

# PULP

Parallel Ultra Low Power

## **HERO Hardware Reference Manual**

**Version 1.0.0**

This document is preliminary and subject to change

Micrel Lab and Multitherman Lab, University of Bologna ,Italy Integrated Systems Lab, ETH Zürich, Switzerland

## Table of Contents

Table of Contents	2
HERO Features	6
Introduction to the HERO architecture	7
System on Chip	9
Cores	9
Memory areas	9
Debug architecture	9
Events and interrupts model	9
Exceptions model	9
Data types supported	9
Event Units	9
DMA (direct memory access )	10
Performance counters	10
Memory map	11
Aliased memory map	11
Cluster aliased address map	11
Device components description	12
RISC-V cores	12
Complex number operations	12
Complex multiplication operations	12
Subtraction of 2 complexes with post rotation by -j	13
Complex conjugate operation	13
Extensions to existing RI5CY vector operations.	13
Addition of vector of half words with post right shift	13
Subtraction of vectors of half words with post right shift	14
Viterbi specific instructions	14
Enhanced shuffling	15
Cluster Subsystem	15
Cluster Subsystem Events	16
Cluster control unit	16
Cluster control unit registers	17
Cluster control unit registers details	17
End Of Computation status register. (EOC)	17
Cluster cores fetch enable configuration register. (FETCH_EN)	17
Cluster clock gate configuration register. (CLOCK_GATE)	19
Cluster cores debug resume register. (DBG_RESUME)	19
Cluster cores debug halt status register. (DBG_HALT_STATUS)	20
Cluster cores debug halt mask configuration register. (DBG_HALT_MASK)	21
Cluster core 0 boot address configuration register. (BOOT_ADDR0)	22
TCDM arbitration policy ch0 for cluster cores configuration register. (TCDM_ARB_POLICY_CH0)	22
TCDM arbitration policy ch1 for DMA/HWCE configuration register. (TCDM_ARB_POLICY_CH1)	22
Read only duplicate of TCDM_ARB_POLICY_CH0 register (TCDM_ARB_POLICY_CH0_REP)	23
Read only duplicate of TCDM_ARB_POLICY_CH1 register (TCDM_ARB_POLICY_CH1_REP)	23
Cluster timer	23
Cluster timer registers	24
Cluster timer registers details	24
Timer Low Configuration register. (CFG_LO)	24
Timer High Configuration register. (CFG_HI)	25
Timer Low counter value register. (CNT_LO)	26
Timer High counter value register. (CNT_HI)	26
Timer Low comparator value register. (CMP_LO)	26
Timer High comparator value register. (CMP_HI)	27
Start Timer Low counting register. (START_LO)	27
Start Timer High counting register. (START_HI)	27
Reset Timer Low counter register. (RESET_LO)	27
Reset Timer High counter register. (RESET_HI)	28

Cluster event unit	28
Cluster event unit registers	28
Cluster event unit registers details	35
Input event mask configuration register. (EVT_MASK)	35
Hardware task dispatcher push command register. (HW_DISPATCH_PUSH_TASK)	35
Hardware task dispatcher pop command register. (HW_DISPATCH_POP_TASK)	36
Hardware mutex 0 non-blocking put command register. (HW_MUTEX_0_MSG_PUT)	36
Hardware mutex 0 blocking get command register. (HW_MUTEX_0_MSG_GET)	36
Cluster Software event 0 trigger command register. (SW_EVENT_0_TRIG)	36
Cluster Software event 0 trigger and wait command register. (SW_EVENT_0_TRIG_WAIT)	36
Cluster Software event 0 trigger, wait and clear command register. (SW_EVENT_0_TRIG_WAIT_CLEAR)	37
Cluster SoC peripheral event ID status register. (SOC_PERIPH_EVENT_ID)	37
Cluster hardware barrier 0 trigger mask configuration register. (HW_BARRIER_0_TRIG_MASK)	37
Input event mask update command register with bitwise AND operation. (EVT_MASK_AND)	38
Hardware task dispatcher cluster core team configuration register. (HW_DISPATCH_PUSH_TEAM_CONFIG)	38
Hardware mutex 1 non-blocking put command register. (HW_MUTEX_1_MSG_PUT)	38
Hardware mutex 1 blocking get command register. (HW_MUTEX_1_MSG_GET)	38
Cluster Software event 1 trigger command register. (SW_EVENT_1_TRIG)	39
Cluster Software event 1 trigger and wait command register. (SW_EVENT_1_TRIG_WAIT)	39
Cluster Software event 1 trigger, wait and clear command register. (SW_EVENT_1_TRIG_WAIT_CLEAR)	39
Cluster hardware barrier 0 status register. (HW_BARRIER_0_STATUS)	39
Input event mask update command register with bitwise OR operation. (EVT_MASK_OR)	40
Cluster Software event 2 trigger command register. (SW_EVENT_2_TRIG)	40
Cluster Software event 2 trigger and wait command register. (SW_EVENT_2_TRIG_WAIT)	40
Cluster Software event 2 trigger, wait and clear command register. (SW_EVENT_2_TRIG_WAIT_CLEAR)	40
Cluster hardware barrier summary status register. (HW_BARRIER_0_STATUS_SUM)	41
Interrupt request mask configuration register. (IRQ_MASK)	41
Cluster Software event 3 trigger command register. (SW_EVENT_3_TRIG)	41
Cluster Software event 3 trigger and wait command register. (SW_EVENT_3_TRIG_WAIT)	42
Cluster Software event 3 trigger, wait and clear command register. (SW_EVENT_3_TRIG_WAIT_CLEAR)	42
Cluster hardware barrier 0 target mask configuration register. (HW_BARRIER_0_TARGET_MASK)	42
Interrupt request mask update command register with bitwise AND operation. (IRQ_MASK_AND)	42
Cluster Software event 4 trigger command register. (SW_EVENT_4_TRIG)	43
Cluster Software event 4 trigger and wait command register. (SW_EVENT_4_TRIG_WAIT)	43
Cluster Software event 4 trigger, wait and clear command register. (SW_EVENT_4_TRIG_WAIT_CLEAR)	43
Cluster hardware barrier 0 trigger command register. (HW_BARRIER_0_TRIG)	43
Interrupt request mask update command register with bitwise OR operation. (IRQ_MASK_OR)	44
Cluster Software event 5 trigger command register. (SW_EVENT_5_TRIG)	44
Cluster Software event 5 trigger and wait command register. (SW_EVENT_5_TRIG_WAIT)	44
Cluster Software event 5 trigger, wait and clear command register. (SW_EVENT_5_TRIG_WAIT_CLEAR)	44
Cluster hardware barrier 0 self trigger command register. (HW_BARRIER_0_SELF_TRIG)	45
Cluster cores clock status register. (CLOCK_STATUS)	45
Cluster Software event 6 trigger command register. (SW_EVENT_6_TRIG)	45
Cluster Software event 6 trigger and wait command register. (SW_EVENT_6_TRIG_WAIT)	45
Cluster Software event 6 trigger, wait and clear command register. (SW_EVENT_6_TRIG_WAIT_CLEAR)	46
Cluster hardware barrier 0 trigger and wait command register. (HW_BARRIER_0_TRIG_WAIT)	46
Pending input events status register. (EVENT_BUFFER)	46
Cluster Software event 7 trigger command register. (SW_EVENT_7_TRIG)	47
Cluster Software event 7 trigger and wait command register. (SW_EVENT_7_TRIG_WAIT)	47
Cluster Software event 7 trigger, wait and clear command register. (SW_EVENT_7_TRIG_WAIT_CLEAR)	47
Cluster hardware barrier 0 trigger, wait and clear command register. (HW_BARRIER_0_TRIG_WAIT_CLEAR)	47
Pending input events status register with EVT_MASK applied. (EVENT_BUFFER_MASKED)	48
Cluster hardware barrier 1 trigger mask configuration register. (HW_BARRIER_1_TRIG_MASK)	48
Pending input events status register with IRQ_MASK applied. (EVENT_BUFFER_IRQ_MASKED)	48
Cluster hardware barrier 1 status register. (HW_BARRIER_1_STATUS)	48
Pending input events status clear command register. (EVENT_BUFFER_CLEAR)	49
Cluster hardware barrier summary status register. (HW_BARRIER_1_STATUS_SUM)	49
Software events cluster cores destination mask configuration register. (SW_EVENT_MASK)	49
Cluster hardware barrier 1 target mask configuration register. (HW_BARRIER_1_TARGET_MASK)	49

