



Power Management and Calculation for CrossLinkPlus Devices

Technical Note

FPGA-TN-02111-1.0

September 2019

Disclaimers

Lattice makes no warranty, representation, or guarantee regarding the accuracy of information contained in this document or the suitability of its products for any particular purpose. All information herein is provided AS IS and with all faults, and all risk associated with such information is entirely with Buyer. Buyer shall not rely on any data and performance specifications or parameters provided herein. Products sold by Lattice have been subject to limited testing and it is the Buyer's responsibility to independently determine the suitability of any products and to test and verify the same. No Lattice products should be used in conjunction with mission- or safety-critical or any other application in which the failure of Lattice's product could create a situation where personal injury, death, severe property or environmental damage may occur. The information provided in this document is proprietary to Lattice Semiconductor, and Lattice reserves the right to make any changes to the information in this document or to any products at any time without notice.

Contents

Acronyms in This Document	5
1. Introduction	6
2. Power Management Unit	7
2.1. PMU Block	7
2.1.1. PMU State Machine	8
2.1.2. State Transitions	9
2.1.3. Types of Triggers	10
2.1.4. Using the PMU	10
2.2. PMU Module, Primitive and Attributes	13
2.2.1. PMU Port Definitions and Parameters	14
2.2.2. PMU Control Bus and Primitive Attributes	15
2.3. Using Lattice Diamond Clarity Designer UI to Generate PMU	20
3. User Standby Mode during Normal State	23
3.1. Usage of User Standby Mode	23
3.1.1. User Standby Mode for PLL	24
3.1.2. User Standby Mode for Oscillator	24
3.1.3. User Standby Mode for MIPI DPHY	24
3.1.4. User Standby Mode for Inputs and Outputs	24
4. Power Consumption and Calculation	27
4.1. Power Calculator	27
4.1.1. Typical and Worst Case Process	28
4.1.2. Junction Temperature	28
4.1.3. Maximum Safe Ambient Temperature	28
4.1.4. Operating Temperature Range	28
4.1.5. Dynamic Power Multiplier	29
4.1.6. Peak Startup Current	29
4.1.7. Power Budgeting	29
4.1.8. Device Operational Limits	29
4.1.9. Dynamic Power Savings	30
4.1.10. Activity Factor Calculation	30
4.1.11. Power Calculator Assumptions	30
4.2. Using Power Calculator for CrossLinkPlus Devices	31
4.2.1. Power Calculator in Normal State, and User Standby Mode	31
4.2.2. Power Calculator in Sleep State	32
4.2.3. Calculating Average Power	33
5. Thermal Management	34
5.1. Thermal Impedance and Airflow	34
5.2. Thermal Management in Power Calculator	34
5.3. DELPHI Models	35
6. Conclusion	36
References	37
Technical Support Assistance	38
Revision History	39

Figures

Figure 2.1. PMU Block Diagram	7
Figure 2.2. CrossLinkPlus PMU State Machine	8
Figure 2.3. Using USRWKUPN Pin for SLEEP and Wakeup (functional depiction, not timing).....	11
Figure 2.4. Using GPIO for SLEEP when directly routed to PMU (functional depiction, not timing)	12
Figure 2.5. PMU HDL Primitive – PMU.....	13
Figure 2.6. PMU Control Bus Write Operation	16
Figure 2.7. Clarity Designer PMU Selection	21
Figure 2.8. PMU Instance Name Options Window	21
Figure 2.9. PMU Configuration UI.....	22
Figure 3.1. User Standby Mode Use Case Example Block Diagram	23
Figure 3.2. Primitive for Bank Controller for Inputs – Referenced and Differential	25
Figure 3.3. INRDB Simulation Primitive	25
Figure 3.4. Bank Controller for LVDS Outputs Primitive	26
Figure 3.5. LVDSOB Simulation Primitive	26
Figure 4.1. Power Calculator (Summary Tab for CrossLinkPlus Devices).....	27
Figure 4.2. Power Option Controller Window for User Standby Mode options	31
Figure 4.3. Power Calculator – Power Modes/Avg. Power Tab (Power by Supply)	32
Figure 4.4. Power Calculator – Power Modes/Avg. Power Tab (Power by Blocks).....	33

Tables

Table 2.1. PMU HDL Primitive Port Definitions.....	14
Table 2.2. PMU Control Bus Address Mapping.....	15
Table 2.3. PMU Control Register (PMUCR) bit map.....	16
Table 2.4. PMU Control Register (PMUCR) Component Port Definition	17
Table 2.5. PMU Watch Dog Timer Control Register 1 (PMUWDTCR1) Bit Map.....	18
Table 2.6. PMUWDTCR1 Definitions and Settings	18
Table 2.7. PMU Watch Dog Timer Counter 1 (PMUWDTCNT1) Bit Map	18
Table 2.8. PMU Watch Dog Timer Counter 2 (PMUWDTCNT2) Bit Map	19
Table 2.9. PMU Watch Dog Timer Counter 3 (PMUWDTCNT3) bit map.....	19
Table 2.10. PMU Watch Dog Timer Counter 3 (PMUWDTCNT3) bit map.....	19
Table 3.1. Blocks Supporting User Standby Mode	23
Table 3.2. BCINRD Primitive Attributes	25
Table 3.3. BCINRD Simulation Behavior	25
Table 3.4. BCLVDSOB Primitive Attributes.....	26
Table 3.5. BCLVDSOB Simulation Behavior	26

Acronyms in This Document

A list of acronyms used in this document.

Acronym	Definition
AF%	Activity Factor Percentage
AP	Application Processor
DPM	Dynamic Power Multiplier
EBR	Embedded Block RAM
FPGA	Field Programmable Gate Array
GPIO	General Purpose Input/Output
GSR	Global Set-Reset
HDL	Hardware Descriptive Language (Verilog or VHDL)
I/O	Input Output
IP	Intellectual Property
LUT	Look-Up Table
LVDS	Low Voltage Differential Signaling
NCD	Native Circuit Description
OSC	Oscillator
PLL	Phase Locked Loop
PMU	Power Management Unit
PMUCR	PMU Control Register
PMUWDTCR	PMU Watch Dog Timer Control Register
PMUWDTCNT	PMU Watch Dog Timer Counter
SRAM	Static Random-Access Memory
T_J, T_A, T_B, T_C	Junction, Ambient, Board and Case Temperatures
UI	User Interface
USR	User Interface
WDT	Watch Dog Timer

1. Introduction

CrossLinkPlus™ devices from Lattice Semiconductor help stretch the battery life of an application by lowering the power consumption. CrossLinkPlus is a low power FPGA device with a configurable Power Management Unit (PMU) block that places the device in lower power consumption state, Sleep State, to save power, reduce average power consumption and increase battery life for the applications in which it is used.

This document serves as a usage guide for managing and determining power consumption of the CrossLinkPlus device family. The document details the conceptual and functional description along with a guide to utilize the features of the PMU. Power consumption in Lattice Diamond® software Power Calculator, which is the power saving feature of the CrossLinkPlus device family, is also described.

2. Power Management Unit

The Power Management Unit (PMU) is a configurable IP available in CrossLinkPlus devices. The PMU allows the device to be placed in a lower power consumption state called Sleep State when the device is not active. This helps reduce the average power consumption of the device, which can be beneficial in portable or mobile applications.

The Verilog HDL or VHDL module can be generated through the Clarity Designer in Lattice Diamond software. It is configurable based on application requirements. Each configuration settings help set up options unique to the application and can place the device in Sleep State or wake up to active Normal State. PMU works closely with user applications or application processor, providing dynamic options such as wake up using different signals, or when to wake up using timers.

2.1. PMU Block

Power Management Unit has two active states: Normal State and Sleep State. The device is active and running in Normal State. During Sleep State, the device is placed in a low power consuming state for periods when the device is not required to be active or operational.

Figure 2.1 shows an overview of the PMU IP. The functional and usage descriptions of the PMU are detailed in the sections below.

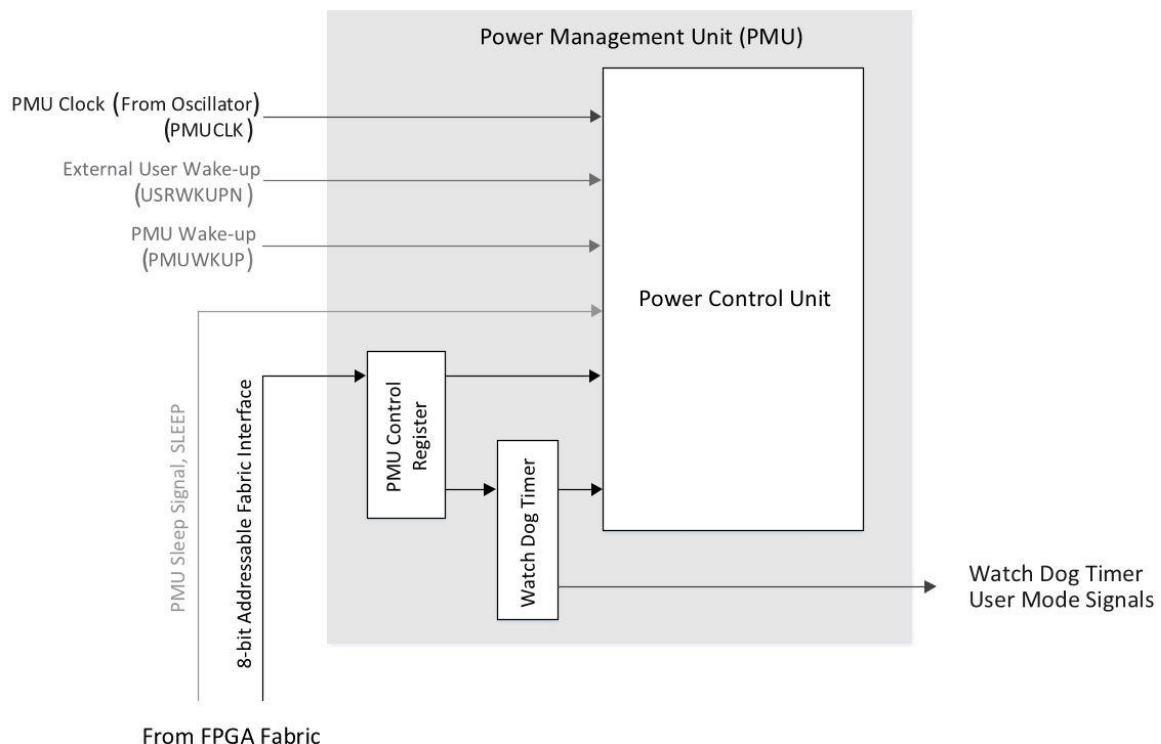


Figure 2.1. PMU Block Diagram

The low speed PMU clock coming in from the internal oscillator is the heart of the PMU. This connection is hardwired from the oscillator to the PMU. The speed of operation for the PMU is fixed at 10 kHz.

The brain of the PMU IP is a Power Control Unit, which essentially controls the various power supplies of the device. Power Control Unit has hardwired connections to internal controls of the power supply switches. These connections are not user accessible.

There are a number of inputs to the Power Control Units. One of the most important ones is the PMU Sleep Signal, SLEEP. This is an active high signal, which when triggered, sends a command to the Power Control Unit to place the device in a low power Sleep State.

There are two wake up signals: External User Wake-Up (USRWKUPN) and PMU Wake-Up (PMUWKUP). PMU Wake-Up (PMUWKUP) is a shared pin (with CDONE pin) that is hardwired to the PMU block.

Along with the Power Control Unit, there are two other blocks inside the PMU: PMU Control Register and Watch Dog Timer.

PMU Control Register

The PMU Control Register, also referred to as PMUCR in this document, stores the configuration settings of the PMU, such as how the PMU wakes up from external pin or Watch Dog Timer expiry, the settings of the Watch Dog Timer, and so on.

PMUCR also has a dedicated 8-bit addressable PMU Control Bus interface to the FPGA fabric. This interface is very useful for applications that demand dynamic update of the PMU configuration as it can be used to update the configuration. The FPGA fabric has to be active, which is in the Normal State, in order to access the PMUCR via this interface. During the Sleep State, the FPGA fabric interface is inactive.

Watch Dog Timer

The Watch Dog Timer (WDT) is the final block of the PMU, and is used to set timers. The PMU can be configured to wake up on the expiry of these WDT timers. An example for a timer would be the timer available in our cell phones. You can set a timer for your device to wake up, say, in 10 minutes. When the PMU places the CrossLinkPlus device in Sleep State, the Watch Dog Timer is available exclusively for the PMU, whether it is set to wake up the device upon expiry or not.

WDT is not used by PMU in the Normal State, but has an option to enable it in user mode. This WDT resource is made available to the application to perform functions that require timer(s). If the SLEEP signal is sent to the PMU, then going to Sleep State takes precedence over the user mode of Watch Dog Timer and it stops all operations and goes to sleep. Any required housekeeping needs to be taken care of at the user end. See the [Using the PMU](#) section.

