowing signals to propagate internally. There is no danger of corrupting the internal state because all clocked elements are still write protected.
8. 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 goes High.
9. The FPGA switches output behavior from the specified SUSPEND Constraint 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 normal FPGA application. By default, the outputs switch over four clock cycles after AWAKE goes High. The outputs are generally released before the clocked elements to allow signals to propagate out of the FPGA.
10. The writable, clocked elements 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 (SRL16), and I/O latches. By default, the clocked elements are released five clock cycles after AWAKE goes High. Generally, the write-protect lock should be held until after outputs are enabled.

It is good design practice to apply a Reset to any design DCMs after exiting the Suspend mode.

PROG_B Programming Pin Always Overrides Suspend Mode

Pulsing the PROG_B programming pin Low always overrides Suspend mode and forces the FPGA to restart configuration. Likewise, power-cycling the FPGA also restarts configuration.

Suspend Mode Timing Example

Table 19-2 provides example, typical timing for the Extended Spartan-3A Family FPGA Suspend feature. Refer to the *Spartan-3A FPGA Family Data Sheet* (DS529), the *Spartan-3AN FPGA Family Data Sheet* (DS557), and the *Spartan-3A DSP FPGA Family Data Sheet* (DS610) as the official sources of these values.

Table 19-2: **Suspend Mode Timing Parameters**

Symbol	Description	Min	Typ	Max	Units
Entering Suspend Mode					
TSUSPENDHIGH_AWAKE	Rising edge of SUSPEND pin to falling edge of AWAKE pin without glitch filter (***suspend_filter*:*No***)	–	7	–	ns
TSUSPENDFILTER	Adjustment to SUSPEND pin rising edge parameters when glitch filter enabled (***suspend_filter*:*Yes***)	+160	+300	+600	ns
TSUSPEND_GWE	Rising edge of SUSPEND pin until FPGA output pins drive their defined SUSPEND constraint behavior	–	10	–	ns
TSUSPEND_GTS	Rising edge of SUSPEND pin to write-protect lock on all writable clocked elements	–	< 5	–	ns
TSUSPEND_DISABLE	Rising edge of the SUSPEND pin to FPGA input pins and interconnect disabled	–	340	–	ns
Exiting Suspend Mode					
TSUSPENDLOW_AWAKE	Falling edge of the SUSPEND pin to rising edge of the AWAKE pin. Does not include DCM lock time.	–	4 to 108	–	µs
TSUSPEND_ENABLE	Falling edge of the SUSPEND pin to FPGA input pins and interconnect re-enabled	–	3.7 to 109	–	µs
TAWAKE_GWE1	Rising edge of the AWAKE pin until write-protect lock released on all writable clocked elements, using ***sw_clk*:*InternalClk*** and ***sw_gwe_cycle*:*1***.	–	67	–	ns
TAWAKE_GWE512	Rising edge of the AWAKE pin until write-protect lock released on all writable clocked elements, using ***sw_clk*:*InternalClk*** and ***sw_gwe_cycle*:*512***.	–	14	–	µs
TAWAKE_GTS1	Rising edge of the AWAKE pin until outputs return to the behavior described in the FPGA application, using***sw_clk*:*InternalClk*** and ***sw_gts_cycle*:*1***.	–	57	–	ns
TAWAKE_GTS512	Rising edge of the AWAKE pin until outputs return to the behavior described in the FPGA application, using***sw_clk*:*InternalClk*** and ***sw_gts_cycle*:*512***.	–	14	–	µs

Enable the Suspend Feature

The Suspend power-saving feature must first be enabled in the FPGA bitstream before it can be used. By default, the Suspend feature is disabled, SUSPEND has no effect, and the AWAKE pin is usable as a general-purpose I/O. When Suspend is enabled, the software will not allow use of the AWAKE pin for I/O.