Software events cluster cores destination mask update command register with bitwise AND operation. (SW_EVENT_MASK_AND)	50
Cluster hardware barrier 1 trigger command register. (HW_BARRIER_1_TRIG)	50
Software events cluster cores destination mask update command register with bitwise OR operation. (SW_EVENT_MASK_OR)	50
Cluster hardware barrier 1 self trigger command register. (HW_BARRIER_1_SELF_TRIG)	50
Input event wait command register. (EVENT_WAIT)	51
Cluster hardware barrier 1 trigger and wait command register. (HW_BARRIER_1_TRIG_WAIT)	51
Input event wait and clear command register. (EVENT_WAIT_CLEAR)	51
Cluster hardware barrier 1 trigger, wait and clear command register. (HW_BARRIER_1_TRIG_WAIT_CLEAR)	51
Cluster hardware barrier 2 trigger mask configuration register. (HW_BARRIER_2_TRIG_MASK)	52
Cluster hardware barrier 2 status register. (HW_BARRIER_2_STATUS)	52
Cluster hardware barrier summary status register. (HW_BARRIER_2_STATUS_SUM)	52
Cluster hardware barrier 2 target mask configuration register. (HW_BARRIER_2_TARGET_MASK)	52
Cluster hardware barrier 2 trigger command register. (HW_BARRIER_2_TRIG)	53
Cluster hardware barrier 2 self trigger command register. (HW_BARRIER_2_SELF_TRIG)	53
Cluster hardware barrier 2 trigger and wait command register. (HW_BARRIER_2_TRIG_WAIT)	53
Cluster hardware barrier 2 trigger, wait and clear command register. (HW_BARRIER_2_TRIG_WAIT_CLEAR)	53
Cluster hardware barrier 3 trigger mask configuration register. (HW_BARRIER_3_TRIG_MASK)	54
Cluster hardware barrier 3 status register. (HW_BARRIER_3_STATUS)	54
Cluster hardware barrier summary status register. (HW_BARRIER_3_STATUS_SUM)	54
Cluster hardware barrier 3 target mask configuration register. (HW_BARRIER_3_TARGET_MASK)	54
Cluster hardware barrier 3 trigger command register. (HW_BARRIER_3_TRIG)	55
Cluster hardware barrier 3 self trigger command register. (HW_BARRIER_3_SELF_TRIG)	55
Cluster hardware barrier 3 trigger and wait command register. (HW_BARRIER_3_TRIG_WAIT)	55
Cluster hardware barrier 3 trigger, wait and clear command register. (HW_BARRIER_3_TRIG_WAIT_CLEAR)	55
Cluster hardware barrier 4 trigger mask configuration register. (HW_BARRIER_4_TRIG_MASK)	56
Cluster hardware barrier 4 status register. (HW_BARRIER_4_STATUS)	56
Cluster hardware barrier summary status register. (HW_BARRIER_4_STATUS_SUM)	56
Cluster hardware barrier 4 target mask configuration register. (HW_BARRIER_4_TARGET_MASK)	56
Cluster hardware barrier 4 trigger command register. (HW_BARRIER_4_TRIG)	57
Cluster hardware barrier 4 self trigger command register. (HW_BARRIER_4_SELF_TRIG)	57
Cluster hardware barrier 4 trigger and wait command register. (HW_BARRIER_4_TRIG_WAIT)	57
Cluster hardware barrier 4 trigger, wait and clear command register. (HW_BARRIER_4_TRIG_WAIT_CLEAR)	57
Cluster hardware barrier 5 trigger mask configuration register. (HW_BARRIER_5_TRIG_MASK)	58
Cluster hardware barrier 5 status register. (HW_BARRIER_5_STATUS)	58
Cluster hardware barrier summary status register. (HW_BARRIER_5_STATUS_SUM)	58
Cluster hardware barrier 5 target mask configuration register. (HW_BARRIER_5_TARGET_MASK)	58
Cluster hardware barrier 5 trigger command register. (HW_BARRIER_5_TRIG)	59
Cluster hardware barrier 5 self trigger command register. (HW_BARRIER_5_SELF_TRIG)	59
Cluster hardware barrier 5 trigger and wait command register. (HW_BARRIER_5_TRIG_WAIT)	59
Cluster hardware barrier 5 trigger, wait and clear command register. (HW_BARRIER_5_TRIG_WAIT_CLEAR)	59
Cluster hardware barrier 6 trigger mask configuration register. (HW_BARRIER_6_TRIG_MASK)	60
Cluster hardware barrier 6 status register. (HW_BARRIER_6_STATUS)	60
Cluster hardware barrier summary status register. (HW_BARRIER_6_STATUS_SUM)	60
Cluster hardware barrier 6 target mask configuration register. (HW_BARRIER_6_TARGET_MASK)	60
Cluster hardware barrier 6 trigger command register. (HW_BARRIER_6_TRIG)	61
Cluster hardware barrier 6 self trigger command register. (HW_BARRIER_6_SELF_TRIG)	61
Cluster hardware barrier 6 trigger and wait command register. (HW_BARRIER_6_TRIG_WAIT)	61
Cluster hardware barrier 6 trigger, wait and clear command register. (HW_BARRIER_6_TRIG_WAIT_CLEAR)	61
Cluster hardware barrier 7 trigger mask configuration register. (HW_BARRIER_7_TRIG_MASK)	62
Cluster hardware barrier 7 status register. (HW_BARRIER_7_STATUS)	62
Cluster hardware barrier summary status register. (HW_BARRIER_7_STATUS_SUM)	62
Cluster hardware barrier 7 target mask configuration register. (HW_BARRIER_7_TARGET_MASK)	62
Cluster hardware barrier 7 trigger command register. (HW_BARRIER_7_TRIG)	63
Cluster hardware barrier 7 self trigger command register. (HW_BARRIER_7_SELF_TRIG)	63
Cluster hardware barrier 7 trigger and wait command register. (HW_BARRIER_7_TRIG_WAIT)	63
Cluster hardware barrier 7 trigger, wait and clear command register. (HW_BARRIER_7_TRIG_WAIT_CLEAR)	63
Cluster instruction cache control unit	64

Cluster instruction cache control unit registers	64
Cluster instruction cache control unit registers details	64
Cluster instruction cache unit enable configuration register. (ENABLE)	64
Cluster instruction cache unit flush command register. (FLUSH)	64
Cluster instruction cache unit selective flush command register. (SEL_FLUSH)	64

# 1 HERO Features

- HERO combines:
  - a hard ARM Cortex-A multicore host processor with
  - a scalable, configurable, and extensible FPGA implementation of an open-source, silicon-proven, cluster-based manycore accelerator.
- The fully open-sourced, heterogeneous software stack of HERO supports:
  - the OpenMP 4.5 Accelerator Model, and shared virtual memory (SVM), which allows for transparent accelerator programming and thereby tremendously simplifies the porting of existing applications to create heterogeneous implementations.
- The base configuration of HERO for the Xilinx Zynq ZC706 Evaluation Kit features the following accelerator configuration:
  - 1 PULP cluster (Mr. Wolf) comprising 8 32-bit RISC-V cores,
  - 128 KiB of shared L1 scratchpad memory,
  - 4 KiB of shared L1 instruction cache,
  - 256 KiB of shared L2 scratchpad and instruction memory, our brand-new IOMMU with
    - an L1 TLB of 32 variable-sized entries, and
    - an L2 TLB of 1024 page-sized entries.

The first release of HERO includes:

- 1 PULP cluster with 8 high-performance cores: extended RISC-V ISA
  - 8 cores that execute in parallel for compute intensive tasks referred to as Cluster
- Memories:
  - A level 1 Memory (128 KB) shared by all the cores in Cluster (0 wait state memory access)
  - Memory Protection Unit
- Debug Mode
  - RISC-V debug unit (not yet available in this release)
- DMA
  - A multi-channel 1D/2D cluster-DMA controls the transactions between the L2 Memory and L1 Memory

## 2 Introduction to the HERO architecture

HERO consists of many different hard- and software components.

On the hardware side, HERO uses an ARM Cortex-A host processor combined with a RISC-V programmable manycore accelerator (PMCA) implemented on FPGA. As PMCA, HERO uses bigPULP, i.e., the big brother of the open-source, multi-core, Parallel Ultra-Low Power (PULP) computing platform jointly developed by ETH Zurich and the University of Bologna. BigPULP is based on the same cluster architecture and source code. Depending on the target FPGA, bigPULP uses one or multiple PULP clusters that share an L2 instruction and data memory, a global interconnect, synchronization infrastructure, as well as the Remap Address Block (RAB) - a software-managed I/O memory management unit - which allows accelerator to coherently access the platform's main memory including support for shared virtual memory (SVM).

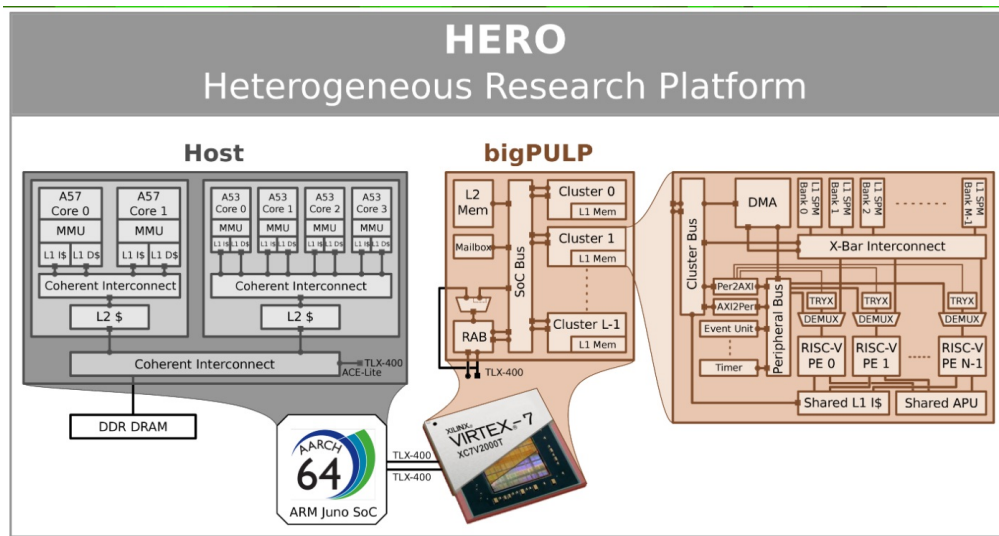


Figure 1. HERO Overview

On the software side, HERO combines two different open-source software stacks. Both comprise various modules and are contained in the HERO SDK.

The programmer writes a single application source file for the host and uses OpenMP directives for parallelization and accelerator offloading. The host-side OpenMP runtime together with the accelerator plugin takes care of the actual offloading of the target section to the accelerator. Lower-level API calls are resolved to the corresponding, device-specific implementation by the target library. The accelerator uses the PULP Software Development Kit (SDK), which also provides accelerator-specific libraries such as the Virtual Memory Management (VMM) library and the accelerator-side OpenMP runtime. The host-side HERO software stack is completed by a user-space runtime library and kernel-level device driver, which take care of system-level tasks such as booting and synchronizing with the accelerator, and allowing the the accelerator manage the shared virtual memory (SVM) subsystem. The host itself runs a full-stack Linux.

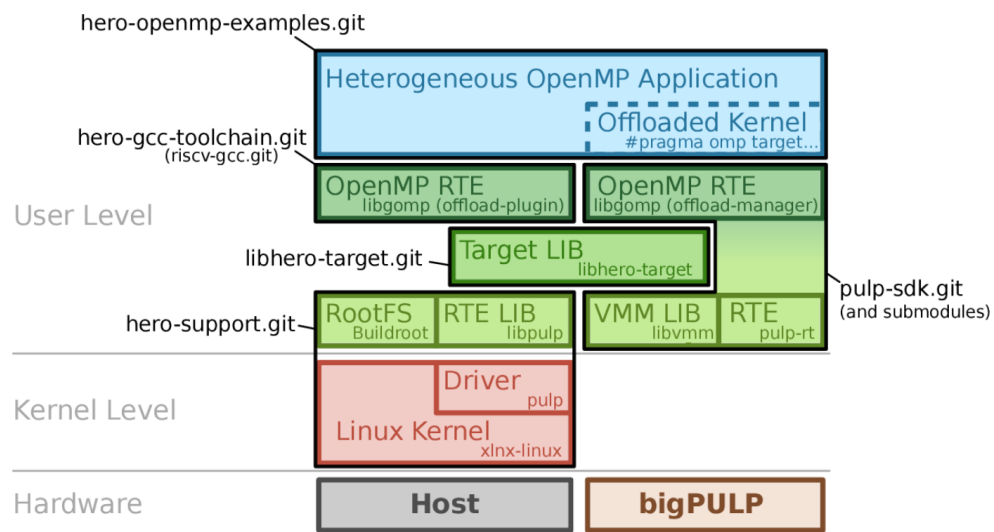


Figure 1. HERO SW stack



## 3 System on Chip

The following figure describes the main functional blocks of HERO :

PULP's hierarchical, demand-driven architecture enables ultra-low-power operation by combining:

- A cluster of 8 cores with an architecture optimized for the execution of vectorized and parallelized algorithms.

All cores and peripherals are power switchable and voltage and frequency adjustable on demand. DC/DC regulators and clock generators with ultra-fast reconfiguration times are integrated. This allows PULP to adapt extremely quickly to the processing/energy requirements of a running application. All elements share access to an L2 memory area. The cluster cores share access to an L1 (TCDM) memory area and instruction cache. Multiple DMA units allow autonomous, fast, low power transfers between cluster L1 and L2 memory and between L2 memory and external peripherals.

### 3.1 Cores

All 8 cores of the cluster share the RV32IMFCXpulp instruction set architecture, while the fabric controller can be configured as either RV32IMC or RV32IMFCXpulp. The I (integer), C (compressed instruction), M (Multiplication and division) and a portion of the supervisor ISA subsets are supported. These standard instruction sets are extended with specific instructions to optimize the performance of signal processing and machine learning algorithms. These extensions include zero-overhead hardware loops, pointer post/pre-modified memory accesses, instructions mixing control flow with computation (min, max, etc), multiply/subtract and accumulate, vector operations, fixed-point operations, bit manipulation and dot product. All of these instruction extensions are optimized by the compiler or can be used 'by hand' (more details in [RISCY User Manual] [https://PULP-platform.org/wp-content/uploads/2017/11/ri5cy\\_user\\_manual.pdf](https://PULP-platform.org/wp-content/uploads/2017/11/ri5cy_user_manual.pdf)).

