Arm® CoreLink™ PCK-600 Power Control Kit

Revision: r0p3

Technical Reference Manual

101150_0003_00_en

Arm® CoreLink™ PCK-600 Power Control Kit

Technical Reference Manual

Copyright © 2018, 2019 Arm Limited or its affiliates. All rights reserved.

Release Information

Document History

Issue	Date	Confidentiality	Change
0000-00	24 January 2018	Non-Confidential	First release for r0p0 BET.
0000-01	02 May 2018	Non-Confidential	First release for r0p0 EAC.
0001-00	05 June 2018	Non-Confidential	First release for r0p1 EAC.
0002-00	27 June 2018	Non-Confidential	First release for r0p2 EAC.
0003-00	30 January 2019	Non-Confidential	First release for r0p3 EAC.

Non-Confidential Proprietary Notice

This document is protected by copyright and other related rights and the practice or implementation of the information contained in this document may be protected by one or more patents or pending patent applications. No part of this document may be reproduced in any form by any means without the express prior written permission of Arm. **No license, express or implied, by estoppel or otherwise to any intellectual property rights is granted by this document unless specifically stated.**

Your access to the information in this document is conditional upon your acceptance that you will not use or permit others to use the information for the purposes of determining whether implementations infringe any third party patents.

THIS DOCUMENT IS PROVIDED "AS IS". ARM PROVIDES NO REPRESENTATIONS AND NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY, SATISFACTORY QUALITY, NON-INFRINGEMENT OR FITNESS FOR A PARTICULAR PURPOSE WITH RESPECT TO THE DOCUMENT. For the avoidance of doubt, Arm makes no representation with respect to, and has undertaken no analysis to identify or understand the scope and content of, third party patents, copyrights, trade secrets, or other rights.

This document may include technical inaccuracies or typographical errors.

TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL ARM BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF ARM HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

This document consists solely of commercial items. You shall be responsible for ensuring that any use, duplication or disclosure of this document complies fully with any relevant export laws and regulations to assure that this document or any portion thereof is not exported, directly or indirectly, in violation of such export laws. Use of the word "partner" in reference to Arm's customers is not intended to create or refer to any partnership relationship with any other company. Arm may make changes to this document at any time and without notice.

If any of the provisions contained in these terms conflict with any of the provisions of any click through or signed written agreement covering this document with Arm, then the click through or signed written agreement prevails over and supersedes the conflicting provisions of these terms. This document may be translated into other languages for convenience, and you agree that if there is any conflict between the English version of this document and any translation, the terms of the English version of the Agreement shall prevail.

The Arm corporate logo and words marked with ® or ™ are registered trademarks or trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. All rights reserved. Other brands and names mentioned in this document may be the trademarks of their respective owners. Please follow Arm's trademark usage guidelines at *http://www.arm.com/company/policies/trademarks*.

Copyright © 2018, 2019 Arm Limited (or its affiliates). All rights reserved.

Arm Limited. Company 02557590 registered in England.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

LES-PRE-20349

Confidentiality Status

This document is Non-Confidential. The right to use, copy and disclose this document may be subject to license restrictions in accordance with the terms of the agreement entered into by Arm and the party that Arm delivered this document to.

Unrestricted Access is an Arm internal classification.

Product Status

The information in this document is Final, that is for a developed product.

Web Address

http://www.arm.com

Contents

Arm® CoreLink™ PCK-600 Power Control Kit Technical Reference Manual

Preface

Chapter 1 Introduction

Chapter 2 Functional description

Chapter 3 Programmers model

Appendix A Signal Descriptions

Appendix B Revisions

Preface

This preface introduces the *Arm® CoreLink™ PCK-600 Power Control Kit Technical Reference Manual*.

It contains the following:

- *About this book* on page 7.
- *Feedback* on page 10.

About this book

This book describes the functionality of the components in the Arm® CoreLink™ PCK-600 Power Control Kit. It also provides the programming information and the signal descriptions.

Product revision status

The r*m*p*n* identifier indicates the revision status of the product described in this book, for example, r1p2, where:

r*m*	Identifies the major revision of the product, for example, r1.
p*n*	Identifies the minor revision or modification status of the product, for example, p2.

Intended audience

This book is written for system designers and programmers who are designing or programming a *System on Chip* (SoC) that uses the PCK-600.

Using this book

This book is organized into the following chapters:

Chapter 1 Introduction
This chapter introduces the PCK-600 Power Control Kit.

Chapter 2 Functional description
This chapter describes the functionality of each PCK-600 component.

Chapter 3 Programmers model
This chapter describes the memory regions and registers that the *Power Policy Unit* (PPU) provides.

Appendix A Signal Descriptions
This appendix describes the interface signals that are present for each PCK-600 component.

Appendix B Revisions
This appendix describes the technical changes between released issues of this book.

Glossary

The Arm® Glossary is a list of terms used in Arm documentation, together with definitions for those terms. The Arm Glossary does not contain terms that are industry standard unless the Arm meaning differs from the generally accepted meaning.

See the *Arm® Glossary* for more information.

Typographic conventions

italic
Introduces special terminology, denotes cross-references, and citations.

bold
Highlights interface elements, such as menu names. Denotes signal names. Also used for terms in descriptive lists, where appropriate.

`monospace`
Denotes text that you can enter at the keyboard, such as commands, file and program names, and source code.

`monospace`
Denotes a permitted abbreviation for a command or option. You can enter the underlined text instead of the full command or option name.

`monospace italic`

Denotes arguments to monospace text where the argument is to be replaced by a specific value.

`monospace bold`

Denotes language keywords when used outside example code.

`<and>`

Encloses replaceable terms for assembler syntax where they appear in code or code fragments. For example:

```
MRC p15, 0, <Rd>, <CRn>, <CRm>, <Opcode_2>
```

SMALL CAPITALS

Used in body text for a few terms that have specific technical meanings, that are defined in the *Arm® Glossary*. For example, IMPLEMENTATION DEFINED, IMPLEMENTATION SPECIFIC, UNKNOWN, and UNPREDICTABLE.

Timing diagrams

The following figure explains the components used in timing diagrams. Variations, when they occur, have clear labels. You must not assume any timing information that is not explicit in the diagrams.

Shaded bus and signal areas are undefined, so the bus or signal can assume any value within the shaded area at that time. The actual level is unimportant and does not affect normal operation.

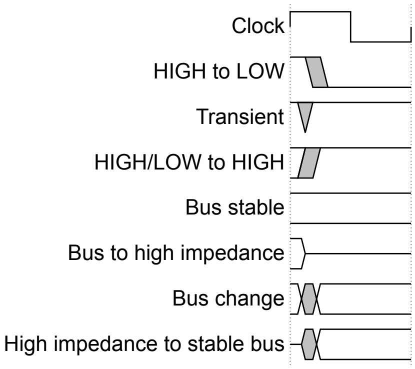

Figure 1 Key to timing diagram conventions

Signals

The signal conventions are:

Signal level

The level of an asserted signal depends on whether the signal is active-HIGH or active-LOW. Asserted means:

- HIGH for active-HIGH signals.
- LOW for active-LOW signals.

Lowercase n

At the start or end of a signal name denotes an active-LOW signal.

Additional reading

This book contains information that is specific to this product. See the following documents for other relevant information.

Arm publications

- *AMBA® Low Power Interface Specification, Arm® Q-Channel and P-Channel Interfaces* (Arm IHI 0068).
- *Arm® AMBA® 3 APB Protocol Specification, v1.0* (Arm IHI 0024B).

The following confidential books are only available to licensees or require registration with Arm:

- *Arm® CoreLink™ PCK-600 Power Control Kit Configuration and Integration Manual* (Arm 101151).
- *Arm® Power Policy Unit Architecture Specification, version 1.1* (Arm DEN 0051).
- *Arm® Clock Controller Architecture Specification, version 1.0* (Arm DEN 0052).
- *Arm® Power Control System Architecture Specification, version 2.0* (Arm DEN 0050).

Other publications

- JEDEC, *Standard Manufacturer's Identification Code*, JEP106.

Feedback

Feedback on this product

If you have any comments or suggestions about this product, contact your supplier and give:

- The product name.
- The product revision or version.
- An explanation with as much information as you can provide. Include symptoms and diagnostic procedures if appropriate.

Feedback on content

If you have comments on content then send an e-mail to *errata@arm.com*. Give:

- The title *Arm CoreLink PCK-600 Power Control Kit Technical Reference Manual*.
- The number 101150_0003_00_en.
- If applicable, the page number(s) to which your comments refer.
- A concise explanation of your comments.

Arm also welcomes general suggestions for additions and improvements.

Note

Arm tests the PDF only in Adobe Acrobat and Acrobat Reader, and cannot guarantee the quality of the represented document when used with any other PDF reader.

Chapter 1
Introduction

This chapter introduces the PCK-600 Power Control Kit.

It contains the following sections:

1.1 About the Power Control Kit

The PCK-600 Power Control Kit provides a set of configurable RTL components for the creation of SoC clock and power control infrastructure. The components use the Arm Q-Channel and P-Channel low power interfaces.

The PCK-600 consists of the following components:

Low Power Distributor Q-Channel (LPD-Q)
The LPD-Q component distributes a Q-Channel from one Q-Channel controller to up to 32 Q-Channel devices.

Low Power Distributor P-Channel (LPD-P)
The LPD-P component distributes a P-Channel from one P-Channel controller to up to 8 P-Channel devices.

Low Power Combiner Q-Channel (LPC-Q)
The LPC-Q component combines the Q-Channels from multiple Q-Channel controllers to multiple Q-Channel devices with common control requirements.

P-Channel to Q-Channel Converter (P2Q)
The P2Q component converts a P-Channel to a Q-Channel.

Clock Controller (CLK-CTRL)
The CLK-CTRL component provides *High-level Clock Gating* (HCG) for a single clock domain.

Power Policy Unit (PPU)
The PPU component is a configurable and programmable P-Channel and Q-Channel power domain controller.

The following figure shows an example system that uses the PCK-600 components to manage three power domains. The PCK-600 components are shown in red and blue.

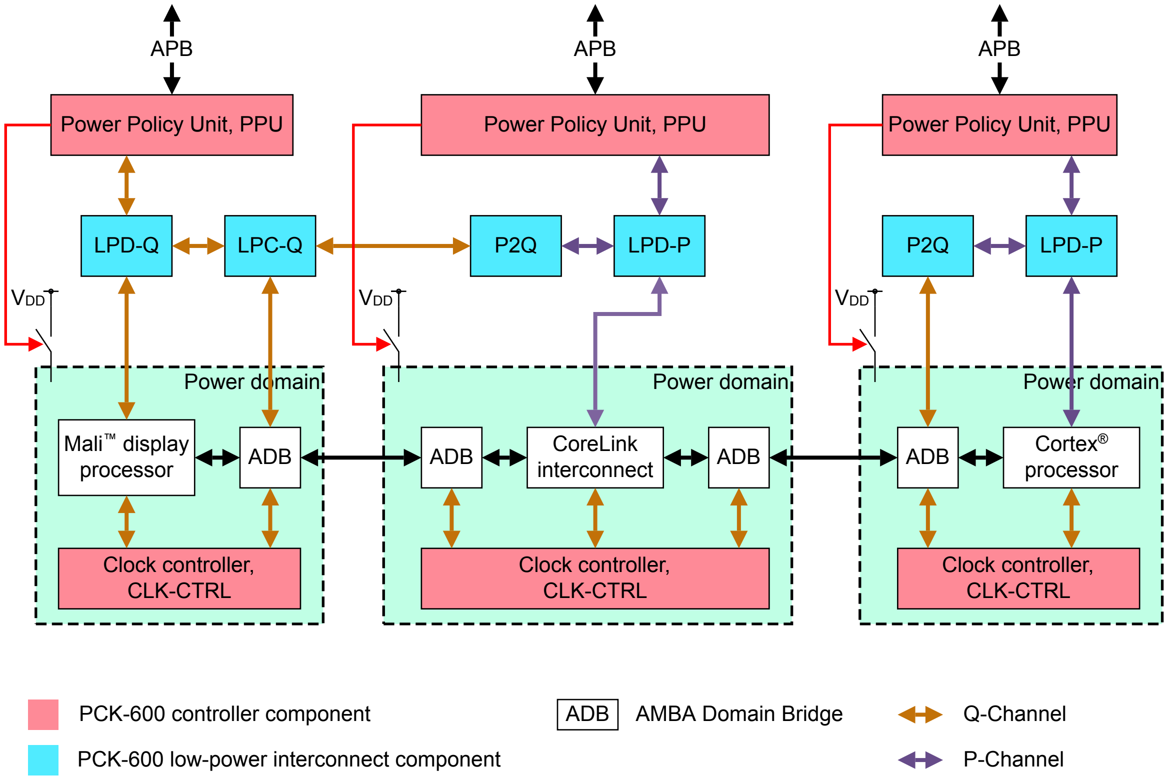

Figure 1-1 Example system that contains PCK-600

1.2 Compliance

The PCK-600 Power Control Kit complies with, or includes components that comply with, the following specifications:

- *Arm® Power Policy Unit Architecture Specification, version 1.1*.
- *Arm® Clock Controller Architecture Specification, version 1.0*.
- *Arm® Power Control System Architecture Specification, version 2.0*.
- *AMBA® Low Power Interface Specification, Arm® Q-Channel and P-Channel Interfaces*.
- *Arm® AMBA® 3 APB Protocol Specification, v1.0*.

This *Technical Reference Manual* (TRM) complements the architecture specifications and protocol specifications. The TRM does not duplicate information from these sources.

1.3 Product documentation

Documentation that is provided with this product includes a *Technical Reference Manual* (TRM) and a *Configuration and Integration Manual* (CIM), together with architecture and protocol information.

For relevant protocol and architectural information that relates to this product, see *Additional reading* on page 8.

The PCK-600 documentation is as follows:

Technical Reference Manual

The TRM describes the functionality and the effects of functional options on the behavior of the PCK-600. It is required at all stages of the design flow. The choices that are made in the design flow can mean that some behaviors that the TRM describes are not relevant. If you are programming the PCK-600, contact the implementer to determine:

- The available build configuration options.
- The address map for the *Power Policy Unit*s (PPUs) in the PCK-600.

The TRM complements architecture and protocol specifications and relevant external standards. It does not duplicate information from these sources.

Configuration and Integration Manual

The CIM describes:

- The available build configuration options.
- How to configure the *Register Transfer Level* (RTL) with the build configuration options.
- How to integrate PCK-600 into a SoC.
- How to implement PCK-600 into your design.
- The processes to validate the configured design.

The Arm product deliverables include reference scripts and information about using them to implement your design.

The CIM is a confidential book that is only available to licensees.

1.4 Product revisions

This section describes the differences in functionality between product revisions:

r0p0	First release.
r0p0-r0p1	• Added support for PPU operating modes. • Updated the REV field to `0x1`. See *3.4.7 Peripheral ID 2* on page 3-52.
r0p1-r0p2	Fixed an LPD-P clock domain crossing issue.
r0p2-r0p3	Fixed an LPD-P issue. Removed restriction on PPU configurations.

Chapter 2

Functional description

This chapter describes the functionality of each PCK-600 component.

It contains the following sections:

2.1 About the LPD-Q Q-Channel Distributor

The *Low Power Distributor Q-Channel* (LPD-Q) component enables a Q-Channel controller to control, and potentially sequence, multiple Q-Channel devices.

The LPD-Q supports from 2-32 device Q-Channel interfaces and can be configured to operate in the following modes:

Q-Channel expander

The controller Q-Channel transition request is broadcast to all device Q-Channels, in parallel. The transition requests that are sent to the devices can complete in any order.

Q-Channel sequencer

A controller Q-Channel transition request is passed sequentially to each device Q-Channel. Each transition request must complete before the LPD-Q can send a transition request to the next device.

The control Q-Channel (**ctrl_*** signals) receives power mode requests from the Q-Channel controller. The LPD-Q uses the device Q-Channels (**dev_*** signals) to send the requests to the devices. The LPD-Q uses **clk_qactive_o** to indicate when it requires a clock signal, **clk**.

The **ctrl_qactive_o** output is the logical OR of the multiple device inputs, **dev_qactive_i<X>**. The path from the **dev_qactive_i<X>** signals to the **ctrl_qactive_o** output is a combinatorial path.

