Arm® Lifecycle Manager (LCM)

Version 1.0

Specification

Non-Confidential

Copyright © 2023 Arm Limited (or its affiliates).
All rights reserved.

Issue 01

107616_0000_01_en

Arm® Lifecycle Manager (LCM)

Specification

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.

Release information

Document history

Issue	Date	Confidentiality	Change
00000-01	31 May 2023	Non-Confidential	Initial release

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, 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.

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 affiliates) 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 https://www.arm.com/company/policies/trademarks.

Copyright © 2023 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|version 21.0)

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.

Feedback

Arm welcomes feedback on this product and its documentation. To provide feedback on the product, create a ticket on https://support.developer.arm.com

To provide feedback on the document, fill the following survey: https://developer.arm.com/documentation-feedback-survey.

Inclusive language commitment

Arm values inclusive communities. Arm recognizes that we and our industry have used language that can be offensive. Arm strives to lead the industry and create change.

We believe that this document contains no offensive language. To report offensive language in this document, email terms@arm.com.

Contents

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

1. Introduction

1.1 Conventions

The following subsections describe conventions used in Arm documents.

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: developer.arm.com/glossary.

Typographic conventions

Arm documentation uses typographical conventions to convey specific meaning.

Convention	Use
italic	Citations.
bold	Interface elements, such as menu names. Terms in descriptive lists, where appropriate.
`monospace`	Text that you can enter at the keyboard, such as commands, file and program names, and source code.
`monospace` <u>`underline`</u>	A permitted abbreviation for a command or option. You can enter the underlined text instead of the full command or option name.
`<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	Terms that have specific technical meanings as defined in the *Arm® Glossary*. For example, **IMPLEMENTATION DEFINED**, **IMPLEMENTATION SPECIFIC**, **UNKNOWN**, and **UNPREDICTABLE**.
⚠ Caution	Recommendations. Not following these recommendations might lead to system failure or damage.
⚠ Warning	Requirements for the system. Not following these requirements might result in system failure or damage.
⚠ Danger	Requirements for the system. Not following these requirements will result in system failure or damage.
✎ Note	An important piece of information that needs your attention.

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.

Convention	Use
Tip	A useful tip that might make it easier, better or faster to perform a task.
Remember	A reminder of something important that relates to the information you are reading.

1.2 Useful resources

This document contains information that is specific to this product. See the following resources for other useful information.

Access to Arm documents depends on their confidentiality:

- Non-Confidential documents are available at developer.arm.com/documentation. Each document link in the following tables goes to the online version of the document.
- Confidential documents are available to licensees only through the product package.

Arm product resources	Document ID	Confidentiality
Arm® CoreLink™ SIE-200 System IP for Embedded Technical Reference Manual	DDI 0571	Non-Confidential

Arm architecture and specifications	Document ID	Confidentiality
AMBA® APB Protocol Specification	IHI 0024	Non-Confidential
Arm® Key Management Unit (KMU) Specification	107715	Non-Confidential
Arm® Platform Security Architecture Trusted Base System Architecture for Arm®v6-M, Arm®v7-M and Arm®v8-M	DEN 0083	Non-Confidential
Arm® v8-M Architecture Reference Manual	DDI 0553	Non-Confidential

Non-Arm resources	Document ID	Organization
FIPS 140-2 Security Requirements for Cryptographic Modules	FIPS 140-2	National Institute of Standards and Technology

Note

Arm tests its PDFs only in Adobe Acrobat and Acrobat Reader. Arm cannot guarantee the quality of its documents when used with any other PDF reader.

Adobe PDF reader products can be downloaded at http://www.adobe.com

1.3 Other information

See the Arm website for other relevant information.

- Arm® Developer.
- Arm® Documentation.
- Technical Support.
- Arm® Glossary.

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

2. Terms and Abbreviations

The following table lists the terms and abbreviations used throughout this document.

Table 2-1: Terms and abbreviations

Term	Description
CM	Chip Manufacturer
DM	Device Manufacturer
DRBG	Deterministic Random Bit Generator
HW	Hardware
HUK	Hardware unique key
KMU	Key management unit
LCM	Lifecycle manager
LCS	Lifecycle state
LFSR	Linear-Feedback Shift register
NVM	Non-Volatile Memory
OEM	Original Equipment Manufacturer
OTP	One-Time Programmable
PCI	Production Chip Indicator
PoR	Power-on reset
PSA	Platform security architecture
RAZ/WI	Read as Zero, Write Ignore
RMA	Return Merchandise Authorization
RoT	Root-of-Trust
SE	Security Enabled
SoC	System-on-Chip
SW	Software
TCI	Test Chip Indicator

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

3. LCM overview

The Arm® Lifecycle Manager (LCM) is a security component that is intended for use in Trusted Execution Environments, such as Armv8-M architectures.

The LCM is aimed at the IoT market or security subsystems that are part of infrastructure, client, or automotive markets. The LCM provides foundational security functions for the system, such as *Root-of-Trust* (RoT) key management, lifecycle management, and Secure debug domain control. These functions allow you to build security services such as Secure boot, Secure firmware update, attestation, Secure storage, and Secure partitioning (for example, Arm Trusted Firmware). The LCM is a critical component for a system that aims for PSA Level 2+ certification.

The LCM hardware core integrates with the hosting platform using APB3 interfaces, including manager ports and a subordinate port. Other interfaces provide more specialized functionality by indicating the current security state of the LCM to the system and controlling the system behavior.

3.1 LCM features and assets

The LCM provides the following security features.

- Secure lifecycle states
- Persistent storage of RoT keys
- Secure export of RoT keys through the APB interface.
- Debug control and hardware feature enable signals, with properties dependent on LCS
- OTP programmer (user) partition for read and write access
- Secure provisioning for manufacturer asset provisioning in a Non-trusted environment

The LCM manages the following assets and Roots-of-Trust:

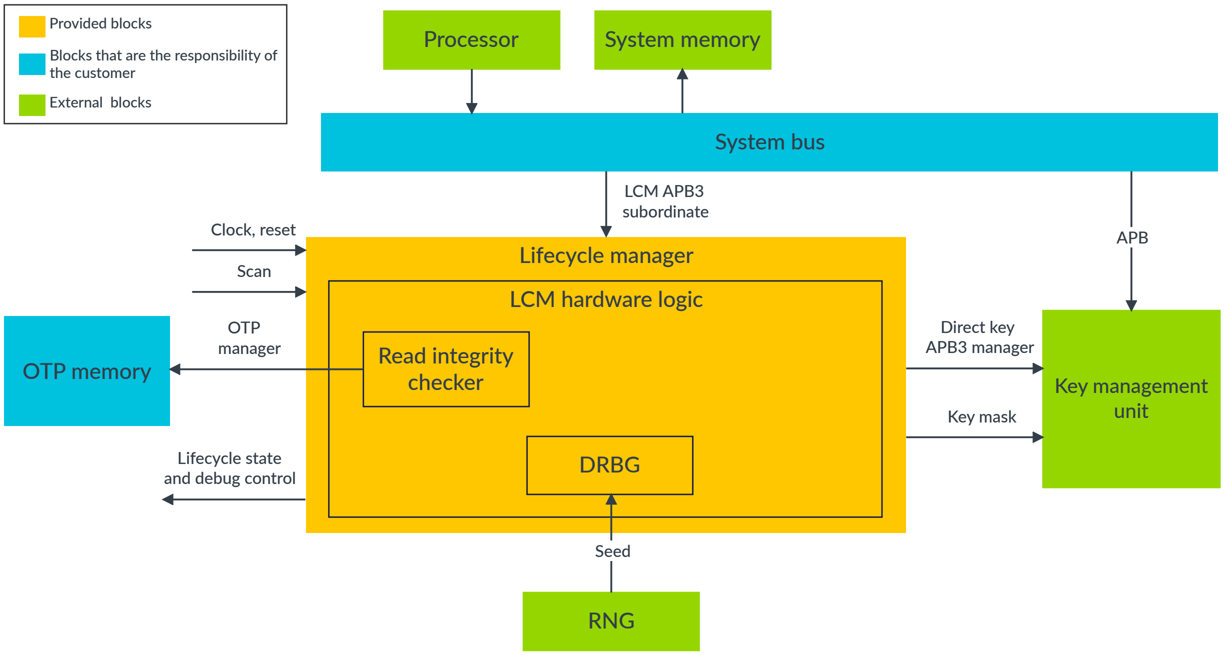
- The Secure lifecycle states, which are encoded and stored securely in the OTP memory
- The device owner's hash of the *Root of Trust Public Key* (ROTPK), which is stored securely in the OTP memory
- The device *Hardware Unique Key* (HUK)
- The *Group Unique Key* (GUK)
- A set of two keys owned by the chip vendor, typically used for code encryption and provisioning
- A set of two keys owned by the *Original Equipment Manufacturer* (OEM) typically used for code encryption and provisioning
- General-purpose OTP flags and signals intended to be used by the chip vendor
- An RTL secret key for Secure provisioning in a Non-trusted environment
- An RTL secret mask to protect confidential fields of the OTP memory
- General-purpose OTP memory partition (code or data), which the programmer can use

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

3.2 Topology

The following top-level figure shows an example system that includes an LCM.

Figure 3-1: LCM top-level diagram



Lifecycle state and debug control unit

The LCM defines a set of lifecycle states (LCS). The LCM also includes a mechanism for controlling debugging capabilities and HW feature enablement, which is defined by the LCS. For more information, see Lifecycle states and Debug Control Unit.

OTP manager

The LCM includes an OTP manager that is responsible for controlling the OTP assets. The OTP manager uses the OTP APB3 manager interface that is connected to the OTP. For more information, see OTP manager.

Read Integrity Checker:

Part of the OTP manager. An engine (state machine) that performs integrity checks on the fields in the OTP that are used to store keys and to determine the LCS.

Key mask

The secret OTP keys are protected using an RTL mask. For more information, see OTP keys integrity protection.

DRBG

The LCM includes a *Deterministic Random Bit Generator* (DRBG) that is used to generate masking values and random delay values. For more information, see DRBG module.

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

LCM APB3 subordinate interface

The LCM includes an APB3 subordinate interface which is used to control the LCM operations, read the LCM FSM state and the LCS, and read and write to the OTP memory. For more information, see LCM APB3 subordinate interface.

Direct key APB3 manager interface

The LCM includes a direct key APB3 manager interface that exports the LCM keys to an external peripheral for key management. For more information, see Direct key APB3 manager interface.

Clock, reset

The LCM requires a single clock input and reset signal that is controlled by the system. For more information, see Clock logic and reset signals.

Scan

The LCM supports scan cell insertion methodology for SoC DFT strategy. For more information, see Scan signals.

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

4. Functional description

The LCM manages the security of the device by using security modes and states to control its behavior and its debug control capabilities.

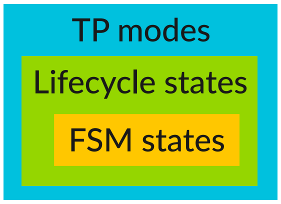
The chip vendor first defines the Test or Production Mode (TP mode) of the device, depending on whether the device is intended for testing purposes (TCI TP mode) or is part of a fully secured production batch (PCI TP mode).

The LCM uses lifecycle states (LCS) to control which features of the device are enabled during the various stages of its life. The chip vendor and the OEM can use Secure provisioning to load their secret keys to the LCM during Chip Manufacturing (CM) LCS and Device Manufacturing (DM) LCS. In Secure Enable LCS Secure provisioning is disabled. Debug features are also disabled in SE LCS, unless the the device is in TCI TP mode. The LCM also includes a Return Merchandise Authorization (RMA) LCS, which is a terminal state for devices that are returned to the manufacturer for analysis of fatal errors.

Within the lifecycle states, the LCM *Finite State Machine* (FSM) uses Reset, Ready, and Fatal Error FSM states to control the behavior of the LCM depending on its internal state and whether any errors have been detected.

The following figure shows the relationships between the security modes and states.

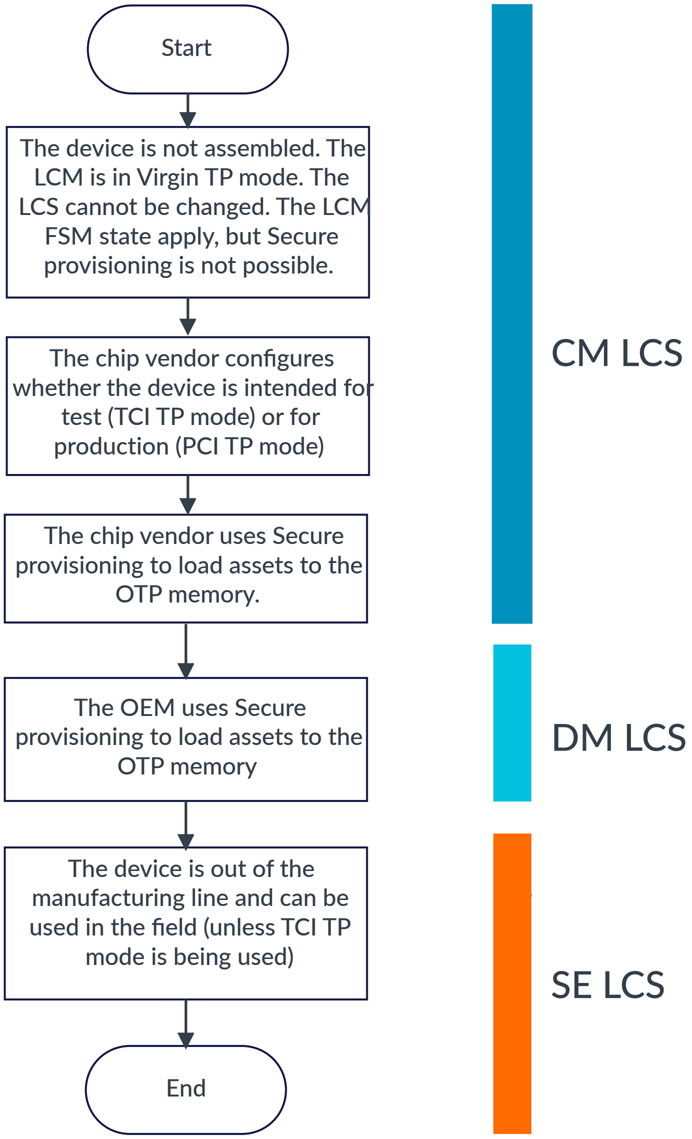
Figure 4-1: LCM security modes and states



The following figure shows a simplified lifetime flow for a device which includes the LCM. At any point in the lifetime of the device, the LCS can be changed to RMA LCS.

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Figure 4-2: LCM simplified flow


Start
The device is not assembled. The LCM is in Virgin TP mode. The LCS cannot be changed. The LCM FSM state apply, but Secure provisioning is not possible.
The chip vendor configures whether the device is intended for test (TCI TP mode) or for production (PCI TP mode)
The chip vendor uses Secure provisioning to load assets to the OTP memory.
The OEM uses Secure provisioning to load assets to the OTP memory
The device is out of the manufacturing line and can be used in the field (unless TCI TP mode is being used)
End
CM LCS
DM LCS
SE LCS

To protect the assets and hardware keys stored in the OTP memory when the device is in PCI TP mode, the LCM uses OTP masking. Keys stored in the OTP memory are further protected with built-in integrity checking functionality.

4.1 TP mode

The LCM includes a mechanism, *Test or Production Mode* (TP mode), that allows the chip vendor to define whether the chip or device is part of a fully secured production batch or is intended for testing purposes.

The LCM defines the following TP modes.

Virgin TP mode

The device is in a non-initialized state. The silicon chip is expected to be blank. The OTP memory is expected to be all zeros (empty). As the first step before initiating the key provisioning process, the programmer must set either the TCI or PCI TP mode value in TP mode Configuration OTP memory field. Otherwise, the OTP fields containing the keys and security configurations cannot be programmed.

Production Chip Indication (PCI) TP mode

The Production Chip Indicator is set. The Secure provisioning functionality of LCM uses the actual secret RTL OTP mask values.

Test Chip Indication (TCI) TP mode

The Test Chip Indicator is set. The RTL secret key and the and RTL OTP masks are disabled, and their related functionality uses zeros instead of the actual RTL secret values. The Secure provisioning functionality of the LCM can still be used, but with zeroized RTL OTP mask values. For more details on the RTL key, see Hardware keys.

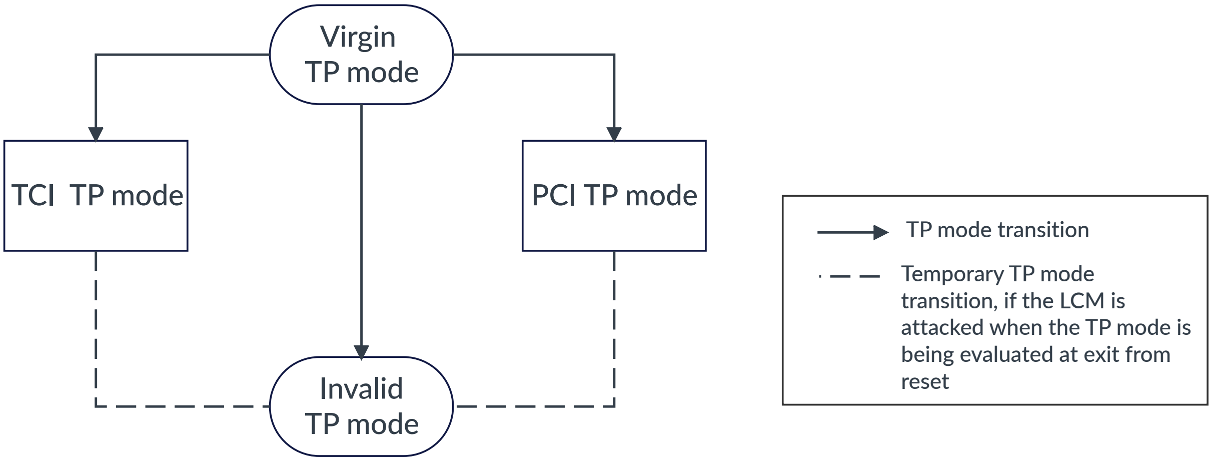
Invalid TP mode

When an Invalid TP mode value is written to the OTP memory, the LCM is permanently locked. Debug is disabled. Intentional transition to Invalid TP mode is possible only from Virgin TP mode. For more information, see LCM lockdown. During certain types of attacks, such as electromagnetic interference, the device may appear to be in Invalid TP mode for the duration of the attack. After reset and if the attack is no longer present, the device returns to its previous TP mode, since no permanent change was made to the value written in the OTP memory.

The *Production Chip Indication* (PCI) TP mode and *Test Chip Indication* (TCI) TP mode affect the Secure services and security mitigations that are used for the provisioning of device secret keys.

The following figure shows the transition between the TP modes.

Figure 4-3: TP mode transitions


Virgin TP mode
TCI TP mode
PCI TP mode
Invalid TP mode
TP mode transition
Temporary TP mode transition, if the LCM is attacked when the TP mode is being evaluated at exit from reset

The value of the TP mode register is reflected through the tp_mode output signals as defined in the LCM generic signals. The tp_mode output must be valid before the lcs_is_valid output is asserted. The value of the TP mode register is readable from the APB-S interface and encoded as defined in the TP_MODE, Test or Production Mode register.

The following table defines the TP mode policies:

Table 4-1: TP mode policies

TP mode	RTL Key usage	RTL mask usage	OTP access to secret keys	Transition to TCI or PCI	DCU default reset value	DCU permanent disable mask
Virgin	Disabled	Disabled	OTP write is disabled	Can be done to both TCI or PCI	DCUs default value is according to CM LCS.	PCI permanent disable mask according to parameter for CM LCS
TCI	Disabled	Disabled	Write or read enabled according to LCS	Disabled	TCI default reset value according to parameter for LCS	TCI permanent disable mask according to parameter for LCS
PCI	Enabled in Secure provisioning	Enabled	Write or read enabled according to LCS and whether Secure provisioning is enabled	Disabled	PCI default reset value according to parameter for LCS	PCI permanent disable mask according to parameter for LCS
Invalid	Disabled	Disabled	OTP memory write is disabled	Disabled	PCI default reset value is according to SE LCS.	PCI permanent disable mask according to SE LCS parameter

4.2 Lifecycle states

The LCM defines four *Lifecycle States* (LCS). The lifecycle states enable the device (both hardware and software) to behave differently in various stages of its life, depending on whether it is in production or in the field. These changes in device behavior protect any security assets after they have been introduced into the device and reduce the risk of IP theft and reverse engineering. The lifecycle states only apply if the TP mode of the device is in Virgin, TCI, or PCI.

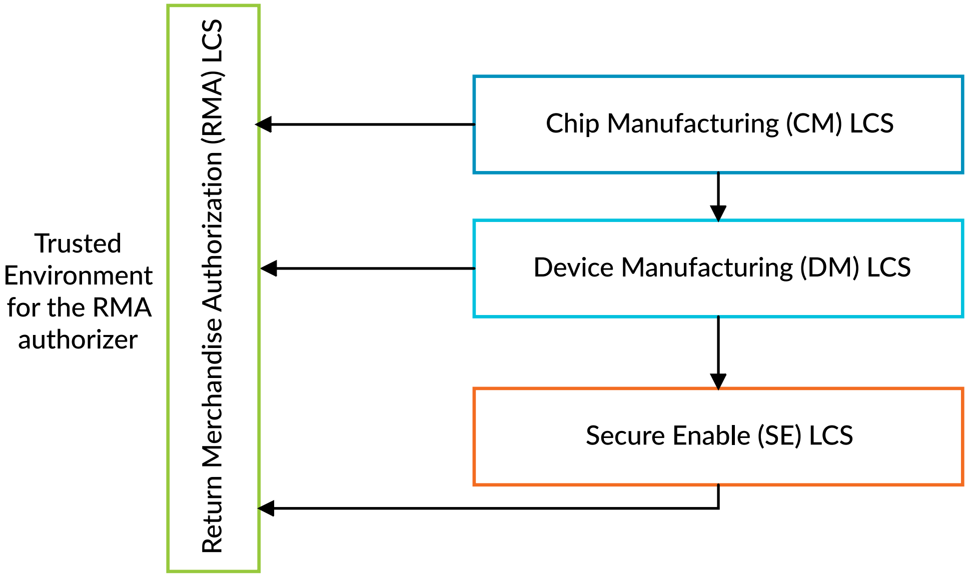
The lifecycle states are:

- Chip Manufacturing LCS
- Device Manufacturing LCS
- Secure Enable LCS
- Return Merchandise Authorization LCS

The following figure shows the possible transitions between the lifecycle states.