### 3.2 Memory areas

There is 1 level of memory internal to HERO, shared by all the cluster cores (128kB). The cluster level 1 memory is banked and connected to the cluster cores via a logarithmic interface that is sized to provide single cycle access in 98% of cases.

Cluster L1 memory supports test-and-set functionality. The test-and-set is an atomic instruction used to write to a memory location and return its old value as a single atomic (i.e. non-interruptible) operation. If multiple processes access the same memory area and if a process is currently performing a test-and-set, no other process may begin another test-and-set until the first process is done.

The instruction caches of the cluster (4kB) will automatically cache instructions as needed. The cluster instruction cache is organized in two levels. A first private icache followed by a shared icache. Generally, the cluster cores will be executing the same area of code on different data hence the shared cluster instruction cache exploits this to reduce memory accesses for loading instructions.

The combination of a high speed shared data and instruction memory in the cluster provides an ideal memory architecture for the execution of code implementing parallelized algorithms.

### 3.3 Debug architecture

To be integrated.

### 3.4 Events and interrupts model

To be completed

### 3.5 Exceptions model

To be completed

### 3.6 Data types supported

The memories are byte addressable so every single data type whose size is a multiple of bytes can be supported either natively if the number of bytes is less or equal than 4 or through software emulation if it is larger.

### 3.7 Event Units

One event unit (EU) are available in HERO, for the [cluster](#).

The EU allows the RISCY cores to be put into sleep mode when waiting for an event to occur. In the EUs, the way of treating incoming events can be controlled. The EU can be instructed to react instantly by jumping to an interrupt routine or to delegate the treatment of the event to a software event task controller.

### 3.8 [DMA \(direct memory access\)](#)

The DMA unit allows the transfer of data between L2 and cluster L1 memory areas. 8 channels can be programmed. Channels can be 1D/2D on the L2 memory and 1D on the cluster L1 side.

### 3.9 Performance counters

Each RISCY cores of the FC and the cluster provide a performance counter. These 32-bit counters can be configured to count the:

- Total number of cycles (also includes the cycles where the core is sleeping)
- Number of cycles the core was active (not sleeping)
- Number of instructions executed
- Number of load data hazards
- Number of jump register data hazards
- Number of cycles waiting for instruction fetches, i.e. number of instructions wasted due to non-ideal caching
- Number of data memory loads executed. Misaligned accesses are counted twice
- Number of data memory stores executed. Misaligned accesses are counted twice
- Number of unconditional jumps (j, jal, jr, jalr)
- Number of both taken and not taken branches
- Number of taken branches
- Number of compressed instructions executed
- Number of memory loads to EXT executed. Misaligned accesses are counted twice. Every non-L1 access is considered external (cluster only)
- Number of memory stores to EXT executed. Misaligned accesses are counted twice. Every non-L1 access is considered external (cluster only)
- Number of cycles used for memory loads to EXT. Every non-L1 access is considered external (cluster only)
- Number of cycles used for memory stores to EXT. Every non-L1 access is considered external (cluster only)
- Number of cycles wasted due to L1/log-interconnect contention (cluster only)
- Number of cycles wasted due to CSR access

## 4 Memory map

The following table describes PULP's memory map. All areas in this map are addressable from any RI5CY cores. The aliased area at the start of the memory map has a different meaning when addressed from FC or Cluster RI5CY cores. See the section on aliasing below.

		Address range
Aliased Memory Area		0x00000000 - 0x003FFFFFFF
	Aliased memory map area	0x1B000000 - 0x1B3FFFFFFF
Cluster Subsystem		0x10000000 - 0x103FFFFFFF
	Cluster L1 RAM (128kB)	0x10000000 - 0x1001FFFF
	Cluster L1 memory test and set unit	0x10100000 - 0x1011FFFF
	<a href="#">Cluster control unit</a>	0x10200000 - 0x102003FF
	<a href="#">Cluster timer</a>	0x10200400 - 0x102007FF
	<a href="#">Cluster event unit</a>	0x10200800 - 0x10200FFF
	<a href="#">Cluster instruction cache control unit</a>	0x10201400 - 0x102017FF
	CL_EU_CORE	0x10204000 - 0x102043FF
	DMA	0x10204400 - 0x10204BFF
L2 Memory		0x1C000000 - 0x1FFFFFFF
	L2 RAM (512kB)	0x1C000000 - 0x1C7FFFFFFF
	L2 RAM PRIVATE CH 0	0x1C000000 - 0x1C007FFF
	L2 RAM PRIVATE CH 0 (SCM)	0x1C006000 - 0x1C007FFF
	L2 RAM PRIVATE CH 1	0x1C008000 - 0x1C00FFFF
	L2 RAM INTERLEAVED (TCDM)	0x1C010000 - 0x1C091FFF

Table 1. Pulp memory map table

### 4.1 Aliased memory map

A reserved section of addresses in the overall memory map have specific meaning when addressed from the FC or Cluster RI5CY cores. These are called aliased addresses. They should be preferred over standard addresses since the access will be faster (1 or 2 clock cycles).

### 4.2 Cluster aliased address map

Functional unit	Aliased address range
Cluster L1 RAM (128kB)	0x1B000000 - 0x1B01FFFF
Cluster L1 memory test and set unit	0x1B100000 - 0x1B11FFFF
<a href="#">Cluster control unit</a>	0x1B200000 - 0x1B2003FF
<a href="#">Cluster timer</a>	0x1B200400 - 0x1B2007FF
<a href="#">Cluster instruction cache control unit</a>	0x1B201400 - 0x1B2017FF
CL_EU_CORE	0x1B204000 - 0x1B2043FF

Table 2. Cluster Subsystem aliased memory map table

## 5 Device components description

### 5.1 RISC-V cores

The FC and cluster cores in PULP are based on the PULP RI5CY core.

RI5CY supports the following instructions:

- Full support for RV32I Base Integer Instruction Set
- Full support for RV32C Standard Extension for Compressed Instructions
- Partial support for RV32M Standard Extension for Integer Multiplication and Division. Multiplication only.
- The Fabric Controller core supports a subset of the draft RISC-V privileged architecture supporting M and U modes.

The RI5CY core design has been extended with the instructions in the sections below. The RI5CY core itself implements extensions to the RISC-V instruction set. The datasheet for the RI5CY core can be found at [https://PULP-platform.org/wp-content/uploads/2017/11/ri5cy\\_user\\_manual.pdf](https://PULP-platform.org/wp-content/uploads/2017/11/ri5cy_user_manual.pdf). Please note that the RI5CY cores in PULP do not contain a floating point unit. For further explanation of the RISC-V instruction format please refer to the RISC-V standards documents.

#### 5.1.1 Complex number operations

*Complex number representation for multiply and subtract operations*

$C = \{Re, Im\}$  represented as a vector of 2 16bits signed numbers. Using gcc vector notation  $C[0]$  is the real part,  $C[1]$  is the imaginary part.

Position in register in little endian:

- Re : bits[15:0]
- Im : bits[31:16]

##### 5.1.1.1 Complex multiplication operations

Mnemonic	Description
pv.cplxmul.s rD, rA, rB	Vector by Vector $rD[15:0] = (rA[15:0] * rB[15:0] - rA[31:16] * rB[31:16]) \gg 15$ $rD[31:16] = (rA[15:0] * rB[31:16] + rA[31:16] * rB[15:0]) \gg 15$
pv.cplxmul.s.div2 rD, rA, rB	Vector by Vector, Div2 $rD[15:0] = (rA[15:0] * rB[15:0] - rA[31:16] * rB[31:16]) \gg 16$ $rD[31:16] = (rA[15:0] * rB[31:16] + rA[31:16] * rB[15:0]) \gg 16$
pv.cplxmul.s.div4 rD, rA, rB	Vector by Vector, Div4 $rD[15:0] = (rA[15:0] * rB[15:0] - rA[31:16] * rB[31:16]) \gg 17$ $rD[31:16] = (rA[15:0] * rB[31:16] + rA[31:16] * rB[15:0]) \gg 17$
pv.cplxmul.s.sc rD, rA, rB	Vector by Scalar $rD[15:0] = (rA[15:0] * rB[15:0] - rA[31:16] * rB[15:0]) \gg 15$ $rD[31:16] = (rA[15:0] * rB[15:0] + rA[31:16] * rB[15:0]) \gg 15$
pv.cplxmul.s.sci	Vector by I6 Scalar $rD[15:0] = (rA[15:0] * ExtS(I6) - rA[31:16] * ExtS(I6)) \gg 15$ $rD[31:16] = (rA[15:0] * ExtS(I6) + rA[31:16] * ExtS(I6)) \gg 15$

Table 3. Complex multiplication operations summary table

35	31	26	25	24	20	19	15	14	12	17	7	6	0
funct5	F	S	rs2	rs1	funct3	rD	opcode						
01010	1	0	src2	src1	000	dest	1010111						
01010	1	0	src2	src1	010	dest	1010111						
01010	1	1	src2	src1	010	dest	1010111						
01010	1	0	src2	src1	100	dest	1010111						
01010	1	Imm6[5:0]s		src1	110	dest	1010111						

Table 4. Complex multiplication operations format table

### 5.1.1.2 Subtraction of 2 complexes with post rotation by -j

$R = \text{subrotmj}(X, Y)$ ,  $R$ ,  $X$ ,  $Y$  complexes.

$R.\text{Re} = X.\text{Im} - Y.\text{Im}$   $R.\text{Im} = Y.\text{Re} - X.\text{Re}$

It can be viewed as  $(XY)$  rotated by 90 degrees (multiplied by  $j$ ).

Mnemonic	Description
pv.subrotmj.h rD, rA, rB	$rD[15:0] = rA[31:16] - rB[31:16]$ $rD[31:16] = rB[15:0] - rA[15:0]$
pv.subrotmj.h.div2 rD, rA, rB	$rD[15:0] = (rA[31:16] - rB[31:16]) \gg 1$ $rD[31:16] = (rB[15:0] - rA[15:0]) \gg 1$
pv.subrotmj.h.div4 rD, rA, rB	$rD[15:0] = (rA[31:16] - rB[31:16]) \gg 2$ $rD[31:16] = (rB[15:0] - rA[15:0]) \gg 2$

Table 5. Subtraction of 2 complexes with post rotation by -j summary table

35	31	26	25	24	20	19	15	14	12	17	7	6	0
funct5	F	S	rs2	rs1	funct3	rD	opcode						
01101	1	0	src2	src1	000	dest	1010111						
01101	1	0	src2	src1	010	dest	1010111						
01101	1	1	src2	src1	010	dest	1010111						

Table 6. Subtraction of 2 complexes with post rotation by -j format table

### 5.1.1.3 Complex conjugate operation

Mnemonic	Description
pv.cplxconj.h rD, rA	$rD[15:0] = rA[15:0]$ $rD[31:16] = -rA[31:16]$

Table 7. Complex conjugate operation summary table

35	31	26	25	24	2019	1514	1217	76	
funct5		F	S	rs2	rs1	funct3		rD	opcode
01011		1	000000		src1	000		dest	1010111

Table 8. Complex conjugate operation format table

## 5.1.2 Extensions to existing RI5CY vector operations.

### 5.1.2.1 Addition of vector of half words with post right shift

Extends the existing pv.add instruction.