The **clk_qactive_o** is a Q-Channel signal that is HIGH:

- When a device has the **dev_qreqn_o<X>** and **dev_qacceptn_i<X>** signals in opposites states.
- When any **dev_qdeny_i<X>** is HIGH.
- When **ctrl_qreqn_i** and **ctrl_qacceptn_o** signals are in opposites states.
- When **ctrl_qdeny_o** is HIGH.

The type of response that the LPD-Q generates to the controller, in response to a quiescence request, depends on the responses that the LPD-Q receives from the devices:

- The LPD-Q accepts a controller quiescence request, if all devices indicate acceptance of the quiescence request by setting **dev_qacceptn_i<X>** LOW. When all the **dev_qacceptn_i<X>** are LOW, then the LPD-Q sets **ctrl_qacceptn_o** LOW.
- The LPD-Q denies a controller quiescence request, either when:
  - A **dev_qdeny_i<X>** asserts, in response to the assertion of **dev_qreqn_o<X>**.
  - A **dev_qactive_i<X>** goes HIGH, between the assertion of all **dev_qreqn_o<X>** signals and the assertion of the last **dev_qacceptn_i<X>**. This behavior only occurs when the `ACTIVE_DENY` configuration parameter is set to 1.

  When either of these denial conditions occur, then the LPD-Q sets **ctrl_qdeny_o** HIGH.

Input resynchronization

The LPD-Q supports optional resynchronization on either or both the controller and device interfaces.

This section contains the following subsections:

- *2.1.1 LPD-Q operating in expander mode* on page 2-18.
- *2.1.2 LPD-Q operating in sequencer mode* on page 2-19.
- *2.1.3 LPD-Q configuration parameters* on page 2-20.

2.1.1 LPD-Q operating in expander mode

If the `SEQUENCER` parameter is set to zero, then the LPD-Q operates in expander mode.

In expander mode, when the LPD-Q receives a quiescent entry or exit request on the control Q-Channel, it then sends the request to all the device Q-Channels. The LPD-Q waits for all devices to accept or a denial to occur, before it generates the response to the controller.

The following figure shows the LPD-Q configured as an expander.

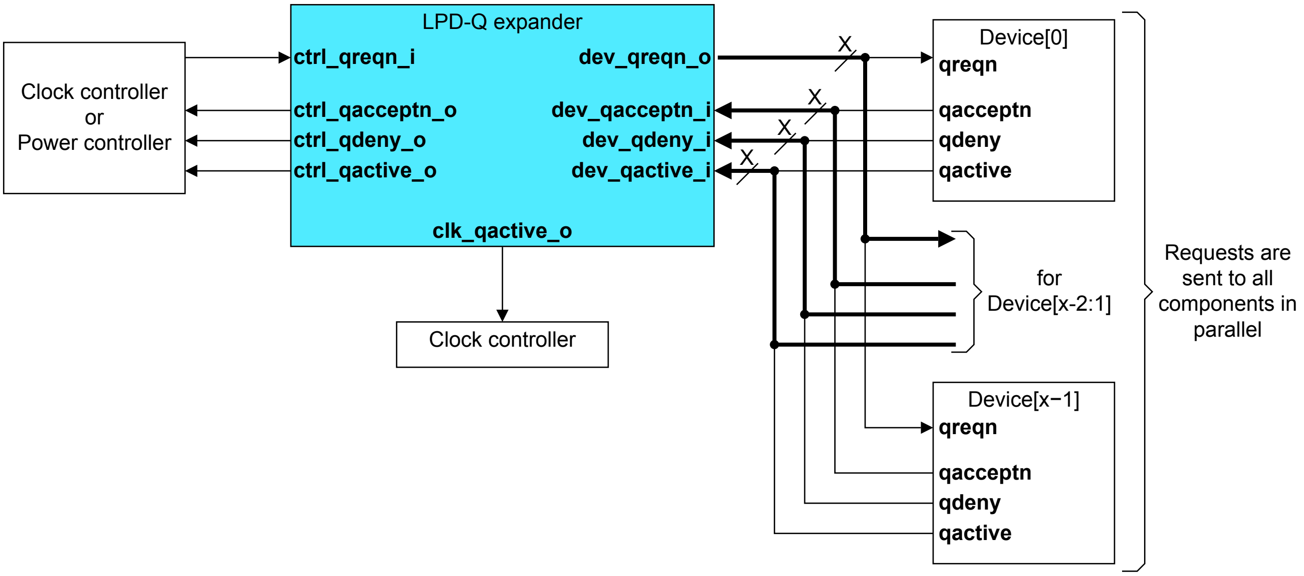

Figure 2-1 Example LPD-Q expander connections

When a denial scenario occurs, the LPD-Q returns all device Q-Channels to the running state. The control Q-Channel does not complete until the devices return to the running state.

2.1.2 LPD-Q operating in sequencer mode

If the `SEQUENCER` parameter is set to one, then the LPD-Q operates in sequencer mode.

In sequencer mode, when the LPD-Q receives a quiescent entry or exit request on the control Q-Channel, it then sequentially sends the request to all the device Q-Channels. The LPD-Q waits for the response from each device before sending a quiescence request to the next device. The LPD-Q waits for all devices to respond before generating the response to the controller.

The following figure shows the LPD-Q configured as a sequencer.

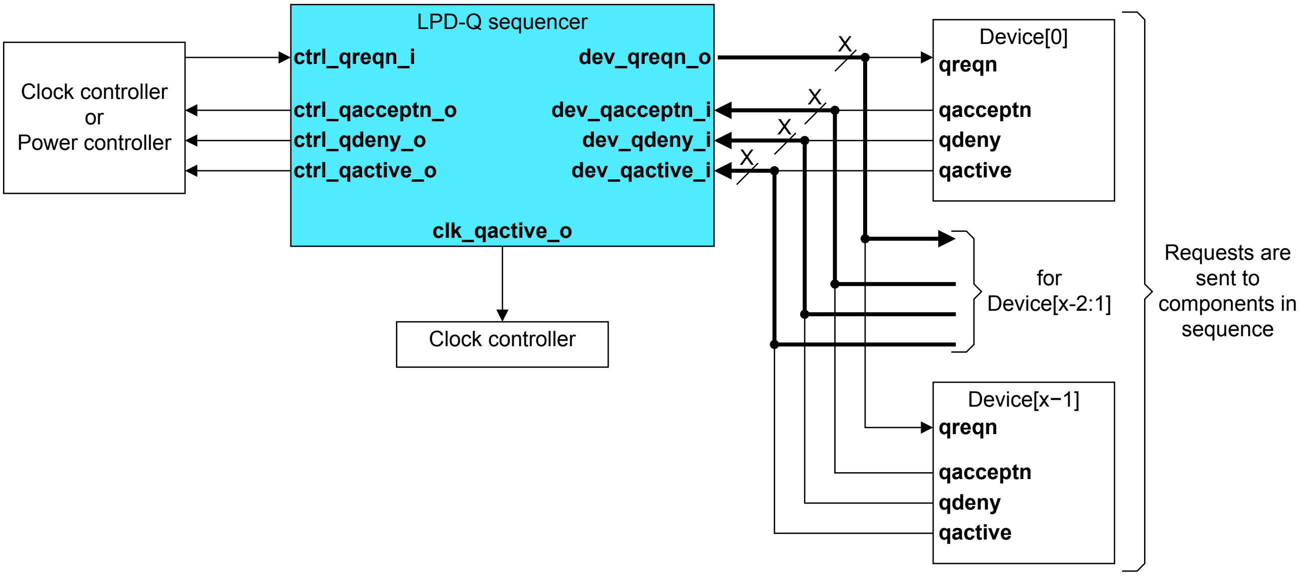

Figure 2-2 Example LPD-Q sequencer connections

Quiescence entry sequence

When the controller issues a quiescence entry request by asserting **ctrl_qreqn_i**, the LPD-Q sends a quiescence entry request on the highest device Q-Channel, **dev_qreqn_o<NUM_QCHL−1>**. If the device accepts the request by setting **dev_qacceptn_i<NUM_QCHL−1>** LOW, the LPD-Q decrements the device number and then sends a quiescence entry request to the next device. The LPD-Q repeats this process until all Q-Channels are quiescent or a denial occurs. The LPD-Q only sends an accept response to the controller, by asserting **ctrl_qacceptn_o**, when all devices accept the quiescence entry requests.

If any device denies the quiescence entry request by asserting **dev_qdeny_i<X>**, or asserts **dev_qactive_i<X>** when `ACTIVE_DENY`=1, then the LPD-Q asserts the **ctrl_qdeny_o** output. The LPD-Q does not send the remaining entry requests to Q-Channels that are in the Q_RUN state. If the quiescence entry request sequence was interrupted due to the assertion of:

dev_qdeny_i<X>

> The LPD-Q returns channels to the Q_RUN state in ascending numerical order, from the device that asserted **dev_qdeny_i<X>** to device [`NUM_QCHL`−1].

dev_qactive_i<X>

> The LPD-Q returns channels to the Q_RUN state, starting with the last channel to enter the Q_STOPPED state, followed by the channels in ascending numerical order, from the last device that accepted to device [`NUM_QCHL`−1].
>
> When a **dev_qactive_i<x>** asserts, the LPD-Q waits until the device Q-Channel to which it is currently requesting quiescence accepts or denies the request, before it returns it to the running state. The point at which this channel returns to Q_RUN might not be in sequence with the other channels.

The LPD-Q only deasserts **ctrl_qdeny_o**, when all device channels return to the Q_RUN state and the controller sets **ctrl_qreqn_i** HIGH.

Quiescence exit sequence

When the controller issues a quiescence exit request, by deasserting **ctrl_qreqn_i**, the LPD-Q sends exit requests in a sequential sequence from device[0] to device[`NUM_QCHL`−1].

2.1.3 LPD-Q configuration parameters

There are multiple configuration parameters that configure or modify the functionality of the Low Power Distributor Q-Channel.

The following table shows the LPD-Q configuration parameters.

Table 2-1 LPD-Q configuration parameters

Parameter	Possible settings	Default	Description
`SEQUENCER`	0, 1	0	0 = The LPD-Q operates as an expander. 1 = The LPD-Q operates as a sequencer.
`NUM_QCHL`	2-32	2	Sets the number of device (**dev_***) Q-Channel interfaces in the LPD-Q.
`CTRL_Q_CH_SYNC`	0, 1	1	0 = A synchronizer is not present on the **ctrl_qreqn_i** input. 1 = A synchronizer is present on the **ctrl_qreqn_i** input.

Table 2-1 LPD-Q configuration parameters (continued)

Parameter	Possible settings	Default	Description
`DEV_Q_CH_SYNC`	0, 1	1	0 = Synchronizers are not present on the **dev_qacceptn_i[N]** or **dev_qdeny_i[N]** inputs. 1 = Synchronizers are present on the **dev_qacceptn_i[N]** and **dev_qdeny_i[N]** inputs.
`ACTIVE_DENY`	0, 1	1	0 = Support for denying a quiescence request by using **dev_qactive_i[N]** is not included. 1 = Support for denying a quiescence request using **dev_qactive_i[N]** is included. Synchronizers are included on the **dev_qactive_i[N]** inputs, where these signals are used internally depending on the value of `DEV_Q_CH_SYNC`. **Note** The path from a device to the controller **QACTIVE** is combinatorial, irrespective of whether synchronizers are present.

2.2 About the LPD-P P-Channel Distributor

The *Low Power Distributor P-Channel* (LPD-P) component enables a P-Channel controller to control, and potentially sequence, multiple P-Channel devices.

The LPD-P supports from 1-8 device P-Channel interfaces and can be configured to operate in the following modes:

P-Channel expander

The controller P-Channel transition request is broadcast to all device P-Channels, in parallel. The transition requests that are sent to the devices can complete in any order.

P-Channel sequencer

A controller P-Channel transition request is passed sequentially to each device P-Channel. Each transition request must complete before the LPD-P can send a transition request to the next device.

The following figure shows the main interfaces on the LPD-P.

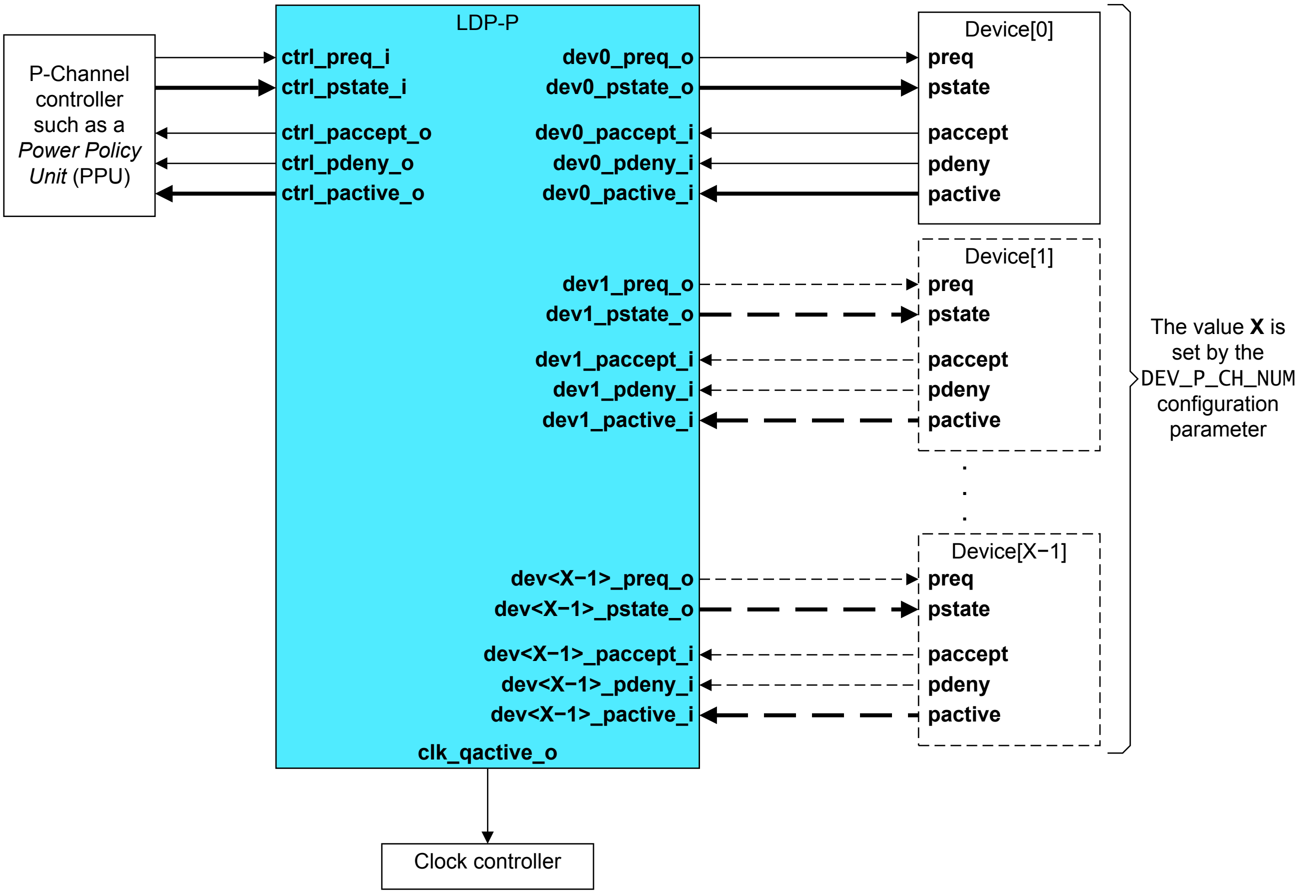

Figure 2-3 LPD-P interfaces

The control P-Channel (**ctrl_*** signals) receives power mode requests from the P-Channel controller. The LPD-P uses the device P-Channels (**dev<X>_*** signals) to send the power mode requests to the devices. The LPD-P uses **clk_qactive_o** to indicate when it requires a clock signal, **clk**.

The control P-Channel supports all the PPU power modes and 16 operating modes. The **ctrl_pactive_o[P_CH_PACTIVE_LEN−1:0]** outputs are the logical OR of the multiple device inputs, **dev<X>_pactive_i[P_CH_PACTIVE_LEN−1:0]**. The path from the **dev<X>_pactive_i** signals to the **ctrl_pactive_o** output is a combinatorial path.

During integration, you can configure which **dev<X>_pactive_i** inputs are included in the OR function. See *2.2.5 PACTIVE remapping* on page 2-26 for more information.

The **clk_qactive_o** is a Q-Channel signal that is HIGH:

- When any **PREQ**, **PACCEPT**, or **PDENY** signal is HIGH.
- After reset deasserts and until the initialization completes.