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Figure 4-4: LCM lifecycle states


Return Merchandise Authorization (RMA) LCS
Trusted Environment for the RMA authorizer
Chip Manufacturing (CM) LCS
Device Manufacturing (DM) LCS
Secure Enable (SE) LCS

The LCM determines the LCS by reading the OTP fields immediately after reset. The LCS must not be affected by any input to LCM, other than the OTP memory-related values. The calculation of the LCS depends on the values of the following fields in OTP memory:

- CM Configuration 1
- CM Configuration 2
- DM Configuration
- CM RMA OTP flag
- DM RMA OTP flag

After the LCS is calculated according to the OTP fields, the LCM stores it in the LCS_VALUE register.

An integrity error or any invalid LCS state in the OTP memory that does not map to the CM, DM, SE, or RMA lifecycle states, causes the LCM to leave the lcs_is_valid signal deasserted and also assert the fatal_err signal. When the fatal_err signal is asserted, LCM is disabled, and the keys are invalidated and not exported.

The following table describes how the lifecycle states defined by this specification map to the PSA lifecycle states.

Table 4-2: LCM lifecycle states and PSA lifecycle states

LCM lifecycle state	PSA lifecycle state
Chip Manufacturing (CM)	Device Assembly and Test
Device Manufacturing (DM)	Provisioning
Secure Enable (SE)	Secured
Return Merchandise Authorization (RMA)	Decommissioned

Manufacturing test access is not blocked in the DM LCS.

4.2.1 Chip Manufacturing LCS

The *Chip Manufacturing* (CM) LCS indicates the device has, typically, not been assembled into a device yet, and has not left production line of the chip vendor.

In CM LCS, Arm recommends all debugging and testing facilities of the chip are enabled (depending on configuration), even those that enable access to security-sensitive information (for example, chip test modes and validation functionality such as "external NOR-flash in place of boot ROM").

The LCM is in CM LCS by default when it is in Virgin TP mode. To perform Secure provisionming, the programmer must set either the TCI or PCI TP mode value in the TP mode Configuration OTP memory field. The LCM cannot move to DM LCS unless Secure provisioning has been performed.

Transition to the next LCS (DM) is done by programming the OTP memory fields with chip vendor-related secrets. However, if only some of the CM configuration 1 or CM Configuration 2 bits are set, it is possible for the device to remain in CM LCS without transitioning to DM LCS. The transition to the RMA LCS is done by programming both the CM and the DM RMA OTP flags.

When the device is in CM LCS, the lcs signal indication value is `0b000`.

In CM LCS, the following capabilities are enabled:

- Debugging and software development capabilities. These capabilities are configurable and could be restricted.
- External-testing capabilities, such as JTAG boundary scan.

Conditions

The device is in CM LCS if the following conditions are met on reset in the OTP memory:

- Bits [31:0] of the CM configuration 1 flag (OTP memory offset `0x00E4`) are not set.
- Bits [7:0] of the CM configuration 2 glag (OTP memory offset `0x00E8`) are not set.
- Bits [15:0] of the DM configuration flag (OTP memory offset `0x00EC`) are not set.
- Either the CM RMA OTP flag (OTP memory offset `0x00F0`) or the DM RMA OTP flag (OTP memory offset `0x00F4`) is not set. For the LCM LCS to change to RMA LCS, both of these flags must be set.

4.2.2 Device Manufacturing LCS

The *Device Manufacturing* (DM) LCS is typically used during device assembly on the production line of the OEM.

In the DM LCS, the following capabilities are enabled:

- Debugging and software development capabilities. These capabilities are configurable and could be restricted.

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.

- External-testing capabilities, such as JTAG boundary scan.

Internal testing capabilities (for example, full-chip internal scan) are typically disabled.

Transition to the next lifecycle state (SE) is done by programming the OTP fields with OEM-related secrets. Transition to the RMA lifecycle is done by programming the CM and DM RMA OTP flags.

When the device is in DM LCS, the lcs signal indication value is `0b001`.

Conditions

The device is in DM LCS if the following conditions are met in the OTP memory on reset:

- Either bits [31:0] of the CM configuration 1 flag (OTP memory offset `0x00E4`) are set to a value other than zero or bits [7:0] of the CM configuration 2 glag (OTP memory offset `0x00E8`) are set to a value other than zero.
- Bits [15:0] of the DM configuration flag (OTP memory offset `0x00EC`) are not set.
- Either the CM RMA OTP flag (OTP memory offset `0x00F0`) is not set or the DM RMA OTP flag (OTP memory offset `0x00F4`) is not set. For the LCM LCS to change to RMA LCS, both of these flags must be set.

4.2.3 Secure Enable LCS

The *Secure Enable* (SE) LCS is used for devices out of the manufacturing line and in the field.

A device in SE state may freely use all security functions, but its debugging and testing capabilities which are based on the Secure debug are typically fully blocked, unless the LCM is in TCI TP mode.

The only possible transition from SE LCS is into RMA LCS state.

When the device is in SE LCS, the lcs signal indication value is `0b101`.

Conditions

The device is in SE LCS if the following conditions are met on reset in the OTP memory:

- Either bits [31:0 of CM configuration 1 flag (OTP memory offset `0x00E4`) are set to a value other than zero or bits [7:0 of CM configuration 2 flag (OTP memory offset `0x00E8`) are set to a value other than zero.
- Bits [15:0] of DM configuration flag (OTP memory offset `0x00EC`) are set to a value other than zero.
- Either the CM RMA OTP flag (OTP memory offset `0x00F0`) is not set or the DM RMA OTP flag (OTP memory offset `0x00F4`) is not set. For the LCM LCS to change to RMA LCS, both of these flags must be set.

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

4.2.4 Return Merchandise Authorization LCS

The *Return Merchandise Authorization* (RMA) lifecycle state is a terminal state for devices that are returned to the manufacturer for analysis of fatal errors.

When a device is put into RMA state, it loses its secret keys and therefore its ability to perform most security functions. However, depending on the partner's use of the PERMANENT_DISABLE_MASK_VAL_LCS_RMA configuration parameters, the device may regain full access to all debugging and testing capabilities. For more information, see Configuration parameters.

Any RMA Flag value that is different than all zeroes or all ones causes the LCM to leave the lcs_is_valid signal deasserted and assert the fatal_err signal.

When the device is in RMA LCS, the lcs signal indication value is `0b111`.

Conditions

The device is in RMA LCS if both of the following conditions are met on reset in the OTP memory:

- The CM RMA OTP flag (OTP memory offset `0x00F0`) is set to all ones.
- The DM RMA OTP flag (OTP memory offset `0x00F4`) is set to all ones.

4.2.4.1 Transition into RMA LCS

The LCM supports a two-phase RMA LCS transition to allow both the chip vendor and the OEM to be involved in the RMA transition. RMA LCS is set only if the CM RMA OTP flag and the DM RMA OTP flags are set. One flag is programmed by the chip vendor and the other by the OEM.

The chip vendor can lock the DM RMA OTP flag using the DMA_RMA_LOCK register. The OEM can set the DM RMA OTP flag only if it was not locked by the chip vendor in advance. For security reasons, the DM RMA OTP flag should only be unlocked after successful authorization by the chip vendor boot loader code.

The DM RMA OTP flag lock feature prevents unauthorized transitions into RMA LCS.

4.3 LCM FSM states

The LCM *Finite State Machine* (FSM) transitions between three FSM states depending on whether the device is reset, operational, or if an error has been detected. The FSM checks the values stored in the LCM OTP memory are correct.

Reset FSM state

Reset FSM state occurs after LCM performs a Cold reset. The transition between Reset FSM state and Ready FSM state forms the LCM start-up flow.

Ready FSM state

Ready FSM state is is the default FSM state once the LCM start-up flow is completed. Once for each power cycle, the LCM can transition from Ready FSM state to Secure provisioning. However, the transition from Ready FSM state to Secure provisioning can only happen in CM or DM LCS.

Fatal Error FSM state

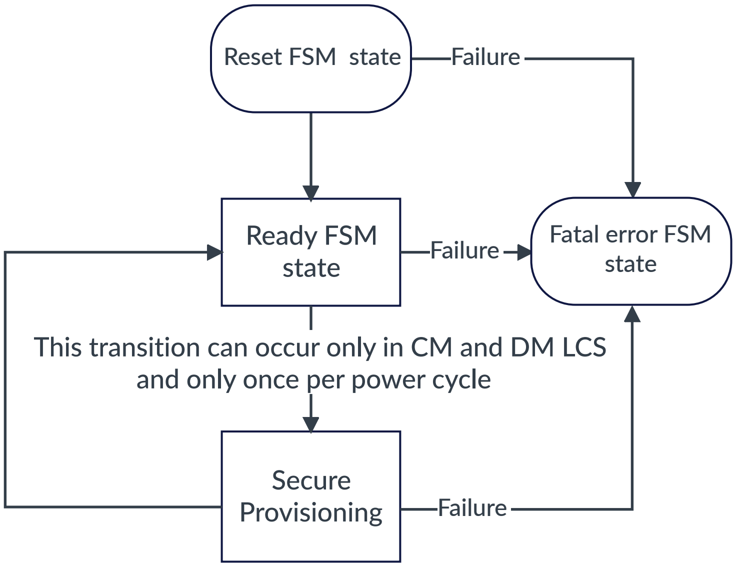
Fatal Error FSM state is triggered by the LCM FSM or can by initiated by software. In Fatal Error FSM state, writes to the LCM are disabled.

The FSM states are different from the LCS states. Reset, Ready, and Fatal Error FSM states refer to the operation of the LCM, not necessarily the system as a whole. They can occur when the device is in CM, DM, SE, or RMA LCS.

Figure 4-5: LCM FSM states on page 24 provides a high-level overview of the possible transitions between the different FSM states. The FSM transitions are triggered by the FSM itself except the transition to Secure provisioning, which is controlled by software.

The LCM FSM can also move to Fatal Error FSM state, when the programmer sets the FATAL_ERR register to the FATAL_ERR_SET value (`0xFA7A_1EEE`) through the APB-S interface.

Figure 4-5: LCM FSM states



Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

4.3.1 Reset FSM state

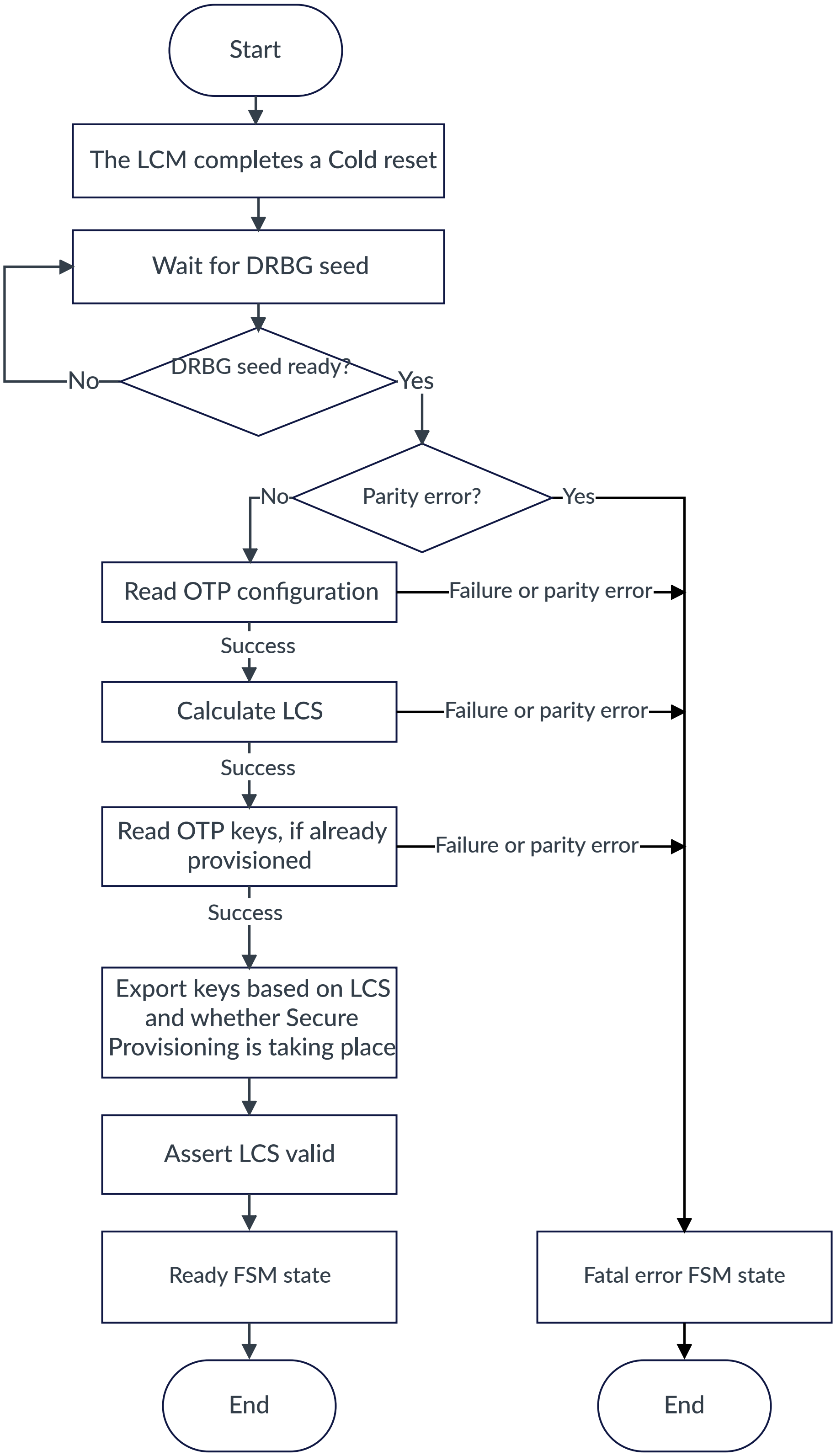
Reset FSM state occurs when the LCM completes a Cold reset. On reset, regardless of the LCS, the LCM copies the 16b *General Purpose Persistent Configuration* (GPPC) field from the OTP memory to the GPPC register. The GPPC register reflects its value at the GPPC signals interface.

Reset FSM state occurs in all lifecycle states. The LCM can transition from Reset FSM state either to Ready FSM state or to Fatal Error FSM state. The transition between Reset FSM state and Ready FSM state forms the LCM start-up flow.

The following figure shows the high-level flow of Reset FSM state, which is triggered when the LCM completes Cold reset.

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Figure 4-6: LCM Reset FSM state



Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

4.3.2 Ready FSM state

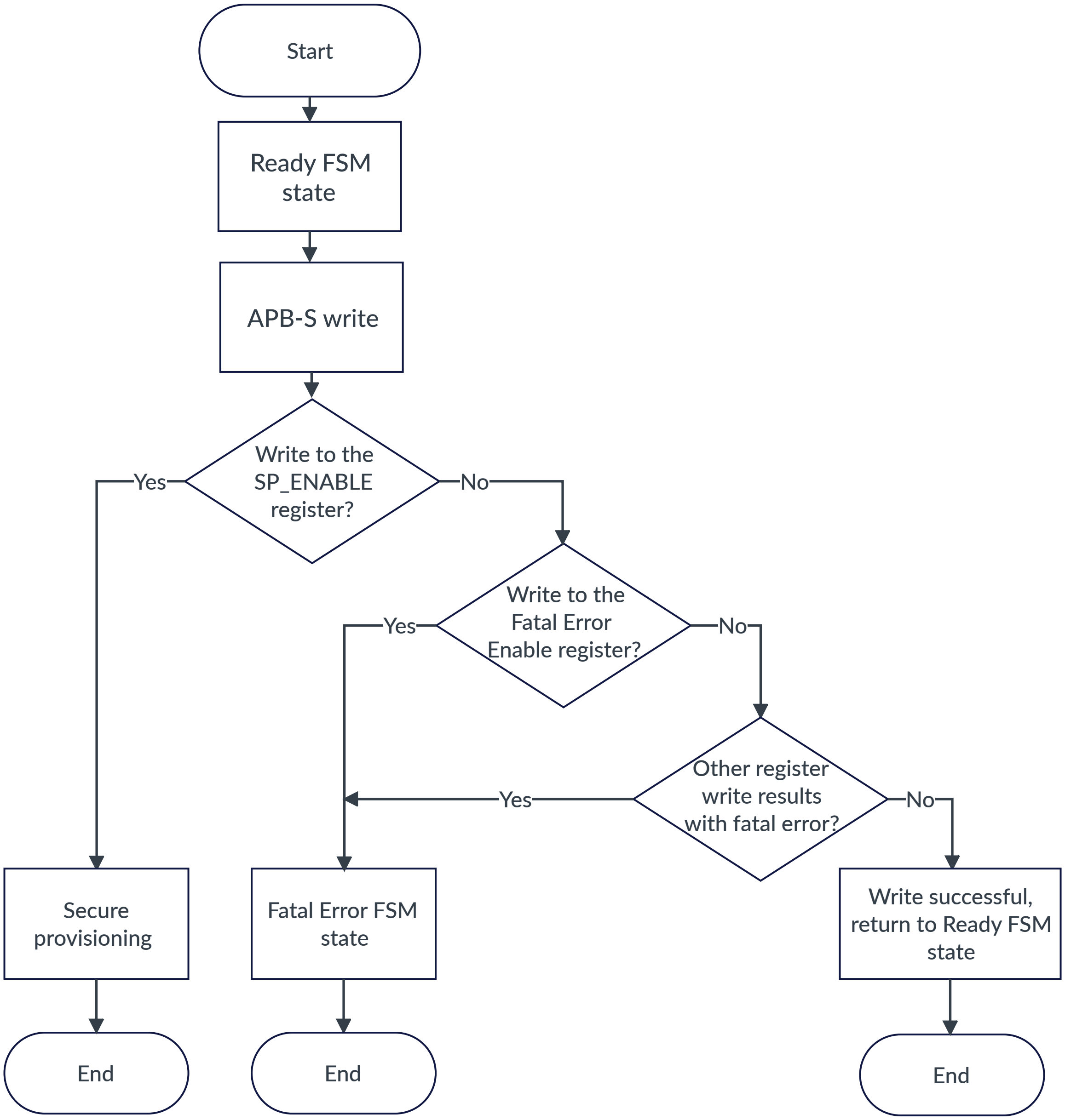
Once LCM start-up flow is complete, the LCM is in Ready FSM state.

The Ready FSM state can occur in all lifecycle states. After a successful read or write, the LCM remains in Ready FSM state. Transition to Fatal Error FSM state occurs if there is a parity mismatch, or if the internal register duplication compare fails, or if the programmer sets the FATAL_ERR register to the FATAL_ERR_SET value (`0xFA7A_1EEE`).

The following figure shows a high-level flow for a Write in Ready FSM State.

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Figure 4-7: A Write in LCM Ready FSM state


Start
Ready FSM state
APB-S write
Write to the SP_ENABLE register?
Yes
No
Write to the Fatal Error Enable register?
Yes
No
Other register write results with fatal error?
Yes
No
Secure provisioning
Fatal Error FSM state
Write successful, return to Ready FSM state
End
End
End

For more information about the transition between Ready FSM state and Secure provisioning, see Secure provisioning.

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

4.3.2.1 Secure provisioning

The LCM allows the chip vendor and OEM to provision their assets into the OTP memory in an untrusted environment, for example during production.

Once for each power cycle, the LCM can transition from Ready FSM state to Secure provisioning. However, the transition from Ready FSM state to Secure provisioning can only happen in CM or DM LCS.

The LCM includes a Secure provisioning reset request signal, sp_rst_req, to indicate to the system that Secure provisioning was enabled.

If the provisioning environment is not Trusted, the LCM provides an RTL group key that can be used to protect the assets. The RTL key is usable and exported only after Secure provisioning is enabled and only if the LCM is in PCI TP mode. The RTL key is protected by the LCM, which does not allow the scan interface and the debug interface to access it.

To enable the RTL key and initiate the Secure provisioning flow, the programmer should set the SP_ENABLE register to the specified value (`0x5EC1_0E1E`) from the APB-S interface. The LCM asserts the sp_rst_req interface signal, validates the KRTL HW key, and exports all valid keys by writing to the direct key APB interface. The LCM then enables Secure provisioning with the DCU settings deasserted according to value of the DCU_SP_DISABLE_MASK_VAL configuration parameter in PCI TP mode. If no errors occur, lcs_is_valid remains asserted.