2.1.1. PMU State Machine

There are two PMU states in CrossLinkPlus devices: Normal State and Sleep State. Both are mutually exclusive and the device can be placed in either one of the states. [Figure 2.2](#) shows the PMU State Machine and triggers that help transition one to the other.

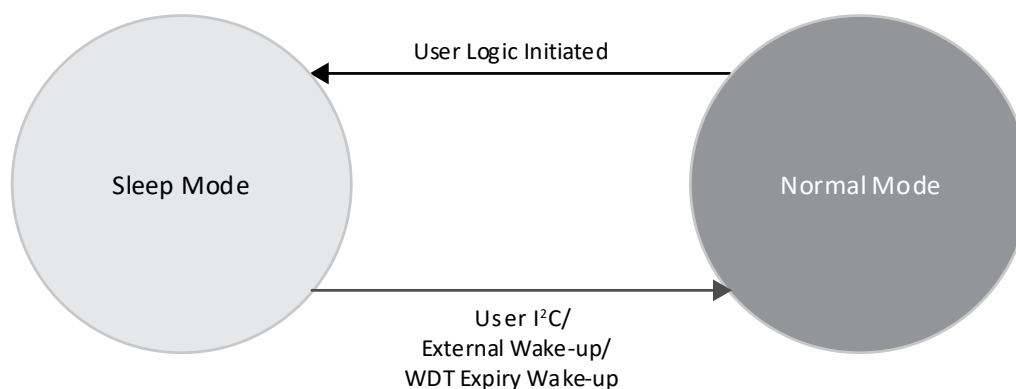


Figure 2.2. CrossLinkPlus PMU State Machine

Each state is discussed in detail below. It is important to understand the transition to each states, and the triggers that change the state from one to the other.

2.1.1.1. Normal State

All elements of the device are active in the Normal State to the extent required by the design. The device is fully active and performing as required by the application.

The device is at its highest power consumption in the Normal State. There are certain blocks of the device that can be placed in User Standby mode if they are not required by the application. See the [User Standby Mode during Normal](#)

State section. These blocks, PLL, I/O, OSC, and MIPI DPHY, can be placed in a lower power state when they are neither used nor needed by the application. This helps save some power in the Normal State; however, User Standby Mode is different from the Sleep State. In the User Standby Mode, these blocks have a smaller static leakage current associated with them. There is no dynamic current because these are inactive.

PMU Sleep State, on the other hand, makes the device inactive and places everything in the lower power. Sleep State overrides the User Standby mode.

2.1.1.2. Sleep State

In the Sleep State, the device is power gated and is not operational. The configuration of the device, the application bitstream that the device is programmed with, and EBR contents are retained. When it transitions to Normal State, the device operates with these contents preserved.

The device is at its lowest power during the Sleep State. This helps reduce the overall power consumption for the device.

The PMU block stays powered on and active along with the associated I/O during Sleep State.

Retention in Sleep State

- CrossLinkPlus devices retain configuration SRAM, EBR contents when in Sleep State.
- All registers within the IPs, EBR registers for example, and PFU registers does not retain their state.
- User I²C registers retain their state.
- The I/O holds the value during the Sleep State. The last value of each I/O is held to avoid issues with other devices connected downstream from the CrossLinkPlus device.
- Differential I/O does not hold their value; all differential I/O is tri-stated.

2.1.2. State Transitions

Figure 2.2 shows the two state transitions to move from one state to the other.

2.1.2.1. Normal to Sleep

This transition happens when you request the device to go to sleep. The trigger can come from an external source or generated within the application when the SLEEP signal is toggled high and the PMU goes to Sleep State.

The PMU can only be placed in Sleep State through the SLEEP signal coming in from the FPGA fabric. There are no other external pins wired directly to place the device to sleep.

Example

Assume an external trigger or signal coming from an Application Processor (AP), requesting the CrossLinkPlus device to be placed in Sleep State. This signal comes in through an I/O and initiates a *go to sleep* command to the application logic. This logic can then initiate the SLEEP signal to PMU through the fabric interface, and the PMU places the device in Sleep State. The signal can be routed directly to the PMU SLEEP input via FPGA fabric in a simple scenario. For more complex design requirements, Lattice recommends a more graceful shutdown of the functions and operations being performed, and then initiate the SLEEP signal to the PMU using a state machine. See the [Using the PMU](#) section for more details.

2.1.2.2. Sleep to Normal

Before looking at triggers needed to wake up the device from the Sleep State, let us look at the device in this state. PMU block, along with associated I/O, are active; the FPGA fabric has been power gated and inactive; the configuration or programming of the device along with EBR contents are preserved; and the I/O is holding the state.

We can wake the device up in numerous ways. One simple way is by using the External User Wake-Up (USRWKUPN) signal. This option can be enabled in the module UI generated through Lattice Diamond Clarity Designer. USRWKUPN is an active low signal where when you toggle it low, it wakes up the device (adherence to timing requirements to wake up the device is required). USRWKUPN is a dedicated pin, and if required by the application, must be enabled in the Clarity Designer UI.

Another way to wake up the device is through the expiration of the Watch Dog Timer (WDT). The Watch Dog Timer is configured through the Clarity Designer UI based on the application requirement. At the expiration of the Watch Dog Timer, it sends a trigger to the Power Control Unit to wake up. In the Diamond Clarity Designer UI, you can specify initialization values of the WDT counters. See the [Using Lattice Diamond Clarity Designer UI to Generate PMU](#) section for more details. If left blank, these are set to 0 second. The values of the timers can be updated dynamically; this has to be taken care of through the application.

The device wake up can also be initiated internally through Always-On power domain blocks – PSEQ Control Block and via user I²C.

The device can always be reconfigured, even in Sleep State, but this wakes it up. This is **not** a recommended method to wake up, because all the changes made in the device are lost. EBR contents and PMUCR settings could be updated during operation of the device. It reconfigures the device to its very first original bitstream and initializes everything. Although not recommended, it can be used to wake up the device.

Sleep State to Normal State transition has multiple triggers; however, there is no priority when it comes to these triggers. All triggers cause the same transition, and hence are dealt with on a first come first serve basis.

2.1.3. Types of Triggers

There are a number of triggers that cause the PMU to change the device from Normal State to Sleep State, and vice-versa. These triggers can be categorized into two main types based on how the trigger is originated.

2.1.3.1. Internal Triggers

These are triggers generated from the logic within the fabric.

Looking at the transition from Sleep State to the Normal State, it is seen that Watch Dog Timer expiry is an internal trigger.

The trigger that changes the device state from Normal to Sleep can be internal if generated within the FPGA fabric. This also contains decoded commands in internal Always-On logic that are sent from user I²C port from an external signal.

2.1.3.2. External Triggers

These are triggers generated from external pin on General Purpose I/O (GPIO) pins.

For example, the Application Processor may require the device to go to Sleep. This signal, when directly routed to the SLEEP port of the PMU, can be considered an External Trigger.

2.1.3.3. Recommendations

It is recommended that all triggers be internal, including ones changing from Normal State to Sleep State. This is further discussed in the [Using the PMU](#) section. Even if the request to place the device to Sleep State is coming off-chip, it is recommended that the request be processed and housekeeping taken care of, before generating a SLEEP signal.

The triggers causing the device to wake up from Sleep State to Normal State can be either internal, WDT expiry, or external, user I²C/ External. The device cannot be woken up via FPGA fabric because the FPGA fabric is powered down.

2.1.4. Using the PMU

The previous sections have touched on a few recommendations for changing states of the device. This section covers typical and recommended use case for the PMU. It is not mandatory to strictly adhere to this use case, but is provided as a guideline and best practices model.

It is essential to include the PMU IP into the FPGA design in order to place the CrossLinkPlus device into the lower power Sleep State. Lattice Diamond Clarity Designer tool can be used to configure the PMU. See the [Using Lattice Diamond Clarity Designer UI to Generate PMU](#) section for more details. When instantiated in the design, PMU is always On, and uses the low speed 10-kHz clock from internal oscillator of the device to perform its operations.

2.1.4.1. Recommended Use Case

A recommended use case scenario is controlling the PMU through a user implemented state machine. This state machine is unique to each application and controls the interface to the PMU through the FPGA fabric interface, SLEEP and 8-bit Addressable PMUCR interface.

An external event or request to go to sleep can come to this state machine. The state machine then processes the request, manages the operation(s) that the CrossLinkPlus device is handling, performs housekeeping of these operation(s), updates the PMUCR, if needed, and then initiates the SLEEP signal.

The available PMU IP does not implement any fixed state machines, because the requirements of each application is unique. PMU IP is a more open and simple interface that can be interfaced with the user state machine that processes the state transitions request. Any dynamic updates, as needed, for the PMU configuration can be done through 8-bit addressable PMUCR. See [PMU Control Register \(PMUCR\)](#) section for more details. For example, the application processor can request the device to go to sleep for 2 hours. If the 2-hour time is different from what was initially programmed in the WDT, the state machine updates the WDT with the new 2-hour time and then initiate SLEEP.

PMUCR can be used to enable and set Watch Dog Timer, and to enable and disable various triggers. This interface is referred to as System Control Bus Interface (SCI), and is accessed via the fabric using CIB interface signals. Usage of SCI interface is discussed in the [PMU Control Bus and Primitive Attributes](#) section.

Lastly, based on PMU configuration, the device can be woken up using I²C interface, or using PMUWKUP or USRWKUPN pins.

2.1.4.2. Using SLEEP Signal

Lattice recommends that SLEEP signal in CrossLinkPlus devices be used under the recommended settings. When not used correctly, the device can stay in permanent Sleep State and not wake up.

Here are a couple of scenarios when SLEEP is directly routed to the PMU.

Routing SLEEP signal to USRWKUPN Pin

In CrossLinkPlus devices, the USRWKUPN pin is a dedicated dual purpose shared pin.

One of the functions of the pin is to be used as User Wake Up port. This function on the pin is hardwired and directly connected to the Power Management Unit (PMU). The other function of this pin is as a General Purpose I/O. To use this pin as User Wake Up port, persistence on this pin must be turned On. Otherwise, it acts as a General Purpose Input/Output (GPIO) pin. Turning On persistence on this pin makes it available as GPIO during Normal State, and as User Wake Up pin during Sleep State.

Take an example when we need to just place the CrossLinkPlus device to sleep. The device is operating in the Normal State; and the SLEEP signal is directly routed to USRWKUPN pin.

The pin can then be used to place the device to Sleep State by toggling the signal high. When the device is in Sleep State, this pin resumes its wake up function (USRWKUPN) and can be used to wake up the device. USRWKUPN is active low and toggling this pin low initiates the trigger for the device to wake up.

So in Normal State, the pin acts as trigger to change state from Normal to Sleep, and when in Sleep State, it can be used to transition from Sleep to Normal.

[Figure 2.3](#) shows a functional depiction of how the change in signal puts the device to sleep and wake up when routed to the same pin (USRWKUPN).

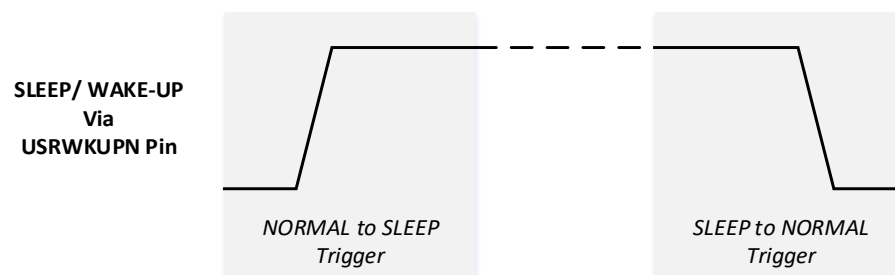


Figure 2.3. Using USRWKUPN Pin for SLEEP and Wakeup (functional depiction, not timing)

Routing SLEEP signal to GPIO

Another use case scenario is when SLEEP signal is routed to a General Purpose Input/Output (or GPIO) for the CrossLinkPlus device.

Take an example where the SLEEP signal needs to be directly routed to PMU. Routing SLEEP signal this way should always be avoided under all conditions, because it will not wake up the device once placed into Sleep State.

Let us look at what effects directly routing SLEEP signal has and a recommended workaround.

The device is operating in the Normal State; and the SLEEP signal is coming in to the PMU via GPIO routed through FPGA fabric. SLEEP signal is going high, and the PMU places the device in Sleep State.

Now, the device needs to be woken up via any trigger. The fabric comes alive and GPIOs would be holding value before going to sleep. The signal drives high, routed through FPGA fabric, and to PMU that triggers the device to go to sleep. The device gets stuck in this permanent Sleep State when the SLEEP signal is using GPIO.

Lattice recommends that you follow the recommended method of using application specific state machine that are discussed earlier in this section. If you need to use a GPIO for the SLEEP signal that needs to be directly fed into the PMU, then register this signal, which is reset by a Global Set-Reset (GSR). By employing a GSR reset register, the PMU never sees the high when waking up the device and this can avoid glitches or placing the device in Sleep State inadvertently. Figure 2.4 shows the block diagram for the same.