Via User Constraints File (UCF)

Suspend mode is enabled and the SUSPEND Input Glitch Filter option is defined using a CONFIG statement in a user constraints file (UCF). Table 19-3 shows the available options. This is the recommended method for enabling Suspend mode as this constraint also automatically reserves the AWAKE pin.

```
CONFIG ENABLE_SUSPEND = “FILTERED” ;
```

Figure 19-5: **UCF Constraint Defining Suspend Mode Behavior for an I/O pin**

Table 19-3: **Available Options for the ENABLE_SUSPEND Constraint**

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	

Via BitGen

Caution! 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

Define the I/O Behavior During Suspend Mode

Use a SUSPEND Constraint to define the behavior of each pin to be programmed differently than the default, 3STATE.

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 19-4. The default behavior is for the pin to be high impedance during Suspend mode although other options are available.

Table 19-4: **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 "true" differential I/O standards (LVDS, RSDS, mini-LVDS, PPDS, TMDS) are high impedance, using one of the 3STATE attributes described in Table 19-4. The DRIVE_LAST_VALUE attribute is not supported for differential output drivers.

Treat the pseudo-differential I/O standards, such as BLVDS, LVPECL, DIFF_HSTL, and DIFF_SSTL, as two single-ended I/O pins. All the attributes apply as for "Single-Ended I/O Standards" although the settings must be set appropriately for 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 Constraint

The SUSPEND constraint allows each pin to have an individually defined behavior during Suspend mode. The available options are in Table 19-4, page 495.

UCF Example

Figure 19-6 shows an example UCF constraint that defines the Suspend mode behavior for a specific pin. The SUSPEND constraint can be included on the same UCF line as other constraints for a pin.

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

Figure 19-6: **UCF Constraint Defining Suspend Mode Behavior for an I/O pin**

More Information

For additional information on the SUSPEND constraint, see the Constraints Guide for the latest software version (www.xilinx.com/support/software_manuals.htm).

- **Constraints Guide** for ISE® 10 Software
  http://toolbox.xilinx.com/docsan/xilinx10/books/docs/cgd/cgd.pdf

Application State Retained during Suspend Mode

When entering Suspend mode, all writable clocked elements are write-protected. The state of all clocked memory elements is retained during Suspend mode.

- Logic block flip-flops
- I/O block latches and flip-flops
- Logic block distributed RAM (LUT RAM)
- Logic block shift registers (SRL16)
- Block RAM and registers

When exiting Suspend mode, all writable clocked elements are re-enabled, controlled by the ***sw_gwe_cycle*** setting.

An additional bitstream option called ***en_sw_gsr*** controls whether all clocked elements are globally set or reset when the FPGA awakens from Suspend mode. By default, ***en_sw_gsr:No***, which means that clocked elements are not set or reset when the FPGA awakens and all states are preserved.

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 (sw_clk)

The wake-up timing when exiting Suspend mode is controlled by a selectable clock source as shown in Figure 19-7 and described in Table 19-5. The clock source is defined by up to two bitstream generator options, ***sw_clk*** and sometimes StartupClk.