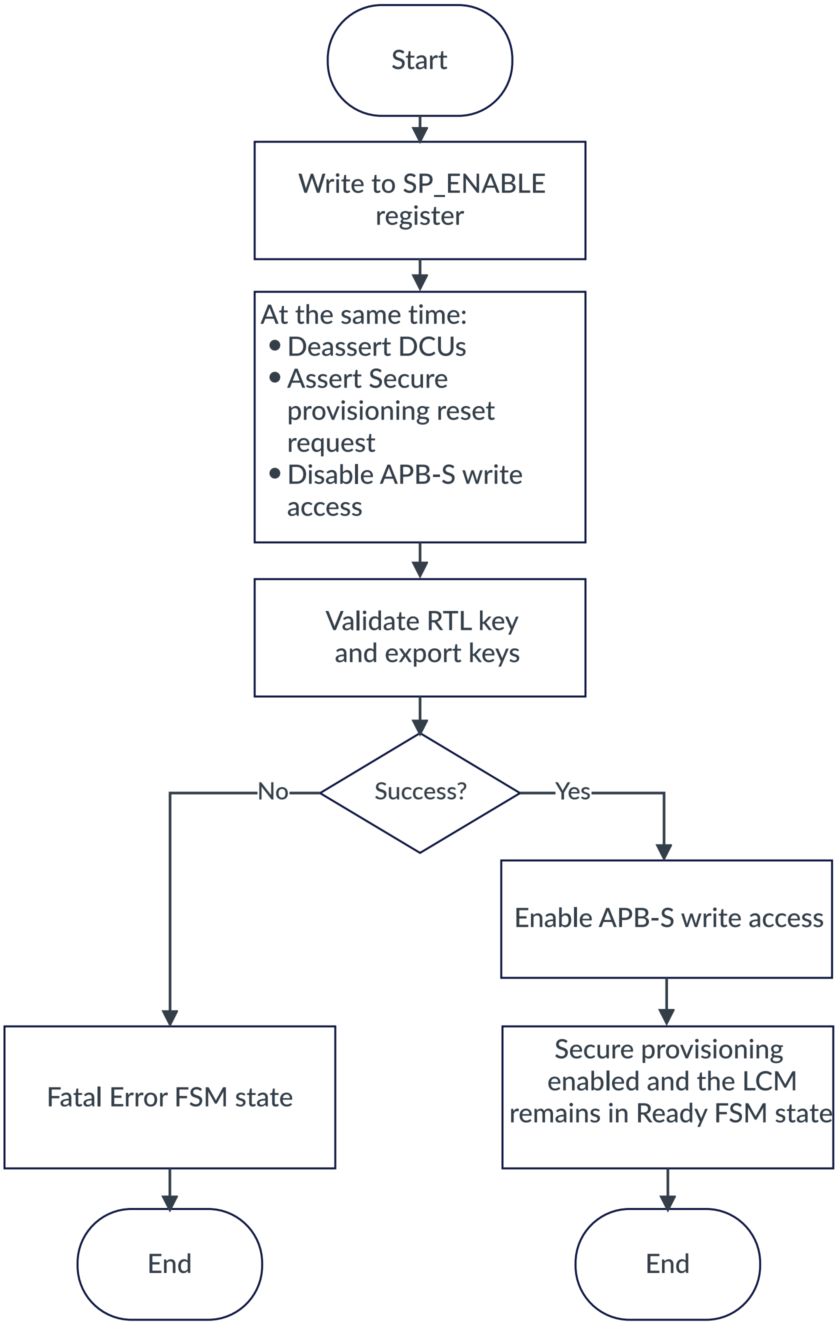
The sp_rst_req signal remains asserted when LCM Secure provisioning is enabled until the next reset cycle.

After a Secure provisioning reset request, Arm expects that all software returns to the most Trusted boot loader state, for example, the ROM bootloader. Following the reset request, the LCM re-exports all valid keys through the direct key APB3 manager interface.

The following figure shows the behavior of the LCM in Secure provisioning.

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Figure 4-8: LCM FSM Ready transition to Secure provisioning


Start
Write to SP_ENABLE register
At the same time:
• Deassert DCUs
• Assert Secure provisioning reset request
• Disable APB-S write access
Validate RTL key and export keys
Success?
No
Yes
Enable APB-S write access
Fatal Error FSM state
Secure provisioning enabled and the LCM remains in Ready FSM state
End
End

Related information

Secure provisioning programmer flow on page 105

RoT key export on page 39

Hardware keys on page 32

Direct key APB3 manager interface on page 56

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

4.3.3 Fatal Error FSM state

Fatal Error FSM state is triggered by the LCM FSM or can be initiated by software. When the LCM is in Fatal Error FSM satate, the fatal_err signal is asserted. A fatal error does not always trigger SW to transition the device to RMA LCS. A reset of the device may recover the LCM, if the fatal error has been caused by a transient fault.

Fatal Error FSM state can occur in all LCS states.

The LCM determines a Fatal Error FSM state, if one of the following conditions applies:

- The OTP read integrity check fails
- The internal register parity check fails
- The internal register duplication compare fails
- The lifecycle state is invalid
- The TP Mode state is invalid.

If the programmable FATAL_ERR_EN field in the FATAL_ERR register is set to FATAL_ERR_SET value or the LCM enters Fatal Error FSM state, the following actions are taken:

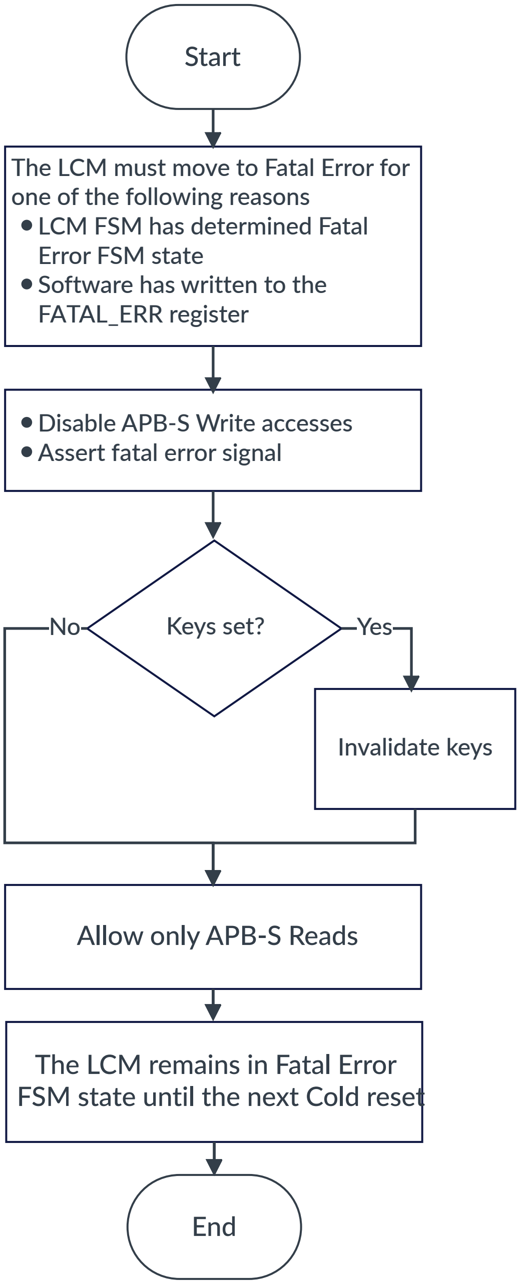
- All keys are invalidated
- The fatal_err signal is asserted
- APB-S write transactions return a bus error
- The LCM remains in Fatal Error FSM and it cannot change its FSM state until Cold reset

In Fatal Error FSM state, all LCM indications are treated as invalid. The lcs_is_valid signal indication remains asserted. Reads are permitted in Fatal Error FSM state.

The following figure shows the behaviors occuring when the LCM moves to Fatal Error FSM state.

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Figure 4-9: LCM Fatal Error FSM state


Start
The LCM must move to Fatal Error for one of the following reasons
● LCM FSM has determined Fatal Error FSM state
● Software has written to the FATAL_ERR register
● Disable APB-S Write accesses
● Assert fatal error signal
Keys set?
No
Yes
Invalidate keys
Allow only APB-S Reads
The LCM remains in Fatal Error FSM state until the next Cold reset
End

4.4 Hardware keys

The LCM hardware embedded secret keys are stored in the OTP memory, except for one key stored in RTL. Only the LCM can access the OTP and RTL keys until they are exported through

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

the direct key manager interface, where they are managed by a dedicated module, such as a *Key Management Unit* (KMU) or cryptographic engine.

On reset and after determining the lifecycle status:

- The LCM copies the OTP keys into the HW key shadow registers, according to the conditions defined in the Table 4-4: OTP memory map on page 42. The LCM includes HW key shadow registers for all OTP and RTL keys as defined in Table 4-3: HW Keys on page 33.
- All valid key registers are exported through the direct key APB3 manager interface before the lcs_is_valid signal is asserted, according to the conditions described in the Table 4-3: HW Keys on page 33. Invalid key registers should not be exported through the direct key APB3 manager interface.

When the FATAL_ERR register is set to FATAL_ERR_SET value by the programmer through the APB-S interface, all the LCM key registers are invalidated and zeroized.

When the fatal_err signal is asserted by the LCM, the OTP keys are not copied to the HW key shadow registers and if they have already been copied, the key registers are invalidated and zeroized.

The following assets must not be accessible through the APB-S interface or leak to any other blocks other than the LCM HW key shadow registers:

- HW keys in OTP memory
- HW key shadow registers
- RTL key (KRTL)

If the unmasked OTP key is all ones or all zeros:

- The LCM does not copy the key from OTP to the shadow register, so that if lcs_is_valid is asserted, the HW key shadow register is zeroized and invalidated.
- The LCM asserts the readable key zero count error indication register bit and the LCM HW key shadow register is zeroized and invalidated.

The LCM supports seven 256-bit HW keys:

Table 4-3: HW Keys

Key ID	Address offset (in bytes)	Key name	Key type
0	`0x00`	RTL Key (KRTL)	RTL
1	`0x20`	Hardware Unique Key (HUK)	OTP
2	`0x40`	Group Unique Key (GUK)	OTP
3	`0x60`	Chip vendor Provisioning key (KP_CM)	OTP
4	`0x80`	Chip vendor Code Encryption key (KCE_CM)	OTP
5	`0xA0`	OEM Provisioning key (KP_DM)	OTP
6	`0xC0`	OEM Code Encryption key (KCE_DM)	OTP

4.4.1 RTL key, KRTL

The RTL key (KRTL) is owned by the chip vendor and configured into the RTL. The KRTL is used for Secure provisioning during CM or DM LCS in PCI TP mode and is not available in other lifecycle states and TP modes.

Key ID

0

Key export address offset (in bytes)

`0x00`

Key type

RTL key

Guidance

To avoid security vulnerabilities, you should use the RTL key only for key derivation purposes.

Key is invalid and shadow register is zeroized, if any of the following conditions are true

- FATAL_ERR register is set to FATAL_ERR_SET value by the programmer
- LCM fatal_err signal asserted by LCM
- LCS = SE or RMA
- TP mode = TCI or Virgin

Export key through direct key APB interface, if all of the following conditions are true

- LCM not in Fatal Error FSM state
- Key is valid
- LCS = CM or DM
- TP mode = PCI

4.4.2 HW Unique Key, HUK

The HW Unique Key (HUK) is randomized on the device and provisioned into the OTP memory.

Key ID

1

Key export address offset (in bytes)

`0x20`

Key type

OTP key

Guidance

To avoid security vulnerabilities, you should use the HUK only for key derivation purposes. The HUK is revealed only to the system that contains the LCM and the OTP memory.

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

The HUK is invalid and its HW key shadow register is zeroized, if any of the following conditions are true

- FATAL_ERR register is set to FATAL_ERR_SET value by the programmer
- LCM fatal_err signal asserted by LCM
- Number of zeros mismatch
- All key bits are set
- All key bits are cleared
- LCS = CM or RMA
- TP mode = Virgin

Export key through direct key APB interface, if all of the following conditions are true

- LCM not in Fatal Error FSM state
- Key is valid
- LCS = DM or SE
- TP mode = TCI or PCI

4.4.3 Group Unique Key, GUK

The Group Unique Key (GUK) is owned by the chip vendor and provisioned into the OTP memory.

Key ID

2

Key export address offset (in bytes)

`0x40`

Key type

OTP key

Guidance

To avoid security vulnerabilities, you should use the GUK only for key derivation purposes. The GUK is known only to a group of systems containing the LCM and the OTP memory.

The GUK is invalid and its HW key shadow register is zeroized, if any of the following conditions are true

- FATAL_ERR register is set to FATAL_ERR_SET value by the programmer
- LCM fatal_err signal asserted by LCM
- Number of zeros mismatch
- All key bits are set
- All key bits are cleared
- LCS = CM or RMA
- TP mode = Virgin

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Export key through direct key APB interface, if all of the following conditions are true

- LCM not in Fatal Error FSM state
- Key is valid
- LCS = DM or SE
- TP mode = TCI or PCI

4.4.4 Chip vendor Provisioning key, KP_CM

The chip vendor Provisioning key (KP_CM) is owned by the chip vendor and provisioned into the OTP memory.

Key ID

3

Key export address offset (in bytes)

`0x60`

Key type

OTP key

Guidance

To avoid security vulnerabilities, you should use the KP_CM only for key derivation purposes.

The KP_CM is invalid and its HW key shadow register is zeroized, if any of the following conditions are true

- FATAL_ERR register is set to FATAL_ERR_SET value by the programmer
- LCM fatal_err signal asserted by LCM
- Number of zeros mismatch
- All key bits are set
- All key bits are cleared
- LCS = CM or RMA
- TP mode = Virgin

Export key through direct key APB interface, if all of the following conditions are true

- LCM not in Fatal Error FSM state
- Key is valid
- LCS = DM or SE
- TP mode = TCI or PCI

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

4.4.5 Chip vendor code encryption key, KCE_CM

The chip vendor code encryption key (KCE_CM) owned by the chip vendor and provisioned into the OTP memory.

Key ID

4

Key export address offset (in bytes)

`0x80`

Key type

OTP key

Guidance

To avoid security vulnerabilities, you should use the KCE_CM only for key derivation purposes.

The KCE_CM is invalid and its HW key shadow register is zeroized, if any of the following conditions are true

- FATAL_ERR register is set to FATAL_ERR_SET value by the programmer
- LCM fatal_err signal asserted by LCM
- Number of zeros mismatch
- All key bits are set
- All key bits are cleared
- LCS = CM or RMA
- TP mode = Virgin

Export key through direct key APB interface, if all of the following conditions are true

- LCM not in Fatal Error FSM state
- Key is valid
- LCS = DM or SE
- TP mode = TCI or PCI

4.4.6 OEM Provisioning Key, KP_DM

The OEM Provisioning Key (KP_DM) is owned by the OEM and provisioned into the OTP memory.

Key ID

5

Key export address offset (in bytes)

`0xA0`

Key type

OTP key

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Guidance

To avoid security vulnerabilities, you should use the KP_DM only for key derivation purposes.

The KP_DM is invalid and its HW key shadow register is zeroized, if any of the following conditions are true

- FATAL_ERR register is set to FATAL_ERR_SET value by the programmer
- LCM fatal_err signal asserted by LCM
- Number of zeros mismatch
- All key bits are set
- All key bits are cleared
- LCS = CM or DM or RMA
- TP mode = Virgin

Export key through direct key APB interface, if all of the following conditions are true

- LCM not in Fatal Error FSM state
- Key is valid
- LCS = SE
- TP mode = TCI or PCI

4.4.7 OEM code encryption key, KCE_DM

The OEM code encryption key (KCE_DM) is owned by the OEM and provisioned into the OTP memory.

Key ID

6

Key export address offset (in bytes)

`0xC0`

Key type

OTP key

Guidance

To avoid security vulnerabilities, you should use the KCE_DM only for key derivation purposes.

The KCE_DM is invalid and its HW key shadow register is zeroized, if any of the following conditions are true

- FATAL_ERR register is set to FATAL_ERR_SET value by the programmer
- LCM fatal_err signal asserted by LCM
- Number of zeros mismatch
- All key bits are set

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- All key bits are cleared
- LCS = CM or DM or RMA
- TP mode = Virgin

Export key through direct key APB interface, if all of the following conditions are true

- LCM not in Fatal Error FSM state
- Key is valid
- LCS = SE
- TP mode = TCI or PCI

4.4.8 RoT key export

The LCM includes an APB interface used to export the KRTL and the secret keys stored in the OTP memory to a system peripheral that can manage the keys, such as a *Key Management Unit* (KMU) or a cryptographic engine, which complies with the LCM key export design.

For more information about the KMU, see the *Arm® Key Management Unit (KMU) Specification*.

Arm expects that the interface is implemented as a point-to-point path between the LCM and the KMU. Neither the processor nor any other manager port with access to any interconnect has visibility to the transactions on this APB interface.

The LCM does not provide key locking or key access control capabilities for the programmer during boot time or runtime of the system firmware. Since the LCM cannot provide access control to the secret keys during boot time or runtime, Arm recommends that you connect a KMU to the LCM direct key APB3 manager interface. Arm assumes that the KMU manages usage policies for each key and provides key locking capabilities to the programmer.

4.4.9 OTP keys integrity protection

The LCM includes built-in integrity checking functionality for some of the OTP secret keys (key IDs one to six).

- HUK
- GUK
- KP_CM
- KCE_CM
- KP_DM
- KCE_DM

Integrity checking protection is required because OTP-programmed values could be forced by an attacker to transition from zero to one (non-programmed to programmed state), assuming that the initial state of the OTP memory is all zeros (non-programmed state).

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Integrity checking is achieved by counting the number of zeros in a key and programming and storing that zero count value alongside each key, so that the zero count value can be checked when the key is accessed in the future. The zero count calculation and the storage of the zero count value alongside each key is known as the integrity code. The zero-count integrity code value from the OTP memory is never exposed to the programmer, because it can leak information about the secret RTL mask value.

For all the OTP secret keys:

- If the number of zeros does not match the zero count value, the key zero count error indications are asserted and the relevant key register are invalidated and zeroized.
- The zero counting occurs only when the key is read from OTP memory (depending on lifecycle), according to the conditions defined in Table 4-3: HW Keys on page 33.

Since the keys are also protected using an RTL mask, the zero counting is performed on the actual values programmed in OTP, after the key bits were XORed with the RTL mask bits.

The RTL mask value is not used in TCI mode, so the zero counting operation is performed on the raw key values. This is reasonable because the key values for TCI mode are test keys rather than confidential production keys.

The masked version of the OTP key value is never exposed to the programmer through the APB-S interface (during provisioning phase).

An APB-S write transaction to the relevant OTP secret key memory addresses and triggers the LCM to calculate and write the integrity code. The LCM then verifies that the integrity codes were written correctly. The verification is done by comparing a precalculated integrity code value to the value returned by a read transaction issued by the LCM. If the comparison is successful, the LCM responds with all zeros data value to the programmer through the APB-S interface. If the comparison fails, the LCM responds with all-ones data value to the programmer through the APB-S interface.

If a key slot value is left blank (all-zeros), the integrity code overflows above the maximal value of 255 zero count. In this case, the LCM does not calculate and store the integrity code and the key is marked as invalid (not used). The comparison operation ignores the invalid key and the comparison is marked as successful.

The programmer interface provides an error status bit for each key to indicate the key comparison failure.

The number of zeroes in the HUK, GUK, KP_CM, KCE_CM, and ROTPK keys

The CM configuration flags (OTP Offset `0x00E4` and `0x00E8`) in the OTP memory indicate the number of zeros in the HUK, GUK, KP_CM, KCE_CM, and ROTPK keys. See Table 4-6: CM configuration 1 (OTP memory offset 00E4) on page 45 and Table 4-7: CM configuration 2 (OTP memory offset 00E8) on page 45 for the bit ranges assigned to each of these keys. These OTP fields are not accessible from the APB-S interface.

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Writes to the CM configuration flags

When the LCS and TP mode match the conditions described in Table 4-4: OTP memory map on page 42, the LCM replaces the value that is written to the field with the HW calculated number of zeros of the HUK OTP words after applying the relevant key XOR masks. If the LCS and TP mode do not match the conditions described in Table 4-4: OTP memory map on page 42, the LCM ignores the write and returns a bus error.

Reads to the CM configuration flags

When the LCS and TP mode match the conditions described in Table 4-4: OTP memory map on page 42, the LCM compares the read value from OTP with the HW calculated number of zeros of the respective OTP key words after applying the relevant key XOR masks, and number of zeros of the respective OTP key words after applying the relevant key XOR masks, and replaces the key bits number of zeros field in the read data with zeros if the comparison succeeds, and with all ones in case of the comparison fails. If the LCS and TP mode do not match the conditions described in Table 4-4: OTP memory map on page 42, the LCM returns a bus error and clears the data from the LCM HW key shadow registers.

4.5 OTP manager

The LCM includes an OTP manager that is responsible for controlling the OTP memory assets. The OTP manager uses the OTP APB3 manager interface that is connected to the OTP memory.

The OTP manager provides the following:

- Reading and integrity verification of the OTP keys (zero count). For more information, see OTP keys integrity protection.
- Providing double read and compare to specific OTP memory addresses. The OTP manager asserts the LCM fatal_err signal if the comparison of the two read results do not match. Only 32-bit read transactions are supported. The protected word addresses are described in Table 4-4: OTP memory map on page 42.
- Providing access control for programmer read and write requests depending on the current LCS and FSM state.

Access to and from OTP fields is described in the Table 4-4: OTP memory map on page 42.

- Any read access to a non-readable OTP word, as described in the Table 4-4: OTP memory map on page 42, returns zeros.
- Any write access to a non-writable OTP word, as described in the Table 4-4: OTP memory map on page 42, is ignored (Security requirement)
- If the DM_RMA_LOCK register bits are all set to ones, the DM RMA Flag must not be writable.

When lcs_is_valid signal is deasserted, accesses through the APB-S interface to the OTP are not served (writes are ignored and an APB error is signaled, the read data is zero).

To avoid OTP data leakage, due to resetting the LCM during an access to the OTP memory, the LCM generates a dummy read after reset. This dummy read cleans the OTP read data bus by reading a non-protected 32-bit word from any byte address in the user region (0x00F8 to OTP_SIZE_IN_BYTES) .

Related information

- OTP APB3 manager interface
- OTP APB3 manager interface signals

4.6 OTP memory map

The following table describes the OTP memory content, access permissions, OTP mask bit allocations, and the OTP double read integrity protection settings.

If the system requires access to the LCM OTP memory, the LCM must be powered on.