Mnemonic	Description
pv.add.h.div2 rD, rA, rB	$rD[15:0] = (rA[15:0] + rB[15:0]) \gg 1$ $rD[31:16] = (rA[31:16] + rB[31:16]) \gg 1$
pv.add.b.div2 rD, rA, rB	$rD[7:0] = (rA[7:0] + rB[7:0]) \gg 1$ $rD[15:8] = (rA[15:8] + rB[15:8]) \gg 1$ $rD[23:16] = (rA[23:16] + rB[23:16]) \gg 1$ $rD[31:24] = (rA[31:24] + rB[31:24]) \gg 1$
pv.add.h.div4 rD, rA, rB	$rD[15:0] = (rA[15:0] + rB[15:0]) \gg 2$ $rD[31:16] = (rA[31:16] + rB[31:16]) \gg 2$

Mnemonic	Description
pv.add.b.div4 rD, rA, rB	$rD[7:0] = (rA[7:0] + rB[7:0]) \gg 2$ $rD[15:8] = (rA[15:8] + rB[15:8]) \gg 2$ $rD[23:16] = (rA[23:16] + rB[23:16]) \gg 2$ $rD[31:24] = (rA[31:24] + rB[31:24]) \gg 2$

Table 9. Addition of vector of half words with post right shift summary table

35	31	26	25	24	20	19	15	14	12	17	7	6	0
funct5	F	S	rs2	rs1	funct3	rD	opcode						
00000	0	0	src2	src1	010	dest	1010111						
00000	0	0	src2	src1	011	dest	1010111						
00000	0	1	src2	src1	010	dest	1010111						
00000	0	1	src2	src1	011	dest	1010111						

Table 10. Addition of vector of half words with post right shift format table

### 5.1.2.2 Subtraction of vectors of half words with post right shift

Extends the existing pv.sub instruction

Mnemonic	Description
pv.sub.h.div2 rD, rA, rB	$rD[15:0] = (rA[15:0] - rB[15:0]) \gg 1$ $rD[31:16] = (rA[31:16] - rB[31:16]) \gg 1$
pv.sub.b.div2 rD, rA, rB	$rD[7:0] = (rA[7:0] - rB[7:0]) \gg 1$ $rD[15:8] = (rA[15:8] - rB[15:8]) \gg 1$ $rD[23:16] = (rA[23:16] - rB[23:16]) \gg 1$ $rD[31:24] = (rA[31:24] - rB[31:24]) \gg 1$
pv.sub.h.div4 rD, rA, rB	$rD[15:0] = (rA[15:0] - rB[15:0]) \gg 2$ $rD[31:16] = (rA[31:16] - rB[31:16]) \gg 2$
pv.sub.b.div4 rD, rA, rB	$rD[7:0] = (rA[7:0] - rB[7:0]) \gg 2$ $rD[15:8] = (rA[15:8] - rB[15:8]) \gg 2$ $rD[23:16] = (rA[23:16] - rB[23:16]) \gg 2$ $rD[31:24] = (rA[31:24] - rB[31:24]) \gg 2$

Table 11. Subtraction of vectors of half words with post right shift summary table

35	31	26	25	24	20	19	15	14	12	17	7	6	0
funct5	F	S	rs2	rs1	funct3	rD	opcode						
00001	0	0	src2	src1	010	dest	1010111						
00001	0	0	src2	src1	011	dest	1010111						
00001	0	1	src2	src1	010	dest	1010111						
00001	0	1	src2	src1	011	dest	1010111						

Table 12. Subtraction of vectors of half words with post right shift format table

### 5.1.3 Viterbi specific instructions

Selection of the survivor path relies on which branch the vector max2 has taken for each of its vector sub elements. The usual approach here is to have a variant of max setting 2 hardware flags to keep trace of which input operand has been selected for the elaboration of the max2. These 2 flags are then used by a selection operation taking 2 input vectors and the 2 flags and produce as an output a vector that is a selection of shifted by 1 to the left inputs with bit0 set according to flags.

Mnemonic	Description
----------	-------------

Mnemonic	Description
pv.vitop.max rD, rA, rB	$rD[31:16] = \max(rA[31:16], rB[31:16])$ $rD[15:0] = \max(rA[15:0], rB[15:0])$ $\$VF0 = (rA[31:16] \leq rB[31:16])$ $\$VF1 = (rA[15:0] \leq rB[15:0])$
pv.vitop.sel rD, rA, rB	$rD[31:16] = ((rA[31:16] << 1) \& \! \$VF0)   (rB[31:16] << 1   \$VF0)$ $rD[15:0] = ((rA[15:0] << 1) \& \! \$VF1)   (rB[15:0] << 1   \$VF1)$

Table 13. Viterbi specific instructions summary table

35	31	26	25	24	20	19	15	14	12	17	7	6	0
funct5	F	S	rs2	rs1	funct3	rD	opcode						
01100	1	0	src2	src1	001	dest	1010111						
01100	1	0	src2	src1	000	dest	1010111						

Table 14. Viterbi specific instructions format table

### 5.1.4 Enhanced shuffling

Extension to RI5CY pv.pack.h rD, rA, rB operation

Mnemonic	Description
pv.pack.h.h rD, rA, rB	$rD[31:16] = rA[31:16]$ $rD[15:0] = rB[31:16]$
pv.pack.l.h rD, rA, rB	$rD[31:16] = rA[15:0]$ $rD[15:0] = rB[15:0]$

Table 15. Enhanced shuffling summary table

35	31	26	25	24	20	19	15	14	12	17	7	6	0
funct5	F	S	rs2	rs1	funct3	rD	opcode						
11010	0	0	src2	src1	110	dest	1010111						
11010	0	0	src2	src1	100	dest	1010111						

Table 16. Enhanced shuffling format table

## 5.2 Cluster Subsystem

### 5.2.1 Cluster Subsystem Events

Event number	Event name	IP instance name	Direction	Description
0	SW_EVT_0	<a href="#">Cluster RISCY cores</a>	Input	Software event 0 from one of the CL_COREs
1	SW_EVT_1	<a href="#">Cluster RISCY cores</a>	Input	Software event 1 from one of the CL_COREs
2	SW_EVT_2	<a href="#">Cluster RISCY cores</a>	Input	Software event 2 from one of the CL_COREs
3	SW_EVT_3	<a href="#">Cluster RISCY cores</a>	Input	Software event 3 from one of the CL_COREs
4	SW_EVT_4	<a href="#">Cluster RISCY cores</a>	Input	Software event 4 from one of the CL_COREs
5	SW_EVT_5	<a href="#">Cluster RISCY cores</a>	Input	Software event 5 from one of the CL_COREs
6	SW_EVT_6	<a href="#">Cluster RISCY cores</a>	Input	Software event 6 from one of the CL_COREs
7	SW_EVT_7	<a href="#">Cluster RISCY cores</a>	Input	Software event 7 from one of the CL_COREs
8	DMA_EVT_0	DMA	Input	DMA event 0
9	DMA_EVT_1	DMA	Input	DMA event 1
10	CL_TIMER_LO_EVT	<a href="#">CL_TIMER_UNIT</a>	Input	Cluster basic timer low event
11	CL_TIMER_HI_EVT	<a href="#">CL_TIMER_UNIT</a>	Input	Cluster basic timer high event
12	Reserved		Input	Reserved
13	Reserved		Input	Reserved
14	Reserved			Reserved
15	Reserved			Reserved
16	BARRIER_EVT	CL_EVENT_UNIT	Input	Barrier event
17	MUTEX_EVT	CL_EVENT_UNIT	Input	Mutex event
18	DISPATCHER_EVT	CL_EVENT_UNIT	Input	Dispatcher event
19	Reserved			Reserved
20	Reserved			Reserved
21	Reserved			Reserved
22	Reserved			Reserved
23	Reserved			Reserved
24	decompressor done event			Reserved
25	Reserved			Reserved
26	Reserved			Reserved
27	periph_fifo_event	SOC_EVENT_GENERATOR	Input	SoC peripheral event
28	Reserved			Reserved
29	Reserved			Reserved
30	Reserved			Reserved
31	Reserved			Reserved

Table 17. Cluster Subsystem Events table

### 5.2.2 Cluster control unit

CL\_CTRL\_UNIT component manages the following features:

- End of Computation status flag
- Configurable fetch activation for all cores of the Cluster
- Configurable core 0 boot address to define where to fetch first instruction in CL\_CORE\_0 after releasing the reset
- Configurable full cluster clock gating
- Configurable Cluster L1 memory arbitration policy
- Cluster cores resume command control



- Cluster cores halt status flags
- Configurable cluster cores debug halt command group mask policy

None

### 5.2.2.1 Cluster control unit registers

Name	Address	Aliased address	Size	Type	Access	Default	Description
<a href="#">EOC</a>	0x10200000	0x1B200000	32	Status	R/W	0x0000	End Of Computation status register.
<a href="#">FETCH_EN</a>	0x10200008	0x1B200008	32	Config	R/W	0x0000	Cluster cores fetch enable configuration register.
<a href="#">CLOCK_GATE</a>	0x10200020	0x1B200020	32	Config	R/W	0x0000	Cluster clock gate configuration register.
<a href="#">DBG_RESUME</a>	0x10200028	0x1B200028	32	Config	W	0x0000	Cluster cores debug resume register.
<a href="#">DBG_HALT_STATUS</a>	0x10200028	0x1B200028	32	Config	R	0x0000	Cluster cores debug halt status register.
<a href="#">DBG_HALT_MASK</a>	0x10200038	0x1B200038	32	Config	R/W	0x0000	Cluster cores debug halt mask configuration register.
<a href="#">BOOT_ADDR0</a>	0x10200040	0x1B200040	32	Config	R/W	0x0000	Cluster core 0 boot address configuration register.
<a href="#">TCDM_ARB_POLICY_CH0</a>	0x10200080	0x1B200080	32	Config	R/W	0x0000	TCDM arbitration policy ch0 for cluster cores configuration register.
<a href="#">TCDM_ARB_POLICY_CH1</a>	0x10200088	0x1B200088	32	Config	R/W	0x0000	TCDM arbitration policy ch1 for DMA/HWCE configuration register.
<a href="#">TCDM_ARB_POLICY_CH0_REP</a>	0x102000C0	0x1B2000C0	32	Config	R/W	0x0000	Read only duplicate of TCDM_ARB_POLICY_CH0 register
<a href="#">TCDM_ARB_POLICY_CH1_REP</a>	0x102000C8	0x1B2000C8	32	Config	R/W	0x0000	Read only duplicate of TCDM_ARB_POLICY_CH1 register

Table 18. Cluster control unit registers table

### 5.2.2.2 Cluster control unit registers details

#### 5.2.2.2.1 End Of Computation status register. (EOC)

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															EOC

Bit 0 - **EOC** (R/W)

End of computation status flag bitfield:

- 0b0: program execution under going
- 0b1: end of computation reached

#### 5.2.2.2.2 Cluster cores fetch enable configuration register. (FETCH\_EN)

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								CORE7	CORE6	CORE5	CORE4	CORE3	CORE2	CORE1	CORE0

#### Bit 7 - **CORE7** (R/W)

Core 7 fetch enable configuration bitfield:

- 0b0: disabled
- 0b1: enabled

#### Bit 6 - **CORE6** (R/W)

Core 6 fetch enable configuration bitfield:

- 0b0: disabled
- 0b1: enabled

#### Bit 5 - **CORE5** (R/W)

Core 5 fetch enable configuration bitfield:

- 0b0: disabled
- 0b1: enabled

#### Bit 4 - **CORE4** (R/W)

Core 4 fetch enable configuration bitfield:

- 0b0: disabled
- 0b1: enabled

#### Bit 3 - **CORE3** (R/W)

Core 3 fetch enable configuration bitfield:

- 0b0: disabled
- 0b1: enabled

#### Bit 2 - **CORE2** (R/W)

Core 2 fetch enable configuration bitfield:

- 0b0: disabled
- 0b1: enabled

#### Bit 1 - **CORE1** (R/W)

Core 1 fetch enable configuration bitfield:

- 0b0: disabled
- 0b1: enabled

Bit 0 - **CORE0** (R/W)

Core 0 fetch enable configuration bitfield:

- *0b0*: disabled
- *0b1*: enabled

**5.2.2.2.3 Cluster clock gate configuration register. (CLOCK\_GATE)***Reset value: 0x0000*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															EN

Bit 0 - **EN** (R/W)

Cluster clock gate configuration bitfield:

- *0b0*: disabled
- *0b1*: enabled

**5.2.2.2.4 Cluster cores debug resume register. (DBG\_RESUME)***Reset value: 0x0000*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								CORE7	CORE6	CORE5	CORE4	CORE3	CORE2	CORE1	CORE0

Bit 7 - **CORE7** (W)

Core 7 debug resume configuration bitfield:

- *0b0*: stay halted
- *0b1*: resume core 7

Bit 6 - **CORE6** (W)

Core 6 debug resume configuration bitfield:

- *0b0*: stay halted
- *0b1*: resume core 6

Bit 5 - **CORE5** (W)

Core 5 debug resume configuration bitfield:

- *0b0*: stay halted
- *0b1*: resume core 5

Bit 4 - **CORE4** (W)

Core 4 debug resume configuration bitfield:

- *0b0*: stay halted
- *0b1*: resume core 4

Bit 3 - **CORE3** (*W*)

Core 3 debug resume configuration bitfield:

- *0b0*: stay halted
- *0b1*: resume core 3

Bit 2 - **CORE2** (*W*)

Core 2 debug resume configuration bitfield:

- *0b0*: stay halted
- *0b1*: resume core 2

Bit 1 - **CORE1** (*W*)

Core 1 debug resume configuration bitfield:

- *0b0*: stay halted
- *0b1*: resume core 1

Bit 0 - **CORE0** (*W*)

Core 0 debug resume configuration bitfield:

- *0b0*: stay halted
- *0b1*: resume core 0

**5.2.2.2.5 Cluster cores debug halt status register. (DBG\_HALT\_STATUS)***Reset value: 0x0000*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								CORE7	CORE6	CORE5	CORE4	CORE3	CORE2	CORE1	CORE0

Bit 7 - **CORE7** (*R*)

Core 7 debug halt status flag bitfield:

- *0b0*: running
- *0b1*: halted

Bit 6 - **CORE6** (*R*)

Core 6 debug halt status flag bitfield:

- *0b0*: running
- *0b1*: halted

Bit 5 - **CORE5** (*R*)

Core 5 debug halt status flag bitfield:

- *0b0*: running
- *0b1*: halted

Bit 4 - **CORE4** (*R*)

Core 4 debug halt status flag bitfield:

- *0b0*: running
- *0b1*: halted

Bit 3 - **CORE3** (*R*)

Core 3 debug halt status flag bitfield:

- *0b0*: running
- *0b1*: halted

Bit 2 - **CORE2** (*R*)

Core 2 debug halt status flag bitfield:

- *0b0*: running
- *0b1*: halted

Bit 1 - **CORE1** (*R*)

Core 1 debug halt status flag bitfield:

- *0b0*: running
- *0b1*: halted

Bit 0 - **CORE0** (*R*)

Core 0 debug halt status flag bitfield:

- *0b0*: running
- *0b1*: halted

**5.2.2.2.6 Cluster cores debug halt mask configuration register. (DBG\_HALT\_MASK)***Reset value: 0x0000*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								CORE7	CORE6	CORE5	CORE4	CORE3	CORE2	CORE1	CORE0

Bit 7 - **CORE7** (*R/W*)

Core 7 debug halt mask bitfield. When bit is set, core will be part of mask group and stopped when one of the members of the group stops.

Bit 6 - **CORE6** (*R/W*)

Core 6 debug halt mask bitfield. When bit is set, core will be part of mask group and stopped when one of the members of the group stops.

Bit 5 - **CORE5** (*R/W*)

Core 5 debug halt mask bitfield. When bit is set, core will be part of mask group and stopped when one of the members of the group stops.

Bit 4 - **CORE4** (R/W)

Core 4 debug halt mask bitfield. When bit is set, core will be part of mask group and stopped when one of the members of the group stops.

Bit 3 - **CORE3** (R/W)

Core 3 debug halt mask bitfield. When bit is set, core will be part of mask group and stopped when one of the members of the group stops.

Bit 2 - **CORE2** (R/W)

Core 2 debug halt mask bitfield. When bit is set, core will be part of mask group and stopped when one of the members of the group stops.

Bit 1 - **CORE1** (R/W)

Core 1 debug halt mask bitfield. When bit is set, core will be part of mask group and stopped when one of the members of the group stops.

Bit 0 - **CORE0** (R/W)

Core 0 debug halt mask bitfield. When bit is set, core will be part of mask group and stopped when one of the members of the group stops.

## 5.2.2.2.7 Cluster core 0 boot address configuration register. (BOOT\_ADDR0)

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BA															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BA															

Bits 31:0 - **BA** (R/W)

Cluster core 0 boot address configuration bitfield.

## 5.2.2.2.8 TCDM arbitration policy ch0 for cluster cores configuration register. (TCDM\_ARB\_POLICY\_CH0)

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															POL

Bit 0 - **POL** (R/W)

TCDM arbitration policy for cluster cores configuration bitfield:

- 0b0: fair round robin
- 0b1: fixed order

## 5.2.2.2.9 TCDM arbitration policy ch1 for DMA/HWCE configuration register. (TCDM\_ARB\_POLICY\_CH1)

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															POL

Bit 0 - **POL** (R/W)

TCDM arbitration policy for DMA/HWCE configuration bitfield:

- *0b0*: fair round robin
- *0b1*: fixed order

**5.2.2.2.10 Read only duplicate of TCDM\_ARB\_POLICY\_CH0 register (TCDM\_ARB\_POLICY\_CH0\_REP)***Reset value: 0x0000*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															POL

Bit 0 - **POL** (R/W)

TCDM arbitration policy for cluster cores configuration bitfield:

- *0b0*: fair round robin
- *0b1*: fixed order

**5.2.2.2.11 Read only duplicate of TCDM\_ARB\_POLICY\_CH1 register (TCDM\_ARB\_POLICY\_CH1\_REP)***Reset value: 0x0000*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															POL

Bit 0 - **POL** (R/W)

TCDM arbitration policy for DMA/HWCE configuration bitfield:

- *0b0*: fair round robin
- *0b1*: fixed order

**5.2.3 Cluster timer**

BASIC TIMER component manages the following features:

- 2 general purpose 32bits up counter timers
- Input trigger sources:
  - FLL clock
  - FLL clock + Prescaler
  - Reference clock at 32kHz
  - External event
- 8bit programmable prescaler to FLL clock
- Counting modes:
  - One shot mode: timer is stopped after first comparison match
  - Continuous mode: timer continues counting after comparison match
  - Cycle mode: timer resets to 0 after comparison match and continues counting

- 64 bit cascaded mode
- Interrupt request generation on comparison match

None

### 5.2.3.1 Cluster timer registers

Name	Address	Aliased address	Size	Type	Access	Default	Description
<a href="#">CFG_LO</a>	0x10200400	0x1B200400	32	Config	R/W	0x0000	Timer Low Configuration register.
<a href="#">CFG_HI</a>	0x10200404	0x1B200404	32	Config	R/W	0x0000	Timer High Configuration register.
<a href="#">CNT_LO</a>	0x10200408	0x1B200408	32	Data	R/W	0x0000	Timer Low counter value register.
<a href="#">CNT_HI</a>	0x1020040C	0x1B20040C	32	Data	R/W	0x0000	Timer High counter value register.
<a href="#">CMP_LO</a>	0x10200410	0x1B200410	32	Config	R/W	0x0000	Timer Low comparator value register.
<a href="#">CMP_HI</a>	0x10200414	0x1B200414	32	Config	R/W	0x0000	Timer High comparator value register.
<a href="#">START_LO</a>	0x10200418	0x1B200418	32	Config	R/W	0x0000	Start Timer Low counting register.
<a href="#">START_HI</a>	0x1020041C	0x1B20041C	32	Config	R/W	0x0000	Start Timer High counting register.
<a href="#">RESET_LO</a>	0x10200420	0x1B200420	32	Config	R/W	0x0000	Reset Timer Low counter register.
<a href="#">RESET_HI</a>	0x10200424	0x1B200424	32	Config	R/W	0x0000	Reset Timer High counter register.

Table 19. Cluster timer registers table

### 5.2.3.2 Cluster timer registers details

#### 5.2.3.2.1 Timer Low Configuration register. (CFG\_LO)

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CASC	Reserved														
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PVAL								CCFG	PEN	ONE_S	MODE	Reserved	IRQEN	RESET	ENABLE

Bit 31 - **CASC** (R/W)

Timer low + Timer high 64bit cascaded mode configuration bitfield.

Bits 15:8 - **PVAL** (R/W)

Timer low prescaler value bitfield.  $F_{\text{timer}} = F_{\text{clk}} / (1 + \text{PRESC\_VAL})$

Bit 7 - **CCFG** (R/W)

Timer low clock source configuration bitfield:

- 0b0: FLL or FLL+Prescaler
- 0b1: Reference clock at 32kHz

Bit 6 - **PEN** (R/W)

Timer low prescaler enable configuration bitfield:

- 0b0: disabled
- 0b1: enabled



Bit 5 - **ONE\_S** (R/W)

Timer low one shot configuration bitfield:

- *0b0*: let Timer low enabled counting when compare match with CMP\_LO occurs.
- *0b1*: disable Timer low when compare match with CMP\_LO occurs.

Bit 4 - **MODE** (R/W)

Timer low continuous mode configuration bitfield:

- *0b0*: Continue mode - continue incrementing Timer low counter when compare match with CMP\_LO occurs.
- *0b1*: Cycle mode - reset Timer low counter when compare match with CMP\_LO occurs.

Bit 2 - **IRQEN** (R/W)

Timer low compare match interrupt enable configuration bitfield:

- *0b0*: disabled
- *0b1*: enabled

Bit 1 - **RESET** (R/W)

Timer low counter reset command bitfield. Cleared after Timer Low reset execution.

Bit 0 - **ENABLE** (R/W)

Timer low enable configuration bitfield:

- *0b0*: disabled
- *0b1*: enabled

**5.2.3.2.2 Timer High Configuration register. (CFG\_HI)***Reset value: 0x0000*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								CLKCFG	PEN	ONE_S	MODE	Reserved	IRQEN	RESET	ENABLE

Bit 7 - **CLKCFG** (R/W)

Timer high clock source configuration bitfield:

- *0b0*: FLL or FLL+Prescaler
- *0b1*: Reference clock at 32kHz

Bit 6 - **PEN** (R/W)

Timer high prescaler enable configuration bitfield:

- *0b0*: disabled
- *0b1*: enabled

Bit 5 - **ONE\_S** (R/W)

Timer high one shot configuration bitfield:

- *0b0*: let Timer high enabled counting when compare match with CMP\_LO occurs.
- *0b1*: disable Timer high when compare match with CMP\_LO occurs.