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

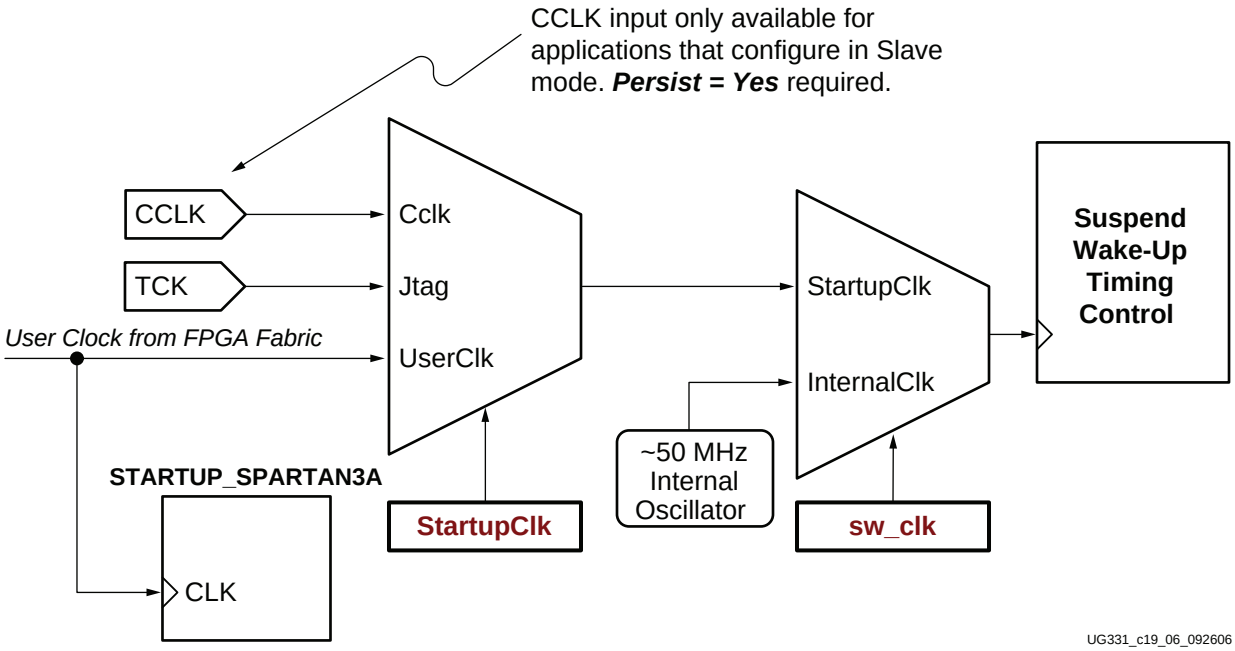

Figure 19-7: **Suspend Mode Wake-Up Timing Control Clock Selection**

- The ***sw_clk*** option is specific to the Suspend feature. By default, ***sw_clk:InternalClk***.
- The ***StartupClk*** option is available on every application. By default ***StartupClk:Cclk***. Consequently, the CCLK pin is the default clock source when exiting Suspend mode.

Table 19-5: **Clock Sources to Wake-Up from Suspend Mode**

sw_clk Setting	StartupClk Setting	Clock Source	Restriction
InternalClk	--	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_SPARTAN3A design primitive	The clock input to the STARTUP design primitive can originate from any nonclocked signal in the FPGA. It cannot originate from a flip-flop source because all clocked elements are write-protected while in Suspend mode.

Switch Outputs from Suspend to Normal Behavior (sw_gts_cycle)

The Suspend/Wake ***sw_gts_cycle*** bitstream option controls when I/O pins are released from their SUSPEND constraint settings and returned to normal operation. The timing is controlled by the "Wake-Up Timing Clock Source (sw_clk)" described above. The default ***sw_gts_cycle*** setting is 4 cycles, but this control can be set for any value between 1 and 1,024 clock cycles.

This control becomes active after the AWAKE pin goes High. After the specified number of clock cycles, all output, open-drain output, and bidirectional I/O pins transition from their Suspend behavior, individually specified using a SUSPEND Constraint, back to the normal behavior specified in the original FPGA application.

It is best to release the outputs before releasing the write-protect lock on all clocked elements.

Release Write Protect on Clocked Elements (sw_gwe_cycle)

The Suspend/Wake ***sw_gwe_cycle*** bitstream option controls when the write-protect lock is released on all clocked elements.

The timing is controlled by the Wake-Up Timing Clock Source (sw_clk) described above. The default ***sw_gwe_cycle*** setting is 5 cycles, but this control can be set for any value between 1 and 1,024 clock cycles.

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

If the ***en_sw_gsr:yes*** option was set, then the clocked elements are already globally set or reset to the value specified in the original FPGA design before the write-protect lock is released. If ***en_sw_gsr:no***, then the state of the FPGA before entering Suspend mode is preserved.

It is best to release the outputs before releasing the write-protect lock on all clocked elements.

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
- PROG_B pin

SUSPEND Pin

When the Suspend feature is enabled (see "Enable the Suspend Feature," page 494), 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 19-6 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.