Table 4-4: OTP memory map

Byte address	Field description	Writable from APB-S if any of the conditions are true	Readable from APB-S if any of the conditions are true	HW LCM operation	OTP mask bits	Protected by double read and compare?
0x0000 - 0x001F	HUK	1. LCS = CM AND TP mode = TCI 2. LCS = CM AND TP mode = PCI AND Secure Prov Enabled 3. LCS = RMA	1. LCS = CM AND TP mode = TCI 2. LCS = CM AND TP mode = PCI AND Secure Prov Enabled	If (LCS = DM) or (LCM = SE), then copy to a HW key shadow register on reset and count zeros	255:0	Yes
0x0020 - 0x003F	GUK	1. LCS = CM AND TP mode = TCI 2. LCS = CM AND TP mode = PCI AND Secure Prov Enabled 3. LCS = RMA	1. LCS = CM AND TP mode = TCI 2. LCS = CM AND TP mode = PCI AND Secure Prov Enabled	If (LCS = DM) or (LCM = SE), then copy to a HW key shadow register on reset and count zeros	511:256	Yes
0x0040 - 0x005F	KP_CM	1. LCS = CM AND TP mode = TCI 2. LCS = CM AND TP mode = PCI AND Secure Prov Enabled 3. LCS = RMA	1. LCS = CM AND TP mode = TCI 2. LCS = CM AND TP mode = PCI AND Secure Prov Enabled	If (LCS = DM) or (LCM = SE), then copy to a HW key shadow register on reset and count zeros	767:512	Yes
0x0060 - 0x007F	KCE_CM	1. LCS = CM AND TP mode = TCI 2. LCS = CM AND TP mode = PCI AND Secure Prov Enabled 3. LCS = RMA	1. LCS = CM AND TP mode = TCI 2. LCS = CM AND TP mode = PCI AND Secure Prov Enabled	If (LCS = DM) or (LCM = SE), then copy to a HW key shadow register on reset and count zeros	1023:768	Yes

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Byte address	Field description	Writable from APB-S if any of the conditions are true	Readable from APB-S if any of the conditions are true	HW LCM operation	OTP mask bits	Protected by double read and compare?
0x0080 - 0x009F	KP_DM	1. LCS = DM AND TP mode = TCI 2. LCS = DM AND TP mode = PCI AND Secure Prov Enabled 3. LCS = RMA	1. LCS = DM AND TP mode = TCI 2. LCS = DM AND TP mode = PCI AND Secure Prov Enabled (In CM it should be all zeros)	If (LCM = SE), then copy to a HW key shadow register on reset and count zeros	1279:1024	Yes
0x00A0 - 0x00BF	KCE_DM	1. LCS = DM AND TP mode = TCI 2. LCS = DM AND TP mode = PCI AND Secure Prov Enabled 3. LCS = RMA	1. LCS = DM AND TP mode = TCI 2. LCS = DM AND TP mode = PCI AND Secure Prov Enabled (In CM it should be all zeros)	If (LCM = SE), then copy to a HW key shadow register on reset and count zeros	1535:1280	Yes
0x00C0 - 0x00DF	Root- of-Trust Public Key (ROTPK). This key is described in *Arm® Platform Security Architecture Trusted Base System Architecture for Arm®v6-M, Arm®v7-M and Arm®v8-M*.	1. LCS = CM 2. LCS = RMA	Yes	None	Not masked	Yes
0x00E0 - 0 x00E3	TP mode Configuration	1. LCS = CM AND TP mode = Virgin	Yes	Read to determine the TP Mode.	Not masked	Yes
0x00E4 - 0x00E7	CM Configuration 1	Not writable. If (LCS = CM AND TP mode = TCI) or (LCS = CM AND TP mode = PCI AND Secure Prov Enabled), then writing any value to this word will trigger writing the HW calculated number of zeros for the relevant keys.	Not readable. If (LCS = CM AND TP mode = TCI) or (LCS = CM AND TP mode = PCI AND Secure Prov Enabled), then reading from this word will return the number of zeros comparison result after comparing all relevant keys. Possible values per key zero count are: 0x00: Comparison successful. 0xFF: Comparison failed.	1. Read to determine the LCS. Must be zero for CM LCS. 2. Compared with the actual number of zero bits in the CM Configuration 1 keys	Not masked	Yes

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Byte address	Field description	Writable from APB-S if any of the conditions are true	Readable from APB-S if any of the conditions are true	HW LCM operation	OTP mask bits	Protected by double read and compare?
0x00E8 -0x00EB	CM Configuration 2	1. LCS = CM	Yes	1. Read to determine the LCS. Must be zero for CM LCS. 2. Read to determine GPPC register value and reflect in output signals.	Not masked	Yes
0x00EC - 0x00EF	DM Configuration	Not writable. If (LCS = DM AND TP mode = TCI) or (LCS = DM AND TP mode = PCI AND Secure Prov Enabled), then writing any value to this word will trigger writing the HW calculated number of zeros for the relevant keys.	Not readable. If (LCS = DM AND TP mode = TCI) or (LCS = DM AND TP mode = PCI AND Secure Prov Enabled), then reading from this word will return the number of zeros comparison result after comparing all relevant keys. Possible values per key zero count are: 0x00: Comparison successful. 0xFF: Comparison failed.	1. Read to determine the LCS. Must be zero for DM LCS. 2. Compared with the actual number of zero bits in the DM configuration keys.	Not masked	Yes
0x00F0 - 0x00F3	CM RMA OTP flag	1. LCS = CM 2. LCS = DM 3. LCS = SE	Yes	Read to determine the LCS. Must be set for RMA.	Not masked	Yes
0x00F4 - 0x00F7	DM RMA OTP flag	1. LCS = CM AND DM_RMA_LOCK register = 0 2. LCS = DM AND DM_RMA_LOCK register = 0 3. LCS = SE AND DM_RMA_LOCK register = 0	Yes	Read to determine the LCS. Must be set for RMA.	Not masked	Yes
0x00F8 - 0xEFFF	User defined data	Yes	Yes	None	Not masked	No

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

The following table describes the TP mode configuration.

Table 4-5: TP Mode configuration (OTP Offset `0x00E0`)

Bits	Description	Possible values
[31:0]	TP mode configuration	• `0x0000_0000` - Virgin TP mode • `0x0000_FFFF` - TCI TP mode • `0xFFFF_0000` - PCI TP mode • Any other value - Invalid TP Mode

The following table describes the first part of the CM configuration OTP memory fields.

Table 4-6: CM configuration 1 (OTP memory offset `0x00E4`)

Bits	Description	Possible values
[7:0]	Number of zero bits in HUK	Any value. All zeros or all ones invalidates the key.
[15:8]	Number of zero bits in GUK	Any value. All zeros or all ones invalidates the key.
[23:16]	Number of zero bits in KP_CM	Any value. All zeros or all ones invalidates the key.
[31:24]	Number of zero bits in KCE_CM	Any value. All zeros or all ones invalidates the key.

The following table describes the second part of the CM configuration OTP memory fields.

Table 4-7: CM configuration 2 (OTP memory offset `0x00E8`)

Bits	Description	Possible values
[7:0]	Number of zero bits in ROTPK	Any value. All zeros or all ones invalidates the key.
[23:8]	General Purpose Persistent Configuration (GPPC) flags	Any value
[31:24]	Reserved	Any value

The following table describes the DM configuration.

Table 4-8: DM configuration (OTP memory offset `0x00EC`)

Bits	Description	Possible values
[7:0]	Number of zero bits in KP_DM	Any value. All zeros or all ones invalidates the key.
[15:8]	Number of zero bits in KCE_DM	Any value. All zeros or all ones invalidates the key.
[31:16]	Reserved	Any value

The following table describes the CM RMA OTP flag.

Table 4-9: CM RMA OTP flag (OTP memory offset `0x00F0`)

Bits	Description	Possible values
[31:0]	CM RMA OTP flag	• `0x0000_0000` - CM RMA LCS disabled • `0xFFFF_FFFF` - CM RMA LCS enabled All other values are invalid LCS.

The following table describes the DM RMA OTP flag.

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Table 4-10: DM RMA OTP flag (OTP memory offset `0x00F4`)

Bits	Description	Possible values
[31:0]	DM RMA OTP flag	• `0x0000_0000` - DM RMA LCS disabled • `0xFFFF_FFFF` - DM RMA LCS enabled All other values are invalid LCS.

4.7 OTP memory masking

The LCM includes a permanent RTL mask of random bits to protect the assets stored in the OTP memory, for example, from an attack using a *Scanning Electron Microscope* (SEM). This permanent RTL mask is XORed with the relevant secret bits during reads and writes of the OTP memory, so that the values that are programmed to the OTP memory are masked from potential attackers.

The RTL of the LCM includes a modifiable 1536 bit OTP_MASK_VAL configuration parameter. The OTP_MASK_VAL parameter must not be accessible through the APB-S interface. For more information, see Configuration parameters. The OTP mask bits for the respective field addresses are described in Table 4-11: OTP masking bit assignment on page 46.

Reads and writes to the OTP memory are XORed with the OTP mask only if the LCM is in PCI TP Mode.

Masked writes

If the TP Mode is PCI and if LCS is not RMA, when writing to an OTP field, any value is XORed with the matching part of the OTP mask before it is written to the OTP memory. The matching OTP memory bits are defined in the “The relevant OTP_MASK_VAL bits” column in Table 4-11: OTP masking bit assignment on page 46.

Masked reads

If TP Mode is PCI and if LCS is not RMA, when a value is read from the OTP memory it is XORed the LCM, with the matching part of the OTP mask before being returned to the processor or LCM hardware. The matching OTP memory bits are defined in the “The relevant OTP_MASK_VAL bits” column in Table 4-11: OTP masking bit assignment on page 46.

When the LCS is RMA, a write to an allowed word results in setting the OTP word to all ones (after performing read-modify-write).

The following table describes the OTP mask bit assignment per OTP 32-bit words:

Table 4-11: OTP masking bit assignment

OTP address	Field description	The relevant OTP_MASK_VAL bits
`0x00`	HUK bits 31:0	31:0
`0x04`	HUK bits 63:32	63:32
`0x08`	HUK bits 95:64	95:64
`0x0C`	HUK bits 127:96	127:96
`0x10`	HUK bits 159:128	159:128

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.

OTP address	Field description	The relevant OTP_MASK_VAL bits
0x14	HUK bits 191:160	191:160
0x18	HUK bits 223:192	223:192
0x1C	HUK bits 255:224	255:224
0x20	GUK bits 31:0	287:256
0x24	GUK bits 63:32	319:288
0x28	GUK bits 95:64	351:320
0x2C	GUK bits 127:96	383:352
0x30	GUK bits 159:128	415:384
0x34	GUK bits 191:160	447:416
0x38	GUK bits 223:192	479:448
0x3C	GUK bits 255:224	511:480
0x40	KP_CM bits 31:0	543:512
0x44	KP_CM bits 63:32	575:544
0x48	KP_CM bits 95:64	607:576
0x4C	KP_CM bits 127:96	639:608
0x50	KP_CM bits 159:128	671:640
0x54	KP_CM bits 191:160	703:672
0x58	KP_CM bits 223:192	735:704
0x5C	KP_CM bits 255:224	767:736
0x60	KCE_CM bits 31:0	799:768
0x64	KCE_CM bits 63:32	831:800
0x68	KCE_CM bits 95:64	863:832
0x6C	KCE_CM bits 127:96	895:864
0x70	KCE_CM bits 159:128	927:896
0x74	KCE_CM bits 191:160	959:928
0x78	KCE_CM bits 223:192	991:960
0x7C	KCE_CM bits 255:224	1023:992
0x80	KP_DM bits 31:0	1055:1024
0x84	KP_DM bits 63:32	1087:1056
0x88	KP_DM bits 95:64	1119:1088
0x8C	KP_DM bits 127:96	1151:1120
0x90	KP_DM bits 159:128	1183:1152
0x94	KP_DM bits 191:160	1215:1184
0x98	KP_DM bits 223:192	1247:1216
0x9C	KP_DM bits 255:224	1279:1248
0xA0	KCE_DM bits 31:0	1311:1280
0xA4	KCE_DM bits 63:32	1343:1312
0xA8	KCE_DM bits 95:64	1375:1344
0xAC	KCE_DM bits 127:96	1407:1376
0xB0	KCE_DM bits 159:128	1439:1408

OTP address	Field description	The relevant OTP_MASK_VAL bits
0xB4	KCE_DM bits 191:160	1471:1440
0xB8	KCE_DM bits 223:192	1503:1472
0xBC	KCE_DM bits 255:224	1535:1504

4.8 DRBG module

The LCM includes a *Deterministic Random Bit Generator* (DRBG) that is used to generate masking values and random delay values. The DRBG is implemented using a *Linear-Feedback Shift Register* (LFSR) unit. The masking values are used during key export transaction through the LCM Direct Key APB3 manager interface.

The LFSR generates a 16-bit pseudorandom mask for each 32-bit data word written on the direct key APB3 manager interface. The LFSR generates a random delay of between zero and seven cycles for the OTP read integrity module. The masks provide protection against leakage of key material by XORing the exported key value with the mask. The mask changes every cycle to increase the difficulty of side-channel attacks.

The random delay values are used during OTP double read operations so that a random delay is used between two sequential read transactions from the OTP fields.

The DRBG uses an external seed value for the initial state. The seed is set through the LFSR seed signal interface.

The module is seeded from a 64-bit wide signal interface external source which is sampled after reset. The LCM performs the following steps after reset:

1. The LCM waits for the LFSR to be ready.
2. The LFSR waits for lfsr_valid input signal to be set and then it samples the lfsr_data input signals and becomes ready.
3. The LCM performs the first access to the OTP memory.

4.9 Debug Control Unit

The LCM includes a mechanism for controlling debugging capabilities and HW feature enablement.

The *Debug Control Unit* (DCU) includes a set of HW masks that allows DCU signals to control bitwise settings and locking capabilities. This control is accomplished by a combination of programmer registers for enablement of the DCU signals, programmer registers for locking the DCU signals, and RTL DCU configuration parameters for each LCS.

- DCU registers for enablement or locking of the DCU signals
    - 128-bit DCU register (DCU_EN<0-3>). For more information, see DCU_EN.
    - 128-bit DCU lock register (DCU_LOCK<0-3>). For more information, see DCU_LOCK.

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- 128-bit DCU_DISABLE_MASK parameter value register (DCU_DISABLE_MASK<0-3>). For more information, see DCU_SP_DISABLE_MASK.

Caution

Do not use the DAUTHCTRL register, as defined by the *Arm® v8-M Architecture Reference Manual*, for debug control. Using the DAUTHCTRL register overrides the Secure debug mechanism.

- RTL DCU configuration parameters. For more information about these configuration parameters, see Configuration parameters
  - DCU Reset Default configuration parameter for each LCS and for the TCI and PCI TP modes. There are eight 128-bit DEFAULT_DCU_VAL_LCS configuration parameters. Their default values are described in Configuration parameters.
  - DCU Permanent Disable Mask configuration parameter for each LCS and for the TCI and PCI TP modes. There are eight 128-bit DCU_PERMANENT_DISABLE_MASK_VAL_LCS configuration parameters.
  - DCU Secure Provisioning Disable Mask configuration parameter that is applied during Secure Provisioning. Set the 128-bit DCU_SP_DISABLE_MASK_VAL parameter with the DCU bits that should be disabled during Secure provisioning. For more information about the logic used to deassert the DCU during Secure provisioning, see Figure 4-14: The logic for deasserting the DCU signal on entry to Secure provisioning on page 53.

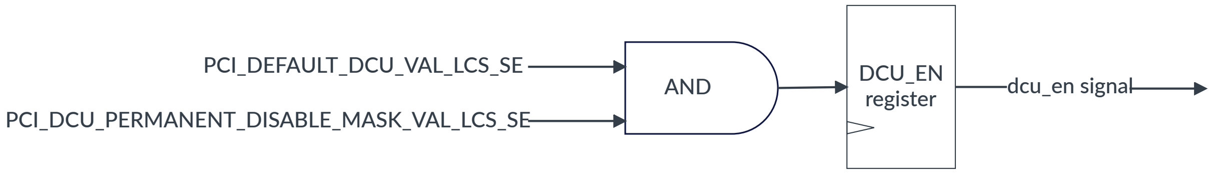
The DCU signals are a set of 128-bit control flags. The specific interpretation of these flags depends on the system integration. The partner may also choose to connect one or more of these outputs to debug control signals or to feature enablement control signals that enable a specific SoC feature.

4.9.1 DCU in LCM Reset FSM state

During LCM FSM reset state, reads and writes to LCM registers are disabled. The configuration parameters used to calculate the DCU_EN register depend on whether lcs_is_valid is asserted:

- After reset and until lcs_is_valid is asserted, the DCU_EN register and the DCU signals value is calculated by ANDing the PCI_DEFAULT_DCU_VAL_LCS_SE and the PCI_DCU_PERMANENT_DISABLE_MASK_VAL_LCS_SE configuration parameter values. The following figure shows the logic for calculating the DCU_EN register value in LCM Reset FSM state after reset and until lcs_is_valid is asserted.

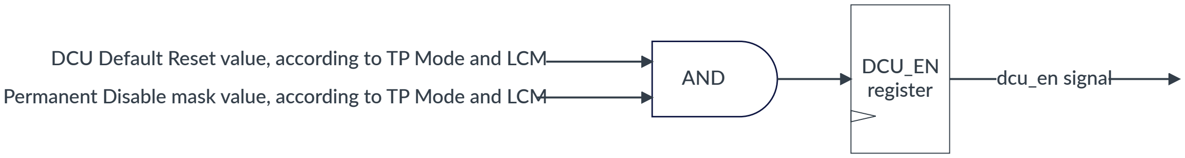
Figure 4-10: DCU_EN register value in LCM Reset FSM state after reset and until lcs_is_valid is asserted


PCI_DEFAULT_DCU_VAL_LCS_SE
PCI_DCU_PERMANENT_DISABLE_MASK_VAL_LCS_SE
AND
DCU_EN register
dcu_en signal

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.

- After the LCS is calculated and lcs_is_valid is asserted, the DCU signals and the DCU_EN register are set to the Default Reset DCU configuration parameter value (according to the LCS and TP Mode) ANDed with the Permanent Disable Mask value (according to the LCS and TP mode). The following figure shows the logic for calclulating the DCU_EN register value in LCM Reset FSM state after the LCS is calculated.

Figure 4-11: DCU_EN register value in LCM Reset FSM state after the LCS is calculated


DCU Default Reset value, according to TP Mode and LCM
Permanent Disable mask value, according to TP Mode and LCM
AND
DCU_EN register
dcu_en signal

Related information

- Configuration parameters

4.9.2 DCU in LCM Ready FSM state

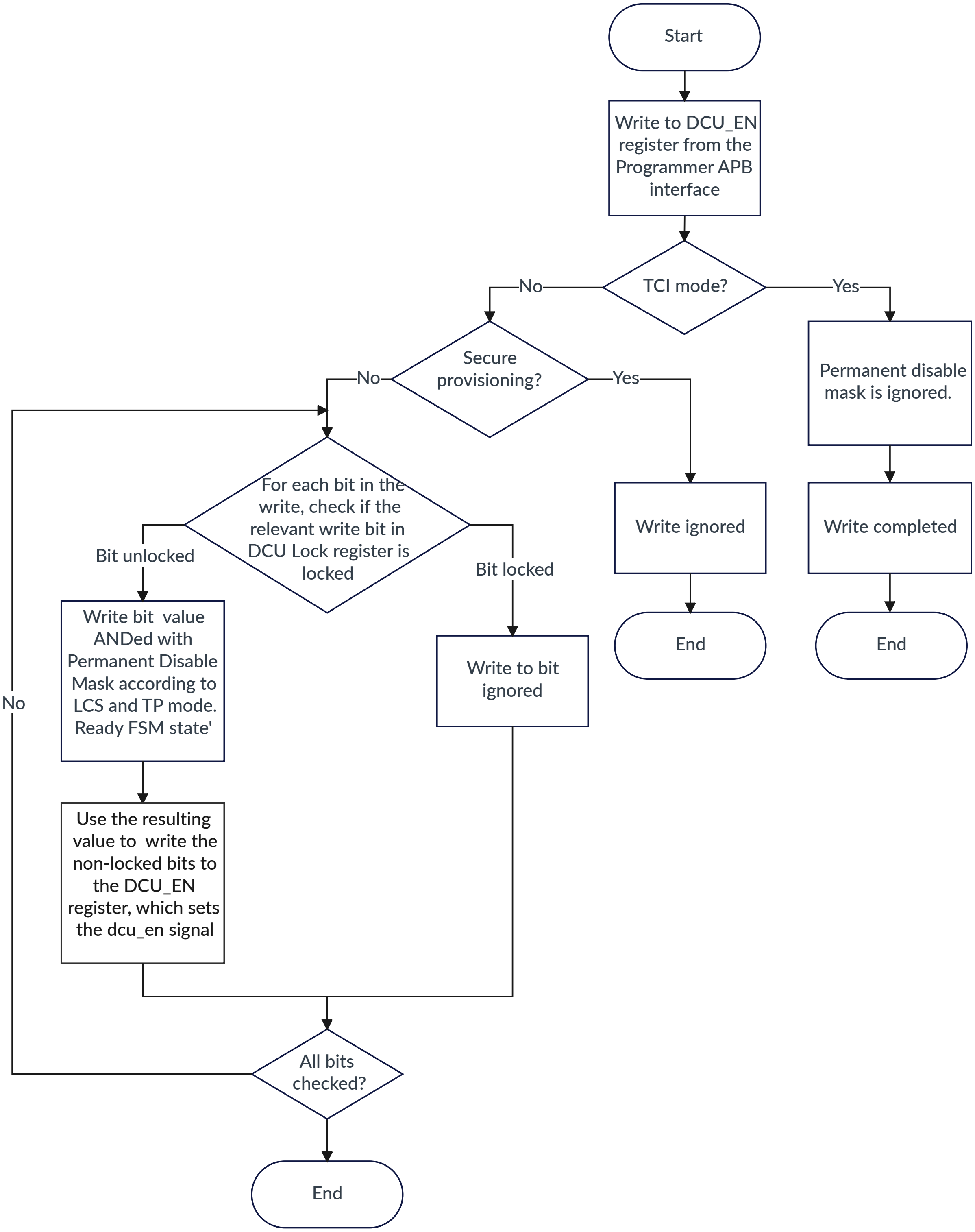
The DCU flow behaviour depends on the TP mode of the LCM and whether the LCM is in Secure provisioning.