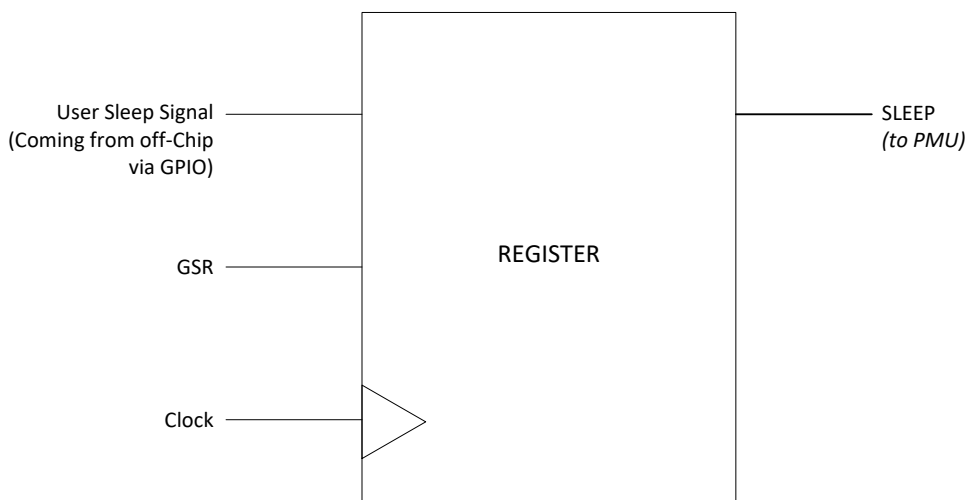


Figure 2.4. Using GPIO for SLEEP when directly routed to PMU (functional depiction, not timing)

CrossLinkPlus devices retain configuration SRAM, EBR contents when in Sleep state. All the registers within the IPs, for example, EBR registers, and PFU registers do not retain their state during the sleep mode. User I²C registers retain their state.

2.1.4.3. Global Clocks, Memories and PMU

Most FPGA designs use memories. The CrossLinkPlus device has built in Embedded Block RAM (EBR) blocks that are complemented by the LUT-based Distributed Memories.

When creating designs that employ EBRs or Distributed RAMs, use global clocks. Use of non-global clocks can potentially cause issues with memory contents retention/corruption issues when changing states. When using non-global clocks, all read or write operations must be completed and stopped before entering the Sleep mode.

The guidelines under [Recommended Use Case](#) section should be strictly followed when using non-global clocks in the designs. User state machine should take care of housekeeping when going into Sleep State. All activities related to memory contents must be stopped before the device goes into Sleep State to avoid potential memory corruption issues.

Lattice Diamond software identifies this situation and generates an appropriate warning for the user.

2.1.4.4. State of the I/O

CrossLinkPlus devices hold the state of the I/O during Sleep State. The last value of each I/O is held to avoid any issues with other devices connected downstream from the CrossLinkPlus devices. The Differential I/O does not hold their value and are all tri-stated.

When the CrossLinkPlus device is coming out of Sleep State, which is device is waking up, the fabric becomes active and it is reset. As the fabric comes out of reset, it starts to drive some of the I/O; the values it drives to can potentially conflict with the last frozen state of the I/O.

This, however, does not affect how the CrossLinkPlus device operates. To avoid potential conflict between frozen I/O and what the fabric drives coming out of the Sleep State, we recommend some housekeeping before placing the device to sleep; such as wrapping up the operations being performed, and ensuring the I/O states before and after Sleep State are the same.

2.1.4.5. Synthesis Attribute for PMU Primitive

You can enable the PMU in your code. Watch Dog Timer user mode in Normal State is optional for PMU.

To avoid optimization of PMU when the user mode for Watch Dog Timer is not enabled, the wrapper should include the `syn_noprune` attribute for the instance. This attribute prevents instance optimization for black-box module with unused output ports.

Here is an example of the syntax:

```
/* synthesis syn_noprune=1 */
```

2.2. PMU Module, Primitive and Attributes

The HDL module gets generated for the PMU and serves as PMU Primitive. It can be used in your design via module generated by Clarity Designer, or as direct PMU Primitive instantiation. The block level diagram for HDL module is shown in [Figure 2.5](#).

Certain inputs and outputs are optional based on the application/design requirements. Clarity Designer provides these options and allows configurations needed for each unique application.

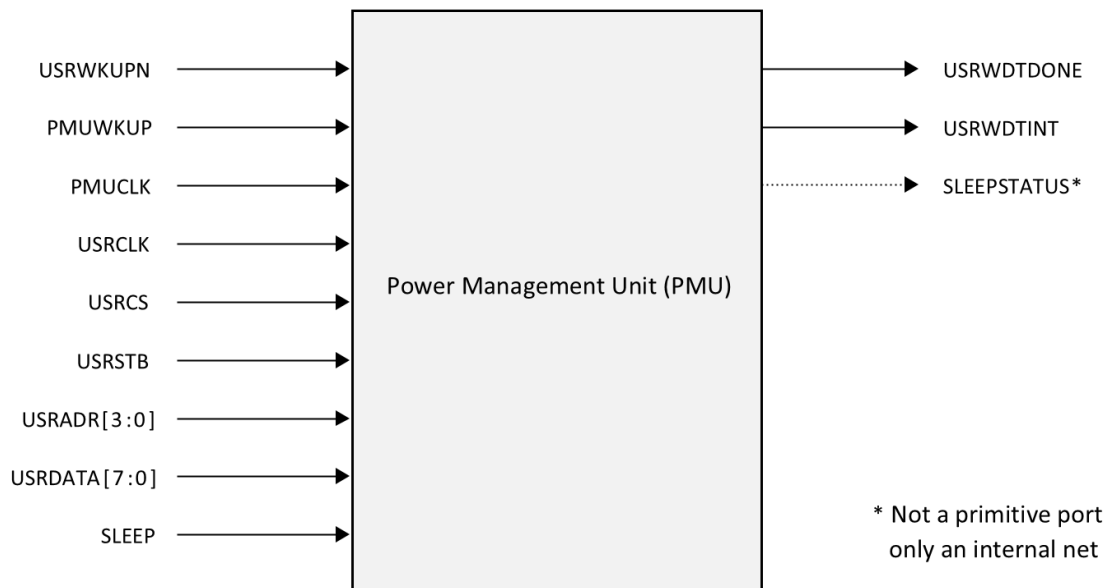


Figure 2.5. PMU HDL Primitive – PMU

2.2.1. PMU Port Definitions and Parameters

Table 2.1 describes the ports available in the PMU, along with their functions, and default values.

When the PMU is not used in a design, it has certain default settings that can be different from the settings when it is used in a design. Table 2.1 provides values for both cases. The FPGA fabric interface that comprises of PMU Control Register and the SLEEP signal is tied low when PMU is not in use. When used, this interface is available to update the PMU settings as needed.

Watch Dog Timer (WDT) User Mode makes the WDT available for use in Normal State. During Sleep State, WDT is available for use by PMU and not available for the application.

Table 2.1. PMU HDL Primitive Port Definitions

Port Name	Description	Width	Direction	Signal Source	Port Default	Optional Port Requirements	Default UI Options	Notes
USRWKUPN	User Provided Wake Up Signal	1	Input	External Pin	1 – Disabled	0 – Enabled 1 – Disabled	1 – Disabled	1, 3
PMUWKUP	PMU Wake Up Signal coming from Configuration Logic	1	Input	Configuration Logic/ User Pin	0 – Disabled	1– Enabled 0 – Disabled	1– Enabled	2, 3
PMUCLK	PMU Clock coming in from Internal Oscillator	1	Input	Internal Oscillator	0	—	—	—
USRCLK	PMU Control Register User Interface – Clock, Chip Select, Standby, Address and Data	1	Input	From FPGA Fabric	0	—	—	—
USRCS		1	Input	From FPGA Fabric	0	—	—	—
USRSTB		1	Input	From FPGA Fabric	0	—	—	—
USRADR[3:0]		4	Input	From FPGA Fabric	0	—	—	—
USRDATA[7:0]		8	Input	From FPGA Fabric	0	—	—	—
SLEEP	Sleep Signal	1	Input	From FPGA Fabric	0	—	—	—
USRWDTDONE	Watch Dog Timer User Mode Done Signal	1	Output	To FPGA Fabric	—	1– Enabled 0 – Disabled	0 – Disabled	3
USRWDTINT	Watch Dog Timer User Mode Interrupt Signal	1	Output	To FPGA Fabric	—	1– Enabled 0 – Disabled	0 – Disabled	3

Notes:

- If you do not select the User Wakeup option in the interface, this port is set to DISABLED (or 1). See the [Using Lattice Diamond Clarity Designer UI to Generate PMU](#) section for more details.
- PMUWKUP is connected to I2C0 (of the two) user I²C which is hardwired to the PMU block.
- The ports available in the wrapper change according to the selections you made in the interface section. Some of these options are as follows:
 - USRWKUPN (active low signal) is disabled by default. Selections in the interface change it as follows:
 - When Enabled (0), the port is available in the wrapper.
 - When Disabled, the port is not brought out as user port, and is connected to '1' inside the wrapper.
 - PMUWKUP is the user I²C wake up. This port is available in the wrapper when you enable the User I²C Wake Up.
 - When Enabled (1), the port is available in the wrapper.
 - When Disabled, the port is not brought out as user port, and is connected to '0' inside the wrapper.
 - USRWDTDONE is the user mode for WDT in Normal mode.
 - When Enabled (1), the port is available in the wrapper.
 - When Disabled, the port is not brought out as user port (and can be left unconnected).
 - USRWDTINT is the trigger for the user mode for WDT in Normal mode.
 - When Enabled (1), the port is available in the wrapper.
 - When Disabled, the port is not brought out as user port (and can be left unconnected).

2.2.2. PMU Control Bus and Primitive Attributes

PMU Control Bus is an 8-bit addressable fabric interface through which the application can access various registers and settings in the PMU. This interface can be used to update the PMU Control Register (PMUCR), before placing the device in the Sleep mode. Updating the PMUCR provides an ability to change these settings. If you do not update the PMUCR before going to Sleep State, then the original settings used during configuration of the PMU are kept. If the PMUCR is updated, then new settings take effect, and the device operation is updated accordingly. Changes made to these settings when device is operational are effective, as long as the device is not rebooted. If the device is rebooted, the initial settings are programmed in the registers.

Additionally, the PMU Control Bus interface can be used to update PMU Watch Dog Control Register 1 (PMUWDTCR1). If the Watch Dog Timers (WDT) is being used in the design, you are required to preset the three 32-bit WDT Count values for the PMU Watch Dog Counters (PMUWDTCNT1, PMUWDTCNT2, and PMUWDTCNT3). The details of each of these registers are discussed in the sections that follow.

One of the features of the PMU is that the settings and the attributes can be changed dynamically through the 8-bit addressable PMU Control Bus interface to the block, based on the application requirement; for example, an alarm or timer settings that can be set for different times. Understanding of these attributes is very essential if you want to change these settings.

Table 2.2 provides the list of addresses and their corresponding usage through the PMU Control Bus interface. Each address corresponds to the 8-bit data that can be written on to those registers. Some addresses are not available in the CrossLinkPlus device family and are indicated as N/A or Not Available.

Table 2.2. PMU Control Bus Address Mapping

Address USRADR[3:0]	Name	Data USRDATA[7:0]	Usage	Access
0000	N/A	—	Not Available	—
0001	PMUCR	8-bit	PMU Control Register	Write
0010	N/A	—	Not Available	—
0011	PMUWDTCR1	8-bit	PMU Watch Dog Timer Control Register1	Write
0100	PMUWDTCNT1_3	8-bit	PMU WDT Count Register1_3	Write
0101	PMUWDTCNT1_2	8-bit	PMU WDT Count Register1_2	Write
0110	PMUWDTCNT1_1	8-bit	PMU WDT Count Register1_1	Write
0111	PMUWDTCNT1_0	8-bit	PMU WDT Count Register1_0	Write
1000	PMUWDTCNT2_3	8-bit	PMU WDT Count Register2_3	Write
1001	PMUWDTCNT2_2	8-bit	PMU WDT Count Register2_2	Write
1010	PMUWDTCNT2_1	8-bit	PMU WDT Count Register2_1	Write
1011	PMUWDTCNT2_0	8-bit	PMU WDT Count Register2_0	Write
1100	PMUWDTCNT3_3	8-bit	PMU WDT Count Register3_3	Write
1101	PMUWDTCNT3_2	8-bit	PMU WDT Count Register3_2	Write
1110	PMUWDTCNT3_1	8-bit	PMU WDT Count Register3_1	Write
1111	PMUWDTCNT3_0	8-bit	PMU WDT Count Register3_0	Write

Typical PMU Control Bus write operation is demonstrated in [Figure 2.6](#).