Table 19-6: **SUSPEND Pin Functionality**

en_suspend Setting	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.
yes 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.
	1	Force the FPGA to enter power-saving Suspend mode.

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 PUDC_B control has no affect 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.

Via User Constraints File (UCF)

The SUSPEND filter is set as part of the ENABLE_SUSPEND constraint, as described in "Via User Constraints File (UCF)," page 494.

Bitstream Generator (BitGen) Option

The filter can also be enabled via a bitstream generator option:

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

Effect on FPGA Configuration

Suspend mode is activated by an FPGA configuration bitstream option. Consequently, the SUSPEND pin has no effect on configuration.

If Suspend mode is enabled in the bitstream and the SUSPEND pin is High, the FPGA successfully configures and then immediately enters Suspend mode. The FPGA's DONE pin will be High, but the AWAKE pin will be Low.

Tie SUSPEND to GND if not Using Suspend Mode

If not using Suspend mode, connect the SUSPEND pin to GND. Do not leave the pin floating.

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 Enabled

If the Suspend feature is enabled, then the AWAKE pin indicates the present state of the FPGA, as summarized in Table 19-7. The AWAKE pin cannot be used by the FPGA application as a general-purpose I/O pin.

Table 19-7: **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 19-8. This behavior is controlled by a bitstream generator (BitGen) option:

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

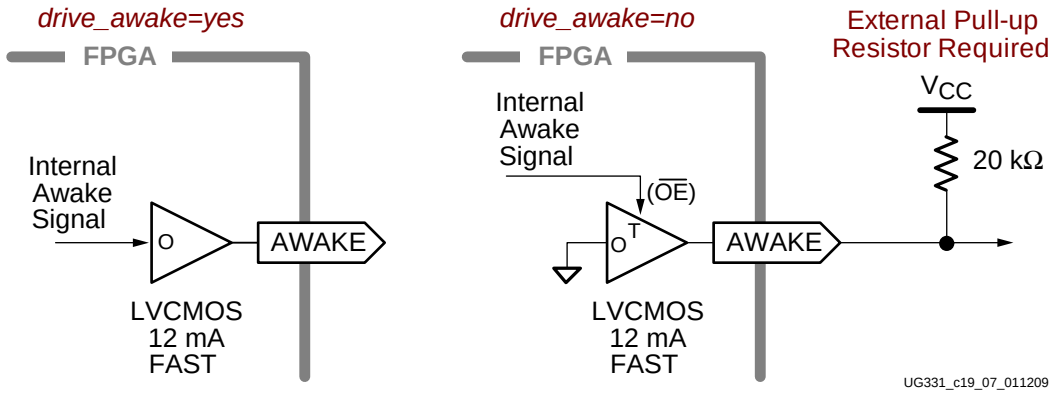

Figure 19-8: **AWAKE Output Drive Options if Suspend Mode Enabled**

The AWAKE output pin is supplied by the V_{CCO} power rail on bank 2 when Suspend mode is enabled.

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

Controlling Wake-Up from an External Source

By default ***drive_awake*:*no***. When ***drive_awake*:*no***, 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. The resistor value should be high to minimize the amount of current flow during Suspend mode. 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 until AWAKE goes High.

Holding the AWAKE pin Low delays the transition from Suspend mode to Active mode and allows an external controller to decide when to awaken the FPGA.

JTAG Operations Allowed During Suspend Mode

Table 19-8 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 19-8: **JTAG Operations Allowed during Suspend Mode**

Boundary Scan Command	Description
IDCODE	Read the JTAG ID code that describes the Spartan-3A/3AN/3A DSP 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.