- In PCI TP mode, Virgin, or Invalid TP mode the DCU signals and the DCU_EN register are ANDed with the relevant Permanent Disable Mask parameter, based on the LCS, so that the relevant DCU signals are forced to zero. See Figure 4-12: DCU in LCM Ready FSM state on page 51
- The DCU signals are deasserted when the LCM enters Secure provisioning. For more information, see Figure 4-14: The logic for deasserting the DCU signal on entry to Secure provisioning on page 53. Writes to DCU_EN are not allowed when LCM is in Secure Provisioning, unless the LCM is in TCI TP mode.
- In TCI TP mode, the DCU signals and the DCU_EN register are not ANDed with the DCU_SP_DISABLE_MASK parameter, even if LCM Secure provisioning mode is enabled.

Figure 4-12: DCU in LCM Ready FSM state on page 51 shows a high-level flow for setting the DCU_EN register value and the associated dcu_en.

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

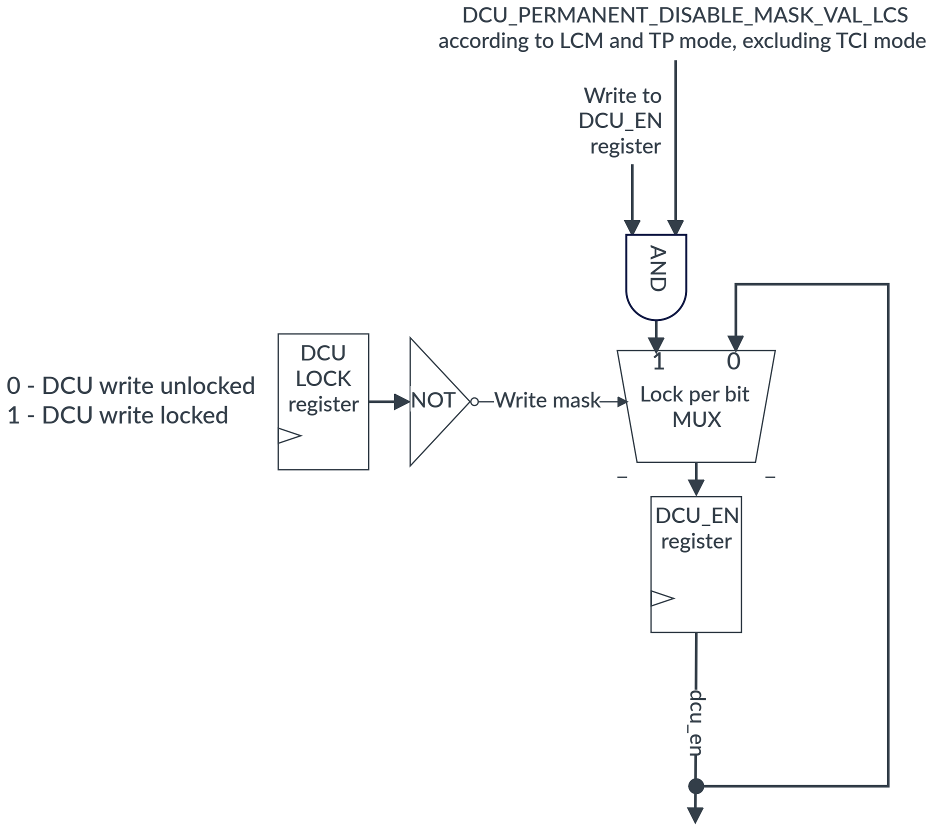
Figure 4-12: DCU in LCM Ready FSM state


Start
Write to DCU_EN register from the Programmer APB interface
TCI mode?
No
Yes
Permanent disable mask is ignored.
Write completed
End
Secure provisioning?
No
Yes
Write ignored
End
For each bit in the write, check if the relevant write bit in DCU Lock register is locked
Bit unlocked
Bit locked
Write bit value ANDed with Permanent Disable Mask according to LCS and TP mode. Ready FSM state'
Write to bit ignored
Use the resulting value to write the non-locked bits to the DCU_EN register, which sets the dcu_en signal
All bits checked?
No
End

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Figure 4-13: The logic for calculating the DCU signal in LCM Ready FSM state on page 52 defines the logic for the calculation of the DCU signal for Ready FSM state for all TP modes apart from TCI. If any bit of the DCU_LOCK register is set, then the feedback loop is selected for that bit, and the DCU_EN register bit value remains unchanged. Otherwise, the DCU signal for the unlocked bit is changed to reflect the write value ANDed with the Permanent Disable Mask parameter according to the LCS and TP mode.

Figure 4-13: The logic for calculating the DCU signal in LCM Ready FSM state



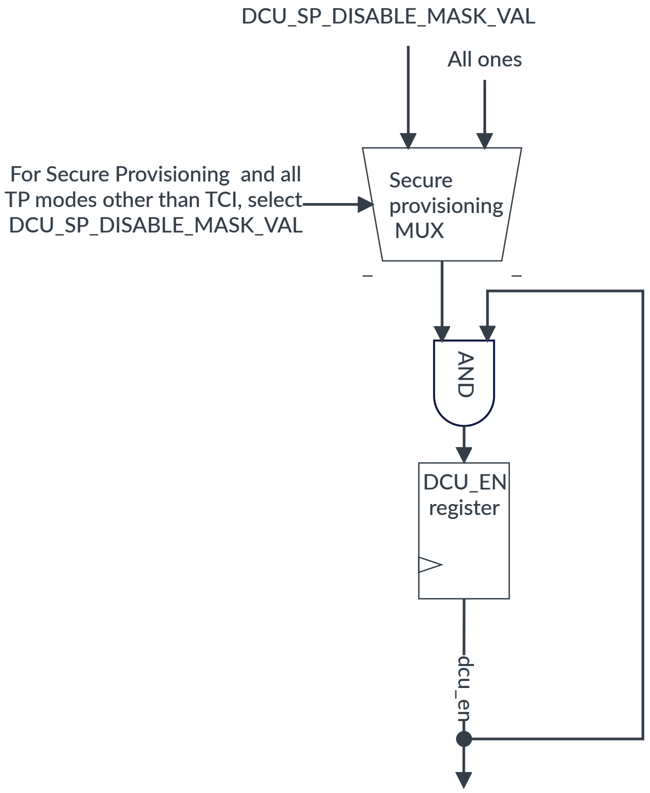
Unless the LCM is in TCI TP mode, the DCU signals are deasserted when the LCM enters Secure provisioning.

Figure 4-14: The logic for deasserting the DCU signal on entry to Secure provisioning on page 53 shows the logic for deasserting the DCU when the LCM Ready FSM state moves into Secure provisioning.

If the LCM is in TCI TP mode, the Secure provisioning MUX selects all ones, and DCU_EN value remains unchanged. In all other TP modes when the LCM is in Secure provisioning, the MUX selects the DCU_SP_DISABLE_MASK_VAL and forces the relevant DCU signal bits to zero.

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Figure 4-14: The logic for deasserting the DCU signal on entry to Secure provisioning


DCU_SP_DISABLE_MASK_VAL
All ones
For Secure Provisioning and all TP modes other than TCI, select DCU_SP_DISABLE_MASK_VAL
Secure provisioning MUX
AND
DCU_EN register
dcu_en

Related information

- Configuration parameters

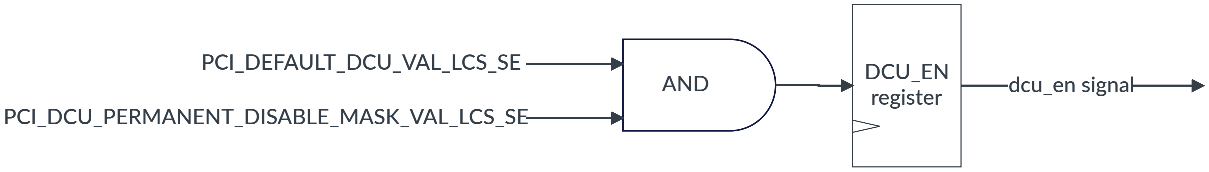
4.9.3 DCU in LCM Fatal Error FSM state

If LCM fatal_err is asserted, the DCU signals and the DCU_EN register value is calculated by ANDing the PCI_DEFAULT_DCU_VAL_LCS_SE and the PCI_DCU_PERMANENT_DISABLE_MASK_VAL_LCS_SE configuration parameter values.

In Fatal Error FSM state, writes to the LCM are disabled.

The following figure defines the logic for the calculation of the DCU signal for Fatal Error FSM state.

Figure 4-15: DCU in LCM Fatal Error FSM state


PCI_DEFAULT_DCU_VAL_LCS_SE
PCI_DCU_PERMANENT_DISABLE_MASK_VAL_LCS_SE
AND
DCU_EN register
dcu_en signal

4.10 General Purpose Persistent Configuration

The LCM includes a *General Purpose Persistent Configuration* (GPPC) register and corresponding OTP memory field that can optionally be used by the chip vendor. The read-only register reflects the value that was programmed in the corresponding OTP memory field. The GPPC allows the chip vendor to control various system features through the OTP persistent configuration.

The GPPC register is readable and non-writable through the APB-S interface. The GPPC register value is reflected in the GPPC output signal, which is described in LCM generic signals.

Related information

- GPPC, General Purpose Persistent Configuration register
- LCM generic signals
- Table 4-7: CM configuration 2 (OTP memory offset 00E8) on page 45

4.11 Register integrity protection

The LCM protects all registers against fault injection attacks by including either register duplication or register parity mitigations. A mismatch error is signalled by the fatal_err signal.

4.12 Hardware interfaces

The LCM includes several HW interfaces that integrate with the system.

4.12.1 LCM APB3 subordinate interface

The LCM includes an APB3 subordinate interface which you can use to control the LCM operations and read the LCS.

The interface provides the following control and status features:

- Direct control of the DCU bits, Secure provisioning, RMA lifecycle transition locking, and fatal error indication.

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- Visibility of the LCM LCS, configuration, and TP mode indications.
- Direct access to the OTP memory. Access to some OTP memory entries is disabled for writes or reads depending on the LCS of the LCM.

The APB3 subordinate interface does not filter transactions according to their security attributes (Secure/Non-secure) or their privilege level attributes. Therefore, Arm recommends that you control the LCM APB3 interface using an Arm TrustZone® *Peripheral Protection Controller* (PPC) for that purpose. For more details regarding APB4 PPCs (which are APB3-compatible), see *Arm® CoreLink™ SIE-200 System IP for Embedded Technical Reference Manual*.

The LCM ignores the APB address (apbs_paddr) bits 1:0. The APB subordinate interface data bus width is 32 bits.

The default value of the APB subordinate pready signal is 0.

The memory map and registers exposed in the APB3 subordinate interface are detailed in the Programmer model.

The APB3 manager accesses the entire OTP memory directly through the APB-S interface (the APB-S Addresses `0x1000-0xFFFF` are directly mapped to the OTP). Accesses are allowed or prevented according to Table 4-4: OTP memory map on page 42.

The following accesses return an APB error.

- Accesses to the reserved regions
- Accesses to the unused OTP area above OTP_SIZE_IN_BYTE limit
- All APB-S writes before the lcs_is_valid is asserted. These writes are not performed.

If an APB error is signaled, the APB read data bus is zeroed. All APB-S write transactions targeting the OTP memory region must be paused, when Secure provisioning is enabled (SP_ENABLE) and until Secure provisioning is successfully entered (all keys are successfully exported).

Related information

APB3 subordinate interface signals on page 65

4.12.2 OTP APB3 manager interface

The LCM includes an APB3 manager interface to the OTP memory.

The OTP APB manager interface data bus width is 32 bits.

The OTP APB manager interface address bus width is eight to 16 bits. The OTP APB manager interface address bus, nvm_apb_paddr, bits 1:0 are always zero.

All accesses (initiated through APB-S) that are mapped to an OTP address larger than the defined OTP size (according to the OTP_SIZE_IN_BYTES parameter) are blocked by HW, and an APB error is returned to the APB-S interface.

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

If a read error is signaled through the OTP APB-S interface by the nvm_apb_pslverr signal, a matching APB error is returned to the APB-S interface.

All writes to the OTP memory (from LCM/APB-S) are translated by the HW to a read-modify-write operation, so that if a bit is set in the OTP memory, the HW zeroizes the relevant bit in the write-data bus. When the write has been translated to a read-modify-write transaction, an APB error is signalled through the OTP APB-S interface by the nvm_apb_pslverr signal. Depending on whether the error applies to the read or the write part of the read-modify-write transaction, either a matching read or write APB error is returned to the APB-S interface.

Related information

OTP manager on page 41

OTP APB3 manager interface signals on page 65

4.12.3 Direct key APB3 manager interface

The LCM includes a direct key APB3 manager interface that exports the LCM keys to an external peripheral for key management. Arm recommends that this interface is connected through a point-to-point path from the LCM to the *Key Management Unit* (KMU) or a cryptographic engine.

The LCM exports all valid HW keys as defined in Table 4-3: HW Keys on page 33 through the interface, after reading the keys from the OTP memory and determining a valid LCS. The LCM exports the keys depending on the lifecycle and TP mode, described in Table 4-3: HW Keys on page 33. The LCM only exports a key if the LCS allows it and if the key has passed the integrity check. The LCM sets lcs_is_valid signal only if all the interface write transactions have completed successfully.

The interface data bus width is 32 bits. The interface supports only 32-bit write transactions.

The interface address bus width is 10 bits. For the interface address bus, apb_paddr, bits 1:0 are always zero.

The LCM exports all valid HW keys depending on Secure provisioning, as defined in Table 4-3: HW Keys on page 33. The direct key interface includes a 16-bit mask sent over the APB user bits that is XORed with each half of the 32-bit word data. The direct key interface masking is enabled or disabled according to the DISABLE_DIRECT_KEY_APB_MASKING configuration parameter. If an export write transaction failed, the LCM invalidates the HW key shadow registers.

Related information

Direct key APB3 manager signals on page 66

4.12.4 LCM generic signal interface

The LCM includes a group of generic signals that indicate to the system the current LCS, the debug control indications, fatal error indications, Secure provisioning reset request, and general-purpose customer-defined control signals.

The lifecycle signals include:

- The lcs and tp_mode signals, which indicate the current lifecycle and the TP mode
- The lcs_is_valid signal, which indicates the validity of the lifecycle
- The fatal_err signal allows the system to know that the LCM was disabled due to a fatal error caused internally by LCM or set by the programmer. The fatal error indication takes precedence over all lifecycle states, meaning that once the fatal error is set, LCM is not functional anymore for the current power cycle. Depending on your system design, the fatal_err signal can be used to reset the system.
- The DCU signals, dcu_en and dcu_lock, which represent the current debug control state and locking state. The dcu_en signal is connected to the debug system control logic. The dcu_lock signal is for information only.
- The Secure provisioning reset request signal, sp_rst_req signal, which is described in Secure provisioning.
- The GPPC signal, gppc, which is described in General Purpose Persistent Configuration.

All the LCM signals are set to their default value one cycle after reset.

1. After reset, the LCM reads the OTP fields and calculates the lifecycle state. Until the lifecycle is determined, LCM always indicates the most restrictive LCS (SE LCS).
2. At the end of this process and after the keys have been exported through the direct key APB interface, LCM indicates the actual LCS and sets the lcs_is_valid signal.
    - If lcs_is_valid signal is not set after exit from reset, the APB-S interface is stalled and software polling is not possible for any of the LCM registers. Software read is only successful if the lcs_is_valid signal is set.
    - If the LCS computation results with an invalid value, the lcs_is_valid signal is not set and instead the fatal_err signal is set.

Related information

- LCM generic signals

4.12.5 LFSR seed signal interface

The LCM includes an interface to set a seed value that is used by the LFSR module. The seed is typically generated by a true random number generator from an entropy source. Arm expects the seed to be different on every power cycle. However, the seed is not required to be of high quality.

Related information

LFSR seed signals on page 67

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

4.12.6 Clock interface

The clock interface includes a single core_clk clock source, which is used for all LCM modules.

Related information

Clock logic and reset signals on page 68

4.12.7 Reset interface

The reset interface consists of the rst_n signal and the LCM asynchronous reset (active-LOW, asynchronous to core_clk). The SoC generates this reset.

Related information

Clock logic and reset signals on page 68

4.12.8 Q-Channel interface

The LCM includes a standard Q-Channel interface to control the LCM clock. The LCM can accept or deny the system request. If there is an active APB-S transaction, a Direct Key APB transaction or a OTP-APB transaction, the Q-Channel clk_qactive active is HIGH.

Related information

Q-Channel interface signals on page 68

4.12.9 Scan interface

The LCM can be tested using common scan-insertion techniques.

Bypass and exclude the OTP memory from scan. The OTP memory array used for the secrets keys, key IDs 1 to 6, (HUK, GUK, KP_CM, KCE_CM, KP_DM and KCE_DM) and their integrity values (stored in the CM and DM configuration fields) should be bypassed and excluded from scan and may only be tested using a pass/fail on-chip BIST.

The persistent memory should not be reset during scan but should be masked.

The flops generating the internal reset, which clears the LCM registers on dftscanmode input transitions, must be excluded from DFT scan.

To prevent the scan logic from breaching implementation security, the LCM expects the system integrator to provide the following signals:

dftscanmode

The system is expected to provide a dftscanmode input to LCM. This signal should be asserted before scanning is initiated and remain asserted whenever scanning of the LCM

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

is possible. Besides the traditional usage of bypassing clocks and reset logic, assertion and deassertion of dftscanmode generates an internal reset cycle that clears all LCM registers. When dftscanmode input is asserted:

- The OTP Mask is invalidated.
- The DCU registers are tied to zero, both when reading their value through the APB-S, and the DCU signal interface.
- The KRTL is masked

dftse

The system is expected to provide a dftse input to LCM. The system should prevent the use of the scan chains or shift enable while dftscanmode is not asserted by ANDing the dftse input into LCM with the dftscanmode.

dftcgen

The system is expected to provide a dftcgen input to LCM.

Assert dftscanmode whenever scanning of the LCM is possible.

Related information

Scan signals on page 68

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

5. Configuration parameters

All the LCM modifiable parameters are defined in the top-level module and propagated into lower-level modules. This allows you to change the parameters in the instantiation of LCM, rather than changing the include files.

The following tables describe the LCM configuration parameters.

Table 5-1: LCM customer-modifiable configuration parameters on page 60 lists the parameter values that must be defined by the customer.

Table 5-1: LCM customer-modifiable configuration parameters

RTL configuration parameter name	Size in bits	Allowed values	Default value	Description
KRTL_VAL	256	All values	NA	This value must be generated using a true random number generator and kept confidential in the design.
OTP_MASK_VAL	1536	All values	NA	This value must be generated using a true random number generator and kept confidential in the design.

Table 5-2: LCM implementation-defined configuration parameters on page 60 lists the parameter values that are defined during the implementation of the LCM.

Table 5-2: LCM implementation-defined configuration parameters

RTL configuration parameter name	Size in bits	Allowed values	Default value	Description
DISABLE_DIRECT_KEY_APB_MASKING	1	0,1	0	When set to one, the Direct Key APB masking feature is disabled.
DCU_SP_DISABLE_MASK_VAL	128	All values	All bits set	The Secure provisioning disable mask of the 128-bit DCU signals. Clearing a bit in the mask forces the relevant DCU signal to zero when LCM is in Secure provisioning (SP_ENABLE=1).
OTP_ADDR_WIDTH	32	8 to 16	14	The size of the OTP address bus width. OTP_ADDR_WIDTH is equal to OTP_SIZE_IN_BYTES rounded to the next bigger or equal power of 2. The supported values are: • 8 (for 256 Byte OTP size) • 9,10,11,12,13,14,15,16 (for 64 KByte OTP size)
OTP_SIZE_IN_BYTES	32	256, 512, 1024, 2048, 4096, 8192, 16384, 32768, 61440	16384	The size of the OTP region accessible from the LCM APB-S interface. The maximum size is 60KB.

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

RTL configuration parameter name	Size in bits	Allowed values	Default value	Description
PCI_DEFAULT_DCU_VAL_LCS_CM	128	All values	All bits set	The default reset value of the 128-bit DCU signals when LCS=CM and TP Mode=PCI. This parameter is loaded to the 128-bit DCU_EN register after the LCM determines the lifecycle state.
PCI_DEFAULT_DCU_VAL_LCS_DM	128	All values	All bits set	The default reset value of the 128-bit DCU signals when LCS=DM and TP Mode=PCI. This parameter is loaded to the 128-bit DCU_EN register after the LCM determines the lifecycle state.
PCI_DEFAULT_DCU_VAL_LCS_SE	128	All values	All bits clear	The default reset value of the 128-bit DCU signals when LCS=SE and TP Mode=PCI. This parameter is loaded to the 128-bit DCU_EN register after the LCM determines the lifecycle state.
PCI_DEFAULT_DCU_VAL_LCS_RMA	128	All values	All bits set	The default reset value of the 128-bit DCU signals when LCS=RMA and TP Mode=PCI. This parameter is loaded to the 128-bit DCU_EN register after the LCM determines the lifecycle state.
PCI_DCU_PERMANENT_DISABLE_MASK_VAL_LCS_CM	128	All values	All bits set	The permanent disable mask of the 128-bit DCU signals when LCS=CM and TP Mode=PCI. Clearing a bit in the mask forces the relevant DCU signal to zero.
PCI_DCU_PERMANENT_DISABLE_MASK_VAL_LCS_DM	128	All values	All bits set	The permanent disable mask of the 128-bit DCU signals when LCS=DM and TP Mode=PCI. Clearing a bit in the mask forces the relevant DCU signal to zero.
PCI_DCU_PERMANENT_DISABLE_MASK_VAL_LCS_SE	128	All values	All bits set	The permanent disable mask of the 128-bit DCU signals when LCS=SE and TP Mode=PCI. Clearing a bit in the mask forces the relevant DCU signal to zero.
PCI_DCU_PERMANENT_DISABLE_MASK_VAL_LCS_RMA	128	All values	All bits set	The permanent disable mask of the 128-bit DCU signals when LCS=RMA and TP Mode=PCI. Clearing a bit in the mask forces the relevant DCU signal to zero.
TCI_DEFAULT_DCU_VAL_LCS_CM	128	All values	All bits set	The default reset value of the 128-bit DCU signals when LCS=CM and TP Mode=TCI. This parameter is loaded to the 128-bit DCU_EN register after the LCM determines the lifecycle state.
TCI_DEFAULT_DCU_VAL_LCS_DM	128	All values	All bits set	The default reset value of the 128-bit DCU signals when LCS=DM and TP Mode=TCI. This parameter is loaded to the 128-bit DCU_EN register after the LCM determines the lifecycle state.
TCI_DEFAULT_DCU_VAL_LCS_SE	128	All values	All bits set	The default reset value of the 128-bit DCU signals when LCS=SE and TP Mode=TCI. This parameter is loaded to the 128-bit DCU_EN register after the LCM determines the lifecycle state.

