芯来科技
NUCLEI

Copyright Notice

The product described herein is subject to continuous development and improvement; information herein is given by Nuclei in good faith but without warranties.

This document is intended only to assist the reader in the use of the product. Nuclei System Technology shall not be liable for any loss or damage arising from the use of any information in this document, or any incorrect use of the product.

Contact Information

Should you have any problems with the information contained herein or any suggestions, please contact Nuclei System Technology by email support@nucleisys.com, or visit “Nuclei User Center” website http://user.nucleisys.com for supports or online discussion.

Revision History

Rev.	Rev. Date	Revised Section	Revised Content
1.0	2020/4/20	N/A	1. Fist English version
1.1.0	2021/8/13	1	1. ADD L2 Cache CCM operations

Contents

Table List

1.CCM Mechanism Introduction

1.1. Nuclei CCM Background

CCM (Cache Control and Mantainence) is defined for software to control and mantainence the internal L1 I/D-Cache and external L2 Cache of the core, software can manage the Cache states flexibly to meet the acutal application scenarios.

1.2. Nuclei CCM Implementations

Nuclei CCM mechanism uses specific CSRs to control and mantainence the Cache. CCM operations have 3 types: by single Address, by ALL and Flush pipeline.

For the 'by single Address' operation, one complete CCM operation flow is:

- Using CSRW to write CSR 'ccm_xbeginaddr', to specify the ADDR of the CCM operation.
- Using CSRW to write CSR 'ccm_xcommand', to specify the CMD type of the CCM operation, CMD type will be list in details in below chapter.
- CCM operation will be triggered at the next cycle of the CMD CSR write operation.
- For some operations, such as Lock, using CSRR to read CSR 'ccm_xdata' to get the result of the CCM operation.

For the 'by ALL' operation, one complete CCM operation flow is:

- Using CSRW to write CSR 'ccm_xcommand', to specify the CMD type of the CCM operation, CMD type will be list in details in below chapter.
- CCM operation will be triggered at the next cycle of the CMD CSR write operation.

Note: the *x* in ccm_xbeginaddr, ccm_xcommand and ccm_xdata can be M/S/U mode, each mode will have its own CSR registers.

1.3. Nuclei CCM CSR registers

Nuclei defines some extend CSR registers for CCM, which are list in below table, M/S/U mode has its own CSR registers, and the permission of the CCM operations in S/U mode is defined in 'ccm_suen'.

Table 1-1 Nuclei CCM CSR Registers List

Type	CSR Addr	R/W	Name	Description

CCM CSRs	0x7CB	MRW	ccm_mbeginaddr	Machine Mode CCM operation ADDR
	0x7CC	MRW	ccm_mcommand	Machine Mode CCM operation CMD
	0x7CD	MRW	ccm_mdata	Machine Mode CCM operation ReadBack DATA
	0x7CE	MRW	ccm_suen	Supervisor/User mode CCM Control
	0x5CB	SRW	ccm_sbeginaddr	Supervisor Mode CCM operation ADDR
	0x5CC	SRW	ccm_scommand	Supervisor Mode CCM operation CMD
	0x5CD	SRW	ccm_sdata	Supervisor Mode CCM operation ReadBack DATA
	0x4CB	URW	ccm_ubeginaddr	User Mode CCM operation ADDR
	0x4CC	URW	ccm_ucommand	User Mode CCM operation CMD
	0x4CD	URW	ccm_udata	User Mode CCM operation ReadBack DATA
	0x4CF	URW	ccm_fpipe	Flush Pipeline CMD

1.3.1. ccm_beginaddr

ccm_beginaddr is to define the ADDR of the CCM operation, this ADDR is treat as VA (Virtual Address), which will be translated to be PA by MMU (if configured) in pipeline, before accessing Cache.

Note, this CSR register will INCR one cacheline in Byte automatically by HW after each CCM operation. (*ccm_beginaddr = ccm_beginaddr + One-Cacheline-Bytes*)

ccm_beginaddr CSR description is list inTable 1 -2。

Table 1-2 ccm_beginaddr CSR Register Content

Field Name	**Bits**	**RW**	**Reset Value**	**Description**
beginaddr	XLEN-1:0	RW	0	CCM operation ADDR, M/S/U mode has its own one.

1.3.2. ccm_command

ccm_command is to define the CMD of the CCM operation, after an valid CMD is defined, the corresponding operation will be triggered at the next cycle, till a completion sent from Cache. ccm_command CSR description is list in TableTable 1 -3。

Table 1-3 ccm_command CSR Register Content

Field Name	**Bits**	**RW**	**Reset Value**	**Description**
Reserved	XLEN-1:5	R	0	-
command	4:0	RW	0	CCM operation CMD, M/S/U mode has its own one.