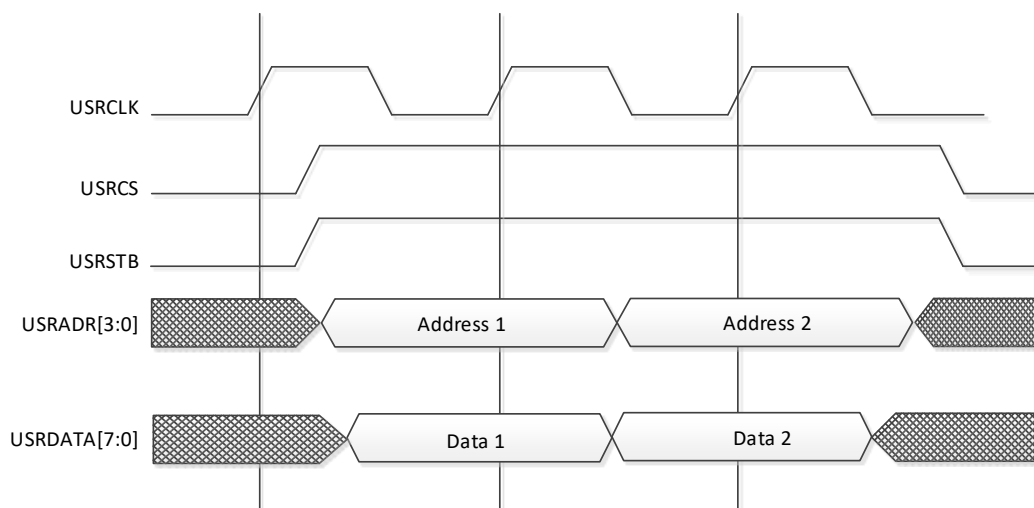


Figure 2.6. PMU Control Bus Write Operation

The PMU Control Bus does not support read operation, since PMU IP does not require data read back functionality.

2.2.2.1. PMU Control Register (PMUCR)

The first address (0001) through the PMU Control Bus interface corresponds to the PMU Control Register (PMUCR). This interface can be used to update the PMUCR before placing the device in the Sleep mode. If you do not update the PMUCR before going to sleep, then the original settings from Lattice Diamond Clarity Designer configuration for the PMU are used. If the PMUCR is updated, then new settings takes into effect, and the device operation is updated accordingly.

PMU Control Register (PMUCR) is an 8-bit user accessible register that can be used to update the settings. These include how PMU wakes up the device through Watch Dog Timer, external wake up signal, and so on.

During the PMU configuration through Lattice Diamond Clarity Designer, the PMUCR is set to a certain value based on the selected UI options. These become the initial settings for the PMU and PMUCR. These can be easily changed through the 8-bit addressable PMU Control Bus interface to the PMU.

[Table 2.3](#) shows what each of the bits of the 8-bit PMUCR stands for. For details on each attribute, check [Table 2.4](#).

Table 2.3. PMU Control Register (PMUCR) bit map

Bit #	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Attribute	PMUEN	UWDTEN	UWDTMD	UWDTINT	EXTEN	N/A	I2CEN	WDTEN

[Table 2.4](#) shows the detailed selections of the all the bits along with various values. PMUCR[7] bit is set to 1 when PMU is instantiated in the design. If PMU is not used, this bit is set to 0 and PMU is disabled.

Table 2.4. PMU Control Register (PMUCR) Component Port Definition

UI Selection	Attribute	PMUCR Bit	Description	Core	UI Option	Value Range	Default Value	Notes
—	PMUEN	PMUCR[7]	PMU Enable	—	—	1– Enabled 0 – Disabled	—	1, 2
Enable User Mode for Watch Dog Timer in <i>Normal State</i>	UWDTEN	PMUCR[6]	User WDT Enable	Core/ Sim	Check Box	1– Enabled 0 – Disabled	0 – Disabled	3
Mode Selection	UWDTMD	PMUCR[5]	User WDT Mode Selection	Core/ Sim	Drop Down Menu	COUNT_ONCE , COUNT_REPE AT	COUNT_ONCE	3
Interrupt Enable	UWDTINT	PMUCR[4]	User WDT Interrupt Enable	Core/ Sim	Check Box	1– Enabled 0 – Disabled	1– Enabled	—
External User Wake Up	EXTEN	PMUCR[3]	External Pin Interrupt Enable	Core/ Sim	Check Box	1– Enabled 0 – Disabled	0 – Disabled	—
—	Not Available	PMUCR[2]	—	—	—	—	0 – Disabled	—
User I2C Wake Up	I2CEN	PMUCR[1]	Slave I2C Interrupt Enable	Core/ Sim	Check Box	1– Enabled 0 – Disabled	1– Enabled	—
Watch Dog Timer (WDT) Expiry Wake Up	WDTEN	PMUCR[0]	Watch Dog Timer Interrupt Enable	Core/ Sim	Check Box	1– Enabled 0 – Disabled	1– Enabled	—

Notes:

- PMUCR [7] is automatically set to ENABLED (1) when PMU is included in the design. If the PMU is not instantiated, then the PMUCR [7] is DISABLED (set to 0). In the disabled state, none of the features or interfaces of the PMU are available.
- When you enable the User mode for the built-in WDT in Normal Mode, UWDTEN is set to ENABLED. The other UI options should get active based on this check box, that is when the User mode for the WDT is ENABLED, and then the Mode Selection and Interrupt Enable Bit option are available for you to set. UWDTEN and WDTEN are applicable in mutually exclusive states (Normal and Sleep), and they apply to the built-in Watch Dog Timer.
- In Normal mode, the Watch Dog Timer is available to access for user logic via PMU Control Bus by enabling the UWDTEN option.
In sleep mode, the user logic is inactive and hence the Watch Dog Timer is not used for user logic. When set properly, WDT can be set to wake up the device from Sleep (to *Normal State*), once the WDT count expires.

2.2.2.2. PMU Watch Dog Timer Control Register 1 (PMUWDTCR1) Settings

The next available address is 0011. It corresponds to the PMU Watch Dog Timer Control Register 1 (PMUWDTCR1). This register is used for selecting the counter that is used by the Watch Dog Counter to wake up the device. The Watch Dog Counter is discussed in the following section.

Although this is an 8-bit register, only the two lowest bits are available to select between the three counters available. [Table 2.5](#) shows the bit mapping for PMUWDTCR1.

Table 2.5. PMU Watch Dog Timer Control Register 1 (PMUWDTCR1) Bit Map

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
N/A						Sleep Counter Selection (SLPCNT)	

Settings for the two bits correspond to the following selections:

- 00 = WDT Disabled.
- 01 = Use PMUWDTCNT1 for WDT for Sleep Mode if WDT is enabled.
- 10 = Use PMUWDTCNT2 for WDT for Sleep Mode if WDT is enabled.
- 11 = Use PMUWDTCNT3 for WDT for Sleep Mode if WDT is enabled.

Table 2.6 lists the options available in the UI for the selection of Watch Dog Timer to use when waking up. When WDT wake option is the selection, the default value is set to Watch Dog Counter # 1.

Table 2.6. PMUWDTCR1 Definitions and Settings

UI Selection	Attribute	PMUWDTCR1 Bits	Description	Core/Sim	UI Option	Value Range	Default Value
Select the Watch Dog Timer (WDT) Counter that wakes up the device	SLPCNT	1:0	Sleep Mode WDT Count Select	Core/Sim	Drop Down Menu	COUNTER_1, COUNTER_2, COUNTER_3	COUNTER_1

As an example, if the data 00110010 is written to the PMUWDTCR1, then the six MSB are ignored (that is 001100) and only the two LSBs are considered (10). These indicate to the PMU that it needs to wake up using the expiry of Watch Dog Counter #2 (PMUWDTCNT2), after Sleep phase initiation.

2.2.2.3. Watch Dog Timer Counters and their Initialization Options

The next set of addresses from 0100 through 1111 on the PMU Control Bus corresponds to the Watch Dog Timer Counters. These are the counters that are set to a specific value in terms of time upon expiry of which the PMU wakes up the device.

There are three Watch Dog Timer Counters. Each counter is 32-bit. Four addresses cover the 32-bit space for each counter, each address corresponding to 8-bits. The three counters are PMUWDTCNT1, PMUWDTCNT2, and PMUWDTCNT3.

The PMU WDT Count Registers, total 32 bits, hold the binary number of PMU clock cycle for built-in Watch Dog Timer to achieve desired timing. For normal application, you should set the counter prior to entering power saving mode, if it is set to be used.

Each counter should be written sequentially from MSB to LSB, for example PMUWDTCNT1_3 to PMUWDTCNT1_0 when writing PMUWDTCNT1. On receiving interrupts, the PMU takes a clock cycle to activate the programmable sequence logic to perform the enter and exit power saving mode, so the logical PMU_WDT count setting should be always larger than 1.

The mapping of each of the counters is shown in the following tables.

PMUWDTCNT1 comprises of four bytes PMUWDTCNT1_3, PMUWDTCNT1_2, PMUWDTCNT1_1, and PMUWDTCNT1_0. The bit mapping is as shown in the Table 2.7.

Table 2.7. PMU Watch Dog Timer Counter 1 (PMUWDTCNT1) Bit Map

PMUWDTCNT1 [31:0]			
PMUWDTCNT1_3 [7:0]	PMUWDTCNT1_2 [7:0]	PMUWDTCNT1_1 [7:0]	PMUWDTCNT1_0 [7:0]

PMUWDCNT2 comprises of four bytes PMUWDCNT2_3, PMUWDCNT2_2, PMUWDCNT2_1, and PMUWDCNT2_0. The bit mapping is shown in the [Table 2.8](#).

Table 2.8. PMU Watch Dog Timer Counter 2 (PMUWDCNT2) Bit Map

PMUWDCNT2 [31:0]			
PMUWDCNT2_3 [7:0]	PMUWDCNT2_2 [7:0]	PMUWDCNT2_1 [7:0]	PMUWDCNT2_0 [7:0]

PMUWDCNT3 comprises of four bytes PMUWDCNT3_3, PMUWDCNT3_2, PMUWDCNT3_1, and PMUWDCNT3_0. The bit mapping is shown in the [Table 2.9](#).

Table 2.9. PMU Watch Dog Timer Counter 3 (PMUWDCNT3) bit map

PMUWDCNT3 [31:0]			
PMUWDCNT3_3 [7:0]	PMUWDCNT3_2 [7:0]	PMUWDCNT3_1 [7:0]	PMUWDCNT3_0 [7:0]

Each of these registers is 32-bit. All values in the 32-bit range are supported and user accessible. [Table 2.10](#) provides the Clarity Designer UI options.

Table 2.10. PMU Watch Dog Timer Counter 3 (PMUWDCNT3) bit map

UI Selection	Attribute	PMUWDCNT Bits	Description	Core/Sim	UI Option	Value Range (hh:mm:ss)	Default Value
WDT Counter 1	PMUWDCNT1 [31:0]	31:0	Preset value for WDT Counter 1	Core/Sim	Text Box	00:00:00 To 08:00:00	00:00:00
WDT Counter 2	PMUWDCNT2[31:0]	31:0	Preset value for WDT Counter 2	Core/Sim	Text Box	00:00:00 To 08:00:00	00:00:00
WDT Counter 3	PMUWDCNT3[31:0]	31:0	Preset value for WDT Counter 3	Core/Sim	Text Box	00:00:00 To 08:00:00	00:00:00

When configuring the PMU module in the Lattice Diamond Clarity Designer, you can preset the values of these registers to a set count value to which you would like your WDT to count. These values serve as the initial values. If the application desires fixed counter values and does not need to change, set these counters in the Clarity Designer. If no value is provided, these are all set to zeros (0s).

Clarity Designer require values to be set in terms of Hours: Minutes: Seconds (HH:MM:SS). At the application level, when accessing the registers through the PMU Control Bus Interface, you need to provide the binary values corresponding to the desired time. The next section describes the conversion between the two.

The maximum value of the counter can be set to eight hours.

PMU Watch Dog Timer Count Calculation

PMU runs on the low frequency clock from the internal oscillator. The default frequency for the Low Speed Oscillator is 10 kHz.

The user input to the WDT counters preset value is in Hours: Minutes: Seconds (HH:MM:SS). Clarity designer automatically converts the given value into corresponding 32-bit number. When the application is accessing the PMUWDCNT registers, 32-bit value is expected.

For directly writing to the PMUWDCNT registers, the time needs to be converted to its corresponding 32-bit value. The formula below can be used for settings these values. This is a simple example that shows the calculation.

Frequency of the PMU Clock, f	=	10 kHz
Time Period, t	=	1/f
	=	100 μ sec
User Input (Time)	=	hh:mm:ss
	=	00:30:00
Total Number of Seconds, T	=	hh*3600+mm*60+ss
	=	0*3600+30*60+0
	=	1800 sec.
Number of clock cycles, n	=	T/t
	=	1800 sec / 100 μ sec
	=	18,000,000
PMUWDCNT0	=	BINARY (n)
	=	BINARY (18,000,000)
	=	00000001 00010010 10101000 10000000

2.3. Using Lattice Diamond Clarity Designer UI to Generate PMU

Lattice Diamond Clarity Designer can be used to generate an HDL wrapper for the PMU.