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

RTL configuration parameter name	Size in bits	Allowed values	Default value	Description
TCI_DEFAULT_DCU_VAL_LCS_RMA	128	All values	All bits set	The default reset value of the 128-bit DCU signals when LCS=RMA and TP Mode=TCI. This parameter is loaded to the 128-bit DCU_EN register after the LCM determines the lifecycle state.
TCI_DCU_PERMANENT_DISABLE_MASK_VAL_LCS_CM	128	All values	All bits set	The permanent disable mask of the 128-bit DCU signals when LCS=CM and TP Mode=TCI. Clearing a bit in the mask forces the relevant DCU signal to zero.
TCI_DCU_PERMANENT_DISABLE_MASK_VAL_LCS_DM	128	All values	All bits set	The permanent disable mask of the 128-bit DCU signals when LCS=DM and TP Mode=TCI. Clearing a bit in the mask forces the relevant DCU signal to zero.
TCI_DCU_PERMANENT_DISABLE_MASK_VAL_LCS_SE	128	All values	All bits set	The permanent disable mask of the 128-bit DCU signals when LCS=SE and TP Mode=TCI. Clearing a bit in the mask forces the relevant DCU signal to zero.
TCI_DCU_PERMANENT_DISABLE_MASK_VAL_LCS_RMA	128	All values	All bits set	The permanent disable mask of the 128-bit DCU signals when LCS=RMA and TP Mode=TCI. Clearing a bit in the mask forces the relevant DCU signal to zero.

6. Integration requirements

To meet the specification, the integration of the LCM must meet some integration requirements.

Key management

Arm expects that the LCM is integrated with a *Key Management Unit* (KMU) or a cryptographic engine that can receive the LCM exported keys and manage their usage.

OTP array

Integrate an OTP array in the system in which all bits are programmed to zeros for a newly manufactured device. If the OTP array bits are all ones, you must include inverters so that LCM reads all zeros from a blank device OTP memory.

High-quality random values

Generate the secret OTP_MASK_VAL and KRTL_VAL parameter values using a *Hardware Security Module* (HSM) that produces high-quality random values. All the bits must be random and not be duplicated.

Caution

Using a low-quality random value is a security risk.

Signal integration

The system software reads the LCS_VALUE register. This register should stall until the lcs_is_valid signal is asserted on boot. If the read is successful, the lcs_is_valid signal indication must be asserted. Otherwise the fatal_err signal is asserted. We recommend that you configure the fatal_err signal to trigger a system reset. The LCS signals should be valid only if the lcs_is_valid signal is asserted.

When the sp_rst_req signal, linked to the SP_ENABLE register, is asserted, you must ensure that this signal causes the reset of the processors, DMAs, and KMU, but it does not reset the device memory. At least one memory region should keep the signed images after reset.

Power

Arm expects that the LCM is integrated as part of the always-on power domain to retain the system LCS throughout hibernate or partial Power down modes. The LCM provides a group of signals to the system that indicate the current LCS and control the debug domains.

DCU

The DCU reset default values for each LCS are described in Configuration parameters. Set the DCU_SP_DISABLE_MASK_VAL parameter with the DCU bits that should be disabled during the Secure provisioning.

Related information

- Scan signals
- Scan interface

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

7. Signal descriptions

This section describes the external signals of the LCM.

Unless explicitly stated, all signals are clocked by core_clk.

7.1 APB3 subordinate interface signals

The APB3 Subordinate interface connects the LCM to the SoC system bus. APB3 Subordinate signals adhere to the APB3 protocol. For more information, see *AMBA® APB Protocol Specification*.

Signal definitions

Table 7-1: APB3 subordinate interface signals

Signal	Direction	Description	Width	Reset default
lcm_apb_paddr	Input	Address. Bits [1:0] are ignored.	15	NA
lcm_apb_psel	Input	Select	1	NA
lcm_apb_penable	Input	Enable	1	NA
lcm_apb_pwrite	Input	Write indication	1	NA
lcm_apb_pwdata	Input	Write data	32	NA
lcm_apb_pready	Output	Ready. Note: Might be deasserted for up to 512 cycles. If there is a direct OTP access, or a software reset request that awaits an OTP access to complete, there might be even more cycles. The number of cycles depends on the number of wait-states when accessing the chosen OTP memory.	1	0
lcm_apb_prdata	Output	Read data	32	0
lcm_apb_pslverr	Output	Error	1	0

Related information

LCM APB3 subordinate interface on page 54

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

7.2 OTP APB3 manager interface signals

LCM uses the OTP APB3 signals to interface with an external *One-Time-Programmable* (OTP) memory. The following table shows the OTP APB3 signals. For more information about APB3 signals, see *AMBA® APB Protocol Specification*.

Signal definitions

Table 7-2: OTP APB3 manager interface signals

Signal	Direction	Description	Width	Reset default
nvm_apb_paddr	Output	Address. Addresses are in bytes. Addresses are always aligned to 32-bit words, so that the 2 least significant bits of the address are zero. Equals 14 when OTP memory size is 16KB.	Log_2 (OTP_SIZE_IN_BYTES)	0
nvm_apb_psel	Output	Select	1	0
nvm_apb_penable	Output	penable signal	1	0
nvm_apb_pwrite	Output	pwrite signal	1	0
nvm_apb_pwdata	Output	Write data	32	0
nvm_apb_pready	Input	pready signal	1	NA
nvm_apb_prdata	Input	Read data	32	NA
nvm_apb_pslverr	Input	Error	1	NA

Related information

OTP manager on page 41
OTP APB3 manager interface on page 55

7.3 Direct key APB3 manager signals

The direct key APB3 manager signals are described in the following table.

Signal definitions

Table 7-3: Direct key APB3 manager interface signals

Signal	Direction	Description	Width	Reset default
dk_apb_paddr	Output	Address. Bits [1:0] are ignored.	10	NA
dk_apb_psel	Output	Select	1	NA
dk_apb_penable	Output	Enable	1	NA
dk_apb_pwrite	Output	Write indication	1	NA
dk_apb_pwdata	Output	Write data	32	NA
dk_apb_pready	Input	Ready	1	0
dk_apb_prdata	Input	Read data	32	0
dk_apb_pslverr	Input	Error	1	0

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Signal	Direction	Description	Width	Reset default
dk_apb_user	Output	Mask value for the 32-bit apbs_pwdata signals. The mask is applied twice as a XOR on both halves of the 32-bit pwdata signals. The existence of this mask depends on the value of the DISABLE_DIRECT_KEY_APB_MASKING configuration parameter.	16	0

Related information

Direct key APB3 manager interface on page 56

7.4 LCM generic signals

The following table describes the generic LCM signals that indicate the lifecycle, debug control, Secure provisioning, GPPC, and error indication.

Signal definitions

Table 7-4: LCM generic signals

Signal	Direction	Description	Width	Reset default
lcs	Output	The LCS indication value. • `0b000` - CM LCS • `0b001` - DM LCS • `0b101` - SE LCS • `0b111` - RMA LCS Any other values are reserved.	3	`0b101`
lcs_is_valid	Output	An indication that the LCS is valid.	1	0
dcu_en	Output	The value of the DCU_EN register.	128	0
dcu_lock	Output	The value of the DCU_LOCK register.	128	0
gppc	Output	The value of the GPPC register, which also reflects the OTP GPPC field.	16	0
tp_mode	Output	The TP mode indication value. • `0b00` - Virgin TP mode • `0b01` - TCI TP mode • `0b10` - PCI TP mode • `0b11` - Invalid TP mode	2	`0b10`
sp_rst_req	Output	A level indication that Secure provisioning is enabled. A Warm reset sequence initiates in the system (for example, processors, DMAs, and KMU).	1	0
fatal_err	Output	Indicates a parity error, an OTP memory integrity error, or another LCM fatal error. You can use the fatal_err indication from the LCM in an SoC, for example by connecting to HW Alarm Logic. With right system design, the fatal_err signal can reset the system. LCS indications must be ignored if fatal_err is asserted.	1	0

7.5 LFSR seed signals

The LFSR seed signals are described in the following table.

Signal definitions

Table 7-5: LFSR seed signals

Signal	Direction	Description	Width	Reset default
lfsr_data	Input	The LFSR seed data bits	64	NA
lfsr_valid	Input	Valid	1	NA

7.6 Clock logic and reset signals

The clock and reset interfaces contain LCM clock and reset signals.

Signal definitions

Table 7-6: Clock logic and reset signals

Signal	Direction	Description	Width	Reset default
rst_n	Input	Asynchronous global reset (negative polarity)	1	NA
core_clk	Input	The main LCM system clock.	1	NA

7.7 Q-Channel interface signals

The Q-Channel signals contain the clock control signals.

Signal definitions

Table 7-7: Q-Channel signals

Signal	Direction	Description	Width	Reset default
clk_qreqn	Input	Indicates when the clock controller issues a clock quiescence entry or exit request to the LCM.	1	NA
clk_qacceptn	Output	Indicates when the LCM accepts the clock quiescence request.	1	0
clk_qdeny	Output	Indicates when the LCM denies the clock quiescence request.	1	0
clk_qactive	Output	Indicates when the LCM is active and when it requests exiting clock quiescence.	1	1

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

7.8 Scan signals

The LCM supports scan cell insertion methodology for SoC DFT strategy. DFT control signals provide high coverage for test strategy for LCM design.

Signal definitions

Table 7-8: Scan signals

Signal	Direction	Description	Width	Reset default
dftscanmode	Input	Control signal indicating that scan mode is in place.	1	NA
dftse	Input	The scan-enable input.	1	NA
dftcgen	Input	Architectural clock gate override for the clock domain.	1	NA

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

8. Programmer model

This chapter provides general information about the LCM register properties.

The following information applies to LCM registers:

- The base address is not fixed and can be different for any particular system implementation. The offset of each register from the base address is fixed.
- Do not attempt to access reserved or unused address locations. Attempting to access these locations can results in unexpected behavior.
- Unless otherwise stated in the accompanying text:
    - Do not modify undefined register bits.
    - Ignore undefined register bits on reads.
    - All register bits are reset to the reset value specified in the register summary table of each block.

Access type is described as follows:

RW

Read/write

RO

Read-only

WO

Write-only

WI

Write ignore

RW1S

Read. Each bit is write once per reset cycle

The LCM registers are accessed using the APB3 subordinate interface.

The programmers model contains regions for control, status indications, and OTP programming.

The registers are grouped into regions. Each region includes a table listing its registers in offset order and subsections describing each register.

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

8.1 Register memory map regions

The LCM programmer memory map includes the following regions.

Table 8-1: LCM programmer memory map regions

Region	Address Range	Region Size (Byte)	Description
LCS register summary	0x0000 - 0x00FF	256	Lifecycle state control registers
DCU register summary	0x0100 - 0x01FF	256	DCU control registers
Reserved	0x0200 - 0x0EFF	3,328	Reserved
PID and CID register summary	0x0F00 - 0x0FFF	256	*Peripheral ID* (PID) *Component ID* (CID) and registers
OTP register region	0x1000 - 0xFFFF	61,440	OTP direct access. Some addresses are not readable and/or writable.

8.2 LCS register summary

The following table shows the registers in the LCS region, in offset order from the base LCM memory address.

Table 8-2: LCS register summary

Offset	Name	Type	Reset	Width	Description
0x0000	LCS_VALUE	RO	0xEEEE_A5A5	32-bit	LCS indication
0x0004	KEY_ERR	RO	0x3F	32-bit	Zero count error status of the OTP keys
0x0008	TP_MODE	RO	0x2222_AA55	32-bit	TP mode
0x000C	FATAL_ERR	RW	0x0	32-bit	LCM Fatal Error FSM state enable
0x0010	DM_RMA_LOCK	RW	0x0	32-bit	DM RMA OTP flag lock enable
0x0014	SP_ENABLE	RW	0x0	32-bit	Secure provisioning enable
0x0018	OTP_ADDR_WIDTH	RO	Configuration dependent	32-bit	Value of the OTP_ADDR_WIDTH configuration parameter that defines the integrated OTP address width.
0x001C	OTP_SIZE_IN_BYTES	RO	Configuration dependent	32-bit	Value of the OTP_SIZE_IN_BYTES configuration parameter that defines the integrated OTP size in bytes
0x0020	GPPC	RO	0x0	32-bit	General Purpose Persistent Configuration (GPPC) value

8.2.1 LCS_VALUE, Lifecycle State Indication register

The register holds the value which indicates the current LCS.

Configurations

This register is available in all configurations.

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Attributes

The register characteristics are:

Width

32

Functional group

LCS register summary

Address offset

`0x0000`

Type

RO

Reset value

`0xEEEE_A5A5`

Bit descriptions

The following table shows the register bit descriptions.

Table 8-3: LCS_VALUE register bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	LCS_VALUE	LCM LCS indication • Chip Manufacturing (CM): `0xCCCC_3C3C` • Device Manufacturing (DM): `0xDDDD_5A5A` • Secure Enabled (SE): `0xEEEE_A5A5` • Return Merchandise Authorization (RMA): `0xFFFF_C3C3` • Invalid: `0xDEAD_BEEF` All other values are reserved.	RO	`0xEEEE_A5A5` (SE)

Related information

Lifecycle states on page 19

8.2.2 KEY_ERR, Error Status of the OTP Keys register

The register holds the zero-count error status of the OTP keys.

Configurations

This register is available in all configurations.

Attributes

The register characteristics are:

Width

32

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.

Functional group

LCS register summary

Address offset

`0x0004`

Type

RO

Reset value

`0x3F`

Bit descriptions

The following table shows the register bit descriptions.

Table 8-4: KEY_ERR register bit descriptions

Bits	Name	Description	Type	Reset
[31:6]	RESERVED	Reserved	RO	0x0
[5]	KCE_DM_ERR	Value of one indicates that the number of zeroes in the relevant loaded key is not equal to the value set in the OTP configuration words, or the key is not valid. The key is not valid if it is all zeros or all ones.	RO	0x1
[4]	KP_DM_ERR	Value of one indicates that the number of zeroes in the relevant loaded key is not equal to the value set in the OTP configuration words, or the key is not valid. The key is not valid if it is all zeros or all ones.	RO	0x1
[3]	KCE_CM_ERR	Value of one indicates that the number of zeroes in the relevant loaded key is not equal to the value set in the OTP configuration words, or the key is not valid. The key is not valid if it is all zeros or all ones.	RO	0x1
[2]	KP_CM_ERR	Value of one indicates that the number of zeroes in the relevant loaded key is not equal to the value set in the OTP configuration words, or the key is not valid. The key is not valid if it is all zeros or all ones.	RO	0x1
[1]	GUK_ERR	Value of one indicates that the number of zeroes in the relevant loaded key is not equal to the value set in the OTP configuration words, or the key is not valid. The key is not valid if it is all zeros or all ones.	RO	0x1
[0]	HUK_ERR	Value of one indicates that the number of zeroes in the relevant loaded key is not equal to the value set in the OTP configuration words, or the key is not valid. The key is not valid if it is all zeros or all ones.	RO	0x1

8.2.3 TP_MODE, Test or Production Mode register

The register holds the value which indicates the current TP mode of the device.

Configurations

This register is available in all configurations.

Attributes

The register characteristics are:

Width

32

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Functional group

LCS register summary

Address offset

`0x0008`

Type

RO

Reset value

`0x2222_AA55`

Bit descriptions

The following table shows the register bit descriptions.

Table 8-5: TP_MODE register bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	TP_MODE	TP mode • Virgin: `0x0000_33CC` • TCI: `0x1111_55AA` • PCI: `0x2222_AA55` • Invalid: `0xDEAD_BEEF` All other values are reserved.	RO	`0x2222_AA55` (PCI)

Related information

TP mode on page 17

8.2.4 FATAL_ERR, the Fatal Error FSM State register

The register controls the Fatal Error FSM state. The register is write-once per LCM reset cycle.

Writing the FATAL_ERR_SET value (`0xFA7A_1EEE`) to this register enables LCM Fatal Error FSM state.

When Fatal Error FSM state is enabled:

- The HW keys are invalidated and zeroized.
- The fatal_err signal is asserted.
- LCM is disabled.

Configurations

This register is available in all configurations.

Attributes

The register characteristics are:

Width

32

Functional group

LCS register summary

Address offset

0x000C

Type

RW

Reset value

0x0

Bit descriptions

The following table shows the register bit descriptions.

Table 8-6: FATAL_ERR register bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	FATAL_ERR_EN	LCM Fatal Error FSM state enable • Writing the value 0xFA7A_1EEE to this register enables LCM Fatal Error mode. Writes of other values are ignored. The 'write-once per LCM reset cycle' is not consumed until the 0xFA7A_1EEE value is written. • This register reads 0x0000_0000, when LCM Fatal Error is disabled. • This register reads 0xFFFF_FFFF, when LCM Fatal Error is enabled. The default value of this field is zero. You can only set the FATAL_ERR field once per LCM reset cycle.	RW	0x0

Related information

Fatal Error FSM state on page 31

8.2.5 DM_RMA_LOCK, DM RMA Flag Lock Enable register

The register controls the DM RMA Lock mode. The register is write-once per LCM reset cycle.

Configurations

This register is available in all configurations.

Attributes

The register characteristics are:

Width

32

Functional group

LCS register summary

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Address offset

0x0010

Type

RW

Reset value

0x0

Bit descriptions

The following table shows the register bit descriptions.

Table 8-7: DM_RMA_LOCK register bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	DM_RMA_LOCK	DM RMA flag lock enable • Writing the DM_RMA_LOCK_SET value, `0xDDAF_10CE`, to this register locks the DM RMA Flag in the OTP for write access. Writes of other values are ignored. The 'write-once per LCM reset cycle' is not consumed until the `0xDDAF_10CE` value is written. • This register reads `0x0000_0000`, when DM RMA flag is unlocked. • This register reads `0xFFFF_FFFF`, when DM RMA flag is locked. The DM_RMA_LOCK field default value is zero.	RW	0x0

8.2.6 SP_ENABLE, Secure Provisioning Enable register

The register controls Secure provisioning. The register is write-once per LCM reset cycle.

This register is writable only if the LCS is CM or DM. In all other cases, register writes are ignored.

After SP_ENABLE is set, and until the Secure provisioning is successfully entered (all keys are successfully exported), all APB-S write transactions are ignored and an APB error is signalled.

When Secure provisioning is enabled:

- If the TP mode is PCI and the LCS is CM or DM, KRTL is enabled.
- Each cleared bit of DCU_SP_DISABLE_MASK configuration parameter forces the respective DCU bit to zero.
- sp_rst_req signal is asserted.
- The valid HW keys are exported through the direct key APB3 manager interface.

Configurations

This register is available in all configurations.

Attributes

The register characteristics are:

Width

32

Functional group

LCS register summary

Address offset

`0x000C`

Type

RW

Reset value

`0x0`

Bit descriptions

The following table shows the register bit descriptions.

Table 8-8: SP_ENABLE register bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	SP_ENABLE	Secure provisioning enable • Writing the value `0x5EC1_0E1E` to this register enables the LCM Secure rrovisioning. Writes of other values are ignored. The 'write-once per LCM reset cycle' is not consumed until the `0x5EC1_0E1E` value is written. • This register will read `0x0000_0000`, when Secure provisioning is disabled. • This register will read `0xFFFF_FFFF`, after Secure Provisioning is enabled and keys relevant to the LCS have been exported. It holds this value until the next reset cycle. The default value of this field is zero.	RW	`0x0`

Related information

Secure provisioning on page 28

8.2.7 OTP_ADDR_WIDTH, OTP Address Width Parameter Value register

The register holds the value of the OTP_ADDR_WIDTH configuration parameter.

All other values are reserved.

Configurations

This register is available in all configurations.

Attributes

The register characteristics are:

Width

32

Functional group

LCS register summary

Address offset

`0x0018`

Type

RO

Reset value

Configuration-dependent

Bit descriptions

The following table shows the register bit descriptions.

Table 8-9: OTP_ADDR_WIDTH register bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	OTP_ADDR_WIDTH	Holds the value of the OTP_ADDR_WIDTH configuration parameter, which defines the integrated OTP address width in words. The supported values are: • 8 (for 256 Byte OTP size) • 9, 10, 11, 12, 13, 14, 15, 16 (for 64 KByte OTP size)	RO	Configuration dependent

Related information

Configuration parameters on page 60

8.2.8 OTP_SIZE_IN_BYTES, OTP Memory Size in Bytes register

The register holds the value of the OTP_SIZE_IN_BYTES configuration parameter, which must be a multiple of four. OTP_ADDR_WIDTH is equal to OTP_SIZE_IN_BYTES rounded to the next bigger or equal power of two. All other values are reserved.

Configurations

This register is available in all configurations.

Attributes

The register characteristics are:

Width

32

Functional group

LCS register summary

Address offset

`0x001C`

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Type

RO

Reset value

Configuration-dependent, defined by the OTP_SIZE_IN_BYTES configuration parameter

Bit descriptions

The following table shows the register bit descriptions.

Table 8-10: OTP_SIZE_IN_BYTES register bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	OTP_SIZE_IN_BYTES	Holds the value of the OTP_SIZE_IN_BYTES parameter, which defines the integrated OTP size, in bytes.	RO	Configuration dependent

Related information

Configuration parameters on page 60

8.2.9 GPPC, General Purpose Persistent Configuration register

The register holds the *General Purpose Persistent Configuration* (GPPC) value.