The type of response that the LPD-P generates to the controller depends on the responses that the LPD-P receives from the devices:

- The LPD-P accepts a controller request, if all devices indicate acceptance of the request by setting **dev<X>_paccept_i** HIGH.
- The LPD-P denies a controller request, when any device denies the request by setting **dev<X>_pdeny_i** HIGH. The **ctrl_pdeny_o** signal remains HIGH until all devices that accepted the request, revert to their previous **PSTATE** value.

System integrations of the LPD-P in non-PCSA compliant architectures are limited to a maximum **PSTATE** length of 4 bits (16 power modes). The **PSTATE[7:4]** bits are reserved for PCSA operating modes.

Input resynchronization

The LPD-P supports optional resynchronization on either or both the controller and device interfaces. The LPD-P can also be configured to set the device **PSTATE** value, one clock cycle before the LPD-P asserts **dev<X>_preq_o**.

This section contains the following subsections:

- *2.2.1 LPD-P initialization* on page 2-23.
- *2.2.2 LPD-P operating in expander mode* on page 2-24.
- *2.2.3 LPD-P operating in sequencer mode* on page 2-24.
- *2.2.4 PSTATE remapping* on page 2-25.
- *2.2.5 PACTIVE remapping* on page 2-26.
- *2.2.6 LPD-P configuration parameters* on page 2-26.

2.2.1 LPD-P initialization

The P-Channel distributor initializes all component P-Channels to `0x00`, which is the PCSA OFF power mode.

If the LPD-P design is used in a non-compliant PCSA architecture, only the lower 4 bits (**PSTATE[3:0]**) are supported for up to 16 unique power modes. **PSTATE[7:4]** bits (PCSA operating modes) are reserved.

In both expander mode and sequencer mode, the LPD-P initializes all devices in parallel by broadcasting a P-Channel request to all downstream devices. This request has **dev<x>_pstate_o** set to `0x00`. The LPD-P then performs one of the following actions:

- If the P-Channel power mode or operating mode value (**ctrl_pstate_i[3:0,7:4]**) presented by the controller during initialization is set to `0x00` then:
  - If no device power mode or operating mode remapping is set, this ends the initialization sequence.
  - If for one or more devices the power mode or operating mode is remapped from `0x00` to a non-zero value, then a second P-Channel request is performed for each device requiring the remapped non-zero value.
- If the P-Channel power mode or operating mode value (**ctrl_pstate_i[3:0,7:4]**) presented by the controller during initialization is set to a value greater than `0x00`, then a second broadcasted P-Channel request is performed to all devices. Any power mode or operating mode re-mapping is taken into account with this request.

The P-Channel distributor performs initialization by asserting all device **PREQ** signals to HIGH on reset de-assertion. This removes dependency on any reset and clock-enable timings which are related to waiting for a T_{INIT} time as the device confirms it has seen the initialization power mode by setting its **PACCEPT** HIGH.

Note

T_{INIT} is the time period that the **PSTATE** value must be sampled at reset de-assertion, defined in component clock-cycles.

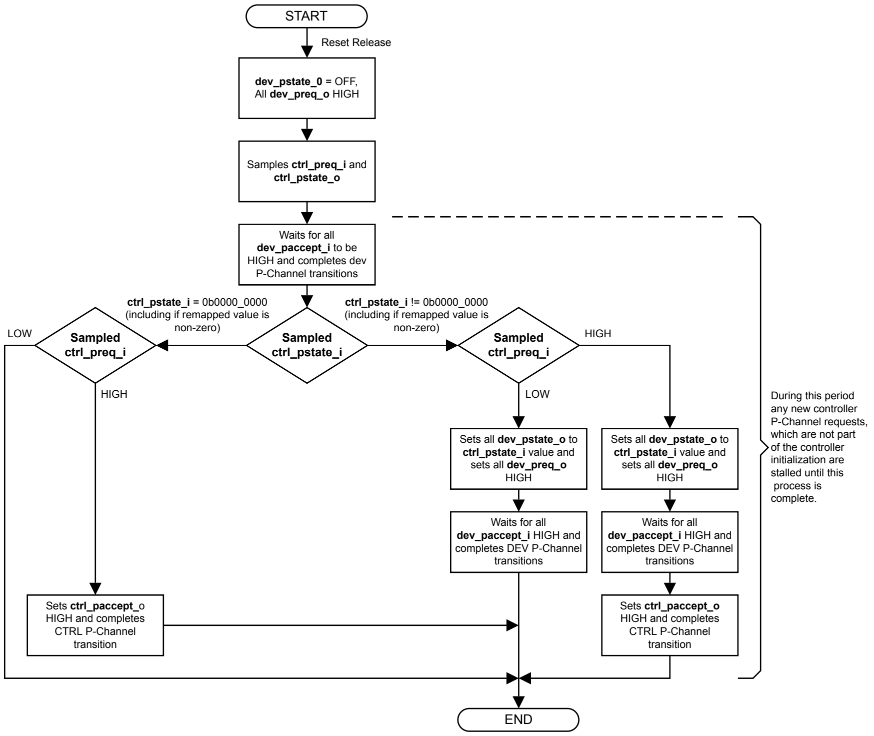

Figure 2-4 Shared LPD-P initialization sequence

2.2.2 LPD-P operating in expander mode

If the `SEQUENCER` parameter is set to zero, then the LPD-P operates in expander mode.

In expander mode, when the LPD-P receives a request on the control P-Channel, it then broadcasts the request to all the device P-Channels. The LPD-P waits for all devices to accept, or a denial to occur, before it generates the response to the controller.

If any device denies the request, the LPD-P reverts all devices that accepted the request to the previously accepted value. The control P-Channel does not complete until the device P-Channels return to the previous state.

2.2.3 LPD-P operating in sequencer mode

If the `SEQUENCER` parameter is set to one, then the LPD-P operates in sequencer mode.

In sequencer mode, when the LPD-P receives a request on the control P-Channel, it then sequentially sends requests to each of the device P-Channels. The LPD-P waits for the response from each device before sending a request to the next device. The LPD-P waits for all devices to respond before generating the response to the controller.

The unmapped (**ctrl_pstate_i[3:0]**) value controls the sequence order as the following entry sequences describe. Also, if `DEV_P_CH_<X>_SAME_EN` == 0 for any device, then determination of auto accept is based on both the mapped operating mode (**dev<X>_pstate_o[7:4]**) and mapped power mode (**dev<X>_pstate_o[3:0]**) values for that device.

Lower power mode entry sequence

When the controller asserts **ctrl_preq_i** and issues a request to a lower power mode, indicated by a lower value on **ctrl_pstate_i[3:0]** than previously, regardless of the operating mode **ctrl_pstate_i[7:4]**, the LPD-P sends the request on the highest device P-Channel, **dev<DEV_P_CH_NUM−1>_preq_o**. If the device accepts the request, then the LPD-P decrements the device number and sends the request to the next device.

If any device denies the request by asserting **dev<X>_pdeny_i**, then the LPD-P asserts the **ctrl_pdeny_o** output. The LPD-P does not send a request to the remaining P-Channels. The LPD-P returns the device P-Channels to the previous state in ascending numerical order, from the last device to accept the request to device [`DEV_P_CH_NUM`−1].

Higher power mode entry sequence

When the controller asserts **ctrl_preq_i** and issues a request to a higher power mode by setting a higher value on **ctrl_pstate_i[3:0]**, regardless of the operating mode **ctrl_pstate_i[7:4]**, then the LPD-P sends the request on the lowest device P-Channel, dev<0>. If the device accepts the request, then the LPD-P increments the device number and then sends the request to the next device.

If any device denies the request by asserting **dev<X>_pdeny_i**, then the LPD-P asserts the **ctrl_pdeny_o** output. The LPD-P does not send a request to the remaining P-Channels. The LPD-P returns the device P-Channels to the previous state in descending numerical order, from the last device to accept the request to device [0].

Same power mode, lower operating mode entry sequence

When the controller asserts **ctrl_preq_i** and issues a request to the same power mode, indicated by the same value on **ctrl_pstate_i[3:0]**, then the operating mode **ctrl_pstate_i[7:4]** controls the sequence order. If the operating mode change is to a lower value, the LPD-P sends the request on the highest device P-Channel, **dev<DEV_P_CH_NUM−1>_preq_o**. If the device accepts the request, then the LPD-P decrements the device number and sends the request to the next device.

If any device denies the request by asserting **dev<X>_pdeny_i**, then the LPD-P asserts the **ctrl_pdeny_o** output. The LPD-P does not send a request to the remaining P-Channels. The LPD-P returns the device P-Channels to the previous state in ascending numerical order, from the last device to accept the request to device [`DEV_P_CH_NUM`−1].

Same power mode, same or higher operating mode entry sequence

When the controller asserts **ctrl_preq_i** and issues a request to the same power mode, indicated by the same value on **ctrl_pstate_i[3:0]**, then the operating mode **ctrl_pstate_i[7:4]** controls the sequence order. If the operating mode change is to the same or higher value, the LPD-P sends the request on the lowest device P-Channel, dev<0>. If the device accepts the request, then the LPD-P increments the device number and then sends the request to the next device.

If any device denies the request by asserting **dev<X>_pdeny_i**, then the LPD-P asserts the **ctrl_pdeny_o** output. The LPD-P does not send a request to the remaining P-Channels. The LPD-P returns the device P-Channels to the previous state in descending numerical order, from the last device to accept the request to device [0].

2.2.4 PSTATE remapping

For each device P-Channel, the LPD-P can be configured to remap the controller P-Channel **PSTATE** value to a different value for the device. This remapping allows devices to use an alternative power mode to the controller power mode.

The **PSTATE** bus provides power mode values on bits[3:0] and operating mode values on bits[7:4]. For each device P-Channel, the LPD-P has a parameter to map the power mode and another parameter to map the operating mode.

For the case, where you configure an LPD-P to support one device P-Channel only, then you create a single P-Channel remapping component.

See *2.2.6 LPD-P configuration parameters* on page 2-26 for more information about the **PSTATE** remapping parameters.

2.2.5 PACTIVE remapping

For each device P-Channel, the LPD-P can be configured to remap the device P-Channel **PACTIVE** input to zero or more controller P-Channel **PACTIVE** outputs.

Typically, the LPD-P generates a **ctrl_pactive_o** output bit from the logical OR of each **dev<X>_pactive_i**, with the same bit number. However, in certain situations, it might be necessary to alter this behavior.

Typically, **PACTIVE[0]** for the OFF power mode is not required because it is requested by default when all other **PACTIVE** bits are LOW. However, the combination of these bits is still supported for devices that might not support the **PACTIVE** mapping in the *Arm® Power Control System Architecture Specification, version 2.0*.

A **dev<X>_pactive_i** input can be assigned to:

- One or more **ctrl_pactive_o** output bits, by setting one or more bits in the configuration parameter.
- Zero **ctrl_pactive_o** output bits, by setting all its parameter bits to `0b0`.

2.2.6 LPD-P configuration parameters

There are multiple configuration parameters that determine the functionality of the Low Power Distributor P-Channel.

The following table shows the LPD-P configuration parameters.

Table 2-2 LPD-P configuration parameters

Parameter	Possible settings	Description
`SEQUENCER`	0, 1	0 = The LPD-P operates as an expander. 1 = The LPD-P operates as a sequencer.
`CTRL_P_CH_SYNC`	0, 1	0 = A synchronizer is not present on the **ctrl_preq_i** input. 1 = A synchronizer is present on the **ctrl_preq_i** input.
`DEV_P_CH_SYNC`	0, 1	0 = Synchronizers are not present on the **dev<X>_paccept_i** or **dev<X>_pdeny_i** inputs. 1 = Synchronizers are present on the **dev<X>_paccept_i** and **dev<X>_pdeny_i** inputs.
`DEV_P_CH_<X>_SAME_EN`	0, 1	Controls whether the P-Channel for device <X> performs a transition to a **PSTATE** value when the device is already in that mode: 0 = The LPD-P does not send a transition on the **dev_*** P-Channel interface. 1 = The LPD-P sends a transition on the **dev_*** P-Channel interface.

Table 2-2 LPD-P configuration parameters (continued)

Parameter	Possible settings	Description
`DEV_P_CH_PREQ_DLY`	0, 1	Controls whether there is a one clock cycle delay between the assertion of **dev<X>_pstate_o** and the assertion of **dev<X>_preq_o**: 0 = Zero clock cycles. 1 = One clock cycle.
`DEV_P_CH_NUM`	1-8	Sets the number of device (**dev_***) P-Channel interfaces in the LPD-P.
`P_CH_PACTIVE_LEN`	1-32	Sets the **PACTIVE** bus width, for all P-Channels.
`P_CH_PSTATE_LEN`	1-8	Sets the **PSTATE** bus width, for all P-Channels.
`DEV_P_CH_<X>_PWR_PSTATE_MAP_<CTRL_PWR_PSTATE>(<DEV_PWR_PSTATE>)`	`0b0000-0b1111` when `P_CH_PSTATE_LEN` ≥ 4. `0b000-0b111` when `P_CH_PSTATE_LEN` == 3. `0b00-0b11` when `P_CH_PSTATE_LEN` == 2. `0b0-0b1` when `P_CH_PSTATE_LEN` == 1.	Sets the device <x> power mode **PSTATE** value to use, when the LPD-P receives a given controller power mode **PSTATE** value: **<CTRL_PWR_PSTATE>** The controller **PSTATE[3:0]** value, which the LPD-P receives on **ctrl_pstate_i[3:0]**. **<DEV_PWR_PSTATE>** The device **PSTATE[3:0]** value, which the LPD-P issues on the **dev<X>_pstate_o[3:0]** output.

Table 2-2 LPD-P configuration parameters (continued)

Parameter	Possible settings	Description
`DEV_P_CH_<X>_OP_PSTATE_MAP_<CTRL_OP_PSTATE>(<DEV_OP_PSTATE>)`	`0b0000`-`0b1111` when `P_CH_PSTATE_LEN` == 8. `0b000`-`0b111` when `P_CH_PSTATE_LEN` == 7. `0b00`-`0b11` when `P_CH_PSTATE_LEN` == 6. `0b0`-`0b1` when `P_CH_PSTATE_LEN` == 5.	Sets the device <x> operating mode **PSTATE** value to use, when the LPD-P receives a given controller operating mode **PSTATE** value: **<CTRL_OP_PSTATE>** The controller **PSTATE[7:4]** value, which the LPD-P receives on **ctrl_pstate_i[7:4]**. **<DEV_OP_PSTATE>** The device **PSTATE[7:4]** value, which the LPD-P issues on the **dev<X>_pstate_o[7:4]** output.
`DEV_P_CH_<X>_PACTIVE_MAP_<N>[P_CH_PACTIVE_LEN]`	0 or 1, for each bit	This parameter controls how the specified **dev<X>_pactive_i[P_CH_PACTIVE_LEN−1:0]** bit contributes to the **ctrl_pactive_o[P_CH_PACTIVE_LEN−1:0]** outputs. There is a parameter for each device P-Channel **PACTIVE** input. <X> is the device P-Channel number. <N> is the device **PACTIVE** bit number, [`P_CH_PACTIVE_LEN`−1:0], for the specified device P-Channel. The parameter contains a bit for each controller **PACTIVE** bit. When a parameter bit is: 0 = The specified **dev<X>_pactive_i[P_CH_PACTIVE_LEN−1:0]** bit is not OR combined to create the **ctrl_pactive_o[P_CH_PACTIVE_LEN−1:0]** bit, as specified by the bit position. 1 = The specified **dev<X>_pactive_i[P_CH_PACTIVE_LEN−1:0]** bit is OR combined to create the **ctrl_pactive_o[P_CH_PACTIVE_LEN−1:0]** bit, as specified by the bit position.

2.3 About the LPC-Q Q-Channel Combiner

The *Low Power Combiner Q-Channel* (LPC-Q) allows two or more Q-Channel controllers to control one or more devices that all have the same control requirements.

When any of the control Q-Channels request quiescence, the LPC-Q moves the device Q-Channels to the quiescent state. The device Q-Channels are brought out of quiescence when the last control Q-Channel exits quiescence.

The following figure shows the main interfaces on the LPC-Q.