1. When a project is created, launch the Clarity Designer using **Tools > Clarity Designer** option or by clicking the  button on the toolbar.
2. Provide the name of the project and select the HDL language of choice. The window appears as [Figure 2.7](#).
3. Select the module you wish to generate. In this case, select **pmu** under **Architecture_Modules**, as shown in [Figure 2.7](#).

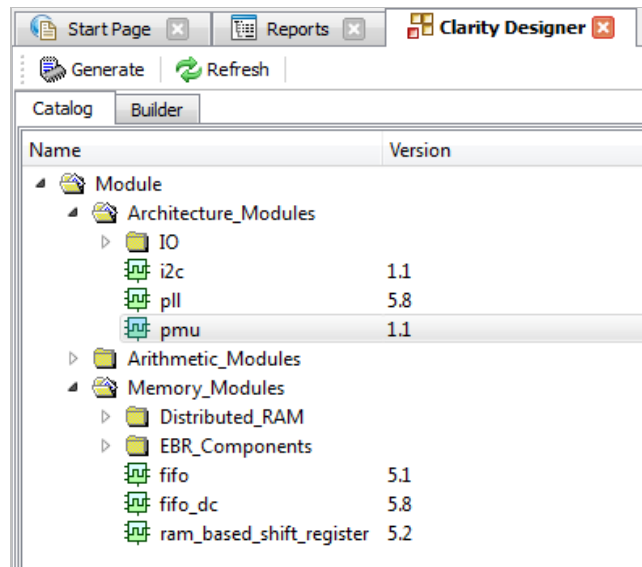


Figure 2.7. Clarity Designer PMU Selection

- Double click on **pmu** to open another window (Figure 2.8) that requires you to provide an instance name. We use *pmu_test_instance* as our instance name for example.
- Click **Customize**.

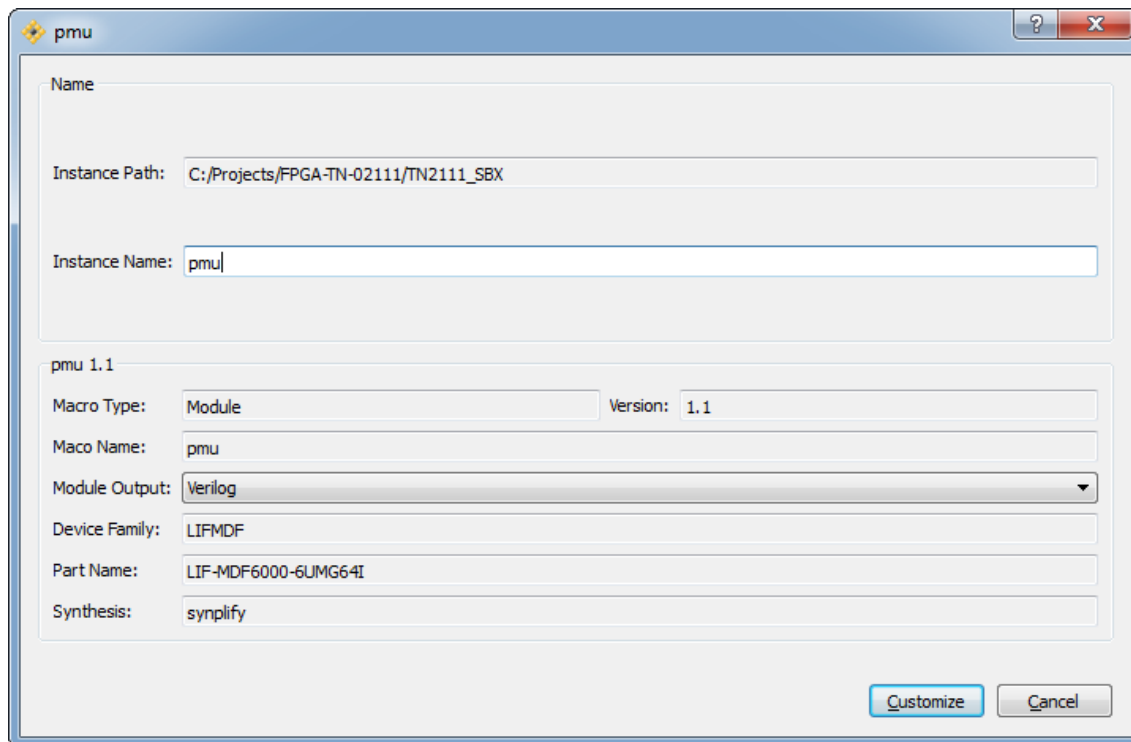


Figure 2.8. PMU Instance Name Options Window

- The next screen shows the various options to configure the PMU. All these options have been discussed in the earlier sections. Figure 2.9 shows the **Configuration UI**.

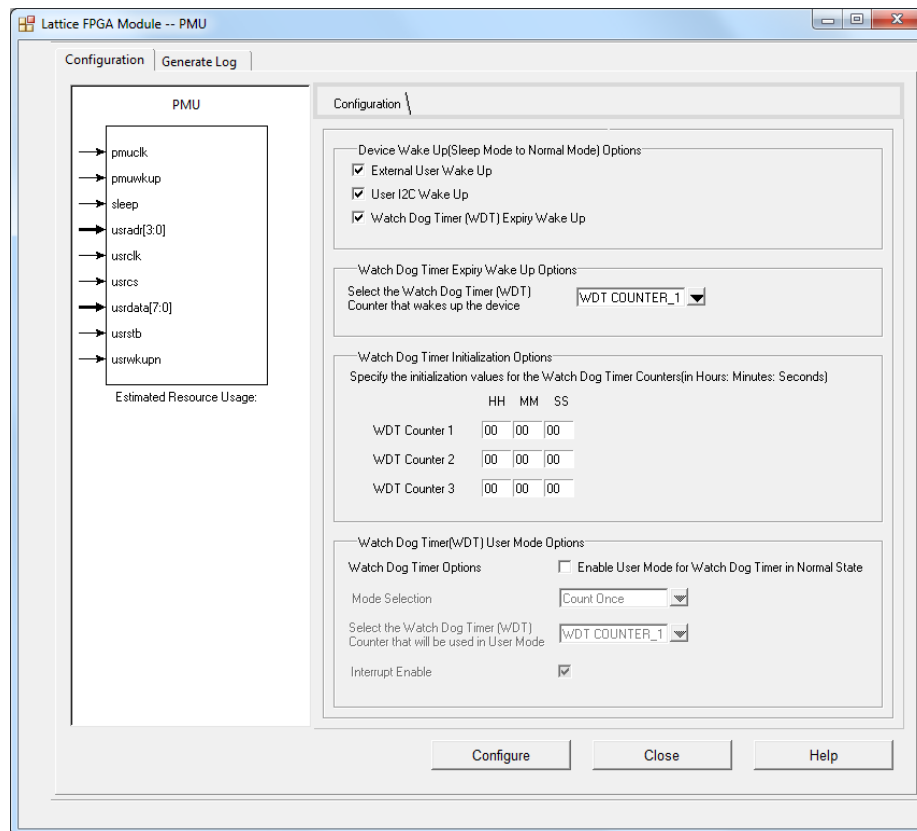


Figure 2.9. PMU Configuration UI

7. When all selections are made, click the **Configure** button and then click the Close button.
8. Click the **Generate** button to include the SBX file in the Diamond project file.

If you need to include the HDL file, you can select the option to *add an existing file*. Then browse to the Clarity Designer project location and add the HDL file.

3. User Standby Mode during Normal State

User Standby Mode for CrossLinkPlus devices is useful in reducing power consumption when the device is in the operation state that is Normal State. User Standby Mode is a dynamically controlled option where some blocks can be placed in a User Standby Mode when the blocks are not required by the application. The reduction in power is achieved by placing these blocks in a low leakage state.

Careful consideration should be taken when enabling the User Standby Mode for the blocks as this renders the blocks non-functional. See [Table 3.1](#) for a list of blocks that may be placed in User Standby Mode.

This feature is available only in Normal State. Sleep State overrides the Standby Mode, and places the whole device in lower power state.

[Table 3.1](#) provides a list of blocks that have User Standby Mode and the signal required to place it in a Standby Mode.

Table 3.1. Blocks Supporting User Standby Mode

Block	User Standby Control Signal
PLL	Through User Standby (USRSTDBY) Signal
I/O	Through LVDS output buffer disable (per bank control) and INR disable (per bank control) (BCINRD and BCLVDSOB signals)
OSC	Through OSC Enable port
MIPI DPHY	Through the USRSTDBY port

3.1. Usage of User Standby Mode

Each block with User Standby Mode option has a signal that can be accessed through FPGA fabric via soft logic. Unlike the Power Management Unit, there is no controller available in the hardware. Application should include a soft controller for accessing User Standby Mode features. This soft controller can place the unused blocks in the User Standby Mode, and hence provide the power savings.

[Figure 3.1](#) gives an example block diagram of the User Standby Mode use case. The diagram shows a soft user logic, such as a state machine, that can place each block in User Standby Mode – together or selectively.

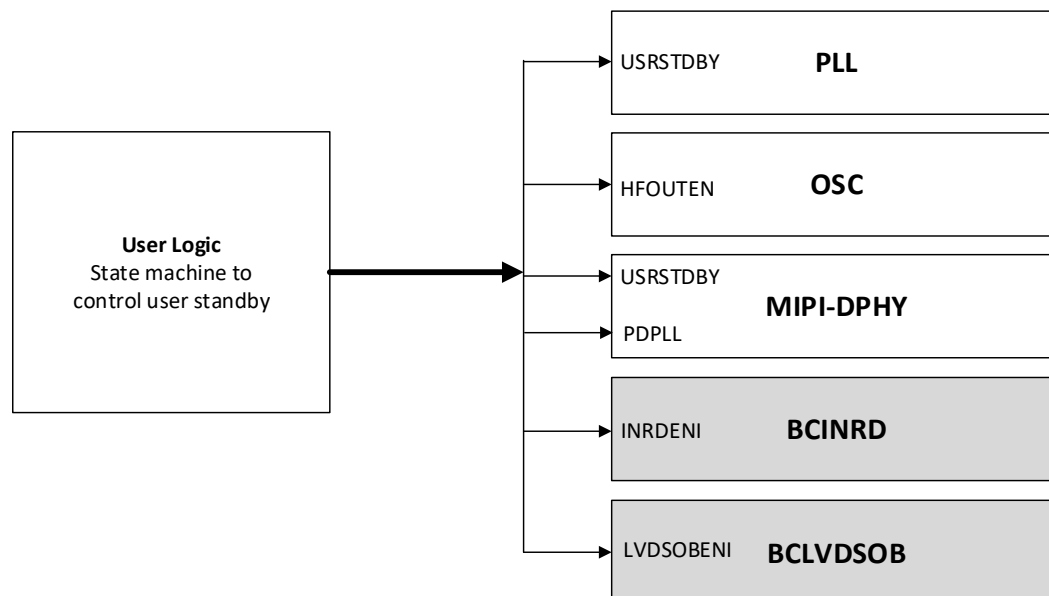


Figure 3.1. User Standby Mode Use Case Example Block Diagram

The following sections discuss how each of the blocks can be placed in User Standby Mode, the signals that can be used, and recommendations.

3.1.1. User Standby Mode for PLL

The CrossLinkPlus device PLL contains a User Standby Mode that allows the PLL to be placed into a standby state to save power when not needed in the design. User Standby Mode is very similar to holding the PLL in reset since the Voltage Controlled Oscillator (VCO) is turned off and needs to regain lock when exiting standby. In both cases, reset and standby modes, the PLL retains its programming.

When generating the PLL using Lattice Diamond Clarity Designer, the standby option can be enabled by checking *Provide Standby Port* under *Optional Ports* tab in the PLL configuration UI. This adds the User Standby Port (USRSTDBY), which is an active high signal.

This port can then be connected to the soft controller for dynamic control during periods when PLL is not required.

3.1.2. User Standby Mode for Oscillator

The CrossLinkPlus device includes an internal oscillator, OSCI, which generates a high frequency output and a low frequency output. The low frequency 10 kHz output is always on and is used by blocks like Power Management Unit (PMU).

The high frequency output can generate a maximum 48 MHz clock, with output divider settings at 1, 2, 4, and 8. The high frequency output clock can be turned off for power savings. The signal that controls the high frequency output clock is HFOUTEN (High Frequency Output Enable).

HFOUTEN is an active high signal, and when high, turns off the high frequency of the internal oscillator.

3.1.3. User Standby Mode for MIPI DPHY

DPHY block in the CrossLinkPlus device can also be placed in a lower User Standby Mode in *Normal State* when it is not required. There are two signals that can place the block in standby: USRSTDBY and PD_PLL.

User Standby signal for DPHY (USRSTDBY) is an active high signal that can power down the DPHY block. USRSTDBY signal is always included in the module generated using Diamond Clarity Designer, during both modes – Transmit and Receive.

Power Down PLL (PD_PLL) is also an active high signal and can power down the PLL for the DPHY block. Powering down the PLL resets the PLL, and thus it can be used to reset the DPHY PLL. PD_PLL signal is always included in the module generated using Lattice Diamond Clarity Designer, during Transmit mode. If the application does not require DPHY PLL block to be powered down, this signal should be tied low (0).