Configurations

This register is available in all configurations.

Attributes

The register characteristics are:

Width

32

Functional group

LCS register summary

Address offset

`0x0020`

Type

RO

Reset value

`0x0`

Bit descriptions

The following table shows the register bit descriptions.

Table 8-11: GPPC register bit descriptions

Bits	Name	Description	Type	Reset
[31:16]	RESERVED	Reserved	RO	0x0

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Bits	Name	Description	Type	Reset
[15:0]	GPPC	The 16-bit GPPC value as reflected in the OTP GPPC field and in the LCM generic signal interface.	RO	0x0

Related information

General Purpose Persistent Configuration on page 54
LCM generic signal interface on page 56
LCM generic signals on page 67

8.3 DCU register summary

In DFT scan mode all DCU_EN and DCU_LOCK registers read all zeros.

When the DCU_EN registers are read, the returned value reflects the actual value of the DCU signals. The actual value of the DCU signals and the registers is affected by both DCU_DISABLE_MASK and DCU_SP_DISABLE_MASK masking operations, and the DCU locking filter.

The following table shows the registers in the DCU region, in offset order from the base LCM memory address.

Table 8-12: DCU register summary

Offset	Name	Type	Reset	Width	Description
0x0100	DCU_EN0	RW	0x0	32-bit	Bits 31:0 of the 128-bit DCU enable register (DCU_EN)
0x0104	DCU_EN1	RW	0x0	32-bit	Bits 63:32 of the 128-bit DCU enable register (DCU_EN)
0x0108	DCU_EN2	RW	0x0	32-bit	Bits 95:64 of the 128-bit DCU enable register (DCU_EN)
0x010C	DCU_EN3	RW	0x0	32-bit	Bits 127:96 of the 128-bit DCU enable register (DCU_EN)
0x0110	DCU_LOCK0	RW1S	0x0	32-bit	Bits 31:0 of the 128-bit DCU lock register (DCU_LOCK)
0x0114	DCU_LOCK1	RW1S	0x0	32-bit	Bits 63:32 of the 128-bit DCU lock register (DCU_LOCK)
0x0118	DCU_LOCK2	RW1S	0x0	32-bit	Bits 95:64 of the 128-bit DCU lock register (DCU_LOCK)
0x011C	DCU_LOCK3	RW1S	0x0	32-bit	Bits 127:96 of the 128-bit DCU lock register (DCU_LOCK)
0x0120	DCU_SP_DISABLE_MASK0	RO	0xFFFF_FFFF	32-bit	Bits 31:0 of the 128-bit DCU_SP_DISABLE_MASK_VAL parameter value
0x0124	DCU_SP_DISABLE_MASK1	RO	0xFFFF_FFFF	32-bit	Bits 63:32 of the 128-bit DCU_SP_DISABLE_MASK_VAL parameter value.
0x0128	DCU_SP_DISABLE_MASK2	RO	0xFFFF_FFFF	32-bit	Bits 95:64 of the 128-bit DCU_SP_DISABLE_MASK_VAL parameter value.
0x012C	DCU_SP_DISABLE_MASK3	RO	0xFFFF_FFFF	32-bit	Bits 127:96 of the 128-bit DCU_SP_DISABLE_MASK_VAL parameter value.
0x0130	DCU_DISABLE_MASK0	RO	0xFFFF_FFFF	32-bit	Bits 31:0 of the 128-bit PERMANENT_DISABLE_MASK parameter value, according to the current LCS.
0x0134	DCU_DISABLE_MASK1	RO	0xFFFF_FFFF	32-bit	Bits 63:32 of the 128-bit PERMANENT_DISABLE_MASK parameter value, according to the current LCS.
0x0138	DCU_DISABLE_MASK2	RO	0xFFFF_FFFF	32-bit	Bits 95:64 of the 128-bit PERMANENT_DISABLE_MASK parameter value, according to the current LCS.

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Offset	Name	Type	Reset	Width	Description
0x013C	DCU_DISABLE_MASK3	RO	0xFFFF_FFFF	32-bit	Bits 127:96 of the 128-bit PERMANENT_DISABLE_MASK parameter value, according to the current LCS.

8.3.1 DCU_EN0, DCU Enable register 0

The register holds bits 31:0 of the 128-bit DCU enable register (DCU_EN). The DCU_EN register is used by software to control the DCU signals. The DCU_EN register is accessible through the APB-S interface for reads and writes in all lifecycle states.

Configurations

This register is available in all configurations.

Attributes

The register characteristics are:

Width

32

Functional group

DCU register summary

Address offset

0x0018

Type

RW

Reset value

0x0

Bit descriptions

The following table shows the register bit descriptions.

Table 8-13: DCU_EN0 register bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	DCU_EN0	Bits [31:0] of the 128-bit DCU enable register (DCU_EN)	RW	0x0

Related information

DCU_EN1 on page 81
DCU_EN2 on page 82
DCU_EN3 on page 83

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

8.3.2 DCU_EN1, DCU Enable register 1

The register holds bits 63:32 of the 128-bit DCU enable register (DCU_EN). The DCU_EN register is accessible for reads and writes in all lifecycle states.

Configurations

This register is available in all configurations.

Attributes

The register characteristics are:

Width

32

Functional group

DCU register summary

Address offset

`0x0104`

Type

RW

Reset value

`0x0`

Bit descriptions

The following table shows the register bit descriptions.

Table 8-14: DCU_EN1 register bit descriptions

Bits	Field Name	Field Description	Reset	Type
[31:0]	DCU_EN1	Bits [63:32] of the 128-bit DCU enable register (DCU_EN)	`0x0`	RW

Related information

DCU_EN0 on page 81
DCU_EN2 on page 82
DCU_EN3 on page 83

8.3.3 DCU_EN2, DCU Enable register 2

The register holds bits 95:64 of the 128-bit DCU enable register (DCU_EN). The DCU_EN register is accessible for reads and writes in all lifecycle states.

Configurations

This register is available in all configurations.

Attributes

The register characteristics are:

Width

32

Functional group

DCU register summary

Address offset

`0x0108`

Type

RW

Reset value

`0x0`

Bit descriptions

The following table shows the register bit descriptions.

Table 8-15: DCU_EN2 register bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	DCU_EN2	Bits [95:64] of the 128-bit DCU enable register (DCU_EN)	RW	`0x0`

Related information

DCU_EN0 on page 81
DCU_EN1 on page 81
DCU_EN3 on page 83

8.3.4 DCU_EN3 DCU Enable register 3

The register holds bits 127:96 of the 128-bit DCU enable register (DCU_EN). The DCU_EN register is accessible for reads and writes in all lifecycle states.

Configurations

This register is available in all configurations.

Attributes

The register characteristics are:

Width

32

Functional group

DCU register summary

Address offset

`0x010C`

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Type

RW

Reset value

`0x0`

Bit descriptions

The following table shows the register bit descriptions.

Table 8-16: DCU_EN3 register bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	DCU_EN3	Bits [127:96] of the 128-bit DCU enable register (DCU_EN)	RW	`0x0`

Related information

DCU_EN0 on page 81
DCU_EN1 on page 81
DCU_EN2 on page 82

8.3.5 DCU_LOCK0, DCU Lock register 0

The register holds bits 31:0 of the 128-bit DCU lock register (DCU_LOCK). The DCU_LOCK register is set by the software and enables the corresponding DCU signals at bit resolution.

Any DCU_EN bit is not writable when the corresponding DCU_LOCK bit is set. The DCU_LOCK register default value is zero (unlocked). The DCU_LOCK register is accessible for reads and writes through the APB-S interface. Each of the 128 DCU_LOCK register bits is set-once per LCM reset cycle.

Configurations

This register is available in all configurations.

Attributes

The register characteristics are:

Width

32

Functional group

DCU register summary

Address offset

`0x0110`

Type

RW1S

Reset value

`0x0`

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Bit descriptions

The following table shows the register bit descriptions.

Table 8-17: DCU_LOCK0 register bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	DCU_LOCK0	Writing 1 locks the corresponding DCU_EN [31:0] bit for modification. Each bit is set-once per LCM reset cycle.	RW1S	`0x0`

Related information

DCU_LOCK1 on page 85
DCU_LOCK2 on page 86
DCU_LOCK3 on page 86

8.3.6 DCU_LOCK1, DCU Lock register 1

The register holds bits 63:32 of the 128-bit DCU lock register (DCU_LOCK).

Configurations

This register is available in all configurations.

Attributes

The register characteristics are:

Width

32

Functional group

DCU register summary

Address offset

`0x0114`

Type

RW1S

Reset value

`0x0`

Bit descriptions

The following table shows the register bit descriptions.

Table 8-18: DCU_LOCK1 register bit descriptions

Bits	Field name	Field description	Type	Reset
[31:0]	DCU_LOCK1	Writing one locks the corresponding DCU_EN [63:32] bit for modification. Each bit is set-once per LCM reset cycle.	RW1S	`0x0`

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Related information

DCU_LOCK0 on page 84
DCU_LOCK2 on page 86
DCU_LOCK3 on page 86

8.3.7 DCU_LOCK2, DCU Lock register 2

The register holds bits 95:64 of the 128-bit DCU lock register (DCU_LOCK).

Configurations

This register is available in all configurations.

Attributes

The register characteristics are:

Width

32

Functional group

DCU register summary

Address offset

0x0118

Type

RW1S

Reset value

0x0

Bit descriptions

The following table shows the register bit descriptions.

Table 8-19: DCU_LOCK2 register bit descriptions

Bits	Field name	Field description	Type	Reset
[31:0]	DCU_LOCK2	Writing one locks the corresponding DCU_EN [95:64] bit for modification. Each bit is set-once per LCM reset cycle.	RW1S	0x0

Related information

DCU_LOCK0 on page 84
DCU_LOCK1 on page 85
DCU_LOCK3 on page 86

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.

8.3.8 DCU_LOCK3, DCU Enable register 3

The register holds bits 127:96 of the 128-bit DCU lock register (DCU_LOCK).

Configurations

This register is available in all configurations.

Attributes

The register characteristics are:

Width

32

Functional group

DCU register summary

Address offset

`0x011C`

Type

RW1S

Reset value

`0x0`

Bit descriptions

The following table shows the register bit descriptions.

Table 8-20: DCU_LOCK3 register bit descriptions

Bits	Field name	Field description	Type	Reset
[31:0]	DCU_LOCK3	Writing one locks the corresponding DCU_EN [127:96] bit for modification. Each bit is set-once per LCM reset cycle.	RW1S	`0x0`

Related information

DCU_EN0 on page 81
DCU_EN1 on page 81
DCU_EN2 on page 82

8.3.9 DCU_SP_DISABLE_MASK0, DCU_SP_DISABLE_MASK Parameter Value register 0

The register holds bits 31:0 of the 128-bit DCU_SP_DISABLE_MASK_VAL parameter value. The DCU_DISABLE_MASK register reads the active permanent disable mask parameter, according

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

to the current LCS, through the APB-S interface. When a bit is zero in a permanent disable mask parameter of the current LCS, the corresponding DCU_EN bit will always be forced to zero.

Configurations

This register is available in all configurations.

Attributes

The register characteristics are:

Width

32

Functional group

DCU register summary

Address offset

`0x0120`

Type

RO

Reset value

The reset value is DCU_SP_DISABLE_MASK_VAL configuration parameter. The default value of DCU_SP_DISABLE_MASK_VAL configuration parameter is `0xFFFF_FFFF`.

Bit descriptions

The following table shows the register bit descriptions.

Table 8-21: DCU_SP_DISABLE_MASK0 register bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	DCU_SP_DISABLE_MASK0	Bits [31:0] of the 128-bit DCU_SP_DISABLE_MASK_VAL parameter value.	RO	The reset value is DCU_SP_DISABLE_MASK_VAL configuration parameter. The default value of DCU_SP_DISABLE_MASK_VAL configuration parameter is `0xFFFF_FFFF`.

Related information

DCU_SP_DISABLE_MASK1 on page 88
DCU_SP_DISABLE_MASK2 on page 89
DCU_SP_DISABLE_MASK3 on page 90

8.3.10 DCU_SP_DISABLE_MASK1, DCU_SP_DISABLE_MASK Parameter Value register 1

The register holds bits 63:32 of the 128-bit DCU_SP_DISABLE_MASK_VAL parameter value.

Configurations

This register is available in all configurations.

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Attributes

The register characteristics are:

Width

32

Functional group

DCU register summary

Address offset

`0x0124`

Type

RO

Reset value

The reset value is DCU_SP_DISABLE_MASK_VAL configuration parameter. The default value of DCU_SP_DISABLE_MASK_VAL configuration parameter is `0xFFFF_FFFF`.

Bit descriptions

The following table shows the register bit descriptions.

Table 8-22: DCU_SP_DISABLE_MASK1 register bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	DCU_SP_DISABLE_MASK1	Bits [63:32] of the 128-bit DCU_SP_DISABLE_MASK_VAL parameter value.	RO	The reset value is DCU_SP_DISABLE_MASK_VAL configuration parameter. The default value of DCU_SP_DISABLE_MASK_VAL configuration parameter is `0xFFFF_FFFF`.

Related information

DCU_SP_DISABLE_MASK0 on page 87
DCU_SP_DISABLE_MASK2 on page 89
DCU_SP_DISABLE_MASK3 on page 90

8.3.11 DCU_SP_DISABLE_MASK2, DCU_SP_DISABLE_MASK Parameter Value register 2

The register holds bits 95:64 of the 128-bit DCU_SP_DISABLE_MASK_VAL parameter value.

Configurations

This register is available in all configurations.

Attributes

The register characteristics are:

Width

32

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Functional group

DCU register summary

Address offset

`0x0128`

Type

RO

Reset value

The reset value is DCU_SP_DISABLE_MASK_VAL configuration parameter. The default value of DCU_SP_DISABLE_MASK_VAL configuration parameter is `0xFFFF_FFFF`.

Bit descriptions

The following table shows the register bit descriptions.

Table 8-23: DCU_SP_DISABLE_MASK2 register bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	DCU_SP_DISABLE_MASK2	Bits [95:64] of the 128-bit DCU_SP_DISABLE_MASK_VAL parameter value.	RO	The reset value is DCU_SP_DISABLE_MASK_VAL configuration parameter. The default value of DCU_SP_DISABLE_MASK_VAL configuration parameter is `0xFFFF_FFFF`.

Related information

DCU_SP_DISABLE_MASK0 on page 87
DCU_SP_DISABLE_MASK1 on page 88
DCU_SP_DISABLE_MASK3 on page 90

8.3.12 DCU_SP_DISABLE_MASK3, DCU_SP_DISABLE_MASK Parameter Value register 3

The register holds bits 127:96 of the 128-bit DCU_SP_DISABLE_MASK_VAL parameter value.

Configurations

This register is available in all configurations.

Attributes

The register characteristics are:

Width

32

Functional group

DCU register summary

Address offset

`0x012C`

Type

RO

Reset value

The reset value is DCU_SP_DISABLE_MASK_VAL configuration parameter. The default value of DCU_SP_DISABLE_MASK_VAL configuration parameter is `0xFFFF_FFFF`.

Bit descriptions

The following table shows the register bit descriptions.

Table 8-24: DCU_SP_DISABLE_MASK3 register bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	DCU_SP_DISABLE_MASK3	Bits [127:96] of the 128-bit DCU_SP_DISABLE_MASK_VAL parameter value.	RO	The reset value is DCU_SP_DISABLE_MASK_VAL configuration parameter. The default value of DCU_SP_DISABLE_MASK_VAL configuration parameter is `0xFFFF_FFFF`.

Related information

DCU_SP_DISABLE_MASK0 on page 87
DCU_SP_DISABLE_MASK1 on page 88
DCU_SP_DISABLE_MASK2 on page 89

8.3.13 DCU_DISABLE_MASK0, DCU_DISABLE_MASK Parameter Value register 0

The register holds bits 31:0 of the 128-bit PERMANENT_DISABLE_MASK parameter value, according to current LCS.

Configurations

This register is available in all configurations.

Attributes

The register characteristics are:

Width

32

Functional group

DCU register summary

Address offset

`0x0130`

Type

RO

Reset value

Configuration-dependent

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Bit descriptions

The following table shows the register bit descriptions.

Table 8-25: DCU_DISABLE_MASK0 register bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	DCU_DISABLE_MASK0	Bits [31:0] of the 128-bit PERMANENT_DISABLE_MASK parameter, according to current LCS.	RO	Configuration dependent

Related information

DCU_DISABLE_MASK1 on page 92
DCU_DISABLE_MASK2 on page 93
DCU_DISABLE_MASK3 on page 93

8.3.14 DCU_DISABLE_MASK1, DCU_DISABLE_MASK Parameter Value register 1

The register holds bits 63:32 of the 128-bit PERMANENT_DISABLE_MASK parameter value, according to current LCS.

Configurations

This register is available in all configurations.

Attributes

The register characteristics are:

Width

32

Functional group

DCU register summary

Address offset

`0x0134`

Type

RO

Reset value

Configuration-dependent

Bit descriptions

The following table shows the register bit descriptions.

Table 8-26: DCU_DISABLE_MASK1 register bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	DCU_DISABLE_MASK1	Bits 63:32 of the 128-bit PERMANENT_DISABLE_MASK parameter, according to current LCS.	RO	Configuration dependent

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Related information

DCU_DISABLE_MASK0 on page 91
DCU_DISABLE_MASK2 on page 93
DCU_DISABLE_MASK3 on page 93

8.3.15 DCU_DISABLE_MASK2, DCU_DISABLE_MASK Parameter Value register 2

The register holds bits 95:64 of the 128-bit PERMANENT_DISABLE_MASK parameter value, according to current LCS.

Configurations

This register is available in all configurations.

Attributes

The register characteristics are:

Width

32

Functional group

DCU register summary

Address offset

`0x0138`

Type

RO

Reset value

Configuration-dependent

Bit descriptions

The following table shows the register bit descriptions.

Table 8-27: DCU_DISABLE_MASK2 register bit descriptions

Bits	Name	Description	Type	Reset
[31:0]	DCU_DISABLE_MASK2	Bits [95:64] of the 128-bit PERMANENT_DISABLE_MASK parameter, according to current LCS.	RO	Configuration dependent

Related information

DCU_DISABLE_MASK0 on page 91
DCU_DISABLE_MASK1 on page 92
DCU_DISABLE_MASK3 on page 93

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

8.3.16 DCU_DISABLE_MASK3, DCU_DISABLE_MASK Parameter Value register 3

The register holds bits 127:96 of the 128-bit PERMANENT_DISABLE_MASK parameter value, according to current LCS.

Configurations

This register is available in all configurations.

Attributes

The register characteristics are:

Width

32

Functional group

DCU register summary

Address offset

`0x013C`

Type

RO

Reset value

Configuration-dependent

Bit descriptions

The following table shows the register bit descriptions.

Table 8-28: DCU_DISABLE_MASK3 register bit descriptions

Bits	Field name	Field description	Reset value	Access type
[31:0]	DCU_DISABLE_MASK3	Bits [127:96] of the 128-bit PERMANENT_DISABLE_MASK parameter, according to current LCS.	Configuration dependent	RO

Related information

DCU_DISABLE_MASK0 on page 91
DCU_DISABLE_MASK1 on page 92
DCU_DISABLE_MASK2 on page 93

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

8.4 PID and CID register summary

The following table shows the registers in the ID region, in offset order from the base LCM memory address.

Table 8-29: PID and CID register summary

Offset	Name	Type	Reset	Width	Description
0x0FD0	Peripheral ID 4	RO	0xF4	32-bit	Peripheral ID 4
0x0FD4	Reserved	RO	0x0	32-bit	Reserved
0x0FD8	Reserved	RO	0x0	32-bit	Reserved
0x0FDC	Reserved	RO	0x0	32-bit	Reserved
0x0FE0	Peripheral ID 0	RO	0xF5	32-bit	Peripheral ID 0
0x0FE4	Peripheral ID 1	RO	0xB0	32-bit	Peripheral ID 1
0x0FE8	Peripheral ID 2	RO	0x0B	32-bit	Peripheral ID 2
0x0FEC	Peripheral ID 3	RO	0x0	32-bit	Peripheral ID 3
0x0FF0	Component ID 0	RO	0xD	32-bit	Component ID 0
0x0FF4	Component ID 1	RO	0xF0	32-bit	Component ID 1
0x0FF8	Component ID 2	RO	0x5	32-bit	Component ID 2
0x0FFC	Component ID 3	RO	0xB1	32-bit	Component ID 3

8.4.1 PIDR4, Peripheral ID register 4

The Peripheral ID4 register returns byte[4] of the peripheral ID.

Configurations

This register is available in all configurations.

Attributes

Width

32

Functional group

PID and CID register summary

Address offset

0x0FD0

Type

RO

Reset value

0x0000_0004

Bit descriptions

The following table shows the register bit descriptions.

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Table 8-30: Peripheral ID 4 register bit descriptions

Bits	Name	Description	Type	Reset
[31:8]	-	Reserved	-	-
[7:4]	SIZE	4KB Count - the number of 4K pages used.< - `0x00`: 4K - `0x01`: 8K - `0x02`: 16K - `0x03`: 32K	RO	`0x0`
[3:0]	DES_2	JEP106 Continuation Code	RO	`0x4`

Related information

Peripheral ID 0 on page 96
Peripheral ID 1 on page 97
Peripheral ID 2 on page 97
Peripheral ID 3 on page 98

8.4.2 PIDR0, Peripheral ID 0 register

The Peripheral ID0 register returns byte[0] of the peripheral ID.

Configurations

This register is available in all configurations.

Attributes

Width

32

Functional group

PID and CID register summary

Address offset

`0x0FE0`

Type

RO

Reset value

`0x0000_00F2`

Bit descriptions

The following table shows the register bit descriptions.