CMD types list in Table 1 -4 :

Table 1-4 ccm_command CMD Types

Type	Operation	Codes	Description
I-Cache operation	INVAL	5b01_000	When I-Cache hit, Unlock and Invalid the specific cacheline in I-Cache; Ignored when I-Cache miss.
	LOCK	5b01_011	When I-Cache hit, Lock the specific cacheline in I-Cache; When I-Cache miss, Refill the specific cacheline then Lock it in I-Cache. Check ccm_data to check the Lock succeed or not.
	UNLOCK	5b01_100	When I-Cache hit, Unlock the specific cacheline in I-Cache; Ignored when I-Cache miss.
	INVAL_ALL	5b01_101	Unlock and Invalid ALL the cacheline in I-Cache.
D-Cache operation	INVAL	5b00_000	When cache hit, Unlock and Invalid the specific cacheline; Ignored when cache miss. Work on both D-Cache and L2 Cache.
	WB	5b 00_001	When cache hit and Dirty, Flush the specific cacheline; When cache miss or hit but not dirty, ignored. Lock bit is not affected. Work on both D-Cache and L2 Cache.
	WBINVAL	5b 00_010	When cache hit, Unlock and Flush and Invalid the specific cacheline; Ignored when cache miss. Work on both D-Cache and L2 Cache.
	LOCK	5b 00_011	When cache hit, Lock the specific cacheline; When cache miss, Refill the specific cacheline then Lock it. Check ccm_data to check the Lock succeed or not. Work on D-Cache only.
	UNLOCK	5b 00_100	When cache hit, Unlock the specific cacheline; Ignored when cache miss. Work on D-Cache only.
	INVAL_ALL	5b 10_111	Unlock and Invalid ALL the cacheline. Work on D-Cache only.
	WB_ALL	5b 00_111	Flush ALL the Valid and Dirty cachelines; Lock bit is not affected. Work on D-Cache only.
	WBINVAL_ALL	5b 00_110	Unlock and Flush and Invalid ALL the Valid and Dirty cachelines. Work on D-Cache only.
	L2_LOCK	5b10_011	When cache hit, Lock the specific cacheline; When cache miss, Refill the specific cacheline then Lock it. Check ccm_data to check the Lock succeed or not. Work on L2 Cache only.

	L2_UNLOCK	5b10_010	When cache hit, Unlock the specific cacheline; Ignored when cache miss. Work on L2 Cache only.
L2 Cache	WB_ALL WBINVAL_ALL		Please refer to the Nuclei ISA Spec for more details for the 'WB_ALL/WBINVAL_ALL' CCM operations on L2 Cache.

All the above list operations are for M mode. Notes:

- ADDR is needed to be specified except for the 'ALL' operations.
- Cacheline is the one hit by the ADDR, the ADDR can point to the head, middle or end of the cacheline.
- S/U mode has the same CMD types as M mode, but for security, D-Cache INVAL_ALL CMD will be upgrade to be WBINVAL_ALL in S/U mode if sinven/uinven is set.
- CCM operations can still work on the Disabled Cache.
- Higher privileged mode can operate on the lower privileged mode CCM CSR registers to trigger the CCM operations; But 'illegal instruction' exception will trigger when the lower privileged mode operates on higher privileged mode CCM CSR registers.

Some special cases for I-Cache CCM operations:

- Permission checking will be done for those 'by ADDR' CCM operations. Permission checking includes: X checking in Page table if VA-PA translation needed, X checking in PMP, X checking in sPMP, Device attribute checking and Non-Cacheable attribute checking. This CCM operation will be ignored if any permission checking fails.
- For LOCK operation, Refill will be triggered if Permission checking passes but Miss in I-Cache, LOCK will fail if Bus Error occurs during Refill and fail info will update into 'ccm_data' CSR register. ECC error (if configured) may occur in Tag Ram when LOCK operation, then LOCK will fail and ECC error info will update into 'sramid' CSR register.
- Software can access the 'ccm_data' CSR register to check the Fail Info details of the LOCK operation, ECC fail info needs to check the 'sramid' CSR register.

Table 1-5 I-Cache Lock Operation Fail Info

Type	Code	Fail Info
I-Cache CCM	0	Lock Succeed
	1	Exceed the Upper entry Num of Lockable

Operation		way (N-Way I-Cache, Lockable is N-1)
	2	PMP/sPMP/Page-Table X permission check fail, or is Device/Non-Cacheable attribute
	3	Refill has Bus Error
	4	reserved

- No Permission checking will be done on ‘INVAL_ALL’, the whole I-Cache will be invalidated directly.