Bit 4 - **MODE** (R/W)

Timer high continuous mode configuration bitfield:

- *0b0*: Continue mode - continue incrementing Timer high counter when compare match with CMP\_LO occurs.
- *0b1*: Cycle mode - reset Timer high counter when compare match with CMP\_LO occurs.

Bit 2 - **IRQEN** (R/W)

Timer high compare match interrupt enable configuration bitfield:

- *0b0*: disabled
- *0b1*: enabled

Bit 1 - **RESET** (W)

Timer high counter reset command bitfield. Cleared after Timer high reset execution.

Bit 0 - **ENABLE** (R/W)

Timer high enable configuration bitfield:

- *0b0*: disabled
- *0b1*: enabled

**5.2.3.2.3 Timer Low counter value register. (CNT\_LO)***Reset value: 0x0000*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNT_LO															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT_LO															

Bits 31:0 - **CNT\_LO** (R/W)

Timer Low counter value bitfield.

**5.2.3.2.4 Timer High counter value register. (CNT\_HI)***Reset value: 0x0000*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNT_HI															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT_HI															

Bits 31:0 - **CNT\_HI** (R/W)

Timer High counter value bitfield.

**5.2.3.2.5 Timer Low comparator value register. (CMP\_LO)***Reset value: 0x0000*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CMP_LO															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMP_LO															

Bits 31:0 - **CMP\_LO** (R/W)

Timer Low comparator value bitfield.

#### 5.2.3.2.6 Timer High comparator value register. (CMP\_HI)

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CMP_HI															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMP_HI															

Bits 31:0 - **CMP\_HI** (R/W)

Timer High comparator value bitfield.

#### 5.2.3.2.7 Start Timer Low counting register. (START\_LO)

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															STRT_LO

Bit 0 - **STRT\_LO** (W)

Timer Low start command bitfield. When executed, CFG\_LO.ENABLE is set.

#### 5.2.3.2.8 Start Timer High counting register. (START\_HI)

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															STRT_HI

Bit 0 - **STRT\_HI** (W)

Timer High start command bitfield. When executed, CFG\_HI.ENABLE is set.

#### 5.2.3.2.9 Reset Timer Low counter register. (RESET\_LO)

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															RST_LO

Bit 0 - **RST\_LO** (W)

Timer Low counter reset command bitfield. When executed, CFG\_LO.RESET is set.

### 5.2.3.2.10 Reset Timer High counter register. (RESET\_HI)

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															RST_HI

Bit 0 - **RST\_HI** (W)

Timer High counter reset command bitfield. When executed, CFG\_HI.RESET is set.

## 5.2.4 Cluster event unit

Cluster event unit component manages the following features:

- Cluster software events generation
- Cluster cores clock gate control
- Wait for event functionality
- Input event mask configuration
- Cluster cores IRQ generation
- 2 hardware mutex
- 8 hardware barriers
- 1 message dispatcher

Events managed by Cluster event unit are:

- 1 SoC peripheral event: when this event occurs, the SoC peripheral events fifo must be read to get the SoC event ID.
- 1 message dispatcher event
- 1 barrier event
- up to 4 hardware accelerator events
- 2 Cluster timer events
- 2 DMA events
- 8 software events that can come from cluster cores directly or external triggering.

None

### 5.2.4.1 Cluster event unit registers

Name	Address	Size	Type	Access	Default	Description
<a href="#">EVT_MASK_CORE0</a>	0x10200800	32	Config	R/W	0x0000	Input event mask configuration register.
<a href="#">EVT_MASK_AND_CORE0</a>	0x10200804	32	Config	W	0x0000	Input event mask update command register with bitwise AND operation.
<a href="#">EVT_MASK_OR_CORE0</a>	0x10200808	32	Config	W	0x0000	Input event mask update command register with bitwise OR operation.
<a href="#">IRQ_MASK_CORE0</a>	0x1020080C	32	Config	R/W	0x0000	Interrupt request mask configuration register.
<a href="#">IRQ_MASK_AND_CORE0</a>	0x10200810	32	Config	W	0x0000	Interrupt request mask update command register with bitwise AND operation.

Name	Address	Size	Type	Access	Default	Description
<a href="#">IRQ_MASK_OR_CORE0</a>	0x10200814	32	Config	W	0x0000	Interrupt request mask update command register with bitwise OR operation.
<a href="#">CLOCK_STATUS_CORE0</a>	0x10200818	32	Config	R	0x0000	Cluster cores clock status register.
<a href="#">EVENT_BUFFER_CORE0</a>	0x1020081C	32	Config	R	0x0000	Pending input events status register.
<a href="#">EVENT_BUFFER_MASKED_CORE0</a>	0x10200820	32	Config	R	0x0000	Pending input events status register with EVT_MASK applied.
<a href="#">EVENT_BUFFER_IRQ_MASKED_CORE0</a>	0x10200824	32	Config	R	0x0000	Pending input events status register with IRQ_MASK applied.
<a href="#">EVENT_BUFFER_CLEAR_CORE0</a>	0x10200828	32	Config	W	0x0000	Pending input events status clear command register.
<a href="#">SW_EVENT_MASK_CORE0</a>	0x1020082C	32	Config	R/W	0x0000	Software events cluster cores destination mask configuration register.
<a href="#">SW_EVENT_MASK_AND_CORE0</a>	0x10200830	32	Config	W	0x0000	Software events cluster cores destination mask update command register with bitwise AND operation.
<a href="#">SW_EVENT_MASK_OR_CORE0</a>	0x10200834	32	Config	W	0x0000	Software events cluster cores destination mask update command register with bitwise OR operation.
<a href="#">EVT_MASK_CORE1</a>	0x10200840	32	Config	R/W	0x0000	Input event mask configuration register.
<a href="#">EVT_MASK_AND_CORE1</a>	0x10200844	32	Config	W	0x0000	Input event mask update command register with bitwise AND operation.
<a href="#">EVT_MASK_OR_CORE1</a>	0x10200848	32	Config	W	0x0000	Input event mask update command register with bitwise OR operation.
<a href="#">IRQ_MASK_CORE1</a>	0x1020084C	32	Config	R/W	0x0000	Interrupt request mask configuration register.
<a href="#">IRQ_MASK_AND_CORE1</a>	0x10200850	32	Config	W	0x0000	Interrupt request mask update command register with bitwise AND operation.
<a href="#">IRQ_MASK_OR_CORE1</a>	0x10200854	32	Config	W	0x0000	Interrupt request mask update command register with bitwise OR operation.
<a href="#">CLOCK_STATUS_CORE1</a>	0x10200858	32	Config	R	0x0000	Cluster cores clock status register.
<a href="#">EVENT_BUFFER_CORE1</a>	0x1020085C	32	Config	R	0x0000	Pending input events status register.
<a href="#">EVENT_BUFFER_MASKED_CORE1</a>	0x10200860	32	Config	R	0x0000	Pending input events status register with EVT_MASK applied.
<a href="#">EVENT_BUFFER_IRQ_MASKED_CORE1</a>	0x10200864	32	Config	R	0x0000	Pending input events status register with IRQ_MASK applied.
<a href="#">EVENT_BUFFER_CLEAR_CORE1</a>	0x10200868	32	Config	W	0x0000	Pending input events status clear command register.
<a href="#">SW_EVENT_MASK_CORE1</a>	0x1020086C	32	Config	R/W	0x0000	Software events cluster cores destination mask configuration register.
<a href="#">SW_EVENT_MASK_AND_CORE1</a>	0x10200870	32	Config	W	0x0000	Software events cluster cores destination mask update command register with bitwise AND operation.
<a href="#">SW_EVENT_MASK_OR_CORE1</a>	0x10200874	32	Config	W	0x0000	Software events cluster cores destination mask update command register with bitwise OR operation.
<a href="#">EVT_MASK_CORE2</a>	0x10200880	32	Config	R/W	0x0000	Input event mask configuration register.

Name	Address	Size	Type	Access	Default	Description
<a href="#">EVT_MASK_AND_CORE2</a>	0x10200884	32	Config	W	0x0000	Input event mask update command register with bitwise AND operation.
<a href="#">EVT_MASK_OR_CORE2</a>	0x10200888	32	Config	W	0x0000	Input event mask update command register with bitwise OR operation.
<a href="#">IRQ_MASK_CORE2</a>	0x1020088C	32	Config	R/W	0x0000	Interrupt request mask configuration register.
<a href="#">IRQ_MASK_AND_CORE2</a>	0x10200890	32	Config	W	0x0000	Interrupt request mask update command register with bitwise AND operation.
<a href="#">IRQ_MASK_OR_CORE2</a>	0x10200894	32	Config	W	0x0000	Interrupt request mask update command register with bitwise OR operation.
<a href="#">CLOCK_STATUS_CORE2</a>	0x10200898	32	Config	R	0x0000	Cluster cores clock status register.
<a href="#">EVENT_BUFFER_CORE2</a>	0x1020089C	32	Config	R	0x0000	Pending input events status register.
<a href="#">EVENT_BUFFER_MASKED_CORE2</a>	0x102008A0	32	Config	R	0x0000	Pending input events status register with EVT_MASK applied.
<a href="#">EVENT_BUFFER_IRQ_MASKED_CORE2</a>	0x102008A4	32	Config	R	0x0000	Pending input events status register with IRQ_MASK applied.
<a href="#">EVENT_BUFFER_CLEAR_CORE2</a>	0x102008A8	32	Config	W	0x0000	Pending input events status clear command register.
<a href="#">SW_EVENT_MASK_CORE2</a>	0x102008AC	32	Config	R/W	0x0000	Software events cluster cores destination mask configuration register.
<a href="#">SW_EVENT_MASK_AND_CORE2</a>	0x102008B0	32	Config	W	0x0000	Software events cluster cores destination mask update command register with bitwise AND operation.
<a href="#">SW_EVENT_MASK_OR_CORE2</a>	0x102008B4	32	Config	W	0x0000	Software events cluster cores destination mask update command register with bitwise OR operation.
<a href="#">EVT_MASK_CORE3</a>	0x102008C0	32	Config	R/W	0x0000	Input event mask configuration register.
<a href="#">EVT_MASK_AND_CORE3</a>	0x102008C4	32	Config	W	0x0000	Input event mask update command register with bitwise AND operation.
<a href="#">EVT_MASK_OR_CORE3</a>	0x102008C8	32	Config	W	0x0000	Input event mask update command register with bitwise OR operation.
<a href="#">IRQ_MASK_CORE3</a>	0x102008CC	32	Config	R/W	0x0000	Interrupt request mask configuration register.
<a href="#">IRQ_MASK_AND_CORE3</a>	0x102008D0	32	Config	W	0x0000	Interrupt request mask update command register with bitwise AND operation.
<a href="#">IRQ_MASK_OR_CORE3</a>	0x102008D4	32	Config	W	0x0000	Interrupt request mask update command register with bitwise OR operation.
<a href="#">CLOCK_STATUS_CORE3</a>	0x102008D8	32	Config	R	0x0000	Cluster cores clock status register.
<a href="#">EVENT_BUFFER_CORE3</a>	0x102008DC	32	Config	R	0x0000	Pending input events status register.
<a href="#">EVENT_BUFFER_MASKED_CORE3</a>	0x102008E0	32	Config	R	0x0000	Pending input events status register with EVT_MASK applied.
<a href="#">EVENT_BUFFER_IRQ_MASKED_CORE3</a>	0x102008E4	32	Config	R	0x0000	Pending input events status register with IRQ_MASK applied.
<a href="#">EVENT_BUFFER_CLEAR_CORE3</a>	0x102008E8	32	Config	W	0x0000	Pending input events status clear command register.
<a href="#">SW_EVENT_MASK_CORE3</a>	0x102008EC	32	Config	R/W	0x0000	Software events cluster cores destination mask configuration register.