Table 8-31: Peripheral ID 0 register bit descriptions

Bits	Name	Description	Type	Reset
[31:8]	-	Reserved	-	`0x00`
[7:0]	Peripheral ID 0	PART_0 - Identification register part number, bits[7:0]	RO	`0xC0`

Related information

Peripheral ID 4 on page 95

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Peripheral ID 1 on page 97
Peripheral ID 2 on page 97
Peripheral ID 3 on page 98

8.4.3 PIDR1, Peripheral ID 1 register

The Peripheral ID1 register returns byte[1] of the peripheral ID.

Configurations

This register is available in all configurations.

Attributes

Width

32

Functional group

PID and CID register summary

Address offset

`0x0FE4`

Type

RO

Reset value

`0x0000_00B0`

Bit descriptions

The following table shows the register bit descriptions.

Table 8-32: Peripheral ID 1 register bit descriptions

Bits	Name	Description	Type	Reset
[31:8]	-	Reserved	-	-
[7:4]	DES_0	JEP106 identification code, bits[3:0].	RO	`0xb`
[3:0]	PART_1	Identification register part number, bits[11:8]	RO	`0x3`

Related information

Peripheral ID 4 on page 95
Peripheral ID 0 on page 96
Peripheral ID 2 on page 97
Peripheral ID 3 on page 98

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

8.4.4 PIDR2, Peripheral ID 02 register

The Peripheral ID2 register returns byte[2] of the peripheral ID.

Configurations

This register is available in all configurations.

Attributes

Width

32

Functional group

PID and CID register summary

Address offset

`0x0FE8`

Type

RO

Reset value

`0x0000_001B`

Bit descriptions

The following table shows the register bit descriptions.

Table 8-33: Peripheral ID 2 register bit descriptions

Bits	Name	Description	Type	Reset
[31:8]	_	Reserved	_	_
[7:4]	REVISION	Revision Code	RO	`0x0`
[3]	JEDEC	JEDEC	RO	`0x1`
[2:0]	DES_1	JEP106 identification code,bits[6:4].	RO	`0x3`

Related information

Peripheral ID 4 on page 95

Peripheral ID 0 on page 96

Peripheral ID 1 on page 97

Peripheral ID 3 on page 98

8.4.5 PIDR3, Peripheral ID 03 register

The Peripheral ID3 register returns byte[3] of the peripheral ID.

Configurations

This register is available in all configurations.

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.

Attributes

Width

32

Functional group

PID and CID register summary

Address offset

`0x0FEC`

Type

RO

Reset value

`0x0000_0000`

Bit descriptions

The following table shows the register bit descriptions.

Table 8-34: Peripheral ID 3 register bit descriptions

Bits	Name	Description	Type	Reset
[31:8]	-	Reserved	-	-
[7:4]	REVAND	Manufacturer revision number	RO	`0x0`
[3:0]	CMOD	Customer Modified	RO	`0x0`

Related information

Peripheral ID 4 on page 95
Peripheral ID 0 on page 96
Peripheral ID 1 on page 97
Peripheral ID 2 on page 97

8.4.6 CIDR0, Component ID 0 register

The Component ID0 register returns byte[0] of the component ID.

Configurations

This register is available in all configurations.

Attributes

Width

32

Functional group

PID and CID register summary

Address offset

`0x0FF0`

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Type

RO

Reset value

`0x0000_000D`

Bit descriptions

The following table shows the register bit descriptions.

Table 8-35: Component ID 0 register bit descriptions

Bits	Name	Description	Type	Reset
[31:8]	-	Reserved	-	-
[7:0]	PRMBL_0	Preamble	RO	`0xD`

Related information

Component ID 1 on page 100
Component ID 2 on page 101
Component ID 3 on page 102

8.4.7 CIDR1, Component ID 1 register

The Component ID1 register returns byte[1] of the component ID.

Configurations

This register is available in all configurations.

Attributes

Width

32

Functional group

PID and CID register summary

Address offset

`0x0FF4`

Type

RO

Reset value

`0x0000_00F0`

Bit descriptions

The following table shows the register bit descriptions.

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Table 8-36: Component ID 1 register bit descriptions

Bits	Name	Description	Type	Reset
[31:8]	-	Reserved	-	-
[7:4]	CLASS	Component class	RO	`0xF`
[3:0]	PRMBL_1	Preamble	RO	`0x0`

Related information

Component ID 0 on page 99
Component ID 2 on page 101
Component ID 3 on page 102

8.4.8 CIDR2, Component ID 2 register

The Component ID2 register returns byte[2] of the component ID.

Configurations

This register is available in all configurations.

Attributes

Width

32

Functional group

PID and CID register summary

Address offset

`0x0FF8`

Type

RO

Reset value

`0x0000_0005`

Bit descriptions

The following table shows the register bit descriptions.

Table 8-37: Component ID 2 register bit descriptions

Bits	Name	Description	Type	Reset
[31:8]	-	Reserved	-	-
[7:0]	PRMBL_2	Preamble	RO	`0x5`

Related information

Component ID 0 on page 99
Component ID 1 on page 100

Component ID 3 on page 102

8.4.9 CIDR3, Component ID 3 register

The Component ID3 register returns byte[3] of the component ID.

Configurations

This register is available in all configurations.

Attributes

Width

32

Functional group

PID and CID register summary

Address offset

`0x0FFC`

Type

RO

Reset value

`0x0000_00B1`

Bit descriptions

The following table shows the register bit descriptions.

Table 8-38: Component ID 3 register bit descriptions

Bits	Name	Description	Type	Reset
[31:8]	-	Reserved	-	-
[7:0]	PRMBL_3	Preamble	RO	`0xb1`

Related information

Component ID 0 on page 99
Component ID 1 on page 100
Component ID 2 on page 101

8.5 OTP register region

The addresses in the OTP register region are mapped to OTP memory at offset `0x0`.

The OTP memory layout, access types, and access control for each lifecycle are defined in Table 4-4: OTP memory map on page 42.

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Note

After reset and until the LCS is determined (lcs_is_valid is asserted), any read or write transactions to the OTP region result in an APB error.

Table 8-39: OTP register region

Address offset	Description
`0x1000-0xFFFF`	OTP memory direct access. Some addresses are not readable and/or writable. For more information, see OTP Memory map. The actual OTP size can be read from the OTP_SIZE_IN_BYTES register field in the LCS register region.

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Appendix A Software guidelines

The following pseudocode sections describe the expected programmer usage of various LCM programming operations. The pseudocode is intended to be used by a programmer to design the SW libraries and drivers that control and interact with the LCM. The pseudocode is also used by HW validation engineers to design the validation flows of the LCM. Programmer flows that vary from the described pseudocode sections may not perform the expected operations in the LCM and may result in a stall, unexpected behavior, or a fatal error condition.

The pseudocode is written as annotations and plain informative text without any specific syntax. Some basic register read/write, OTP read/write functions are defined as follows:

- All register name are listed in Programmer model.
- All OTP offsets are listed in OTP memory map.

```
function ReadRegister(Argument RegisterName)
    Read from register RegisterName.
Return the value read from register.
end
function WriteRegister(Argument RegisterName, Argument Value)
    Write Value to register RegisterName.
end
function ReadOtp(Argument OtpOffset)
    Read one word from OTP offset OtpOffset.
Return the value read from OTP.
End
function WriteOtp(Argument OtpOffset, Argument Value)
    Write one word with value of Value to OTP offset OtpOffset.
end
function ReadOtpWords(Argument OtpOffset)
    Read eight words (256 bits) from OTP offset OtpOffset.
    All words shall be read in order from the lowest offset to the highest.
Return the eight words read from OTP.
end
function WriteOtpWords(Argument OtpOffset, Argument WordArray)
    Write eight words (256 bits) with value of WordArray to OTP offset OtpOffset.
    All words shall be written in order from the lowest offset to the highest.
end
```

Security Guideline

Arm recommends that as additional security protection against fault injection attacks, the register read function performs the read operation twice with random delays of zero to seven cycles between each read operation and compares the results. For the write register function, Arm recommends that you perform a read after writing, and compare the results.

A.1 Read lifecycle state routines

These two routines read the lifecycle state and TP Mode. For an invalid state, the routine returns an error.

```
// Note: After exit from reset, the first read of any of the LCM registers
// before the lcs_is_valid signal is asserted stalls the read response. In
```

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

```
// case of fatal error in the computation of the LCS, a fatal_error signal is
// asserted and with proper hardware implementation would reset the system.

function GetLcs()
    // Check for fatal error
    FatalError = ReadRegister(FATAL_ERR)
    if FatalError == 0xFFFF_FFFF       // Fatal Error
        return ERROR

    // Read LCS value
    Lcs = ReadRegister(LCS_VALUE)
    if Lcs == 0xDEAD_BEEF          // Invalid LCS
        return ERROR
    else
        return Lcs
end

function GetTpMode()
    // Check for fatal error
    FatalError = ReadRegister(FATAL_ERR)
    if FatalError == 0xFFFF_FFFF       // Fatal Error
        return ERROR

    // Read TP Mode value
    TpMode = ReadRegister(TP_MODE)
    if TpMode == 0xDEAD_BEEF           // Invalid TP Mode
        return ERROR
    else
        return TpMode
end
```

A.2 Secure provisioning programmer flow

The Secure provisioning programmer flow describes the sequence of operations required to transition between various lifecycle states which include reading and writing to the LCM registers and as a result programming the OTP memory. The sequence is usually executed in a manufacturing facility or test lab. All the steps must be performed in the order described.

Note

The APB-S interface pauses all transactions until the LCS is calculated. Therefore, the latency of the first read operation after reset may be long.

A.2.1 Step 1: Transition from Virgin to PCI or TCI TP mode

The following code describes the transition from Virgin to PCI or TCI TP mode.

```
// Reset event occurs

// LCS must be CM
Lcs = GetLcs()
if Lcs != 0xCCCC_3C3C     // CM LCS
    return ERROR
// TP Mode must be Virgin
TpMode = GetTpMode()
if TpMode != 0x0000_33CC      // Virgin TP Mode
    return ERROR
```

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

```
// Set TCI or PCI TP Mode by writing to the OTP memory and verify
if TCI
    WriteOtp(0x00E0, 0x0000_FFFF)   // Write TCI to TP Mode Configuration in the
 OTP memory
    TpMode = ReadOtp(0x00E0)
    if TpMode != 0x0000_FFFF
        return ERROR
if PCI
    WriteOtp(0x00E0, 0xFFFF_0000)   // Write PCI to TP Mode Configuration in the
 OTP memory
    TpMode = ReadOtp(0x00E0)
    if TpMode != 0xFFFF_0000
        return ERROR


// Check for fatal error
FatalError = ReadRegister(FATAL_ERR)
if FatalError == 0xFFFF_FFFF    // Fatal Error
    return ERROR

// Perform reset and verify if TP Mode is TCI or PCI as expected
```

A.2.2 Step 2: Transition from CM to DM LCS

The following code describes the transition from CM to DM LCS.

```
// Reset event occurs

// LCS must be CM
Lcs = GetLcs()
if Lcs != 0xCCCC_3C3C   // CM LCS
    return ERROR
TpMode = GetTpMode()
// TP Mode must be either TCI or PCI
if (TpMode != 0x1111_55AA) AND (TpMode != 0x2222_AA55)
    return ERROR

// Enter Secure provisioning only if not already set
if (ReadRegister(SP_ENABLE) == 0)
    WriteRegister(SP_ENABLE, 0x5EC1_0E1E)

// At this point the LCM sets the sp_rst_req signal
// In typical implementations the system goes through a Warm reset cycle.
// However, the LCM is not reset. The KMU is reset and SW is reset back to the
// initial boot loader. The provisioning execution flow
// continues as described below.
// There are systems where the propagation of the sp_rst_req signal into Warm reset
// takes time. During that time the LCM reads the LCS from the OTP memory again and
 then
// (in case of no error) attempts to export keys to the KMU. But because the Warm
// reset is delayed and the KMU is soon to be reset, there is a push back mechanism
// that prevents the LCM writes to the KMU until the Warm reset happens. During
// that time (when the LCM attempts to export keys) the LCM push back any
// register access.
// Software must wait 2000 cycles after setting the
// SP_ENABLE until it accesses the LCM registers again. Software attempts to access
 the LCM registers without waiting the required 2000 cycles cause unpredictable
 behaviour.


wait(2000 clocks)


// Wait until Secure provisioning tranistion is complete, in case Warm reset cycle
// didn’t occur.
do
    SpMode = ReadRegister(SP_ENABLE)
    FatalError = ReadRegister(FATAL_ERR)
while (SpMode == 0) AND (FatalError == 0)

// Check for fatal error
if FatalError == 0xFFFF_FFFF    // Fatal Error
    return ERROR

// Make sure LCM transitioned to Secure provisioning
if SpMode != 0xFFFF_FFFF    // If Secure provisioning is not enabled
    return ERROR


// If TP mode is PCI, the RTL key may now be used during Secure provisioning
// to decrypt and verify the confidential keys.
// If TP mode is TCI, the RTL key is not available and only non-confidential
// test keys should be provisioned.

// Write keys to the OTP memory
// The following functions write a 256-bit value to the specfied OTP offset.
// The key is written word by word in order.
// The key bits value must not equal all zeros all ones.
// If some keys are not required for usage, dummy values, such as 0x01020304,
// must be provisioned for all words. In this case,
// we recommend that the bootloader disable any usage of those keys, such
// as invalidating a HW keyslot in the KMU.
// HUK, GUK, KP_CM and KCE_CM are 256-bit values
WriteOtpWords(0x0000, HUK)
WriteOtpWords(0x0020, GUK)
WriteOtpWords(0x0040, KP_CM)
WriteOtpWords(0x0060, KCE_CM)

// Verify keys were written correctly.
// The below function read a 256 bit value from the specified OTP offset.
if ReadOtpWords(0x0000) != HUK
    return ERROR
if ReadOtpWords(0x0020) != GUK
    return ERROR
if ReadOtpWords(0x0040) != KP_CM
    return ERROR
if ReadOtpWords(0x0060) != KCE_CM
    return ERROR

// Trigger LCM zero count programming in CM Configuration 1
WriteOtp(0x00E4, 0xFFFF_FFFF)
// Verify success comparison of CM Configuration 1
CompareResult = ReadOtp(0x00E4)
if CompareResult != 0
        return ERROR

// Write ROTPK in the OTP memory
// The value is written word by word in order.
// The ROTPK bits value must not equal all zeros or all ones.
// If some keys are not required for usage, dummy values, such as 0x01020304,
// must be provisioned for all words. In this case,
// we recommend that the bootloader disable any usage of those keys, such
// as invalidating a HW keyslot in the KMU.
// ROTPK is a 256-bit value
WriteOtpWords(0x00C0, ROTPK)

// Verify ROTPK was written correctly
// The following function reads a 256-bit value from the specified OTP memory
 offset.
if ReadOtpWords(0x00C0) != ROTPK
    return ERROR
```

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

```
// Write ROTPK number of zeros and GPPC value to CM Configuration 2
// CmConfig2Value is a 32-bit value that contains the number of zeros
// in ROTPK and GPPC value
WriteOtp(0x00E8, CmConfig2Value)
// Verify value was written correctly
Result = ReadOtp(0x00E8)
if Result != CmConfig2Value
        return ERROR

// Perform reset and verify that the LCS is DM as expected
```

A.2.3 Step 3: Transition from DM to SE LCS

The following code describes the transition from DM to SE LCS.

```
// Reset event occurs

// LCS must be DM
Lcs = GetLcs()
if Lcs != 0xDDDD_5A5A   // DM LCS
    return ERROR
TpMode = GetTpMode()
// TP Mode must be either TCI or PCI
if (TpMode != 0x1111_55AA) AND (TpMode != 0x2222_AA55)
    return ERROR

// Enter Secure provisioning
if (ReadRegister(SP_ENABLE) == 0)
WriteRegister(SP_ENABLE, 0x5EC1_0E1E)

// At this point the LCM sets the sp_rst_req signal.
// In typical implementations the system goes through a Warm reset cycle.
// However, the LCM would not be reset. The KMU is reset and SW would be reset back
 to the
// initial boot loader and the Secure provisioning execution flow should
// then continue as described below.

// There are systems where the propagation of the sp_rst_req into Warm reset
// takes time. During that time the LCM reads the LCS from the OTP memory again and
 then
// (in case of no error) attempts to export keys to the KMU. But because the Warm
// reset is delayed and the KMU is soon to be reset, there is a push back mechanism
// that prevents the LCM from writing to the KMU until the Warm reset happens.
// While the LCM attempts to export keys, the LCM pushes back any
// register access.
// Software must wait 2000 cycles from after setting the
// SP_ENABLE until it accesses the LCM registers again. Software attempts to access
 the LCM registers without waiting the required 2000 cycles cause unpredictable
 behaviour.


wait(2000 clocks)


// Wait until Secure provisioning transition is complete, in case Warm reset cycle
// didn’t occur.
do
    SpMode = ReadRegister(SP_ENABLE)
    FatalError = ReadRegister(FATAL_ERR)
while (SpMode == 0) AND (FatalError == 0)

// Check for fatal error
if FatalError == 0xFFFF_FFFF    // Fatal Error
    return ERROR

// Make sure LCM transitioned to Secure provisioning
```

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

```
if SpMode != 0xFFFF_FFFF    // If Secure provisioning is not enabled
    return ERROR

// If TP mode is PCI, the RTL key may now be used during Secure provisioning
// to decrypt and verify the confidential keys.
// If TP mode is TCI, the RTL key is not available and only non-confidential
// test keys should be provisioned.

// Write keys to the OTP memory
// The following functions write a 256-bit value to the specfied OTP memory offset.
// The key is written word by word in order.
// The key bits value must not equal all zeros or all ones.
// If some keys are not required for usage, dummy values, such as 0x01020304,
// should be provisioned for all words. In this case,
// we recommends that the bootloader disable any usage of those keys, such
// as invalidating a HW keyslot in the KMU.
// KP_DM and KCE_DM are 256-bit values
WriteOtpWords(0x0080, KP_DM)
WriteOtpWords(0x00A0, KCE_DM)

// Verify the keys were written correctly.
// The below function read a 256-bit value from the specified OTP memory offset.
if ReadOtpWords(0x0080) != KP_DM
    return ERROR
if ReadOtpWords(0x00A0) != KCE_DM
    return ERROR

// Trigger LCM zero count programming in DM Config
WriteOtp(0x00EC, 0xFFFF_FFFF)
// Verify success comparison of CM Config 1
CompareResult = ReadOtp(0x00EC)
if CompareResult != 0
        return ERROR

// Perform reset and verify that the LCS is SE as expected
```

A.2.4 Step 4: Transition from any LCS into RMA (Phase 1 - CM)

The following code describes the first phase of the transition from any LCS into RMA LCS.

```
// Reset event occurs

// The LCS must not be in RMA
Lcs = GetLcs()
if Lcs == 0xFFFF_C3C3   // RMA LCS
    return ERROR
// TP Mode must aleady be set to either TCI or PCI
TpMode = GetTpMode()
if TpMode == 0x0000_33CC    // TP mode is Virgin
    return ERROR

// Write CM RMA OTP flag
WriteOtp(0x00F0, 0xFFFF_FFFF)
// Verify value was written correctly
Result = ReadOtp(0x00F0)
if Result != 0xFFFF_FFFF
        return ERROR
```

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

A.2.5 Transition from any LCS into RMA (Phase 2 - DM)

The following code describes the second phase of the transition from any LCS into RMA LCS.

```
// Reset event occurs

// LCS must not be in RMA
Lcs = GetLcs()
if Lcs == 0xFFFF_C3C3   // RMA LCS
    return ERROR
// TP Mode must aleady be set to either TCI or PCI
TpMode = GetTpMode()
if TpMode == 0x0000_33CC    // TP Mode is Virgin
    return ERROR

// It is possible to lock any write operations to the “DM RMA Flag”
// OTP offset until the next reset cycle by writing the
 DM_RMA_LOCK_SET_VALUE, 0xDDAF_10CE, to the DM_RMA_LOCK
// register. In case the DM RMA lock is not enabled, the below operation
// could continue as normal.

// Write DM RMA flag
WriteOtp(0x00F4, 0xFFFF_FFFF)
// Verify value was written correctly
Result = ReadOtp(0x00F4)
if Result != 0xFFFF_FFFF
        return ERROR
```

Security Guideline

Although the LCM invalidates HW keys on entry to RMA LCS, it does not overwrite them. After the LCS has successfully transitioned into RMA, to comply with *FIPS 140-2 Security Requirements for Cryptographic Modules* requirements the boot loader must overwrite the six confidential keys by writing all ones to OTP memory addresses `0x0000-0x00BF`, in word accesses. The boot loader may also overwrite the ROTPK (privacy assurance) by writing all ones to OTP memory addresses `0x00C0-0x00DF`. In some OTP memory implementations overwriting an OTP word is not possible. In that case, the design does not not comply with FIPS requirements.

A.2.6 LCM lockdown

LCM lockdown happens when the TP mode is set to invalid. It permanently disables the LCM. LCM lockdown can only be set for devices that are in CM LCS and in Virgin TP mode. While RMA LCS allows access to debug, LCM lockdown effectively bricks the device. Using LCM lockdown allows the programmer to protect secret assets from attackers if a system failure has been detected. The device is effectively destroyed.

The lockdown is performed by setting an invalid value to the TP mode configuration word in OTP memory. Once the Invalid TP mode is programmed to an invalid value, all future reset cycles result in the LCM Fatal Error FSM state, which securely disables the LCM. This process is irreversible.

```
// Reset event occurs
// LCS must be CM
Lcs = GetLcs()
if Lcs != 0xCCCC_3C3C   // CM LCS
    return ERROR
```

Copyright © 2023 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

```
// TP mode must be Virgin
TpMode = GetTpMode()
if TpMode != 0x0000_33CC    // Virgin TP Mode
    return ERROR
// Set TP Mode to invalid value and verify
WriteOtp(0x00E0, 0xFFFF_FFFF)   // Write invalid value to TP Mode Config in OTP
 memory
TpMode = ReadOtp(0x00E0)
if TpMode != 0xFFFF_FFFF
    return ERROR
```