Some special cases for D-Cache and L2 Cache CCM operations:

- Permission checking will be done for those ‘by ADDR’ CCM operations. Permission checking includes (except for LOCK/UNLOCK/L2_LOCK/L2_UNLOCK): R checking in Page table if VA-PA translation needed, W checking in PMP, W checking in sPMP, Device attribute checking and Non-Cacheable attribute checking. This CCM operation will be ignored if any permission checking fails or permission checking pass but miss in Cache.

- For LOCK/UNLOCK operation, permission checking will include: R checking in Page table if VA-PA translation needed, R checking in PMP, R checking in sPMP, Device attribute checking and Non-Cacheable attribute checking. This CCM operation will be ignored if any permission checking fails.

- For L2_LOCK/L2_UNLOCK operation, permission checking will include: R checking in Page table if VA-PA translation needed, X or R checking in PMP, X or R checking in sPMP, Device attribute checking and Non-Cacheable attribute checking. This CCM operation will be ignored if any permission checking fails.

- For LOCK/L2_LOCK operation, Refill will be triggered if Permission checking passes but Miss in D-Cache/L2 Cache, LOCK will fail if Bus Error occurs during Refill and fail info will update into ‘ccm_data’ CSR register. ECC error (if configured) may occur in Tag Ram when LOCK/L2_LOCK operation, then LOCK/L2_LOCK will fail and fail info will update into ‘sramid’ CSR register for LOCK operation or ‘L2C_FATAL_CNT’ register for L2_LOCK operation but not ‘sramid’ CSR register.

- Software can access the ‘ccm_data’ CSR register to check the Fail Info details of the LOCK/L2_LOCK operation.

Table 1-6 D-Cache Lock Operation Fail Info

Type	Code	Fail Info
D-Cache	0	Lock Succeed
	1	Exceed the Upper entry Num of Lockable way

CCM Operation		(N-Way D-Cache, Lockable is N-1)
	2	PMP/sPMP/Page-Table X permission check fail, or is Device/Non-Cacheable attribute
	3	Refill has Bus Error
	4	Reserved

- For ‘INVAL_ALL’ operation, PMP/sPMP W checking will be done on each cacheline when ‘ALL’ operation is enabled under the corresponding mode, if failed, ‘INV’ will be upgrade to be ‘WB+INV’ operation.
- No permission checking will be done on other ‘by ALL’ CCM operations if ‘ALL’ operation is enabled under the corresponding mode.
- WB/Flush will be triggered during some of the CCM operations, if Bus error occurs during WB/Flush, error info will be recorded and report to Core asynchronously; (for ‘by ALL’ operations, the cacheline in which bus error occurs will be skipped)

1.3.3. ccm_data

ccm_data is to record the result of the LOCK operation.

For LOCK operation, 0 in ‘ccm_data’ indicates succeed, non-zero value indicates failed. I-Cache Lock fail info can refer to Table 1 -5 , D-Cache and L2 Cache Lock fail info can refer to Error: Reference source not found.

ccm_data CSR content in Table 1 -7.

Table 1-7 ccm_data content

Field Name	Bits	RW	Reset Value	Description
Reserved	XLEN-1:3	R	0	-
data	2:0	RW	0	CCM DATA, M/S/U has its own one.

1.3.4. ccm_suen

ccm_suen is to control CCM operations in S/U mode, CCM operations can be permitted only when ccm_suen is 1, or ‘illegal instruction’ exception will be reported. This CSR will only exist in M mode when S/U mode is configured.

ccm_suen CSR content in Table 1 -8.

Table 1-8 ccm_suen content

Field Name	Bits	RW	Reset Value	Description
Reserved	XLEN-1:26	R	0	-
swballen	25	RW	0	S-mode WB_ALL enable

uwballen	24	RW	0	U-mode WB_ALL enable
Reserved	23:18	R	0	
sinvallen	17	RW	0	S-mode INVAL_ALL/WBINVAL_ALL enable
uinvallen	16	RW	0	U-mode INVAL_ALL/WBINVAL_ALL enable
Reserved	15:10	R	0	
sinven	9	RW	0	S-mode INVAL/WBINVAL enable
uinven	8	RW	0	U-mode INVAL/WBINVAL enable
Reserved	7:2	R	0	
sen	1	RW	0	S-mode CCM operations enable
uen	0	RW	0	U-mode CCM operations enable

1.3.5. ccm_fpipe

ccm_fpipe is to flush the pipeline after CCM operations on Cache, to ensure the latest instructions or data can be seen by pipeline.

ccm_fpipe CSR content in Table 1 -9.

Table 1-9 ccm_fpipe content

Field Name	**Bits**	**RW**	**Reset Value**	**Description**
fpipe	XLEN-1:0	RW	0	CCM Flush Pipeline, Wirte any value to this CSR will trigger pipeline flush, Read this CSR will return 0.