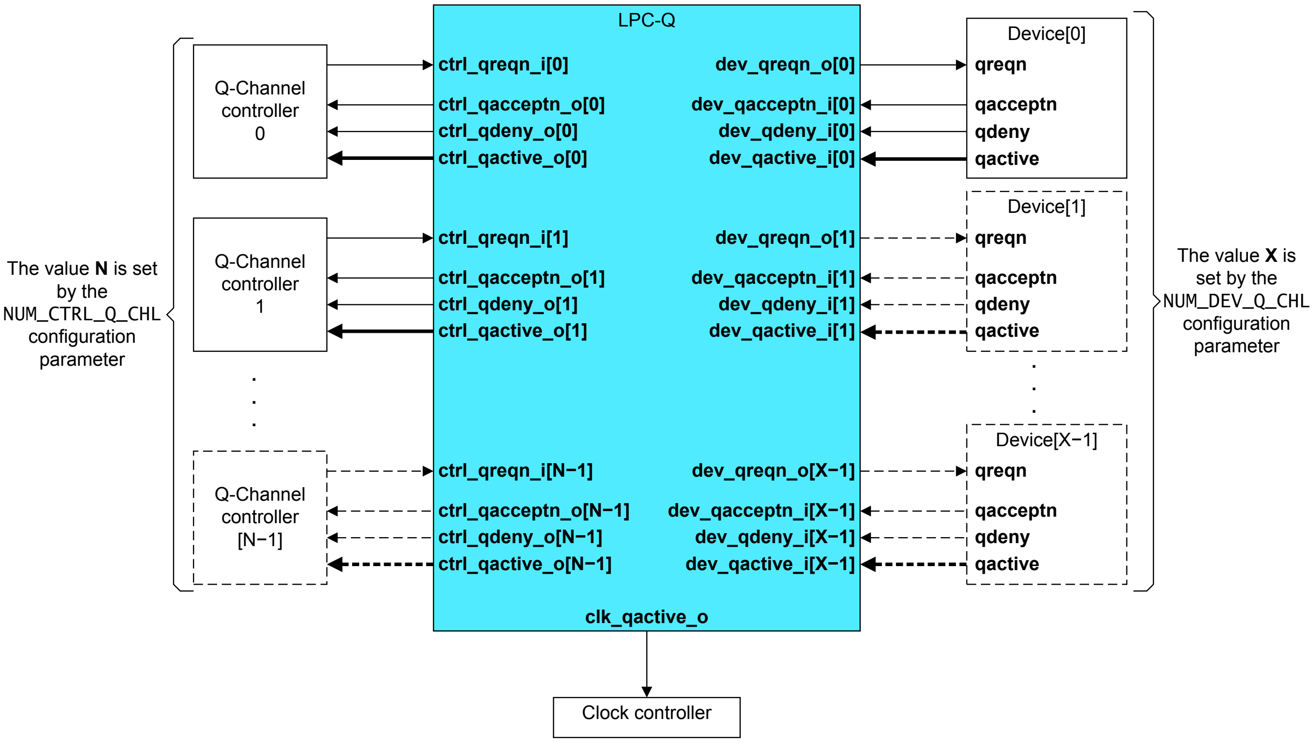

Figure 2-5 LPC-Q interfaces

The **ctrl_qactive_o[<N>]** outputs are the logical OR of one or more device inputs, **dev_qactive_i[<X>]**. The path from the **dev_qactive_i[<X>]** signals to a **ctrl_qactive_o[<N>]** output is a combinatorial path.

The **clk_qactive_o** is a Q-Channel signal that is HIGH when any of the following conditions occur:

- A device has the **dev_qreqn_o[<X>]** and **dev_qacceptn_i[<X>]** signals in opposites states.
- Any **dev_qdeny_i[<X>]** is HIGH.
- A controller has the **ctrl_qreqn_i[<N>]** and **ctrl_qacceptn_o[<N>]** signals in opposites states.
- Any **ctrl_qdeny_o[<N>]** is HIGH.

You might use an LPC-Q where there is a cross-domain component that must be in a quiescent state whenever either of the associated domains goes to a quiescent power mode such as full retention or off, and there is no fixed relationship between the domains. For example, a cross-domain component such as a protocol domain bridge that operates across two power domains, where the bridge must be put into a quiescent mode before either domain is powered down.

2.3.1 LPC-Q configuration parameters

There are several configuration parameters that determine the functionality of the Low Power Combiner Q-Channel.

The following table shows the LPC-Q configuration parameters.

Table 2-3 LPC-Q configuration parameters

Parameter	Possible settings	Default	Description
`NUM_CTRL_Q_CHL`	2-32	2	Sets the number of control interfaces, **ctrl_**, in the LPC-Q.
`NUM_DEV_Q_CHL`	1-32	1	Sets the number of device interfaces, **dev_**, in the LPC-Q.
`CTRL_Q_CH_SYNC`	0, 1	1	0 = Synchronizers are not present on the **ctrl_qreqn_i** inputs. 1 = Synchronizers are present on the **ctrl_qreqn_i** inputs.
`DEV_Q_CH_SYNC`	0, 1	1	0 = Synchronizers are not present on the **dev_qacceptn_i** and **dev_qdeny_i** inputs. 1 = Synchronizers are present on the **dev_qacceptn_i** and **dev_qdeny_i** inputs.

2.4 About the P2Q Converter

The *P-Channel to Q-Channel Converter* (P2Q) converts a single P-Channel to a single Q-Channel. The P2Q uses the maximum 8-bit width for **PSTATE** and 32-bit for **PACTIVE**.

The following figure shows the main interfaces on the P2Q.

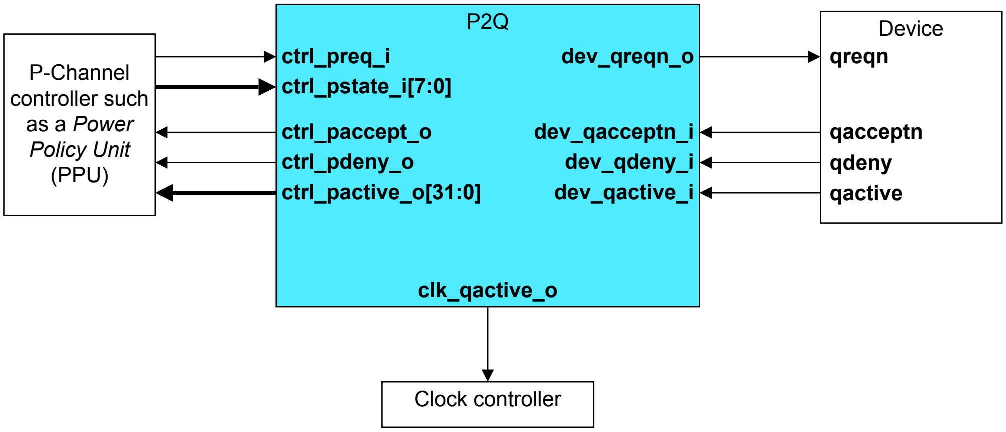

Figure 2-6 P2Q interfaces

The P2Q receives power mode requests on its control P-Channel. The P2Q uses the device Q-Channel (**dev_*** signals) to send requests to the device. The P2Q uses **clk_qactive_o** to indicate when it requires a clock signal, **clk**.

The control P-Channel supports all the PPU power modes and 16 operating modes. The integrator uses the `CTRL_P_CH_PWR_PSTATE_MAP[15:0]` and `CTRL_P_CH_OP_PSTATE_MAP[15:0]` configuration parameters, to choose which power modes and operating modes generate a quiescence request. See *2.4.2 P2Q configuration parameters* on page 2-32.

clk_qactive_o is a Q-Channel signal that is HIGH:

- When any of **ctrl_preq_i**, **ctrl_paccept_o**, or **ctrl_pdeny_o** are HIGH.
- When **dev_qreqn_o** and **dev_qacceptn_i** are in opposite states.
- When **dev_qdeny_i** is HIGH.
- After reset deasserts and until the initialization completes.

When the P2Q becomes idle, then **clk_qactive_o** goes LOW and the **clk** input can be gated to reduce dynamic power.

2.4.1 P2Q initialization

The P2Q Converter goes through an initialization phase on reset de-assertion, where it can drive both the master side and the slave side of the interface.

At the start of the initialization phase the P2Q Converter samples the values of **ctrl_preq_i** and **ctrl_pstate_i** signals. The P2Q Converter takes the CDC synchronizer on the **ctrl_preq_i** into account before sampling the **ctrl_pstate_i** value. The sampled value of the **ctrl_pstate_i** determines the path that the P2Q Converter takes through the state machine, as shown in *Figure 2-7 P2Q initialization sequence* on page 2-32.

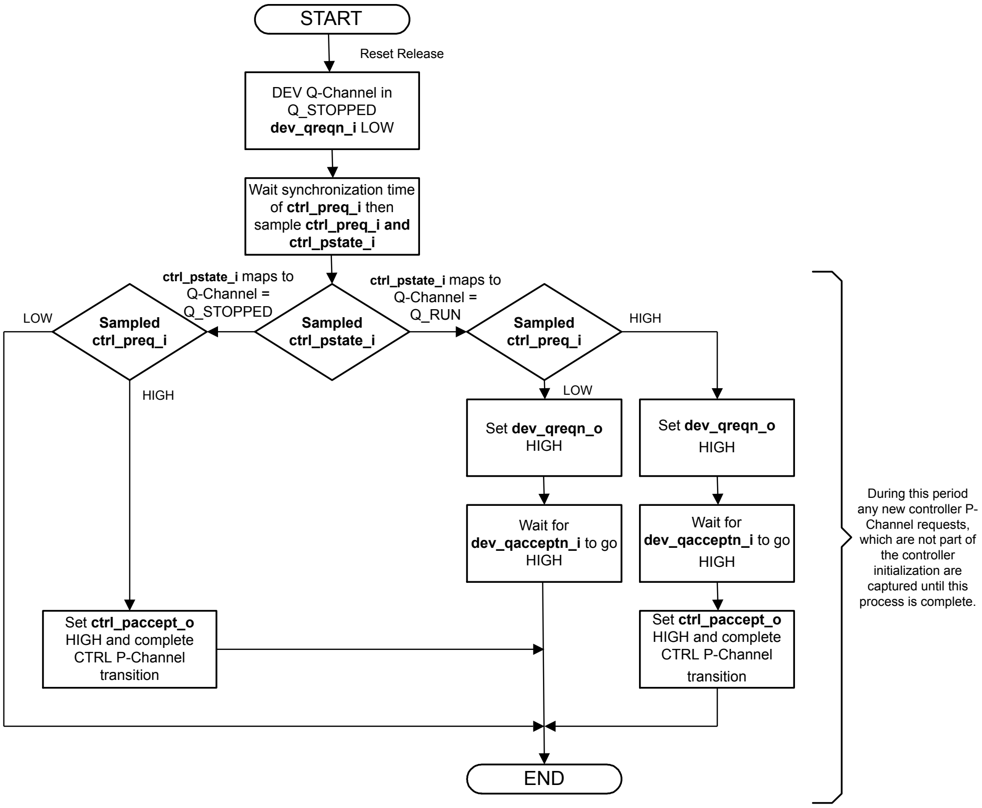

Figure 2-7 P2Q initialization sequence

2.4.2 P2Q configuration parameters

There are multiple configuration parameters that modify the functionality of the P-Channel to Q-Channel Converter.

The following table shows the P2Q configuration parameters.

Table 2-4 P2Q configuration parameters

Parameter	**Possible settings**	**Default**	**Description**
`CTRL_P_CH_PWR_PSTATE_MAP[15:0]`	0 or 1, for each bit	`0b00000001_11110000`.	The value of the **ctrl_pstate_i[3:0]** input signal represents one of 16 possible power modes. This parameter assigns a single bit to each power mode. Therefore, you can select the device Q-Channel to be in the running or quiescent state, for each power mode: 0 = The Q-Channel is quiescent, or if active then quiescence is requested. 1 = The Q-Channel is running, or if quiescent then running is requested. However, this state only occurs if `CTRL_P_CH_OP_PSTATE_MAP[X]` == 1, where X is the P-Channel operating mode, which **ctrl_pstate_i[7:4]** supplies. For example, if `CTRL_P_CH_PWR_PSTATE_MAP[5]` == 0, then when **ctrl_pstate_i[3:0]** == `0b0101`, the Q-Channel is quiescent or quiescence is requested if it is active.
`CTRL_P_CH_OP_PSTATE_MAP[15:0]`	0 or 1, for each bit	`0b00000001_11110000`.	The value of the **ctrl_pstate_i[7:4]** input signal represents one of 16 possible operating modes. This parameter assigns a single bit to each operating mode. Therefore, you can select the Q-Channel to be in the running or quiescent state, for each operating mode: 0 = The Q-Channel is quiescent, or if active then quiescence is requested. 1 = The Q-Channel is running, or if quiescent then running is requested. However, this state only occurs if `CTRL_P_CH_PWR_PSTATE_MAP[X]` == 1, where X is the P-Channel power mode, which **ctrl_pstate_i[3:0]** supplies. For example, if `CTRL_P_CH_OP_PSTATE_MAP[7]` == 0, then when **ctrl_pstate_i[7:4]** == `0b0111` the Q-Channel is quiescent or quiescence is requested if it is active.

Table 2-4 P2Q configuration parameters (continued)

Parameter	Possible settings	Default	Description
`CTRL_P_CH_PACTIVE_MAP[31:0]`	0 or 1, for each bit	`0b11111111_11111111 11111111_11111111`.	This parameter controls whether the Q-Channel **dev_qactive_i** drives each bit in the **ctrl_pactive_o[31:0]** output: 0 = The corresponding bit in **ctrl_pactive_o[31:0]** is tied LOW. 1 = The corresponding bit in **ctrl_pactive_o[31:0]** is set to the value of **dev_qactive_i**.
`CTRL_P_CH_SYNC`	0, 1	1	0 = A synchronizer is not present on the **ctrl_preq_i** input. 1 = A synchronizer is present on the **ctrl_preq_i** input.
`DEV_Q_CH_SYNC`	0, 1	1	0 = Synchronizers are not present on the **dev_qacceptn_i** or **dev_qdeny_i** inputs. 1 = Synchronizers are present on the **dev_qacceptn_i** and **dev_qdeny_i** inputs.

2.5 About the CLK-CTRL

The *Clock Controller* (CLK-CTRL) provides high-level clock gating for devices in a clock domain that support Q-Channel *Low Power Interface* (LPI) clock gating. The CLK-CTRL uses the Q-Channels to ensure that the devices are in a quiescent state before it gates the clock. The CLK-CTRL also ensures that the clock is running, before it allows a device to exit the quiescent state.

The following figure shows the main interfaces on the CLK-CTRL.

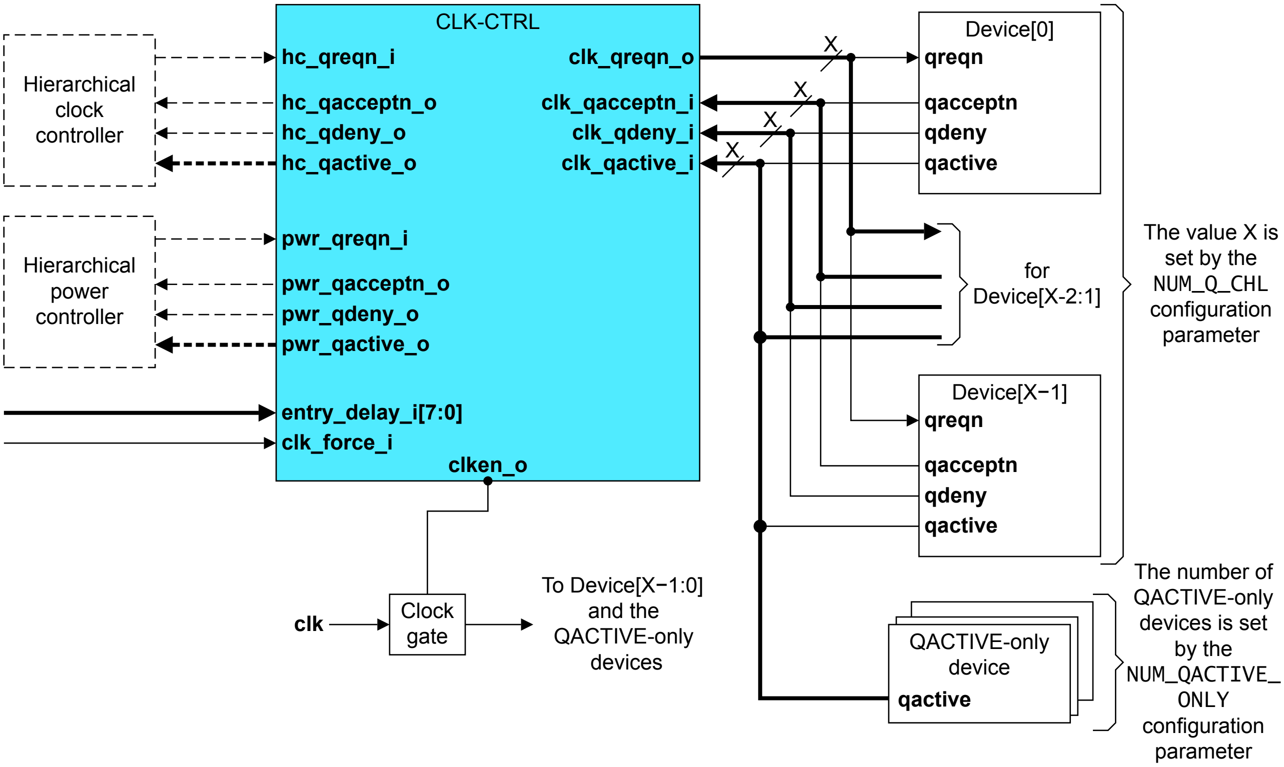

Figure 2-8 CLK-CTRL interfaces

The CLK-CTRL monitors the Q-Channel **clk_qactive_i** inputs, to know when it must perform the Q-Channel requests. When all the Q-Channels are in the quiescent state, the CLK-CTRL sets **clken_o** LOW, to stop the clock. When any **clk_qactive_i** goes HIGH, the CLK-CTRL enables the clock and moves all the device Q-Channels to the running state. You can apply 0-255 **clk** cycles of hysteresis to the quiescence entry, by altering the state of the **entry_delay_i[7:0]** configuration input.