3.1.4. User Standby Mode for Inputs and Outputs

Referenced, differential and LVDS I/O standards consume more power than other I/O standards and are not always required to be active. Bank Controller allows you to turn these I/O off dynamically on a per-bank selection. The Dynamic INRD (input referenced and differential I/O) is used to turn off referenced and differential inputs. Dynamic LVDS control is used to turn off the LVDS output driver.

These options are available during *Normal State* only. In order to use these feature, the application must instantiate the primitives that control these bank controllers. Primitives and corresponding simulation behavior are discussed in the sections below. The control signals to enable and disable this feature has to be controlled by user logic.

Each of the bank controllers – INRD and LVDSOB work independent of each other.

3.1.4.1. Bank Controller for Input Reference and Differential

In the CrossLinkPlus device, the Dynamic Bank Controller is used to power down banks that have Referenced and Differential inputs. The control is dynamic and if needed can release these inputs, including bidirectional I/O. The following sections discuss the primitive and simulation of BCINRD.

BCINRD Primitive

BCINRD Dynamic Bank Controller is represented with primitives as shown in Figure 3.2. Each instantiation controls a single bank. If the control is required for both banks, two BCINRD primitives have to be instantiated, one for each bank (bank 1 and bank 2).

This disables the referenced and differential receivers and any bank controller reference circuits that consume dynamic power, resulting in reduction in power consumption by these circuits. The Inputs Reference and Differential Enable (or INRDENI) signal is an active high signal that can place these inputs in User Standby Mode.

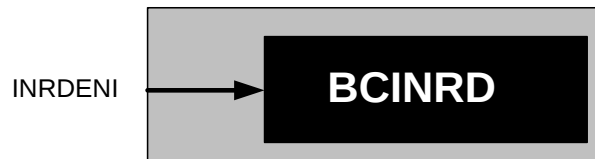


Figure 3.2. Primitive for Bank Controller for Inputs – Referenced and Differential

BCINRD Attributes

Table 3.2 shows the attributes for the BCINRD, with values 1 and 2 representing each bank.

Table 3.2. BCINRD Primitive Attributes

Port	Attribute	Value	Default	Description
INRDENI	—	—	—	Dynamic signal to enable and disable Bank INRD
—	BANKID	1, 2	2	Bank ID for each bank for Dynamic power control INRD

BCINRD Simulation

For application utilizing BCINRD, Lattice Diamond software only supports post-Place & Route simulation. The functional diagram for the simulation model and simulation behavior for each inputs is as shown in Figure 3.3.

For functional simulation, which is before Place & Route, the design requires you to manually instantiate the simulation primitives as per Figure 3.3 (INRDB) on their inputs and bidirectional buffers that are affected by the bank controller.

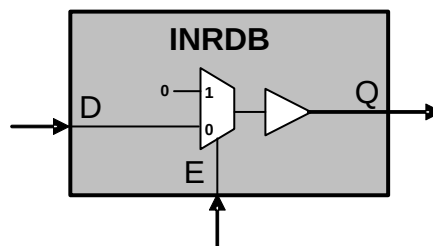


Figure 3.3. INRDB Simulation Primitive

Table 3.3 shows the behavior of the INRDB simulation primitive.

Table 3.3. BCINRD Simulation Behavior

Port	Port Name for Input Buffer	Output
D	Input	—
E	Enable	—
Q	Output	0, when E =1 D, when E=0

3.1.4.2. Bank Controller for LVDS Output Buffers

In the CrossLinkPlus device, the Dynamic Bank Controller is used to power down banks that have LVDS Outputs. The control is dynamic and can release these outputs as needed. The primitive and simulation of BCINRD is detailed next.

BCLVDSOB Primitive

BCLVDSOB Dynamic Bank Controller is represented with primitives as shown in Figure 3.4. Each instantiation controls a single bank. If the control is required for both banks, two BCLVDSOB primitives have to be instantiated, one for each bank (Bank 1 and Bank 2).

This disables the LVDS output receiver circuits that consume dynamic power, resulting in reduction in power consumption by these circuits. The LVDS Output Enable (or LVDSENI) signal is an active high signal that can place these outputs in User Standby Mode.

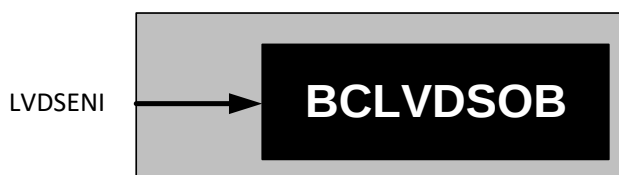


Figure 3.4. Bank Controller for LVDS Outputs Primitive

BCLVDSOB Attributes

Table 3.4 shows the attributes for the BCLVDSOB, with values 1 and 2 representing each bank.

Table 3.4. BCLVDSOB Primitive Attributes

Port	Attribute	Value	Default	Description
LVDSENI	—	—	—	Dynamic signal to enable and disable Bank LVDS Outputs
—	BANKID	1, 2	2	Bank ID for each bank for Dynamic power control LVDS Outputs

BCLVDSOB Simulation

For application utilizing BCLVDSOB, Lattice Diamond software only supports post-Place & Route simulation. The functional diagram for the simulation model and simulation behavior for each inputs is as shown in Figure 3.5.

For functional simulation, which is before Place & Route, the design requires you to manually instantiate the simulation primitives as per Figure 3.5 (LVDSOB) on their output buffers that are affected by the bank controller.

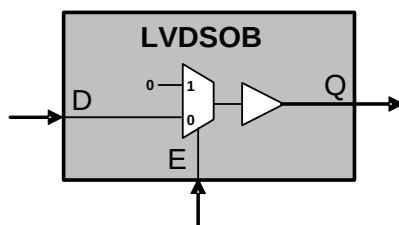


Figure 3.5. LVDSOB Simulation Primitive

Table 3.5 shows the behavior of the LVDSOB simulation primitive.

Table 3.5. BCLVDSOB Simulation Behavior

Port	Port Name for Output Buffer	Output
D	Input	—
E	Enable	—
Q	Output	Z (High Impedance), when E=1; D, when E=0

4. Power Consumption and Calculation

Diamond software includes a Power Calculator tool that can determine the power consumption of the CrossLinkPlus devices. Power Calculator is the fastest power simulation tool available in the industry. It offers two modes – *Estimation mode* (for what-if analysis) and *Calculation mode* (for the more accurate application-specific power consumption by importing NCD design files). The engine performs each calculation quickly and accurately.

When running the Power Calculator tool in Estimation mode, you can provide estimates of the utilization of various components and the tool provides an estimate of the power consumption. This is a good start, especially for what-if analyses and device selection.

Calculation mode, on the other hand, is a more accurate approach, where you can import the actual device utilization by importing the post place and route netlist design file or NCD file. Additionally, Power Calculator supports features like Trace Report (or TWR) import, to get the clock frequencies for various clocks. Trace Report only includes frequencies of the clocks nets that are constrained in the Preference file. You are still required to provide frequencies of the clocks that are not included in Preference file and hence the Trace Report.

The default Activity Factor (AF%) for dynamic power calculation is set to 10% in the Power Calculator. You can change the default AF for the entire project or for each clock net individually. Activity Factor is discussed in detail in the [Activity Factor Calculation](#) section.

4.1. Power Calculator

Refer to the Lattice Diamond Tutorial under Diamond Startup Page. When you go through the procedure, the Lattice Power Calculator main window appears as shown in [Figure 4.1](#)

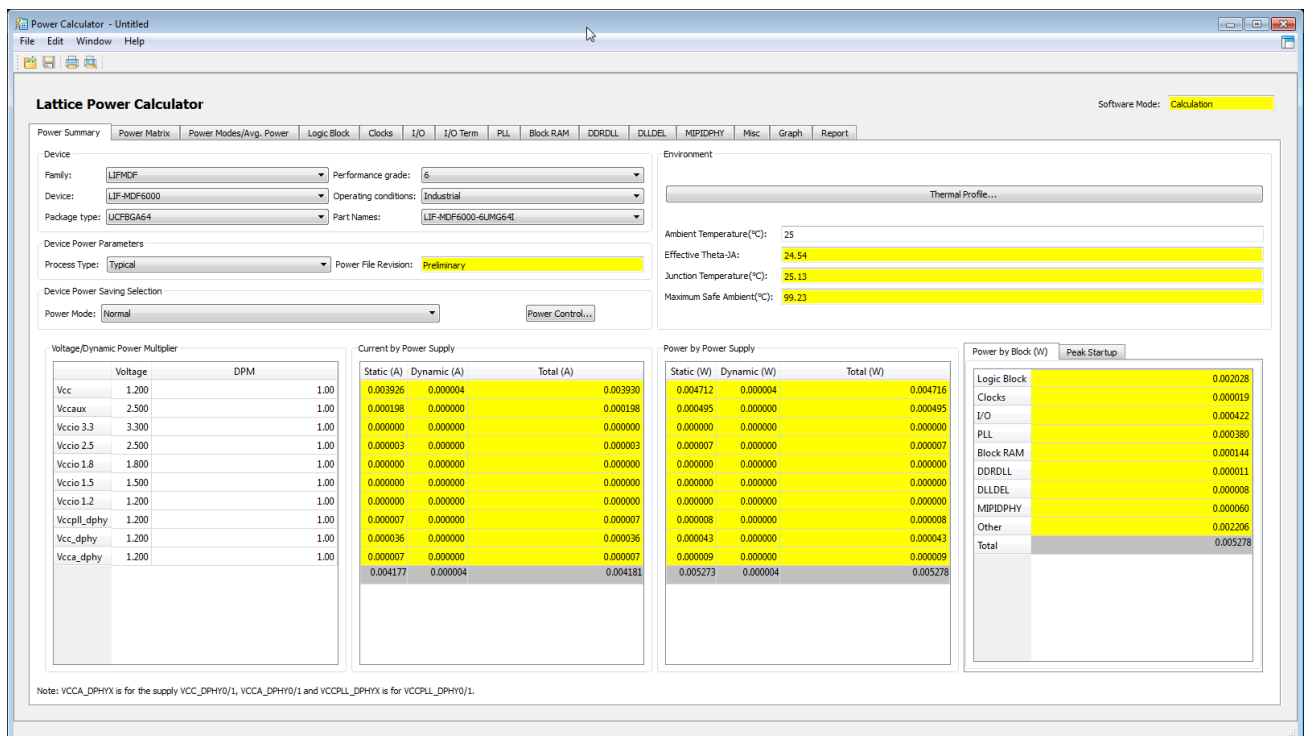


Figure 4.1. Power Calculator (Summary Tab for CrossLinkPlus Devices)

It is important to understand how the options available in the Power Calculator affect the power. For example, if the ambient temperature is changed, the junction temperature is affected according to the following equation:

$$T_J = T_A + \theta_{JA_EFFECTIVE} * P$$

where T_J and T_A are the junction and ambient temperatures, respectively, and P is the power.

$\theta_{JA_EFFECTIVE}$ is the effective thermal impedance between the die and its environment.

The junction temperature is directly dependent on the ambient temperature. An increase in T_A increases T_J and result in an increase of the static leakage component.

Power can be affected by selecting the Process Type or the frequency at which the application runs. Process primarily changes the static leakage or Static Power, and frequency changes dynamic power increasing the frequency of toggling increases the dynamic component of power.

4.1.1. Typical and Worst Case Process

Process variation is a naturally occurring variation in the attributes of transistors, such as length, widths, oxide thickness, when integrated circuits are fabricated. Process variation causes measurable and predictable variance in the output performance of all circuits.

Lattice Diamond Power Calculator provides option to select the Typical (the mean current/power of distribution) and Worst Case (the maximum current/power of distribution) as a result of the variation. Process variation primarily affects the static leakage component of the device.

Process Type selection in the Power Calculator allows you to understand the current and power variation of the devices, and predicts accurate results that are application specific. These can be used for power supply design, battery capacity design, and thermal management for each application.

4.1.2. Junction Temperature

Junction temperature is the temperature of the die during operation. It is one of the most important factors that affects the device power. For a fixed junction temperature, voltage and device package combination, static power is fixed.

Ambient temperature affects the junction temperature. Devices operating in a high-temperature environment have higher leakage since their junction temperature is higher. Power Calculator models the interdependence of ambient and junction temperature. When you provides an ambient temperature, it is rolled into an algorithm that calculates the junction temperature and power through an iterative process to find the thermal equilibrium of the system (device running with the design) with respect to its environment (T_A , airflow and so on).

Thermal impedance plays an important role in determining the devices behavior, and the tool that takes it into account to calculate maximum safe ambient temperature. See the [Thermal Management](#) section for details on how the Power Calculator uses thermal impedances.

4.1.3. Maximum Safe Ambient Temperature

Maximum Safe Ambient Temperature is one of the most important numbers displayed in the Summary tab of the Power Calculator. This is the maximum ambient temperature at which the device application can run without violating the junction temperature limits for the grade of the commercial or industrial device.