Post-Configuration CRC Limitations When Using Suspend Mode

If an application uses the post-configuration CRC feature and an error occurs, do not enter Suspend mode. The FPGA will not wake from Suspend mode without reprogramming, such as asserting PROG_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. If the post-configuration CRC feature is enabled, externally gate the SUSPEND pin input with the INIT_B pin. The post-configuration CRC feature signals an error by driving the INIT_B pin Low. The external gate ensures that the SUSPEND pin cannot drive High when the INIT_B pin is Low. Enable the SUSPEND Input Glitch Filter to avoid a possible race condition between the SUSPEND and INIT_B pins.

For more information, see the "Configuration CRC" chapter in UG332: *Spartan-3 Generation Configuration User Guide.*

Suspend Mode Bitstream Generator Options

Table 19-9 summarizes the various bitstream options associated with Suspend mode.

Table 19-9: **Suspend Mode Bitstream Generator Options**

Suspend Mode Bitstream Options	Options (*default*)	Description
en_suspend	***No***	Suspend mode is not used in this application. Connect the SUSPEND pin to GND.
	Yes	Enables the power-saving Suspend feature, controlled by the SUSPEND pin.
drive_awake	***No***	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 present status by actively driving the AWAKE output.
suspend_filter	***Yes***	Enables the glitch filter on the SUSPEND pin.
	No	Disables the glitch filter on the SUSPEND pin.
en_sw_gsr	***No***	The state of all clocked elements in the FPGA is preserved.
	Yes	Pulses the GSR signal during wake-up, setting or resetting all clocked elements, as originally specified in the FPGA application. The GSR pulse occurs before the AWAKE pin goes High and before the ***sw_gwe_cycle*** and ***sw_gts_cycle*** settings are active.
sw_clk	***StartupClk***	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	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 elements (flip-flops, block RAM, etc.). The default value is five clock cycles after the AWAKE pin goes High. Generally, this value is equal to or greater than the ***sw_gts_cycle*** setting.
sw_gts_cycle	1,..,***4***,...,1024	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 Constraint 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.

FPGA Voltage Requirements During Suspend Mode

During Suspend mode, the V_{CCINT} and V_{CCAUX} rails must remain powered at their specified data sheet levels. V_{CCO} for bank 2 should also be maintained since it powers the AWAKE pin. However, the V_{CCO} supply to the other three I/O banks can be potentially turned off to conserve additional power, depending on system requirements. Optionally,

V_{CCO} can be reduced to 1.0V during SUSPEND mode, but this also affects the voltage levels for any output pin with a SUSPEND=DRIVE_LAST_VALUE constraint.

The FPGA's power-on reset (POR) circuit continues to monitor the V_{CCINT} and V_{CCAUX} supplies. 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.

Supply Requirements During Suspend Mode

When entering Suspend mode, the FPGA exhibits the following characteristics on the V_{CCINT} and V_{CCAUX} power rails:

- The current required on the V_{CCAUX} supply drops significantly as the internal FPGA circuits powered by V_{CCAUX} are internally switched over to the V_{CCINT} supply during Suspend mode.
- The current required on the V_{CCINT} supply increases slightly from its quiescent current level.

Hibernate

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 application state.

Forcing FPGA to Quiescent Current Levels

Pulse PROG_B Low to achieve the quiescent current levels. Driving PROG_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 PROG_B, disable the internal pull-up resistors on all I/Os by driving the pull-up resistor control input High. The specific signal name varies by product family: PUDC_B for the Extended Spartan-3A family, HSWAP for the Spartan-3E family, or HSWAP_EN for Spartan-3 devices. Holding PROG_B Low continues clearing the configuration memory. To minimize quiescent current, release PROG_B High but hold off configuration 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 PROG_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 there are a few options. If using Spartan-3A, Spartan-3AN, or Spartan-3A DSP FPGAs, use the Extended Spartan-3A Family Suspend Mode. If using Spartan-3E or Spartan-3 FPGAs, do not assert PROG_B. If all other test conditions are met (for example, no internal switching, I/Os are off), quiescent current levels are close to or slightly above the data sheet quiescent levels. Ensure that internal pull-up and pull-down resistors on I/O pins are disabled.

Entering Hibernate State

Hibernate starts with the approach described in "Forcing FPGA to Quiescent Current Levels." 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) and V_{CCAUX} (auxiliary) power supplies.