You can disable the clock gating feature by setting **clk_force_i** HIGH.

The `NUM_Q_CHL` configuration parameter controls how many Q-Channel interfaces are on the CLK-CTRL.

The `NUM_QACTIVE_ONLY` configuration parameter controls how many **QACTIVE**-only interfaces are on the CLK-CTRL.

The CLK-CTRL has two optional hierarchical control Q-Channels. You can connect a higher-level clock or power controller to the following hierarchical Q-Channels:

Hierarchical clock control Q-Channel

Allows a higher-level clock controller to make requests to the CLK-CTRL. Signals on this interface have a **hc_** prefix such as **hc_qreqn_i**.

Hierarchical power control Q-Channel

Allows a higher-level power controller to make requests to the CLK-CTRL. Signals on this interface have a **pwr_** prefix such as **pwr_qreqn_i**.

See the *Arm® Clock Controller Architecture Specification, version 1.0* for more information about the CLK-CTRL functionality.

2.5.1 CLK-CTRL configuration parameters

There are multiple configuration parameters that modify the functionality of the Clock Controller.

The following table shows the CLK-CTRL configuration parameters.

Table 2-5 CLK-CTRL configuration parameters

Parameter	Possible settings	Default	Description
`NUM_Q_CHL`	1-8	1	Sets the number of clock device Q-Channel interfaces in the CLK-CTRL.
`NUM_QACTIVE_ONLY`	0-32	1	Sets the number of **QACTIVE** only Q-Channels.
`HC_Q_CH_SYNC`	0, 1	1	0 = A synchronizer is not present on the **hc_qreqn_i** input. 1 = A synchronizer is present on the **hc_qreqn_i** input.
`PWR_Q_CH_SYNC`	0, 1	1	0 = A synchronizer is not present on the **pwr_qreqn_i** input. 1 = A synchronizer is present on the **pwr_qreqn_i** input.
`CLK_Q_CH_SYNC`	0, 1	1	0 = Synchronizers are not present on the **clk_qacceptn_i[N]** or **clk_qdeny_i[N]** inputs. 1 = Synchronizers are present on the **clk_qacceptn_i[N]** and **clk_qdeny_i[N]** inputs. Where [N] = [`NUM_Q_CHL`−1:0].
`ACTIVE_DENY_EN`	0, 1	1	0 = Support for denying a quiescence request by using **QACTIVE** is not included. 1 = Support for denying a quiescence request using **QACTIVE** is included. Synchronizers are included on the **clk_qactive_i[N]** inputs, where these signals are used internally depending on the value of `CLK_Q_CH_SYNC`. **Note** The path from a device to the controller **QACTIVE** is combinatorial, irrespective of whether synchronizers are present.

2.6 About the PPU

The PPU takes a software-programmed power domain policy and then controls the low-level hardware control signals. The PPU enables re-usability by separating technology-specific functionality from common device and software interfacing.

The PPU has the following interfaces:

Software interface

For high-level policy control and configuration. See *Chapter 3 Programmers model* on page 3-44 for more information.

Clock control interface

For high-level clock control.

Device control interface

For low-level device control and for ensuring device quiescence. The interfaces are:

- The device interface, that consists of one or more *Low Power Interface*s (LPIs).
- The device control interface, that includes clock enables, resets, and isolation control.

***Power Control State Machine* (PCSM) interface**

For controlling low-level technology-specific power switch and retention controls.

The following figure shows the PPU interfaces.

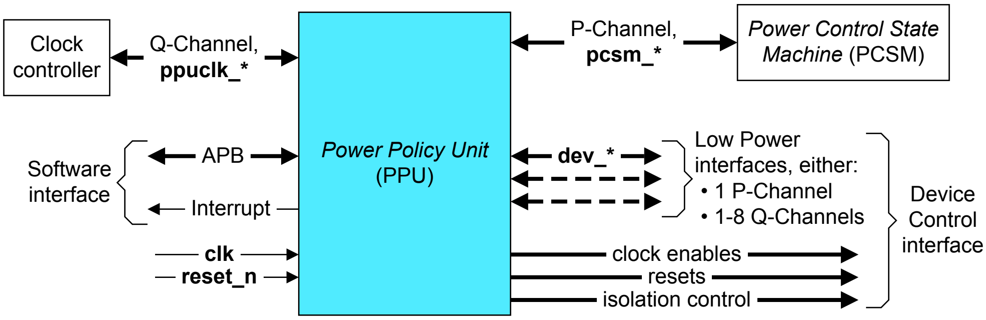

Figure 2-9 PPU interfaces

The PPU provides technology-independent hardware and software interfaces for controlling domain power modes in co-ordination with device quiescence. The device interface uses either a single P-Channel or one or more Q-Channels. A PPU that uses one or more Q-Channels as the device interface is called a Q-Channel PPU. A PPU that uses a P-Channel as the device interface is called a P-Channel PPU. See the *AMBA® Low Power Interface Specification, Arm® Q-Channel and P-Channel Interfaces* for more information about the LPI.

The *Power Control State Machine* (PCSM) is a technology-dependent state machine for the sequencing of power switch chains and retention controls, that can include RAM and register retention. The PCSM executes power mode changes under PPU direction. The interface between the PPU and the PCSM is a P-Channel.

The following figure shows a high-level illustration of how the PPU and PCSM controls connect to each other, and to a power-gated domain. The dotted lines indicate the implementation-dependent components and signal connections.

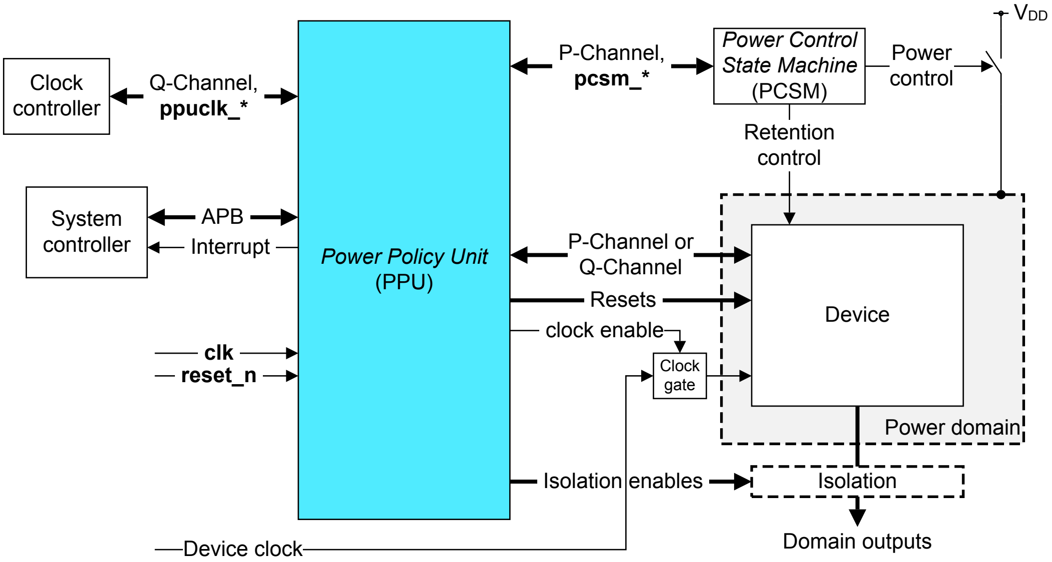

Figure 2-10 Example PPU connections to a power-gated domain

The PPU is a configurable component that can support different power domain scenarios. Many PPU features are optional or configurable. Software can read the PPU Identification Registers to discover which features a PPU supports.

PPU operation

The PPU uses power modes, such as on (ON), off (OFF), and full retention (FULL_RET), to represent the various power conditions of a domain. It has extensive support to reflect the various combinations of logic and memory power states into which a domain can be set.

Software can use these modes as either:

Static policy	A request to enter a mode directly.
Dynamic policy	Sets the minimum mode, so the PPU can autonomously change mode above this minimum, depending on the hardware inputs.

A P-Channel PPU also supports operating modes, which are configurations of the power modes. The meaning of each operating mode is specific to one or more components within the domain.

See the *Arm® Power Policy Unit Architecture Specification, version 1.1* for more information about the PPU.

2.6.1 PPU configuration parameters

There are multiple configuration parameters that configure or modify the functionality of the Power Policy Unit.

Table 2-6 PPU configuration parameters

Parameter	Permitted values	Usage constraints	Description
`DEV_PREQ_DLY`	0-3	Ignored if `DEVCHAN_CFG` > 0.	Sets the delay between **dev_pstate_o** and **dev_preq_o**, in PPU **clk** cycles.
`PCSM_PREQ_DLY`	0-3	-	Sets the delay between **pcsm_pstate_o** and **pcsm_preq_o**, in PPU **clk** cycles.

Table 2-6 PPU configuration parameters (continued)

Parameter	Permitted values	Usage constraints	Description
`ISO_CLKEN_DLY_CFG`	0-255	-	Sets the default value of the software-programmable PPU_DCDR0.ISO_CLKEN_DLY register field, in PPU **clk** cycles. The ISO_CLKEN_DLY value controls the delay between the deassertion of an isolation enable signal and the assertion of a clock enable signal.
`CLKEN_RST_DLY_CFG`	0-255	-	Sets the default value of the software-programmable PPU_DCDR0.CLKEN_RST_DLY register field, in PPU **clk** cycles. The CLKEN_RST_DLY value controls the delay between the assertion of a clock enable signal and the deassertion of a device reset signal.
`RST_HWSTAT_DLY_CFG`	0-255	-	Sets the default value of the software-programmable PPU_DCDR0.RST_HWSTAT_DLY register field, in PPU **clk** cycles. The RST_HWSTAT_DLY value controls the delay between the deassertion of a device reset signal and a transition of the **ppuhwstat_o[20:0]** signal.
`CLKEN_ISO_DLY_CFG`	0-255	-	Sets the default value of the software-programmable PPU_DCDR1.CLKEN_ISO_DLY register field, in PPU **clk** cycles. The CLKEN_ISO_DLY value controls the delay between the deassertion of a clock enable signal and the assertion of an isolation enable signal.
`ISO_RST_DLY_CFG`	0-255	-	Sets the default value of the software-programmable PPU_DCDR1.ISO_RST_DLY register field, in PPU **clk** cycles. The ISO_RST_DLY value controls the delay between the assertion of an isolation enable signal and the assertion of a device reset signal.
`DEVCHAN_CFG`	0-8	-	Number and type of device interface LPI: 0 = A single P-Channel. 1-8 = A Q-Channel PPU with `DEVCHAN_CFG` Q-Channel interfaces.
`DEF_PWR_POLICY`	`0b0000`, `0b1000`	-	Default value of the PPU_PWPR.PWR_POLICY register field.

Table 2-6 PPU configuration parameters (continued)

Parameter	Permitted values	Usage constraints	Description
`DEF_PWR_DYN_EN`	0, 1	`DEF_PWR_POLICY` mode must support dynamic.	Default value of the PPU_PWPR.PWR_DYN_EN register bit.
`DEF_OP_POLICY`	0-`NUM_OPMODE_CFG`	Ignored if `DEVCHAN_CFG` > 0 or `NUM_OPMODE_CFG` == 0.	Default value of the PPU_PWPR.OP_POLICY register field.
`DEF_OP_DYN_EN`	0, 1	Ignored if `DEVCHAN_CFG` > 0 or `NUM_OPMODE_CFG` == 0.	Default value of the PPU_PWPR.OP_DYN_EN register bit.
`FUNC_RET_RAM_REG_CFG`	0, 1	Must be 0 when `STA_FUNC_RET_SPT_CFG` == 0.	Enable FUNC_RET RAM retention configuration register, PPU_FUNRR.
`FULL_RET_RAM_REG_CFG`	0, 1	Must be 0 when `STA_FULL_RET_SPT_CFG` == 0.	Enable FULL_RET RAM retention configuration register, PPU_FULRR.
`MEM_RET_RAM_REG_CFG`	0, 1	Must be 0 when `STA_MEM_RET_SPT_CFG` == 0.	Enable MEM_RET RAM retention configuration register, PPU_MEMRR.
`WARM_RST_DEVREQEN_CFG`	0, 1	-	Default value of the PPU_PTCR.WARM_RST_DEVREQEN register bit.
`DBG_RECOV_PORST_CFG`	0, 1	Ignored if `STA_DBG_RECOV_SPT_CFG` == 0.	Default value of the PPU_PTCR.DBG_RECOV_PORST_EN register bit.
`NUM_OPMODE_CFG`	0-15	Ignored if `DEVCHAN_CFG` > 0.	Number of operating modes.
`OP_ACTIVE_CFG`	0, 1	Ignored if `NUM_OPMODE_CFG` == 0, otherwise: When `OP_ACTIVE_CFG` == 0, then `NUM_OPMODE_CFG` must be between 1 and 8. When `OP_ACTIVE_CFG` == 1, then `NUM_OPMODE_CFG` must be 1, 3, 7, or 15.	Operating mode active configuration. Controls how the PPU responds to **PACTIVE[31:16]**: 0 = Ladder use model. 1 = Independent use model. See the *Arm® Power Policy Unit Architecture Specification, version 1.1* for more information.
`STA_POLICY_OP_IRQ_CFG`	0, 1	Must be 0 when `NUM_OPMODE_CFG` == 0.	Enable static operating policy transition completion.
`STA_POLICY_PWR_IRQ_CFG`	0, 1	Must be 0 when `NUM_OPMODE_CFG` == 0.	Enable static power policy transition completion.
`LOCK_CFG`	0, 1	Must be 0 when `DYN_OFF_SPT_CFG` == 0 && `DYN_OFF_EMU_SPT_CFG` == 0 && `DYN_MEM_RET_SPT_CFG` == 0 && `DYN_MEM_RET_EMU_SPT_CFG` == 0.	Enable Lock support.
`SW_DEV_DEL_CFG`	0, 1	-	Enables software to write to the PPU_DCDR0 and PPU_DCDR1 registers and alter the device control delay parameters.