Power Calculator uses an algorithm to accurately predict this temperature. The algorithm adjusts itself as you change options such as voltage, process, frequency, activity factor, thermal impedance and so on (or any factor that may affect the power dissipation of the device).

Thus, it becomes extremely important to provide more accurate inputs to the tool for more accurate predicted results.

4.1.4. Operating Temperature Range

When designing a system, you must make sure a device operates at specified temperatures within the system environment. This is particularly important to consider before a system is designed. With Power Calculator, you can predict device thermodynamics and estimate the dynamic power budget. The ability to estimate device operating temperature prior to board design also allows you to better plan for power budgeting and thermal management.

Although total power, ambient temperature, thermal resistance and airflow all contribute to device thermodynamics, the junction temperature specified in Operating Conditions in the Data Sheet is the key to device operation.

The allowed junction temperature range is 0°C to 85°C for commercial grade devices and –40°C to 100°C for industrial grade devices. If the junction temperature of the die is not within these temperature ranges, the performance, reliability and functionality can get affected.

4.1.5. Dynamic Power Multiplier

In general, for semiconductor devices, the dynamic power consumption is independent of the variation in process and temperature. Power Calculator follows this rule, and hence any changes to either process or temperature will not change the dynamic current/ power.

In order to provide you with an option to add some safeguards, Power Calculator includes the Dynamic Power Multiplier (DPM) column right next to the voltage supplies on the Summary tab. DPM allows you to add a multiplier for the dynamic current for each individual power supplies. This multiplier is included in the dynamic current equation.

Dynamic Power Multiplier has a default value 1 that means the dynamic power is what is predicted by the Power Calculator. If you want to add, say 20%, additional dynamic power, the DPM can be set to 1.2 (1 + 20%) and it can be placed against the appropriate power supply. This increases the dynamic power for that supply by 20% and provides you some guard band.

4.1.6. Peak Startup Current

The bottom right panel of the Summary tab in Lattice Diamond Power Calculator includes the Peak Startup tab. This tab provides an important piece of information, especially for designing the capacity for power supply or batteries.

Peak Startup current tab provides the current CrossLinkPlus device pulls on each power supply when it is powered on. In certain cases, as applicable, this tab also includes the time period for which this current lasts.

4.1.7. Power Budgeting

Both Peak Startup current and device operational current should be looked into when budgeting for power supply or battery capacity. It is recommended that the higher of the two numbers be used as reference for capacity calculations.

4.1.8. Device Operational Limits

Power Calculator provides the power dissipation of a design under a given set of conditions. It also predicts the junction temperature (T_j) for the design. Any time this junction temperature is outside the limits specified in Data Sheet, the viability of operating the device at this junction temperature must be re-evaluated.

A commercial device is likely to show speed degradation with a junction temperature above 85°C. An industrial device at a junction temperature degrades above 100°C. It is required that the die temperature be kept below these limits to achieve the guaranteed speed operation.

Operating a device at a higher temperature also means a higher static current and hence power. The difference between static current and total current, both static and dynamic currents, at a given temperature provides the power budget available. This is also very useful in power supply or battery capacity designs.

If the device runs at a current higher than this budget, the total ICC is also higher. This causes the die temperature to rise above the specified operating conditions. The four factors of power, ambient temperature, thermal resistance and airflow, can also be varied and controlled to reduce the junction temperature of the device. Power Calculator is a powerful tool to help system designers to properly budget the FPGA power that, in turn, helps improve overall system reliability.

Power Calculator clearly indicates when the application is running at higher power than recommended. If the Junction temperature goes beyond the grade limits, the box turns red in the summary tab. The Junction temperature calculations are provided up to 125°C, which is also the reliability limit of the CrossLinkPlus devices. If the Junction temperature is beyond 125°C under the conditions provided, the tool indicates it by 125+. This shows that the device achieves thermal equilibrium beyond 125°C.

In certain cases, the thermal equilibrium cannot be achieved by the application under given conditions. In such situation, the device can continue to generate heat and not be able to dissipate. Power Calculator has a built-in algorithm to determine such situations. This is indicated with 125++ under junction temperature calculation. This situation should be avoided under all circumstances, as this can cause permanent damage to the device.

4.1.9. Dynamic Power Savings

The Power Calculator dynamically estimates the power when the Bank Controller and other power save features are implemented in a design. CrossLinkPlus device has a number of options to control power; these can range from Bank Controllers to the disabling individual PLLs/ DLLs.

On the Summary tab of Power Calculator, there is a Power Controller button. Clicking the button launches a window to perform what-if analysis to evaluate power consumption when different dynamic power options are used.

See the [Using Power Calculator for CrossLinkPlus Devices](#) section on page 31 for details on how to use these features.

4.1.10. Activity Factor Calculation

The Activity Factor % (or AF%) is defined as the percentage of frequency (or time) that a signal is active or toggling the output. Most resources associated with a clock domain are running or toggling at some percentage of the frequency at which the clock is running. You must provide this value as a percentage under the AF% column in the Power Calculator tool.

Another term for I/O is the I/O Toggle Rate. The AF% is applicable to the PFU, Routing, and Memory Read Write Ports, and so on. The activity of I/O is determined by the signals that you provided (in the case of inputs) or as an output of the design (in the case of outputs). The rates at which the I/O toggle define their activity. The I/O Toggle Rate or the I/O Toggle Frequency is a better measure of their activity.

The Toggle Rate (or TR) in MHz of the output is defined in the following equation:

$$\text{Toggle Rate (MHz)} = 1/2 * f * \text{AF\%}$$

You are required to provide the TR (MHz) value for the I/O instead of providing the frequency and AF% for other resources. AF can be calculated for each routing resource, output or PFU. However, this involves long calculations. The general recommendation for a design occupying roughly 30% to 70% of the device is an AF% between 15% and 25%. This is an average value. The accurate value of an AF depends upon clock frequency, stimulus to the design and the final output.

Power Calculator allows you to import a VCD file from their simulation to accurately assess the activity factor of their design (Edit > Open Simulation File). The VCD file is based on Post-P&R simulation and it is an ASCII file generated by the simulator. Please check the simulation tool documentation on how to generate VCD file. The AF calculated from the VCD file is based on how accurate the test-bench or stimulus is for the simulation.

4.1.11. Power Calculator Assumptions

The following are the assumptions made by the Power Calculator.

- The Power Calculator tool uses default ambient temperature of 25°C (room temperature). This default temperature can be changed.
- You can define the ambient temperature (T_A) for device junction temperature (T_J) calculation based on the power estimation. T_J is calculated from the user-entered T_A and the power calculation of typical room temperature.
- I/O power consumption is based on an output loading of 5 pF. Designers have the ability to change this capacitive loading.
- You can estimate power dissipation and current for each type of power supply (V_{CC} and V_{CCIO}).
- The nominal V_{CC} is used by default to calculate power consumption. A lower or higher V_{CC} can be chosen from a list of available values.
- θ_{JA} can be changed to better estimate the operating system manually or by entering Airflow in Linear Feet per Minute (LFM) along with a Heat Sink options.
- The default value of the I/O types for CrossLinkPlus devices is LVCMOS25, 8 mA.
- The activity factor (AF) is defined as the toggle rate of the registered output. For example, assuming that the input of a flip-flop is changing at every clock cycle, 100% AF of a flip-flop running at 100 MHz, is 50 MHz.

- The default activity factor for logic is 10%.
- You can import the VCD file from their simulation to get activity factor based on the simulation. It is to be noted that the AF from VCD is as good as the coverage in the simulation.
- Unused I/O is configured as LVCMOS Inputs with weak internal pull-ups. These are generally powered off the VCCIO that is connected to the bank.
- You have an option to import the Frequency from trace report (TWR) or preference file (LPF).
- The operating junction temperature range for commercial grade devices is 0°C to 85°C, and Industrial is –40°C to 100°C For details, refer to the DC Electrical Characteristics section of [CrossLinkPlus Family Data Sheet \(FPGA-DS-02054\)](#).
- For thermal impedance, Power Calculator default value is based on a medium size board (6" x 6", 6 layers), with no heatsink and 200 LFM airflow. This can be changed as needed under Thermal Profile section in the tool.

4.2. Using Power Calculator for CrossLinkPlus Devices

The CrossLinkPlus device has *Normal State*, along with the User Standby Mode, and the *Sleep State*. Power Calculator can calculate the power consumption of the device in various states and options available to you. The easiest way is to either import the NCD file in calculation mode, or providing resource utilization in estimation mode. Once the tool is setup with resources, provide frequency or import using TWR file, update activity factor, check the thermal environment of the system and update ambient temperature of operation.

This section covers ways to determine power consumption of various states and modes for the CrossLinkPlus devices.

4.2.1. Power Calculator in Normal State, and User Standby Mode

Power Calculator can calculate power in the *Normal State*, and its corresponding User Standby Mode. After importing the NCD file, and providing all the inputs, the power consumption of the device is calculated when it is operational. Summary tab provides the static and dynamic, current/power.

The *Normal State* power can be with or without the User Standby Mode enabled. By default, the User Standby Mode is disabled; thus, the *Normal State* power consumption when the Power Calculator is first setup is without User Standby Mode power savings. If the application includes User Standby Mode blocks, Lattice Diamond Power Calculator has option to toggle these options to switch *Normal State* power consumption with and without User Standby Mode. These options can be toggled using the Power Controller button (next to the Power Mode selection on Summary tab).

[Figure 4.2](#) shows the Power Option Controller window tabs (Inputs/Outputs and Other).

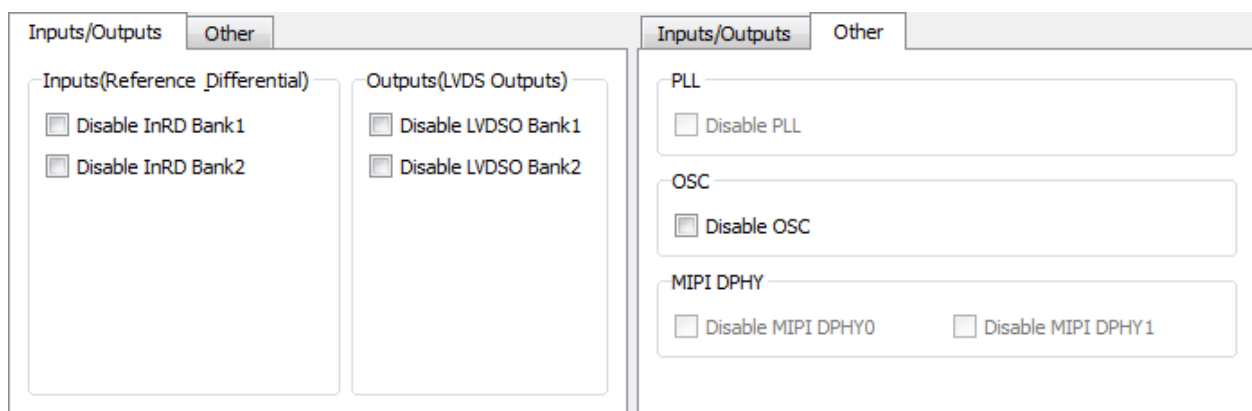


Figure 4.2. Power Option Controller Window for User Standby Mode options

[Figure 4.2](#) uses an example application that includes Bank Controllers for I/O, and Oscillator Control. By default, these options are unchecked, that is User Standby Mode is not enabled. By checking the checkboxes, the Power Calculator tool recalculates power for *Normal State* with User Standby Mode enabled.

Power Calculator provides power with or without User Standby Mode mutually exclusively. It does not provide power for both options together. The *Normal State* power used in average power consumption calculations (discussed in the following section) is based on the options with or without User Standby Mode, but never both.

4.2.2. Power Calculator in Sleep State

Power Calculator has a built-in feature that calculates the power consumption of the CrossLinkPlus device during *Sleep State*. Device in this state is not completely turned off. Parts of the device are still on, such as the Power Management Unit (PMU), the circuitry that retains the SRAM contents that stores device configuration, EBR and Distributed RAM contents, I/O states, and so on. Hence, the *Sleep State* is not a device powered off state, and has some minimal leakage.

In order to get the *Sleep State* power consumption, go to Power Modes/Avg. Power tab. The right hand side pane of the tab indicates the current and power for each supply in the *Sleep State*.

The left hand pane provides the power consumption for the application, which is the same as what is summarized on the Summary tab. This power matches the power if some of the User Standby features are enabled. If some power controller options are selected, they become part of the *Normal State* power, and that is what gets populated here.

Figure 4.3 shows the Power by Supply sub-tab of the Power Modes/ Avg. Power tab.