Power is supplied to V_{CCO} lines throughout the hibernation period. Figure 19-9 shows how to put Spartan-3 generation FPGAs into Hibernate.

During the Hibernation period, the V_{CCINT} and V_{CCAUX} rails are turned off. Power FETs with low "on" resistance are recommended to perform the switching action. Configuration data is lost upon entering Hibernate; therefore, the device will reconfigure after exiting the state.

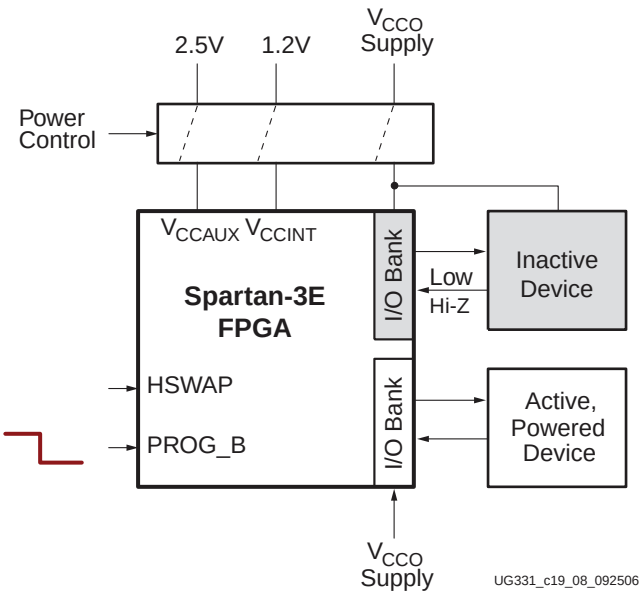

Figure 19-9: **Spartan-3E Hibernate Example**

Holding the PROG_B input Low during the transition into Hibernation period keeps all FPGA output drivers in a high-impedance state. Release PROG_B after re-applying power to the V_{CCINT} and V_{CCAUX} rails. See "Design Considerations," page 507 for recommended levels on Dedicated and Dual-Purpose pins.

Extended Spartan-3A Family FPGA: Turn Off V_{CCO}

Extended Spartan-3A family I/O pins have a floating-well structure, providing full hot-swap/hot-insertion capability. When a an Extended Spartan-3A family 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.

Spartan-3E and Spartan-3 FPGAs: Maintain V_{CCO} on I/O Banks Connected to Powered External Devices

Each user I/O or input-only pin on Spartan-3E and Spartan-3 FPGAs has a power diode between the pin and the associated V_{CCO} rail. The power diodes are present on all signal-carrying pins all of the time. In general, it is safest to maintain V_{CCO} power for all banks throughout the Hibernation period to keep the power diodes inside the I/O block turned off when signals are applied to the pins. In Hibernate, the powered V_{CCO} rails account for little current because the I/Os are in a high-impedance state.

Under certain conditions, it is also possible to switch off the V_{CCO} rail to a particular bank. This action eliminates the V_{CCO} current for those banks (a few milliamperes). There are various ways to achieve this, as shown for the "Inactive Device" in Figure 19-9.

1. Turn off power to any external devices connected to a particular FPGA I/O bank. If both the FPGA I/O bank and the external device are unpowered, there is no current flow.
2. If the external device is powered but the FPGA I/O bank is not, then ensure that all signals driving into the FPGA are either high-impedance (Hi-Z) or that they are under 0.5V. Both cases ensure that there is no current flow through the FPGA's power diodes. Voltages higher than 0.5V can turn on the power diodes. Keep the diodes off to prevent "reverse current" from flowing into the V_{CCO} rail.