Table 2-6 PPU configuration parameters (continued)

Parameter	Permitted values	Usage constraints	Description
`PWR_MODE_ENTRY_DEL_CFG`	0, 1	-	Enables software to write to the PPU_EDTR0 and PPU_EDTR1 registers and alter the power mode entry delay parameters.
`STA_OFF_EMU_SPT_CFG`	0, 1	-	Enable static OFF_EMU.
`STA_MEM_RET_SPT_CFG`	0, 1	-	Enable static MEM_RET.
`STA_MEM_RET_EMU_SPT_CFG`	0, 1	Must be 0 when `STA_MEM_RET_SPT_CFG` == 0.	Enable static MEM_RET_EMU.
`STA_LGC_RET_SPT_CFG`	0, 1	Must be 0 when the following expression is false: `DEVCHAN_CFG` == 0 && `STA_MEM_OFF_SPT_CFG` == 1.	Enable static LOGIC_RET.
`STA_FULL_RET_SPT_CFG`	0, 1	-	Enable static FULL_RET.
`STA_MEM_OFF_SPT_CFG`	0, 1	-	Enable static MEM_OFF.
`STA_FUNC_RET_SPT_CFG`	0, 1	-	Enable static FUNC_RET.
`STA_DBG_RECOV_SPT_CFG`	0, 1	Must be 0 when `DEVCHAN_CFG` > 0.	Enable static DBG_RECOV.
`DYN_OFF_SPT_CFG`	0, 1	Must be 0 when `DYN_ON_SPT_CFG` == 0.	Enable dynamic OFF.
`DYN_OFF_EMU_SPT_CFG`	0, 1	Must be 0 when the following expression is false: `DYN_ON_SPT_CFG` == 1 && `STA_OFF_EMU_SPT_CFG` == 1 && `DYN_OFF_SPT_CFG` == 1.	Enable dynamic OFF_EMU.
`DYN_MEM_RET_SPT_CFG`	0, 1	Must be 0 when the following expression is false: `DYN_ON_SPT_CFG` == 1 && `STA_MEM_RET_SPT_CFG` == 1. Must be 1 when the following expression is true: `DEF_PWR_DYN_EN` == 1 && `OFF_MEM_RET_TRANS_CFG` == 1.	Enable dynamic MEM_RET.

Table 2-6 PPU configuration parameters (continued)

Parameter	Permitted values	Usage constraints	Description
`DYN_MEM_RET_EMU_SPT_CFG`	0, 1	Must be 0 when the following expression is false: `DYN_ON_SPT_CFG == 1` && `STA_MEM_RET_EMU_SPT_CFG` == 1 && `DYN_MEM_RET_SPT_CFG` == 1 && `STA_MEM_RET_SPT_CFG` == 1.	Enable dynamic MEM_RET_EMU.
`DYN_LGC_RET_SPT_CFG`	0, 1	Must be 0 when the following expression is false: `DYN_ON_SPT_CFG` == 1 && `STA_LGC_RET_SPT_CFG` == 1 && `DYN_MEM_OFF_SPT_CFG` == 1 && `STA_MEM_OFF_SPT_CFG` == 1 && `DEVCHAN_CFG` == 0.	Enable dynamic LOGIC_RET.
`DYN_FULL_RET_SPT_CFG`	0, 1	Must be 0 when the following expression is false: `DYN_ON_SPT_CFG` == 1 && `STA_FULL_RET_SPT_CFG` == 1.	Enable dynamic FULL_RET.
`DYN_MEM_OFF_SPT_CFG`	0, 1	Must be 0 when the following expression is false: `DYN_ON_SPT_CFG` == 1 && `STA_MEM_OFF_SPT_CFG` == 1.	Enable dynamic MEM_OFF.
`DYN_FUNC_RET_SPT_CFG`	0, 1	Must be 0 when the following expression is false: `DYN_ON_SPT_CFG` == 1 && `STA_FUNC_RET_SPT_CFG` == 1.	Enable dynamic FUNC_RET.
`DYN_ON_SPT_CFG`	0, 1	-	Enable dynamic ON.
`DYN_WRM_RST_SPT_CFG`	0, 1	Must be 0 when the following expression is false: `DYN_ON_SPT_CFG` == 1 && `DEVCHAN_CFG` == 0.	Enable dynamic WARM_RST.
`DEV_SYNC_EN`	0, 1	-	If set to 1, adds synchronizers to either: • **dev_paccept_i** and **dev_pdeny_i** when `DEVCHAN_CFG` == 0. • **dev_qacceptn_i[DEVCHAN_CFG − 1:0]** and **dev_qdeny_i[DEVCHAN_CFG − 1:0]** when `DEVCHAN_CFG` > 0.

Table 2-6 PPU configuration parameters (continued)

Parameter	Permitted values	Usage constraints	Description
`DEV_ACTIVE_SYNC_EN`	0, 1	-	If set to 1, adds synchronizers to either: • **dev_pactive_i** when `DEVCHAN_CFG` == 0. • **dev_qactive_i[DEVCHAN_CFG – 1:0]** when `DEVCHAN_CFG` > 0.
`PCSM_SYNC_EN`	0, 1	-	Add synchronizer on **pcsm_paccept_i**.
`QCLK_SYNC_EN`	0, 1	-	Add synchronizer on **ppuclk_qreqn_i**.
`OFF_MEM_RET_TRANS_CFG`	0, 1	Must be 0 when `STA_MEM_RET_SPT_CFG` == 0.	Enable direct transitions from OFF to MEM_RET.
`PCSM_OFF_INIT`	0, 1	-	Enables a PCSM initialization handshake when the default policy is OFF.
`OPMODE_PCSM_SPT_CFG`	0, 1	Ignored if `NUM_OPMODE_CFG` == 0.	Enables OPMODE bits on the PCSM and PCSM handshakes on OPMODE only transitions.
`UARCH`	0-2	Value of 2 is only allowed if `PWR_MODE_ENTRY_DEL_CFG` == 1 or `NUM_OPMODE_CFG` > 0.	Defines the microarchitecture of the design: 0 = Minimum area. 1 = Balance. 2 = Performance.

Chapter 3
Programmers model

This chapter describes the memory regions and registers that the *Power Policy Unit* (PPU) provides.

Note

The CLK-CTRL, LPC-Q, LPD-P, LPD-Q, and P2Q have no registers that software can program.

It contains the following sections:

3.1 About the programmers model

The following information applies to all registers:

- Do not attempt to access reserved or unused address locations. Attempting to access these locations can result in unpredictable behavior.
- Unless otherwise stated in the accompanying text:
  - Do not modify undefined register bits.
  - Ignore undefined register bits on reads.
  - Unless otherwise specified, all register bits are reset to a logic 0 by a system or power up reset.
- The following describes the access type:

RW	Read and write.
RO	Read-only.
WO	Write-only.

3.2 Register summary

The *Power Policy Unit* (PPU) registers occupy a 4KB region.

The following table shows the PPU registers in offset order from the base memory address. See the *Arm® Power Policy Unit Architecture Specification, version 1.1* for information about the registers that this document does not describe.

Table 3-1 PPU register summary

Offset	Name	Type	Width	Description
`0x000`	PPU_PWPR	RW	32	Power Policy Register.
`0x004`	PPU_PMER	RW	32	Power Mode Emulation Register.
`0x008`	PPU_PWSR	RO	32	Power Status Register.
`0x00C`	-	-	-	Reserved.
`0x010`	PPU_DISR	RO	32	Device Interface Input Current Status Register.
`0x014`	PPU_MISR	RO	32	Miscellaneous Input Current Status Register.
`0x018`	PPU_STSR	RO	32	Stored Status Register.
`0x01C`	PPU_UNLK	RW	32	Unlock register.
`0x020`	PPU_PWCR	RW	32	Power Configuration Register.
`0x024`	PPU_PTCR	RW	32	Power Mode Transition Configuration Register.
`0x028-0x02C`	-	-	-	Reserved.
`0x030`	PPU_IMR	RW	32	Interrupt Mask Register.
`0x034`	PPU_AIMR	RW	32	Additional Interrupt Mask Register.
`0x038`	PPU_ISR	RW	32	Interrupt Status Register.
`0x03C`	PPU_AISR	RW	32	Additional Interrupt Status Register.
`0x040`	PPU_IESR	RW	32	Input Edge Sensitivity Register.
`0x044`	PPU_OPSR	RW	32	Operating Mode Active Edge Sensitivity Register.
`0x048-0x04C`	-	-	-	Reserved.
`0x050`	PPU_FUNRR	RW	32	Functional Retention RAM Configuration Register.
`0x054`	PPU_FULRR	RW	32	Full Retention RAM Configuration Register.
`0x058`	PPU_MEMRR	RW	32	Memory Retention RAM Configuration Register.
`0x05C-0x15C`	-	-	-	Reserved.
`0x160`	PPU_EDTR0	RW	32	Power Mode Entry Delay Register 0.
`0x164`	PPU_EDTR1	RW	32	Power Mode Entry Delay Register 1.
`0x168-0x016C`	-	-	-	Reserved.
`0x170`	PPU_DCDR0	RW	32	Device Control Delay Configuration Register 0.
`0x174`	PPU_DCDR1	RW	32	Device Control Delay Configuration Register 1.
`0x178-0xFAC`	-	-	-	Reserved.
`0xFB0`	PPU_IDR0	RO	32	PPU Identification Register 0.

Table 3-1 PPU register summary (continued)

Offset	Name	Type	Width	Description
`0xFB4`	PPU_IDR1	RO	32	PPU Identification Register 1.
`0xFB8-0xFC4`	-	-	-	Reserved.
`0xFC8`	PPU_IIDR	RO	32	*3.3 Implementation Identification Register, PPU_IIDR* on page 3-48.
`0xFCC`	PPU_AIDR	RO	32	Architecture Identification Register.
`0xFD0`	PID4	RO	32	*3.4.1 Peripheral ID 4* on page 3-49.
`0xFD4`	PID5	RO	32	*3.4.2 Peripheral ID 5* on page 3-49.
`0xFD8`	PID6	RO	32	*3.4.3 Peripheral ID 6* on page 3-50.
`0xFDC`	PID7	RO	32	*3.4.4 Peripheral ID 7* on page 3-50.
`0xFE0`	PID0	RO	32	*3.4.5 Peripheral ID 0* on page 3-51.
`0xFE4`	PID1	RO	32	*3.4.6 Peripheral ID 1* on page 3-51.
`0xFE8`	PID2	RO	32	*3.4.7 Peripheral ID 2* on page 3-52.
`0xFEC`	PID3	RO	32	*3.4.8 Peripheral ID 3* on page 3-53.
`0xFF0`	ID0	RO	32	*3.4.9 Component ID 0* on page 3-53.
`0xFF4`	ID1	RO	32	*3.4.10 Component ID 1* on page 3-54.
`0xFF8`	ID2	RO	32	*3.4.11 Component ID 2* on page 3-54.
`0xFFC`	ID3	RO	32	*3.4.12 Component ID 3* on page 3-54.

3.3 Implementation Identification Register, PPU_IIDR

The PPU_IIDR register provides information about the implementer and implementation of the PPU.

Usage constraints	There are no usage constraints.
Configurations	Available in all configurations.
Attributes	See *3.2 Register summary* on page 3-46.

The following figure shows the bit assignments.

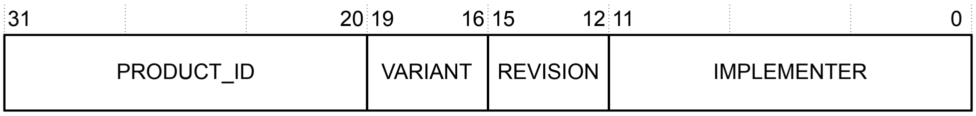

Figure 3-1 PPU_IIDR bit assignments

The following table shows the bit assignments.

Table 3-2 PPU_IIDR bit assignments

Bits	Name	Default	Description
[31:20]	PRODUCT_ID	`0x0B6`	Identifies the PPU component.
[19:16]	VARIANT	`0x1`	Returns the PID2.REV field value.
[15:12]	REVISION	`0x0`	Returns the PID3.REVAND field value.
[11:0]	IMPLEMENTER	`0x43B`	Implementer identification. **[11:8]** The JEP106 continuation code of the implementer. **[7]** Always 0. **[6:0]** The JEP106 identity code of the implementer. For an Arm implementation, bits[11:0] are `0x43B`.

Related reference

3.4.7 Peripheral ID 2 on page 3-52

3.4.8 Peripheral ID 3 on page 3-53

3.4 Implementation-defined identification registers

The PPU has some ID registers that are at the end of the 4KB memory region. Software can use these registers to discover which components are present in an SoC.

This section contains the following subsections:

3.4.1 Peripheral ID 4

Peripheral ID 4 Register.

Usage constraints	There are no usage constraints.
Configurations	Available in all configurations.
Attributes	See *3.2 Register summary* on page 3-46.

The following figure shows the bit assignments.

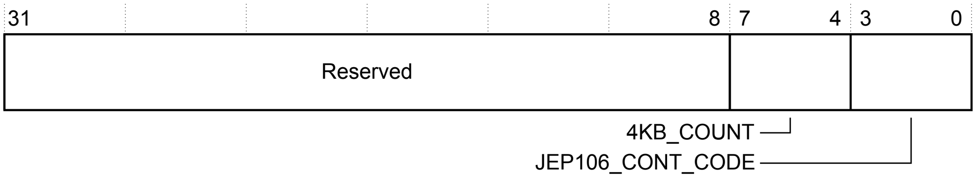

Figure 3-2 Peripheral ID 4 Register bit assignments

The following table shows the bit assignments.

Table 3-3 Peripheral ID 4 Register bit assignments

Bits	Name	Default	Function
[31:8]	-	`0x0`	Reserved.
[7:4]	4KB_COUNT	`0x0`	Indicates that the PPU registers occupy a single 4KB page.
[3:0]	JEP106_CONT_CODE	`0x4`	Indicates how many Continuation Codes (`0x7F`) an Arm device requires. For identifying an Arm device or product, the *Standard Manufacturer's Identification Code* specifies a requirement of four Continuation Codes.

3.4.2 Peripheral ID 5

Peripheral ID 5 Register.

Usage constraints	There are no usage constraints.
Configurations	Available in all configurations.

Attributes	See *3.2 Register summary* on page 3-46.

The following figure shows the bit assignments.

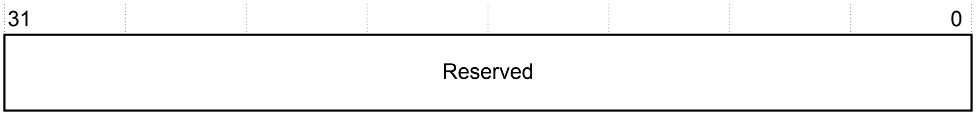

Figure 3-3 Peripheral ID 5 Register bit assignments

The following table shows the bit assignments.

Table 3-4 Peripheral ID 5 Register bit assignments

Bits	Name	Default	Function
[31:0]	-	`0x0`	Reserved.

3.4.3 Peripheral ID 6

Peripheral ID 6 Register.

Usage constraints	There are no usage constraints.
Configurations	Available in all configurations.
Attributes	See *3.2 Register summary* on page 3-46.

The following figure shows the bit assignments.

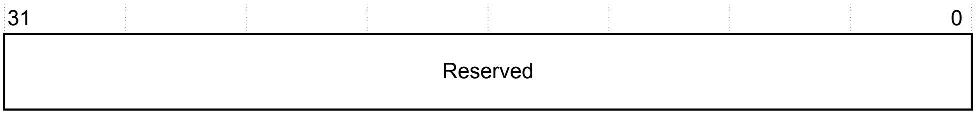

Figure 3-4 Peripheral ID 6 Register bit assignments

The following table shows the bit assignments.

Table 3-5 Peripheral ID 6 Register bit assignments

Bits	Name	Default	Function
[31:0]	-	`0x0`	Reserved.

3.4.4 Peripheral ID 7

Peripheral ID 7 Register.

Usage constraints	There are no usage constraints.
Configurations	Available in all configurations.
Attributes	See *3.2 Register summary* on page 3-46.

The following figure shows the bit assignments.

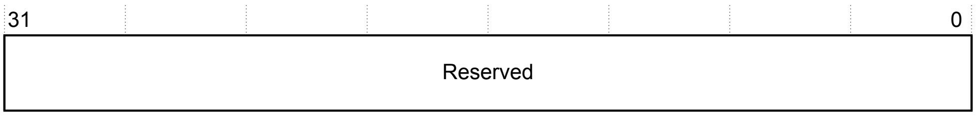

Figure 3-5 Peripheral ID 7 Register bit assignments

The following table shows the bit assignments.

Table 3-6 Peripheral ID 7 Register bit assignments

Bits	Name	Default	Function
[31:0]	-	`0x0`	Reserved.

3.4.5 Peripheral ID 0

Peripheral ID 0 Register.

Usage constraints There are no usage constraints.

Configurations Available in all configurations.

Attributes See *3.2 Register summary* on page 3-46.

The following figure shows the bit assignments.

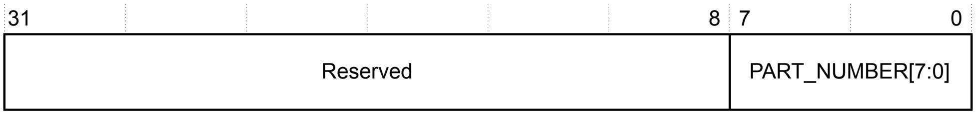

Figure 3-6 Peripheral ID 0 Register bit assignments

The following table shows the bit assignments.