Name	Address	Size	Type	Access	Default	Description
<a href="#">SW_EVENT_MASK_AND_CORE3</a>	0x102008F0	32	Config	W	0x0000	Software events cluster cores destination mask update command register with bitwise AND operation.
<a href="#">SW_EVENT_MASK_OR_CORE3</a>	0x102008F4	32	Config	W	0x0000	Software events cluster cores destination mask update command register with bitwise OR operation.
<a href="#">EVT_MASK_CORE4</a>	0x10200900	32	Config	R/W	0x0000	Input event mask configuration register.
<a href="#">EVT_MASK_AND_CORE4</a>	0x10200904	32	Config	W	0x0000	Input event mask update command register with bitwise AND operation.
<a href="#">EVT_MASK_OR_CORE4</a>	0x10200908	32	Config	W	0x0000	Input event mask update command register with bitwise OR operation.
<a href="#">IRQ_MASK_CORE4</a>	0x1020090C	32	Config	R/W	0x0000	Interrupt request mask configuration register.
<a href="#">IRQ_MASK_AND_CORE4</a>	0x10200910	32	Config	W	0x0000	Interrupt request mask update command register with bitwise AND operation.
<a href="#">IRQ_MASK_OR_CORE4</a>	0x10200914	32	Config	W	0x0000	Interrupt request mask update command register with bitwise OR operation.
<a href="#">CLOCK_STATUS_CORE4</a>	0x10200918	32	Config	R	0x0000	Cluster cores clock status register.
<a href="#">EVENT_BUFFER_CORE4</a>	0x1020091C	32	Config	R	0x0000	Pending input events status register.
<a href="#">EVENT_BUFFER_MASKED_CORE4</a>	0x10200920	32	Config	R	0x0000	Pending input events status register with EVT_MASK applied.
<a href="#">EVENT_BUFFER_IRQ_MASKED_CORE4</a>	0x10200924	32	Config	R	0x0000	Pending input events status register with IRQ_MASK applied.
<a href="#">EVENT_BUFFER_CLEAR_CORE4</a>	0x10200928	32	Config	W	0x0000	Pending input events status clear command register.
<a href="#">SW_EVENT_MASK_CORE4</a>	0x1020092C	32	Config	R/W	0x0000	Software events cluster cores destination mask configuration register.
<a href="#">SW_EVENT_MASK_AND_CORE4</a>	0x10200930	32	Config	W	0x0000	Software events cluster cores destination mask update command register with bitwise AND operation.
<a href="#">SW_EVENT_MASK_OR_CORE4</a>	0x10200934	32	Config	W	0x0000	Software events cluster cores destination mask update command register with bitwise OR operation.
<a href="#">EVT_MASK_CORE5</a>	0x10200940	32	Config	R/W	0x0000	Input event mask configuration register.
<a href="#">EVT_MASK_AND_CORE5</a>	0x10200944	32	Config	W	0x0000	Input event mask update command register with bitwise AND operation.
<a href="#">EVT_MASK_OR_CORE5</a>	0x10200948	32	Config	W	0x0000	Input event mask update command register with bitwise OR operation.
<a href="#">IRQ_MASK_CORE5</a>	0x1020094C	32	Config	R/W	0x0000	Interrupt request mask configuration register.
<a href="#">IRQ_MASK_AND_CORE5</a>	0x10200950	32	Config	W	0x0000	Interrupt request mask update command register with bitwise AND operation.
<a href="#">IRQ_MASK_OR_CORE5</a>	0x10200954	32	Config	W	0x0000	Interrupt request mask update command register with bitwise OR operation.
<a href="#">CLOCK_STATUS_CORE5</a>	0x10200958	32	Config	R	0x0000	Cluster cores clock status register.
<a href="#">EVENT_BUFFER_CORE5</a>	0x1020095C	32	Config	R	0x0000	Pending input events status register.

Name	Address	Size	Type	Access	Default	Description
<a href="#">EVENT_BUFFER_MASKED_CORE5</a>	0x10200960	32	Config	R	0x0000	Pending input events status register with EVT_MASK applied.
<a href="#">EVENT_BUFFER_IRQ_MASKED_CORE5</a>	0x10200964	32	Config	R	0x0000	Pending input events status register with IRQ_MASK applied.
<a href="#">EVENT_BUFFER_CLEAR_CORE5</a>	0x10200968	32	Config	W	0x0000	Pending input events status clear command register.
<a href="#">SW_EVENT_MASK_CORE5</a>	0x1020096C	32	Config	R/W	0x0000	Software events cluster cores destination mask configuration register.
<a href="#">SW_EVENT_MASK_AND_CORE5</a>	0x10200970	32	Config	W	0x0000	Software events cluster cores destination mask update command register with bitwise AND operation.
<a href="#">SW_EVENT_MASK_OR_CORE5</a>	0x10200974	32	Config	W	0x0000	Software events cluster cores destination mask update command register with bitwise OR operation.
<a href="#">EVT_MASK_CORE6</a>	0x10200980	32	Config	R/W	0x0000	Input event mask configuration register.
<a href="#">EVT_MASK_AND_CORE6</a>	0x10200984	32	Config	W	0x0000	Input event mask update command register with bitwise AND operation.
<a href="#">EVT_MASK_OR_CORE6</a>	0x10200988	32	Config	W	0x0000	Input event mask update command register with bitwise OR operation.
<a href="#">IRQ_MASK_CORE6</a>	0x1020098C	32	Config	R/W	0x0000	Interrupt request mask configuration register.
<a href="#">IRQ_MASK_AND_CORE6</a>	0x10200990	32	Config	W	0x0000	Interrupt request mask update command register with bitwise AND operation.
<a href="#">IRQ_MASK_OR_CORE6</a>	0x10200994	32	Config	W	0x0000	Interrupt request mask update command register with bitwise OR operation.
<a href="#">CLOCK_STATUS_CORE6</a>	0x10200998	32	Config	R	0x0000	Cluster cores clock status register.
<a href="#">EVENT_BUFFER_CORE6</a>	0x1020099C	32	Config	R	0x0000	Pending input events status register.
<a href="#">EVENT_BUFFER_MASKED_CORE6</a>	0x102009A0	32	Config	R	0x0000	Pending input events status register with EVT_MASK applied.
<a href="#">EVENT_BUFFER_IRQ_MASKED_CORE6</a>	0x102009A4	32	Config	R	0x0000	Pending input events status register with IRQ_MASK applied.
<a href="#">EVENT_BUFFER_CLEAR_CORE6</a>	0x102009A8	32	Config	W	0x0000	Pending input events status clear command register.
<a href="#">SW_EVENT_MASK_CORE6</a>	0x102009AC	32	Config	R/W	0x0000	Software events cluster cores destination mask configuration register.
<a href="#">SW_EVENT_MASK_AND_CORE6</a>	0x102009B0	32	Config	W	0x0000	Software events cluster cores destination mask update command register with bitwise AND operation.
<a href="#">SW_EVENT_MASK_OR_CORE6</a>	0x102009B4	32	Config	W	0x0000	Software events cluster cores destination mask update command register with bitwise OR operation.
<a href="#">EVT_MASK_CORE7</a>	0x102009C0	32	Config	R/W	0x0000	Input event mask configuration register.
<a href="#">EVT_MASK_AND_CORE7</a>	0x102009C4	32	Config	W	0x0000	Input event mask update command register with bitwise AND operation.
<a href="#">EVT_MASK_OR_CORE7</a>	0x102009C8	32	Config	W	0x0000	Input event mask update command register with bitwise OR operation.
<a href="#">IRQ_MASK_CORE7</a>	0x102009CC	32	Config	R/W	0x0000	Interrupt request mask configuration register.



Name	Address	Size	Type	Access	Default	Description
<a href="#">IRQ_MASK_AND_CORE7</a>	0x102009D0	32	Config	W	0x0000	Interrupt request mask update command register with bitwise AND operation.
<a href="#">IRQ_MASK_OR_CORE7</a>	0x102009D4	32	Config	W	0x0000	Interrupt request mask update command register with bitwise OR operation.
<a href="#">CLOCK_STATUS_CORE7</a>	0x102009D8	32	Config	R	0x0000	Cluster cores clock status register.
<a href="#">EVENT_BUFFER_CORE7</a>	0x102009DC	32	Config	R	0x0000	Pending input events status register.
<a href="#">EVENT_BUFFER_MASKED_CORE7</a>	0x102009E0	32	Config	R	0x0000	Pending input events status register with EVT_MASK applied.
<a href="#">EVENT_BUFFER_IRQ_MASKED_CORE7</a>	0x102009E4	32	Config	R	0x0000	Pending input events status register with IRQ_MASK applied.
<a href="#">EVENT_BUFFER_CLEAR_CORE7</a>	0x102009E8	32	Config	W	0x0000	Pending input events status clear command register.
<a href="#">SW_EVENT_MASK_CORE7</a>	0x102009EC	32	Config	R/W	0x0000	Software events cluster cores destination mask configuration register.
<a href="#">SW_EVENT_MASK_AND_CORE7</a>	0x102009F0	32	Config	W	0x0000	Software events cluster cores destination mask update command register with bitwise AND operation.
<a href="#">SW_EVENT_MASK_OR_CORE7</a>	0x102009F4	32	Config	W	0x0000	Software events cluster cores destination mask update command register with bitwise OR operation.
<a href="#">HW_BARRIER_0_TRIG_MASK</a>	0x10200C00	32	Config	R/W	0x0000	Cluster hardware barrier 0 trigger mask configuration register.
<a href="#">HW_BARRIER_0_STATUS</a>	0x10200C04	32	Status	R	0x0000	Cluster hardware barrier 0 status register.
<a href="#">HW_BARRIER_0_STATUS_SUM</a>	0x10200C08	32	Status	R	0x0000	Cluster hardware barrier summary status register.
<a href="#">HW_BARRIER_0_TARGET_MASK</a>	0x10200C0C	32	Config	R/W	0x0000	Cluster hardware barrier 0 target mask configuration register.
<a href="#">HW_BARRIER_0_TRIG</a>	0x10200C10	32	Config	W	0x0000	Cluster hardware barrier 0 trigger command register.
<a href="#">HW_BARRIER_1_TRIG_MASK</a>	0x10200C20	32	Config	R/W	0x0000	Cluster hardware barrier 1 trigger mask configuration register.
<a href="#">HW_BARRIER_1_STATUS</a>	0x10200C24	32	Status	R	0x0000	Cluster hardware barrier 1 status register.
<a href="#">HW_BARRIER_1_STATUS_SUM</a>	0x10200C28	32	Status	R	0x0000	Cluster hardware barrier summary status register.
<a href="#">HW_BARRIER_1_TARGET_MASK</a>	0x10200C2C	32	Config	R/W	0x0000	Cluster hardware barrier 1 target mask configuration register.
<a href="#">HW_BARRIER_1_TRIG</a>	0x10200C30	32	Config	W	0x0000	Cluster hardware barrier 1 trigger command register.
<a href="#">HW_BARRIER_2_TRIG_MASK</a>	0x10200C40	32	Config	R/W	0x0000	Cluster hardware barrier 2 trigger mask configuration register.
<a href="#">HW_BARRIER_2_STATUS</a>	0x10200C44	32	Status	R	0x0000	Cluster hardware barrier 2 status register.
<a href="#">HW_BARRIER_2_STATUS_SUM</a>	0x10200C48	32	Status	R	0x0000	Cluster hardware barrier summary status register.
<a href="#">HW_BARRIER_2_TARGET_MASK</a>	0x10200C4C	32	Config	R/W	0x0000	Cluster hardware barrier 2 target mask configuration register.
<a href="#">HW_BARRIER_2_TRIG</a>	0x10200C50	32	Config	W	0x0000	Cluster hardware barrier 2 trigger command register.