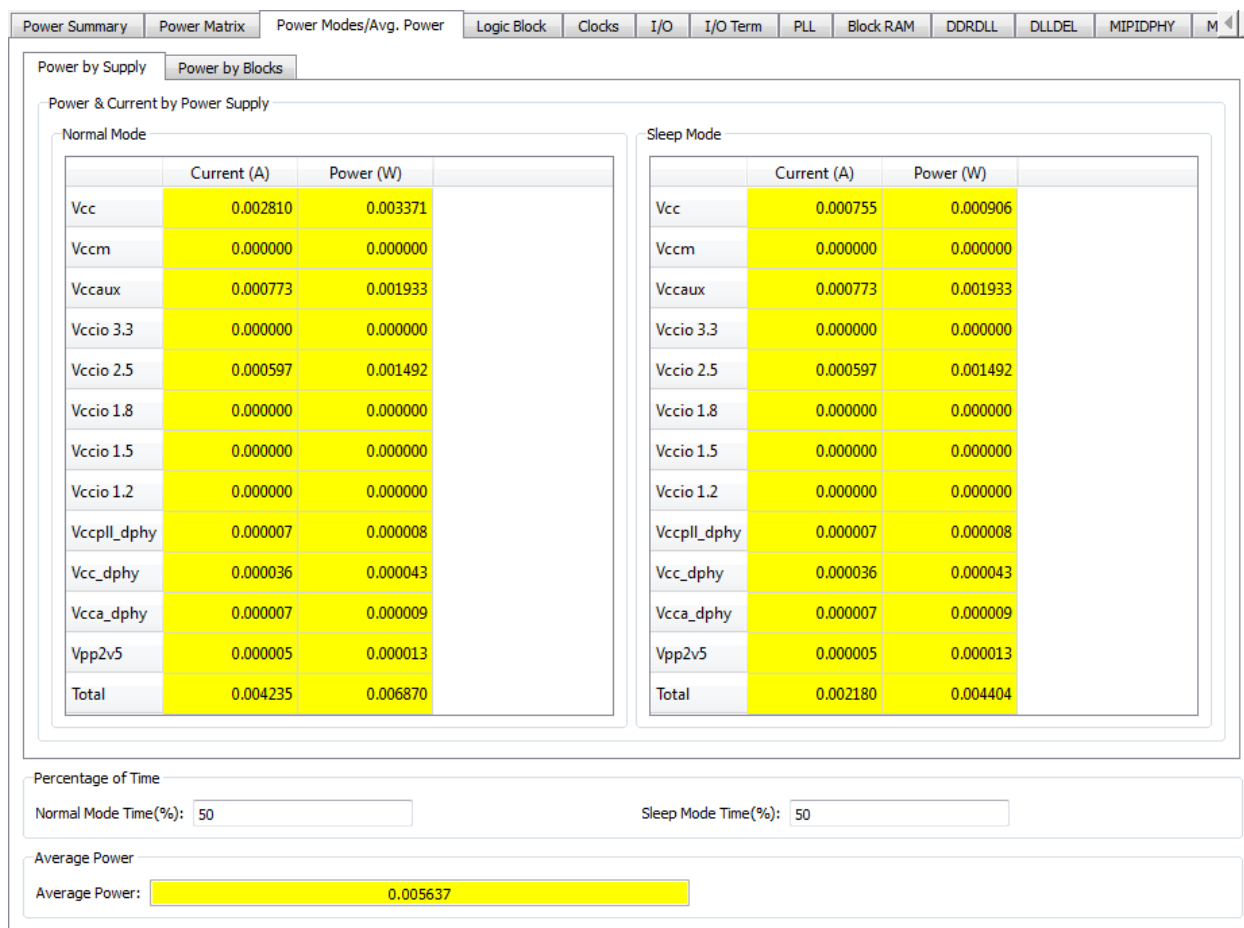


Figure 4.3. Power Calculator – Power Modes/Avg. Power Tab (Power by Supply)

Figure 4.4 shows the Power by Blocks sub-tab of the Power Modes/Avg. Power tab.

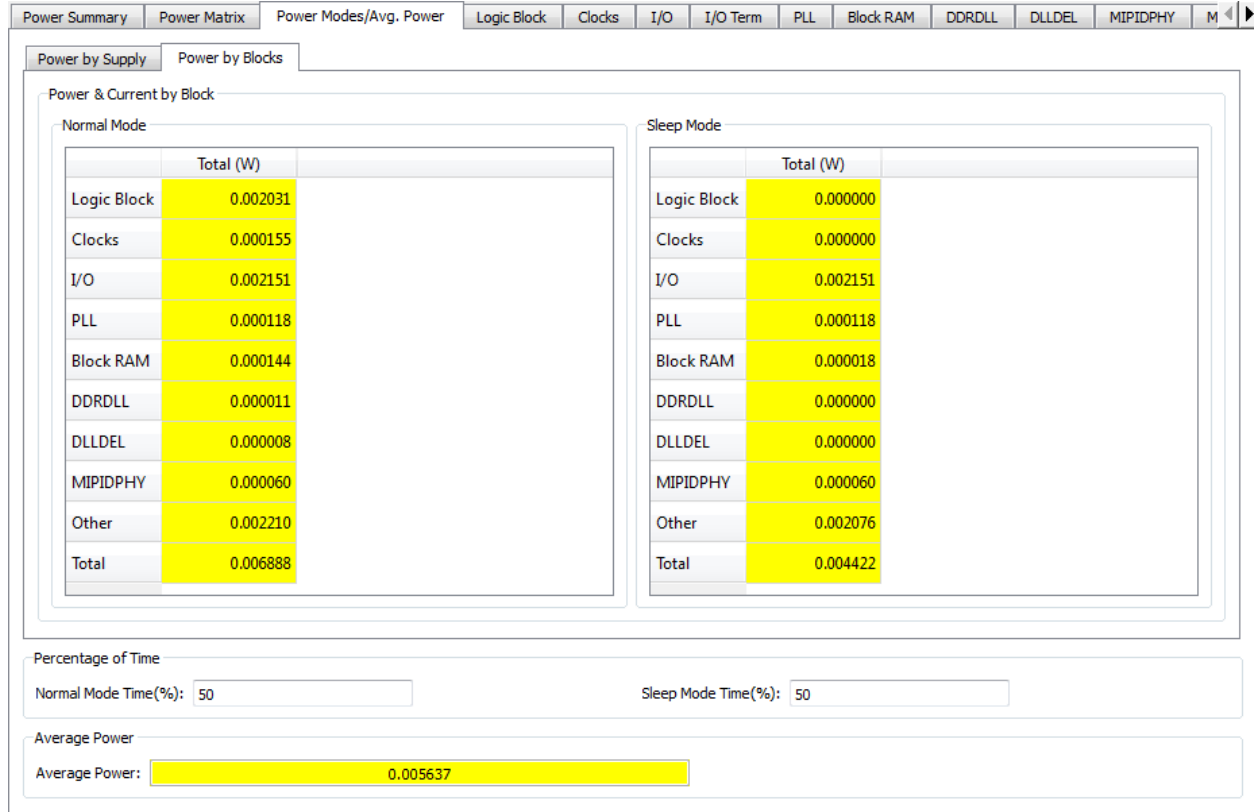


Figure 4.4. Power Calculator – Power Modes/Avg. Power Tab (Power by Blocks)

4.2.3. Calculating Average Power

Application can use the Power Management Unit (PMU) to switch between the *Normal State* and *Sleep State* based on the requirements. When the services of CrossLinkPlus device are not needed, the device can operate at a lower power consuming *Sleep State*. Overall, this reduces the power consumption.

Power Modes/Avg. Power tab also helps calculate the average power consumption of the CrossLinkPlus device. The bottom pane of this tab has Percentage of Time and Average Power sections, as shown in Figure 4.3 and Figure 4.4. Average power is calculated based on the percentage of time the device is in each state.

The average power is consumed as per the following formula:

$$P_{AVG} = P_{NORMAL} * \% \text{ of Time in Normal State} + P_{SLEEP} * \% \text{ of Time in Sleep State}$$

where P_{AVG} , P_{NORMAL} , P_{SLEEP} are average power, *Normal State* power, and *Sleep State* power respectively.

For example, if the device is in each state for half the time (50%), then the Average Power consumption is average of *Normal State* and *Sleep State* power. *Normal State* power can be either with or without the User Standby Mode options.

The default value of the time percentages is set to 50%, and it can be changed based on application.

5. Thermal Management

To improve reliability and prevent device failure, all electronic devices are required to dissipate the heat generated while running. Thermal management refers to the techniques used to improve heat dissipation. The methods include use of heatsinks, fans for air cooling and also other forms of cooling like liquid cooling in modern computers.

CrossLinkPlus devices are designed to be low power devices, and by combining options like using PMU, the average power consumption and hence the heat generated, can be reduced. This section covers understanding and ways to improve thermal management for applications using CrossLinkPlus devices.

5.1. Thermal Impedance and Airflow

A common method for characterizing a packaged device thermal performance is with Thermal Impedance, represented by θ . For a semiconductor device, thermal resistance indicates the steady state temperature rise of the die junction above a given reference for each watt of power (heat) dissipated at the die surface. Its units are degree Celsius per Watt ($^{\circ}\text{C}/\text{W}$).

The most common examples are:

- θ_{JA} , Thermal Resistance Junction-to-Ambient ($\text{in } ^{\circ}\text{C}/\text{W}$)
- θ_{JC} , Thermal Resistance Junction-to-Case ($\text{in } ^{\circ}\text{C}/\text{W}$)
- θ_{JB} , Thermal Resistance Junction-to-Board ($\text{in } ^{\circ}\text{C}/\text{W}$)

Knowing the reference (that is ambient, case, or board) temperature, the power, P , and the relevant θ value, the junction temperature can be calculated per following equations.

- $T_J = T_A + \theta_{JA} * P$
- $T_J = T_C + \theta_{JC} * P$
- $T_J = T_B + \theta_{JB} * P$

where T_J , T_A , T_C and T_B are the junction, ambient, case (or package) and board temperatures ($\text{in } ^{\circ}\text{C}$), respectively. P is the total power dissipation of the device.

θ_{JA} is commonly used with natural and forced convection air-cooled systems. θ_{JC} is useful when the package has a high conductivity case mounted directly to a PCB or heatsink. And θ_{JB} applies when the board temperature adjacent to the package is known.

To improve airflow effectiveness, it is important to maximize the amount of air that flows over the device or the surface area of the heat sink. The airflow around the device can be increased by providing an additional fan or increasing the output of the existing fan. If this is not possible, baffling the airflow to direct it across the device may help. This means the addition of sheet metal or objects to provide the mechanical airflow guides to guide air to the target device. Often the addition of simple baffles can eliminate the need for an extra fan. In addition, the order in which air passes over devices can impact the amount of heat dissipated.

5.2. Thermal Management in Power Calculator

Lattice Diamond Power Calculator utilizes the ambient temperature ($^{\circ}\text{C}$) to calculate the junction temperature ($^{\circ}\text{C}$) based on the θ_{JA} for the targeted device. You can also provide the airflow values (in Linear Feet per Minute or LFM, that is also the same as Cubic Feet per Minute) to obtain a more accurate junction temperature value.

Thermal Environment options in the Power Calculators provide a thermal impedances for JEDEC (4"x4", 2S2P), small (6"x6", 6 layered board), medium (8"x8", 8 layered board) and large (10"x10", or larger board). There are also options to select copper heatsinks of different sized fins. These impedances and options are provided as a guidance. You have an option to provide your own thermal impedance, too.

As each application board can be different, it is recommended that accurate thermal impedance be provided. These can be obtained by measuring using a thermal lab or by using thermal simulation software.

5.3. DELPHI Models

DELPHI Models are thermo-mechanical models that can be used to simulate thermal behavior of the electronic devices in a system. These tools can simulate the thermal behavior of the system and predict the thermal impedance for each component. The thermal impedance obtained using simulation methods can be entered in the Thermal Environment in Power Calculator.

DELPHI Models for CrossLinkPlus device can be downloaded from the web and they are compatible with Mentor Graphics' FloTHERM® and Ansys Icepak tools.

6. Conclusion

CrossLinkPlus devices provide you with a number of options for managing power, and the Power Calculator tool complements by calculating power in different states of the device.

By utilizing the user-friendly interface to access these features in Lattice Diamond software, applications can utilize these features to maximum extent, and have predictable and reliable performance for these devices.

References

For more information, refer to the following documents:

- [CrossLinkPlus Family Data Sheet \(FPGA-DS-02054\)](#)
- [CrossLinkPlus High-Speed I/O Interface \(FPGA-TN-02102\)](#)
- [CrossLinkPlus Hardware Checklist \(FPGA-TN-02105\)](#)
- [CrossLinkPlus Programming and Configuration Usage Guide \(FPGA-TN-02103\)](#)
- [CrossLinkPlus sysCLOCK PLL/DLL Design and Usage Guide \(FPGA-TN-02109\)](#)
- [CrossLinkPlus sysI/O Usage Guide \(FPGA-TN-02108\)](#)
- [CrossLinkPlus Memory Usage Guide \(FPGA-TN-02110\)](#)
- [CrossLinkPlus I2C Hardened IP Usage Guide \(FPGA-TN-02112\)](#)
- [Advanced CrossLinkPlus I2C Hardened IP Reference Guide \(FPGA-TN-02135\)](#)

Technical Support Assistance

Submit a technical support case through www.latticesemi.com/techsupport.

Revision History

Revision 1.0, September 2019

Section	Change Summary
All	Production release.



www.latticesemi.com