Table 3-7 Peripheral ID 0 Register bit assignments

Bits	Name	Default	Function
[31:8]	-	`0x0`	Reserved.
[7:0]	PART_NUMBER[7:0]	`0xB6`	Part number for the PCK-600 PPU. See also PID1.PART_NUMBER[11:8].

3.4.6 Peripheral ID 1

Peripheral ID 1 Register.

Usage constraints There are no usage constraints.

Configurations Available in all configurations.

Attributes See *3.2 Register summary* on page 3-46.

The following figure shows the bit assignments.

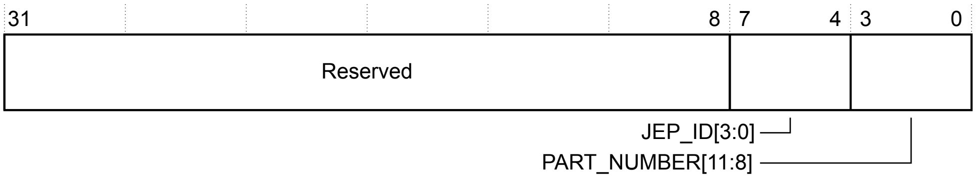

Figure 3-7 Peripheral ID 1 Register bit assignments

The following table shows the bit assignments.

Table 3-8 Peripheral ID 1 Register bit assignments

Bits	Name	Default	Function
[31:8]	-	`0x0`	Reserved.
[7:4]	JEP_ID[3:0]	`0xB`	JEDEC JEP106 ID code. See also PID2.JEP_ID[6:4] and the *Standard Manufacturer's Identification Code*.
[3:0]	PART_NUMBER[11:8]	`0x0`	Part number for the PCK-600 PPU. See also PID0.PART_NUMBER[7:0].

3.4.7 Peripheral ID 2

Peripheral ID 2 Register.

Usage constraints	There are no usage constraints.
Configurations	Available in all configurations.
Attributes	See *3.2 Register summary* on page 3-46.

The following figure shows the bit assignments.

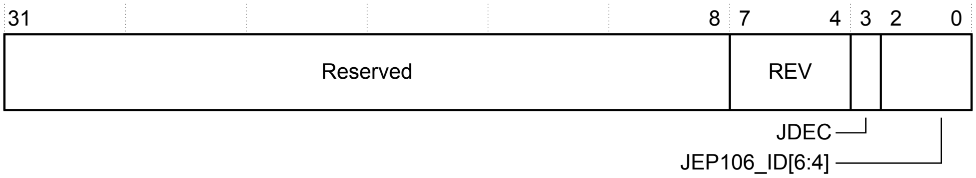

Figure 3-8 Peripheral ID 2 Register bit assignments

The following table shows the bit assignments.

Table 3-9 Peripheral ID 2 Register bit assignments

Bits	Name	Default	Function
[31:8]	-	`0x0`	Reserved.
[7:4]	REV	`0x1`	Revision identifier for the PCK-600 PPU: • `0x0` = r0p0. • `0x1` = r0p1, r0p2.
[3]	JDEC	`0x1`	Indicates the use of a JEDEC-assigned ID value.
[2:0]	JEP106_ID[6:4]	`0x3`	JEDEC JEP106 ID code. See also PID1.JEP_ID[3:0] and the *Standard Manufacturer's Identification Code*.

3.4.8 Peripheral ID 3

Peripheral ID 3 Register.

Usage constraints There are no usage constraints.

Configurations Available in all configurations.

Attributes See *3.2 Register summary* on page 3-46.

The following figure shows the bit assignments.

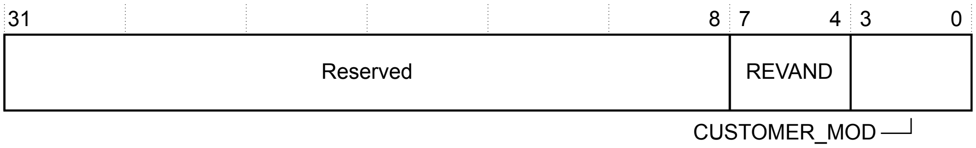

Figure 3-9 Peripheral ID 3 Register bit assignments

The following table shows the bit assignments.

Table 3-10 Peripheral ID 3 Register bit assignments

Bits	Name	Default	Function
[31:8]	-	`0x0`	Reserved.
[7:4]	REVAND	`0x0`	Returns the value of the **ecorevnum_i[3:0]** signal.
[3:0]	CUSTOMER_MOD	`0x0`	Customer modification.

Related reference

A.6 PPU signals on page Appx-A-67

3.4.9 Component ID 0

Component ID 0 Register.

Usage constraints There are no usage constraints.

Configurations Available in all configurations.

Attributes See *3.2 Register summary* on page 3-46.

The following figure shows the bit assignments.

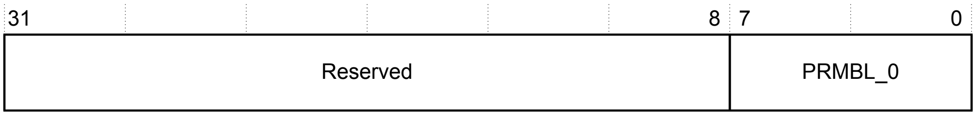

Figure 3-10 Component ID 0 Register bit assignments

The following table shows the bit assignments.

Table 3-11 Component ID 0 Register bit assignments

Bits	Name	Default	Function
[31:8]	-	`0x0`	Reserved.
[7:0]	PRMBL_0	`0x0D`	Preamble 0.

3.4.10 Component ID 1

Component ID 1 Register.

Usage constraints There are no usage constraints.

Configurations Available in all configurations.

Attributes See *3.2 Register summary* on page 3-46.

The following figure shows the bit assignments.

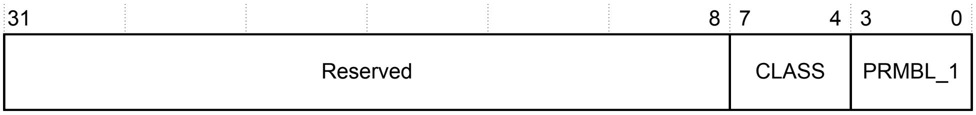

Figure 3-11 Component ID 1 Register bit assignments

The following table shows the bit assignments.

Table 3-12 Component ID 1 Register bit assignments

Bits	**Name**	**Default**	**Function**
[31:8]	-	`0x0`	Reserved.
[7:4]	CLASS	`0xF`	Indicates the component class that this component belongs to.
[3:0]	PRMBL_1	`0x0`	Preamble 1.

3.4.11 Component ID 2

Component ID 2 Register.

Usage constraints There are no usage constraints.

Configurations Available in all configurations.

Attributes See *3.2 Register summary* on page 3-46.

The following figure shows the bit assignments.

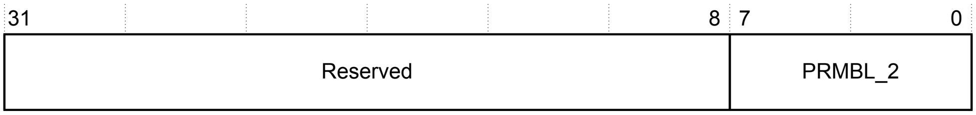

Figure 3-12 Component ID 2 Register bit assignments

The following table shows the bit assignments.

Table 3-13 Component ID 2 Register bit assignments

Bits	**Name**	**Default**	**Function**
[31:8]	-	`0x0`	Reserved.
[7:0]	PRMBL_2	`0x05`	Preamble 2.

3.4.12 Component ID 3

Component ID 3 Register.

Usage constraints There are no usage constraints.

Configurations Available in all configurations.

Attributes See *3.2 Register summary* on page 3-46.

The following figure shows the bit assignments.

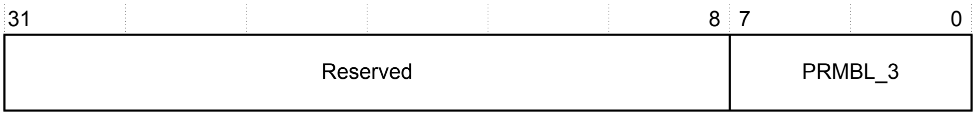

Figure 3-13 Component ID 3 Register bit assignments

The following table shows the bit assignments.

Table 3-14 Component ID 3 Register bit assignments

Bits	Name	Default	Function
[31:8]	-	`0x0`	Reserved.
[7:0]	PRMBL_3	`0xB1`	Preamble 3.

Appendix A

Signal Descriptions

This appendix describes the interface signals that are present for each PCK-600 component.

It contains the following sections:

A.1 LPD-Q Q-Channel Distributor signals

The following tables show the Low Power Distributor Q-Channel signals.

The following table lists the clock and reset signals.

Table A-1 LPD-Q clock and reset signals

Signal	Type	Clock	Description
clk	Input	-	Clock input.
reset_n	Input	**clk**	An active-LOW reset input. **reset_n** can assert asynchronously, but must deassert synchronous to **clk**.

The following table lists the control interface signals.

Table A-2 LPD-Q control interface signals

Signal	Type	Clock	Description
ctrl_qreqn_i	Input	Either: • Asynchronous, when `CTRL_Q_CH_SYNC` == 1. • **clk**, when `CTRL_Q_CH_SYNC` == 0.	This signal indicates when the controller issues a quiescence entry or exit request to the LPD-Q.
ctrl_qacceptn_o	Output	**clk**	This signal indicates when all the LPD-Q devices accept the quiescence request.
ctrl_qdeny_o			This signal indicates when one or more LPD-Q devices deny the quiescence request.
ctrl_qactive_o		Asynchronous	This signal indicates when one or more LPD-Q devices are active or they are requesting to exit from quiescence.

The following table lists the device interface signals.

Table A-3 LPD-Q device interface signals

Signal	Type	Clock	Description
dev_qreqn_o[NUM_QCHL − 1:0]	Output	**clk**	This signal indicates when the LPD-Q issues a quiescence entry or exit request to the LPD-Q devices.
dev_qacceptn_i[NUM_QCHL − 1:0]	Input	Either: • Asynchronous, when `DEV_Q_CH_SYNC` == 1. • **clk**, when `DEV_Q_CH_SYNC` == 0.	This signal indicates when an LPD-Q device accepts the quiescence request.
dev_qdeny_i[NUM_QCHL − 1:0]			This signal indicates when an LPD-Q device denies the quiescence request.
dev_qactive_i[NUM_QCHL − 1:0]		Asynchronous	This signal indicates when an LPD-Q device is active or it is requesting to exit from quiescence.

The following table lists the clock active signal.

Table A-4 LPD-Q clock active signal

Signal	Type	Clock	Description
clk_qactive_o	Output	Asynchronous	This signal indicates when the LPD-Q is active and that it requires the **clk** signal: 0 = The LPD-Q does not require a clock signal. 1 = The LPD-Q requires a clock signal.

The following table lists the *Design for Test* (DFT) signals.

Table A-5 LPD-Q DFT signals

Signal	Type	Clock	Description
dftcgen	Input	**clk**	This signal enables the architectural clock gates, and ensures that the **clk** logic is active during DFT shift mode.

A.2 LPD-P P-Channel Distributor signals

The following tables show the Low Power Distributor P-Channel signals.

The following table lists the clock and reset signals.

Table A-6 LPD-P clock and reset signals

Signal	Type	Clock	Description
clk	Input	-	Clock input.
reset_n	Input	**clk**	An active-LOW reset input. **reset_n** can assert asynchronously, but must deassert synchronous to **clk**.

The following table lists the control interface signals.

Table A-7 LPD-P control interface signals

Signal	Type	Clock	Description
ctrl_preq_i	Input	Either: • Asynchronous, when `CTRL_P_CH_SYNC` == 1. • **clk**, when `CTRL_P_CH_SYNC` == 0.	This signal indicates when the controller issues a power mode request to the LPD-P.
ctrl_pstate_i[P_CH_PSTATE_LEN−1:0]			The power mode and operating mode that is requested when **ctrl_preq_i** is set HIGH.
ctrl_paccept_o	Output	**clk**	This signal indicates when all the LPD-P devices accept the power mode request. At reset, this signal is LOW.
ctrl_pdeny_o			This signal indicates when one or more LPD-P devices deny the power mode request. At reset, this signal is LOW.
ctrl_pactive_o[P_CH_PACTIVE_LEN−1:0]		Asynchronous	This signal indicates the combined power mode that the device **dev<X>_pactive_i[P_CH_PACTIVE_LEN−1:0]** signals request.

The following table lists the device interface signals.

Table A-8 LPD-P device interface signals, where `X` is the number of the interface

Signal, where <X> == 1-`DEV_P_CH_NUM`	Type	Clock	Description
dev<X>_preq_o	Output	**clk**	This signal indicates when the LPD-P issues a power mode request to the LPD-P devices. At reset, this signal is LOW.
dev<X>_pstate_o[P_CH_PSTATE_LEN−1:0]			The power mode and operating mode that is sent to device <X> when **dev<X>_preq_o** is set HIGH. At reset, **dev<X>_pstate_o** are set LOW.

Table A-8 LPD-P device interface signals, where X is the number of the interface (continued)

Signal, where <X> == 1-DEV_P_CH_NUM	Type	Clock	Description
dev<X>_paccept_i	Input	Either: - Asynchronous, when `DEV_P_CH_SYNC` == 1. - **clk**, when `DEV_P_CH_SYNC` == 0.	This signal indicates when an LPD-P device accepts the power mode request.
dev<X>_pdeny_i			This signal indicates when an LPD-P device denies the power mode request.
dev<X>_pactive_i[P_CH_PACTIVE_LEN −1:0]		Asynchronous	This signal indicates the power mode that device <X> requests.

The following table lists the clock active signal.

Table A-9 LPD-P clock active signal

Signal	Type	Clock	Description
clk_qactive_o	Output	Asynchronous	This signal indicates when the LPD-P is active and that it requires the **clk** signal: 0 = The LPD-P does not require a clock signal. 1 = The LPD-P requires a clock signal.

The following table lists the *Design for Test* (DFT) signals.

Table A-10 LPD-P DFT signals

Signal	Type	Clock	Description
dftcgen	Input	**clk**	This signal enables the architectural clock gates, and ensures that the **clk** logic is active during DFT shift mode.

Related reference

2.2.6 LPD-P configuration parameters on page 2-26

A.3 LPC-Q Q-Channel Combiner signals

The following tables show the Low Power Combiner Q-Channel signals.

The following table lists the clock and reset signals.

Table A-11 LPC-Q clock and reset signals

Signal	Type	Clock	Description
clk	Input	-	Clock input.
reset_n	Input	**clk**	An active-LOW reset input. **reset_n** can assert asynchronously, but must deassert synchronous to **clk**.

The following table lists the control interface signals.

Table A-12 LPC-Q control interface signals

Signal	Type	Clock	Description
ctrl_qreqn_i[NUM_CTRL_Q_CHL−1:0]	Input	Either: • Asynchronous, when `CTRL_Q_CH_SYNC` == 1. • **clk**, when `CTRL_Q_CH_SYNC` == 0.	This signal indicates when a controller issues a quiescence entry or exit request to the LPC-Q.
ctrl_qacceptn_o[NUM_CTRL_Q_CHL −1:0]	Output	**clk**	This signal indicates when the LPC-Q accepts the quiescence request.
ctrl_qdeny_o[NUM_CTRL_Q_CHL−1:0]			This signal indicates when the LPC-Q denies the quiescence request.
ctrl_qactive_o[NUM_CTRL_Q_CHL−1:0]		Asynchronous	This signal indicates when one or more LPC-Q devices are active or they are requesting to exit from quiescence.

The following table lists the device interface signals.

Table A-13 LPC-Q device interface signals

Signal	Type	Clock	Description
dev_qreqn_o[NUM_DEV_Q_CHL−1:0]	Output	**clk**	This signal indicates when the LPC-Q issues a quiescence entry or exit request to the LPC-Q devices.
dev_qacceptn_i[[NUM_DEV_Q_CHL −1:0]	Input	Either: • Asynchronous, when `DEV_Q_CH_SYNC` == 1. • **clk**, when `DEV_Q_CH_SYNC` == 0.	This signal indicates when an LPC-Q device accepts the quiescence request.
dev_qdeny_i[NUM_DEV_Q_CHL−1:0]			This signal indicates when an LPC-Q device denies the quiescence request.
dev_qactive_i[NUM_DEV_Q_CHL−1:0]		Asynchronous	This signal indicates when an LPC-Q device is active or it is requesting to exit from quiescence.