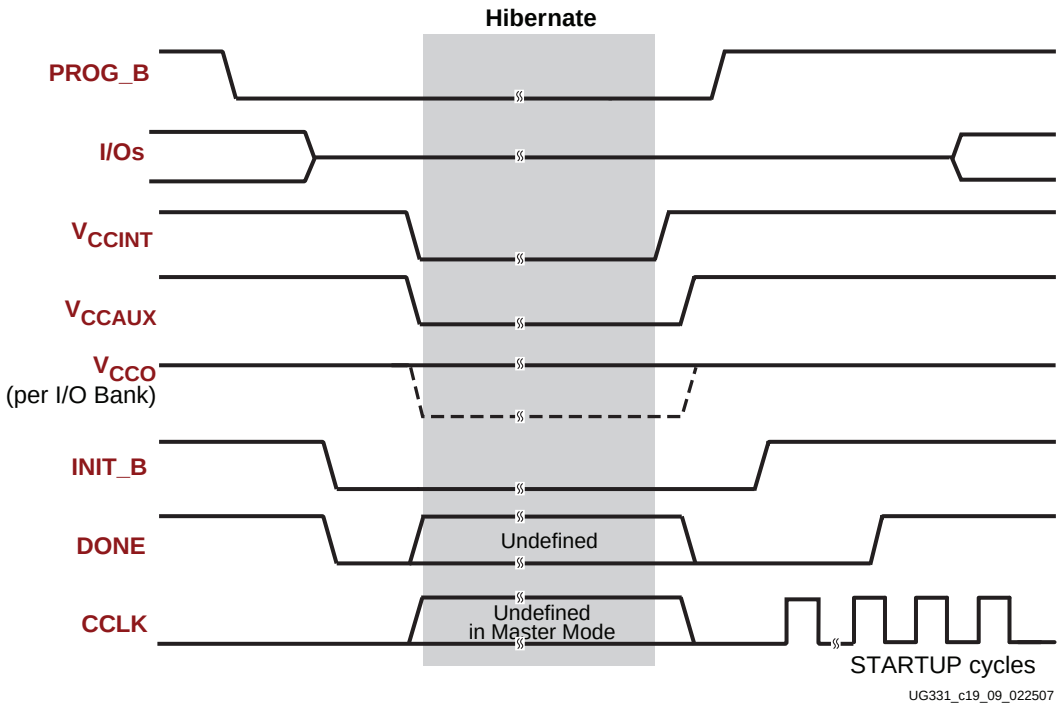

Figure 19-10: **Hibernate Waveform**

Figure 19-10 shows the waveforms for entering and exiting Hibernate. The steps for entering Hibernate are as follows:

1. Pull the PROG_B pin Low to force all user-I/O pins and Input-only 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} and V_{CCAUX} supply rails to the FPGA. Depending on the FPGA product family and the application, it might also be possible to turn off power to the V_{CCO} supply rail.
   - See "Extended Spartan-3A Family FPGA: Turn Off V_{CCO}," page 505
   - See "Spartan-3E and Spartan-3 FPGAs: Maintain V_{CCO} on I/O Banks Connected to Powered External Devices," page 505
4. The FPGA is now in Hibernate. While the FPGA is kept in this state, power consumption rests at the lowest possible level.

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 PROG_B to a High logic level. The rising transition on PROG_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 Startup 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, are powered by a specific V_{CCO} supply input. The Dedicated configuration pins are powered by the V_{CCAUX} supply. If disconnecting power to any of these supplies, consider how that will affect FPGA configuration when power is re-applied.

For specific information on configuration pins and their associated power rails, refer to the "Configuration Pins and Behavior during Configuration" chapter in UG332: *Spartan-3 Generation Configuration User Guide*.

If disconnecting power to V_{CCO} or V_{CCAUX} supplies on Spartan-3 or Spartan-3E FPGAs during Hibernate, do not apply voltages on the pins in excess of 0.5 V to ensure that the power diodes are kept off. This restriction does not apply to Extended Spartan-3A family FPGAs, which have a floating N-well structure for improved hot-swap performance. For more information, see XAPP459: *Eliminating I/O Coupling Effects when Interfacing Large-Swing Single-Ended Signals to User I/O Pins on Spartan-3 Generation FPGAs*.