The following table lists the clock active signal.

Table A-14 LPC-Q clock active signal

Signal	Type	Clock	Description
clk_qactive_o	Output	Asynchronous	This signal indicates when the LPC-Q is active and that it requires the **clk** signal: 0 = The LPC-Q does not require a clock signal. 1 = The LPC-Q requires a clock signal.

The following table lists the *Design for Test* (DFT) signals.

Table A-15 LPC-Q DFT signals

Signal	Type	Clock	Description
dftcgen	Input	**clk**	This signal enables the architectural clock gates, and ensures that the **clk** logic is active during DFT shift mode.

A.4 P2Q Converter signals

The following tables show the P-Channel to Q-Channel Converter signals.

The following table lists the clock and reset signals.

Table A-16 P2Q clock and reset signals

Signal	Type	Clock	Description
clk	Input	-	Clock input.
reset_n	Input	**clk**	An active-LOW reset input. **reset_n** can assert asynchronously, but must deassert synchronous to **clk**.

The following table lists the control interface signals.

Table A-17 P2Q control interface signals

Signal	Type	Clock	Description
ctrl_preq_i	Input	Either: • Asynchronous, when `CTRL_P_CH_SYNC` == 1. • **clk**, when `CTRL_P_CH_SYNC` == 0.	This signal indicates when the controller issues a power mode request to the P2Q.
ctrl_pstate_i[7:0]			The power mode and operating mode that is requested when **ctrl_preq_i** is set HIGH.
ctrl_paccept_o	Output	**clk**	This signal indicates when the P2Q accepts the power mode request. At reset, this signal is LOW.
ctrl_pdeny_o			This signal indicates when the P2Q denies the power mode request. At reset, this signal is LOW.
ctrl_pactive_o[31:0]		Asynchronous	This signal indicates the power mode that the device requests. The power mode depends on the value of **dev_qactive_i** and the configuration of the `CTRL_P_CH_PACTIVE_MAP[31:0]` parameter.

The following table lists the device interface signals.

Table A-18 P2Q device interface signals

Signal	Type	Clock	Description
dev_qreqn_o	Output	**clk**	This signal indicates when the P2Q issues a quiescence entry or exit request to the Q-Channel device. At reset, this signal is LOW.
dev_qacceptn_i	Input	Either: • Asynchronous, when `DEV_Q_CH_SYNC` == 1. • **clk**, when `DEV_Q_CH_SYNC` == 0.	This signal indicates when the Q-Channel device accepts the quiescence request.
dev_qdeny_i			This signal indicates when the Q-Channel device denies the quiescence request.
dev_qactive_i		Asynchronous	This signal indicates when the Q-Channel device is active or it is requesting to exit from quiescence.

The following table lists the clock active signal.

Table A-19 P2Q clock active signal

Signal	Type	Clock	Description
clk_qactive_o	Output	Asynchronous	This signal indicates when the P2Q is active and that it requires the **clk** signal: 0 = The P2Q does not require a clock signal. 1 = The P2Q requires a clock signal.

The following table lists the *Design for Test* (DFT) signals.

Table A-20 P2Q DFT signals

Signal	Type	Clock	Description
dftcgen	Input	**clk**	This signal enables the architectural clock gates, and ensures that the **clk** logic is active during DFT shift mode.

A.5 CLK-CTRL signals

The following tables show the Clock Controller signals.

The following table lists the clock and reset signals.

Table A-21 CLK-CTRL clock and reset signals

Signal	Type	Clock	Description
clk	Input	-	Clock input.
reset_n	Input	**clk**	An active-LOW reset input. **reset_n** can assert asynchronously, but must deassert synchronous to **clk**.

The following table lists the hierarchical clock control signals.

Table A-22 CLK-CTRL hierarchical clock control signals

Signal	Type	Clock	Description
hc_qreqn_i	Input	Either: • Asynchronous, when `HC_Q_CH_SYNC` == 1. • **clk**, when `HC_Q_CH_SYNC` == 0.	This signal indicates when the controller issues a quiescence entry or exit request to the CLK-CTRL.
hc_qacceptn_o	Output	**clk**	This signal indicates when the CLK-CTRL accepts the quiescence request.
hc_qdeny_o			This signal indicates when the CLK-CTRL denies the quiescence request.
hc_qactive_o		Asynchronous	This signal indicates when one or more CLK-CTRL devices are active or they are requesting to exit from quiescence.

The following table lists the hierarchical power control signals.

Table A-23 CLK-CTRL hierarchical power control signals

Signal	Type	Clock	Description
pwr_qreqn_i	Input	Either: • Asynchronous, when `PWR_Q_CH_SYNC` == 1. • **clk**, when `PWR_Q_CH_SYNC` == 0.	This signal indicates when the controller issues a quiescence entry or exit request to the CLK-CTRL.
pwr_qacceptn_o	Output	**clk**	This signal indicates when the CLK-CTRL accepts the quiescence request.
pwr_qdeny_o			This signal indicates when the CLK-CTRL denies the quiescence request.
pwr_qactive_o		Asynchronous	This signal indicates when one or more CLK-CTRL devices are active or they are requesting to exit from quiescence.

The following table lists the clock device interface signals.

Table A-24 CLK-CTRL clock device interface signals

Signal	Type	Clock	Description
clk_qreqn_o[NUM_Q_CHL − 1:0]	Output	**clk**	This signal indicates when the CLK-CTRL issues a quiescence entry or exit request to the CLK-CTRL devices.
clk_qacceptn_i[NUM_Q_CHL − 1:0]	Input	Either: • Asynchronous, when `CLK_Q_CH_SYNC` == 1. • **clk**, when `CLK_Q_CH_SYNC` == 0.	This signal indicates when a CLK-CTRL device accepts the quiescence request.
clk_qdeny_i[NUM_Q_CHL − 1:0]			This signal indicates when a CLK-CTRL device denies the quiescence request.
clk_qactive_i[NUM_Q_CHL + NUM_QACTIVE_ONLY − 1:0]		Asynchronous	This signal indicates when a CLK-CTRL device is active or it is requesting to exit from quiescence.

The following table lists some miscellaneous signals.

Table A-25 CLK-CTRL other signals

Signal	Type	Clock	Description
clken_o	Output	**clk**	When HIGH, this signal can enable the **clk** signal for downstream devices.
clk_force_i	Input	Asynchronous	When HIGH, it disables the CLK-CTRL clock gating mechanism.
entry_delay_i[7:0]	Input	Asynchronous	Sets the value of the quiescence entry delay. Allows you to add 0-255 **clk** cycles of hysteresis to the quiescence entry sequence.

The following table lists the *Design for Test* (DFT) signals.

Table A-26 CLK-CTRL DFT signals

Signal	Type	Clock	Description
dftcgen	Input	**clk**	This signal enables the architectural clock gates, and ensures that the **clk** logic is active during DFT shift mode.

A.6 PPU signals

The following tables show the Power Policy Unit signals.

The following table lists the clock and reset signals.

Table A-27 PPU clock and reset signals

Signal	Type	Clock	Description
clk	Input	-	Clock input.
reset_n	Input	**clk**	An active-LOW reset input. **reset_n** can assert asynchronously, but must deassert synchronous to **clk**.

The following table lists the APB programming interface signals.

Table A-28 PPU programming interface signals

Signal	Type	Clock	Description
psel_i	Input	**clk**	See the *Arm® AMBA® 3 APB Protocol Specification, v1.0* for information about these signals.
penable_i			
paddr_i[31:0]			
pwrite_i			
pwdata_i[31:0]			
pwakeup_i			The PPU uses this wakeup signal as an input for the generation of the **ppuclk_qactive_o** signal. This signal must be driven from a register.
prdata_o[31:0]	Output		See the *Arm® AMBA® 3 APB Protocol Specification, v1.0* for information about these signals.
pready_o			
pslverr_o			

The following table lists the clock Q-Channel signals.

Table A-29 PPU clock Q-Channel signals

Signal	Type	Clock	Description
ppuclk_qreqn_i	Input	Either: • Asynchronous, when `QCLK_SYNC_EN` == 1. • **clk**, when `QCLK_SYNC_EN` == 0.	This signal indicates when the clock controller that controls the PPU **clk** input, issues a quiescence entry or exit request to the PPU.
ppuclk_qacceptn_o	Output	**clk**	This signal indicates when the PPU accepts a quiescence request to gate its clock.
ppuclk_qdeny_o			This signal indicates when the PPU denies a quiescence request to gate its clock.
ppuclk_qactive_o		Asynchronous	This signal indicates when the PPU requires the **clk** signal.

The following table lists the PPU device signals, when the PPU is configured to provide a single P-Channel, that is when `DEVCHAN_CFG` == 0.

Table A-30 PPU device interface signals, when `DEVCHAN_CFG` == 0

Signal	Type	Clock	Description
dev_preq_o	Output	**clk**	This signal indicates when the PPU issues a power mode request to the PPU device. At reset, this signal is LOW.
dev_pstate_o[P_CH_PSTATE_LEN −1:0]			The PPU sends the power mode and operating mode that is requested, when **dev_preq_o** is set HIGH. At reset, this signal is LOW.
dev_paccept_i	Input	Either: • Asynchronous, when `DEV_SYNC_EN` == 1. • **clk**, when `DEV_SYNC_EN` == 0.	This signal indicates when the PPU device accepts the power mode request.
dev_pdeny_i			This signal indicates when the PPU device denies the power mode request.
dev_pactive_i[P_CH_PACTIVE_LEN −1:0]		Either: • Asynchronous, when `DEV_ACTIVE_SYNC_EN` == 1. • **clk**, when `DEV_ACTIVE_SYNC_EN` == 0.	This signal indicates the power mode that the device requests.

The following table lists the PPU device signals, when the PPU is configured to provide one or more Q-Channels, that is when `DEVCHAN_CFG` > 0.

Table A-31 PPU device interface signals, when `DEVCHAN_CFG` > 0

Signal	Type	Clock	Description
dev_qreqn_o[DEVCHAN_CFG − 1:0]	Output	**clk**	This signal indicates when the PPU issues a quiescence entry or exit request to the PPU devices.
dev_qacceptn_i[DEVCHAN_CFG − 1:0]	Input	Either: • Asynchronous, when `DEV_SYNC_EN` == 1. • **clk**, when `DEV_SYNC_EN` == 0.	This signal indicates when a PPU device accepts the quiescence request.
dev_qdeny_i[DEVCHAN_CFG − 1:0]			This signal indicates when a PPU device denies the quiescence request.
dev_qactive_i[DEVCHAN_CFG − 1:0]		Either: • Asynchronous, when `DEV_ACTIVE_SYNC_EN` == 1. • **clk**, when `DEV_ACTIVE_SYNC_EN` == 0.	This signal indicates when a PPU device is active or it is requesting to exit from quiescence.

The following table lists the *Power Control State Machine* (PCSM) P-Channel signals.

Table A-32 PPU PCSM P-Channel signals

Signal	Type	Clock	Description
pcsm_preq_o	Output	**clk**	This signal indicates when the PPU issues a power mode request to the PCSM. At reset, this signal is LOW.
pcsm_pstate_o[P_CH_PSTATE_LEN−1:0]			The power mode and operating mode that the PPU requests, when **pcsm_preq_o** is set HIGH.
pcsm_paccept_i	Input	Either: - Asynchronous, when `PCSM_SYNC_EN` == 1. - **clk**, when `PCSM_SYNC_EN` == 0.	This signal indicates when the PCSM accepts the power mode request.
pcsm_mode_stat_i[P_CH_PSTATE_LEN−1:0]	Input	Pseudo static.	This signal provides the MODESTAT information from the PSCM. The presence of this signal is configuration dependent. If `OFF_MEM_RET_TRANS_CFG` == 0, the signal is not present.

The following table lists the device control signals. See the *Arm® Power Policy Unit Architecture Specification, version 1.1* for more information.

Table A-33 PPU device control signals

Signal	Type	Clock	Description
ppuhwstat_o[x:0]	Output	**clk**	This signal indicates the current mode of the PPU. The width of the bus depends on the value of the `NUM_OPMODE_CFG` parameter. The value of `x` is: - `x` = 15 + (`NUM_OPMODE_CFG`+1).
devclken_o			Device clock enable.
devemuclken_o			Device emulated mode clock enable.
devisolaten_o			Device isolation control.
devemuisolaten_o			Device emulated isolation control.
devwarmresetn_o			The Warm reset for non-retention registers.
devretresetn_o			The Warm reset for retention registers.
devporesetn_o			Device reset.

The following table lists the interrupt and revision signals.

Table A-34 PPU interrupt and revision signals

Signal	Type	Clock	Description
irq_o	Output	**clk**	Interrupt signal.
ecorevnum_i[3:0]	Input		This signal sets the value of the PPU_IIDR.REVISION and PID3.REVAND register fields.

The following table lists the *Design for Test* (DFT) signals.

Table A-35 PPU DFT signals

Signal	Type	Clock	Description
dftcgen	Input	**clk**	This signal enables the architectural clock gates, and ensures that the **clk** logic is active during DFT shift mode.
dftisodisable	Input	Asynchronous	Provides an override for the PPU isolation control signals. Use in DFT mode only.
dftrstdisable	Input	Asynchronous	Provides an override for the PPU reset output signals. Use in DFT mode only.

Related reference

3.4.8 Peripheral ID 3 on page 3-53

3.3 Implementation Identification Register, PPU_IIDR on page 3-48

Appendix B
Revisions

This appendix describes the technical changes between released issues of this book.

It contains the following section:
- *B.1 Revisions* on page Appx-B-72.

B.1 Revisions

This appendix describes changes between released issues of this book.

Table B-1 Issue 0000-00

Change	Location	Affects
First release	-	-

Table B-2 Differences between issue 0000-00 and issue 0000-01

Change	Location	Affects
Added the device interface signals for a PPU that is configured to support Q-Channels.	*Table A-31 PPU device interface signals, when DEVCHAN_CFG > 0* on page Appx-A-68	All revisions
Corrected the number of Q-Channels that the LPD-Q supports.	• *1.1 About the Power Control Kit* on page 1-12. • *2.1 About the LPD-Q Q-Channel Distributor* on page 2-18	All revisions
Updated the unmapped **ctrl_pstate_i[3:0]** value description.	*2.2.3 LPD-P operating in sequencer mode* on page 2-24	All revisions
Corrected the signal names. Added information about the **ctrl_qactive_o[<N>]** and **clk_qactive_o** signals.	*2.3 About the LPC-Q Q-Channel Combiner* on page 2-29	All revisions
Updated the **pcsm_mode_stat_i** description.	*Table A-32 PPU PCSM P-Channel signals* on page Appx-A-69	All revisions
Updated the **ppuhwstat_o** description.	*Table A-33 PPU device control signals* on page Appx-A-69	All revisions

Table B-3 Differences between issue 0000-01 and issue 0001-00

Change	Location	Affects
Updated the clock gate implementation.	*Figure 2-8 CLK-CTRL interfaces* on page 2-35	All revisions
Updated the usage constraints for multiple parameters.	*Table 2-6 PPU configuration parameters* on page 2-38	All revisions
Corrected the description of the VARIANT and REVISION fields.	*Table 3-2 PPU_IIDR bit assignments* on page 3-48	All revisions
Updated the VARIANT field value.	*Table 3-2 PPU_IIDR bit assignments* on page 3-48	r0p1
Updated the REV field value.	*3.4.7 Peripheral ID 2* on page 3-52	r0p1
Corrected the description of the REVAND field.	*Table 3-10 Peripheral ID 3 Register bit assignments* on page 3-53	All revisions

Table B-4 Differences between issue 0001-00 and issue 0002-00

Change	Location	Affects
No technical changes.	-	-

Table B-5 Differences between issue 0002-00 and issue 0003-00

Change	Location	Affects
Added section.	*2.2.1 LPD-P initialization* on page 2-23	r0p3 onwards
Added section	*2.4.1 P2Q initialization* on page 2-31	r0p3 onwards

Table B-5 Differences between issue 0002-00 and issue 0003-00 (continued)

Change	Location	Affects
Added usage contraint for `DYN_MEM_RET_SPT_CFG`.	*2.6.1 PPU configuration parameters* on page 2-38	r0p3
Updated calculation of x in **ppuhwstat_o[x:0]** description.	*A.6 PPU signals* on page Appx-A-67	r0p3