Name	Address	Size	Type	Access	Default	Description
<a href="#">HW_BARRIER_3_TRIG_MASK</a>	0x10200C60	32	Config	R/W	0x0000	Cluster hardware barrier 3 trigger mask configuration register.
<a href="#">HW_BARRIER_3_STATUS</a>	0x10200C64	32	Status	R	0x0000	Cluster hardware barrier 3 status register.
<a href="#">HW_BARRIER_3_STATUS_SUM</a>	0x10200C68	32	Status	R	0x0000	Cluster hardware barrier summary status register.
<a href="#">HW_BARRIER_3_TARGET_MASK</a>	0x10200C6C	32	Config	R/W	0x0000	Cluster hardware barrier 3 target mask configuration register.
<a href="#">HW_BARRIER_3_TRIG</a>	0x10200C70	32	Config	W	0x0000	Cluster hardware barrier 3 trigger command register.
<a href="#">HW_BARRIER_4_TRIG_MASK</a>	0x10200C80	32	Config	R/W	0x0000	Cluster hardware barrier 4 trigger mask configuration register.
<a href="#">HW_BARRIER_4_STATUS</a>	0x10200C84	32	Status	R	0x0000	Cluster hardware barrier 4 status register.
<a href="#">HW_BARRIER_4_STATUS_SUM</a>	0x10200C88	32	Status	R	0x0000	Cluster hardware barrier summary status register.
<a href="#">HW_BARRIER_4_TARGET_MASK</a>	0x10200C8C	32	Config	R/W	0x0000	Cluster hardware barrier 4 target mask configuration register.
<a href="#">HW_BARRIER_4_TRIG</a>	0x10200C90	32	Config	W	0x0000	Cluster hardware barrier 4 trigger command register.
<a href="#">HW_BARRIER_5_TRIG_MASK</a>	0x10200CA0	32	Config	R/W	0x0000	Cluster hardware barrier 5 trigger mask configuration register.
<a href="#">HW_BARRIER_5_STATUS</a>	0x10200CA4	32	Status	R	0x0000	Cluster hardware barrier 5 status register.
<a href="#">HW_BARRIER_5_STATUS_SUM</a>	0x10200CA8	32	Status	R	0x0000	Cluster hardware barrier summary status register.
<a href="#">HW_BARRIER_5_TARGET_MASK</a>	0x10200CAC	32	Config	R/W	0x0000	Cluster hardware barrier 5 target mask configuration register.
<a href="#">HW_BARRIER_5_TRIG</a>	0x10200CB0	32	Config	W	0x0000	Cluster hardware barrier 5 trigger command register.
<a href="#">HW_BARRIER_6_TRIG_MASK</a>	0x10200CC0	32	Config	R/W	0x0000	Cluster hardware barrier 6 trigger mask configuration register.
<a href="#">HW_BARRIER_6_STATUS</a>	0x10200CC4	32	Status	R	0x0000	Cluster hardware barrier 6 status register.
<a href="#">HW_BARRIER_6_STATUS_SUM</a>	0x10200CC8	32	Status	R	0x0000	Cluster hardware barrier summary status register.
<a href="#">HW_BARRIER_6_TARGET_MASK</a>	0x10200CCC	32	Config	R/W	0x0000	Cluster hardware barrier 6 target mask configuration register.
<a href="#">HW_BARRIER_6_TRIG</a>	0x10200CD0	32	Config	W	0x0000	Cluster hardware barrier 6 trigger command register.
<a href="#">HW_BARRIER_7_TRIG_MASK</a>	0x10200CE0	32	Config	R/W	0x0000	Cluster hardware barrier 7 trigger mask configuration register.
<a href="#">HW_BARRIER_7_STATUS</a>	0x10200CE4	32	Status	R	0x0000	Cluster hardware barrier 7 status register.
<a href="#">HW_BARRIER_7_STATUS_SUM</a>	0x10200CE8	32	Status	R	0x0000	Cluster hardware barrier summary status register.
<a href="#">HW_BARRIER_7_TARGET_MASK</a>	0x10200CEC	32	Config	R/W	0x0000	Cluster hardware barrier 7 target mask configuration register.
<a href="#">HW_BARRIER_7_TRIG</a>	0x10200CF0	32	Config	W	0x0000	Cluster hardware barrier 7 trigger command register.
<a href="#">SW_EVENT_0_TRIG</a>	0x10200E00	32	Config	W	0x0000	Cluster Software event 0 trigger command register.
<a href="#">SW_EVENT_1_TRIG</a>	0x10200E04	32	Config	W	0x0000	Cluster Software event 1 trigger command register.

Name	Address	Size	Type	Access	Default	Description
<a href="#">SW_EVENT_2_TRIG</a>	0x10200E08	32	Config	W	0x0000	Cluster Software event 2 trigger command register.
<a href="#">SW_EVENT_3_TRIG</a>	0x10200E0C	32	Config	W	0x0000	Cluster Software event 3 trigger command register.
<a href="#">SW_EVENT_4_TRIG</a>	0x10200E10	32	Config	W	0x0000	Cluster Software event 4 trigger command register.
<a href="#">SW_EVENT_5_TRIG</a>	0x10200E14	32	Config	W	0x0000	Cluster Software event 5 trigger command register.
<a href="#">SW_EVENT_6_TRIG</a>	0x10200E18	32	Config	W	0x0000	Cluster Software event 6 trigger command register.
<a href="#">SW_EVENT_7_TRIG</a>	0x10200E1C	32	Config	W	0x0000	Cluster Software event 7 trigger command register.
<a href="#">SOC_PERIPH_EVENT_ID</a>	0x10200F00	32	Status	R	0x0000	Cluster SoC peripheral event ID status register.

Table 20. Cluster event unit registers table

## 5.2.4.2 Cluster event unit registers details

### 5.2.4.2.1 Input event mask configuration register. (EVT\_MASK)

Reset value: 0x0000

Host access bus: PERIPH/DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EMSOC	Reserved	EMCL													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EMCL															

Bit 31 - **EMSOC** (R/W)

Soc peripheral input event mask configuration bitfield:

- EMSOC[i]=0b0: Input event request i is masked
- EMSOC[i]=0b1: Input event request i is not masked

Bits 29:0 - **EMCL** (R/W)

Cluster internal input event mask configuration bitfield:

- EMCL[i]=0b0: Input event request i is masked
- EMCL[i]=0b1: Input event request i is not masked

### 5.2.4.2.2 Hardware task dispatcher push command register. (HW\_DISPATCH\_PUSH\_TASK)

Reset value: 0x0000

Host access bus: DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MSG															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSG															

Bits 31:0 - **MSG** (W)

Message to dispatch to all cluster cores selected in HW\_DISPATCH\_PUSH\_TEAM\_CONFIG.CT configuration bitfield.

**5.2.4.2.3 Hardware task dispatcher pop command register. (HW\_DISPATCH\_POP\_TASK)***Reset value: 0x0000**Host access bus: DEMUX*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MSG															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSG															

Bits 31:0 - **MSG** (R)

Message dispatched using HW\_DISPATCH\_PUSH\_TASK command and popped by cluster core who issued HW\_DISPATCH\_POP\_TASK command.

**5.2.4.2.4 Hardware mutex 0 non-blocking put command register. (HW\_MUTEX\_0\_MSG\_PUT)***Reset value: 0x0000**Host access bus: DEMUX*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MSG															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSG															

Bits 31:0 - **MSG** (W)

Message pushed when releasing hardware mutex 0 configuration bitfiled. It is a non-blocking access.

**5.2.4.2.5 Hardware mutex 0 blocking get command register. (HW\_MUTEX\_0\_MSG\_GET)***Reset value: 0x0000**Host access bus: DEMUX*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MSG															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSG															

Bits 31:0 - **MSG** (R)

Message popped when taking hardware mutex 0 data bitfiled. It is a blocking access.

**5.2.4.2.6 Cluster Software event 0 trigger command register. (SW\_EVENT\_0\_TRIG)***Reset value: 0x0000**Host access bus: PERIPH/DEMUX*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								SW0T							

Bits 7:0 - **SW0T** (W)

Triggers software event 0 for cluster core i if SW0T[i]=0b1.

**5.2.4.2.7 Cluster Software event 0 trigger and wait command register. (SW\_EVENT\_0\_TRIG\_WAIT)**

Reset value: 0x0000

Host access bus: DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EBM															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EBM															

Bits 31:0 - **EBM** (R)

Triggers software event 0 to all cluster cores targeted in SW\_EVENT\_MASK and gate the issuing cluster core clock until at least one unmasked event occurs. The read content of this bitfield is equivalent to EVENT\_BUFFER\_MASKED.EBM

#### 5.2.4.2.8 Cluster Software event 0 trigger, wait and clear command register. (SW\_EVENT\_0\_TRIG\_WAIT\_CLEAR)

Reset value: 0x0000

Host access bus: DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EBM															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EBM															

Bits 31:0 - **EBM** (R)

Triggers software event 0 to all cluster cores targeted in SW\_EVENT\_MASK and gate the issuing cluster core clock until at least one unmasked event occurs. In addition, EVENT\_BUFFER.EB[i] bits are cleared after the read if EVT\_MASK[i]=0b1. The read content of this bitfield is equivalent to EVENT\_BUFFER\_MASKED.EBM

#### 5.2.4.2.9 Cluster SoC peripheral event ID status register. (SOC\_PERIPH\_EVENT\_ID)

Reset value: 0x0000

Host access bus: PERIPH

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
VALID	Reserved														
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								ID							

Bit 31 - **VALID** (R)

Validity bit of SOC\_PERIPH\_EVENT\_ID.ID bitfield.

Bits 7:0 - **ID** (R)

Oldest SoC peripheral event ID status bitfield.

#### 5.2.4.2.10 Cluster hardware barrier 0 trigger mask configuration register. (HW\_BARRIER\_0\_TRIG\_MASK)

Reset value: 0x0000

Host access bus: PERIPH/DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								HBOTM							

Bits 7:0 - **HB0TM** (R/W)

Trigger mask for hardware barrier 0 bitfield. Hardware barrier 0 will be triggered only if for all HB0TM[i] = 0b1, HW\_BARRIER\_0\_STATUS.HB0S[i]=0b1. HB0TM=0 means that hardware barrier 0 is disabled.

#### 5.2.4.2.11 Input event mask update command register with bitwise AND operation. (EVT\_MASK\_AND)

Reset value: 0x0000

Host access bus: PERIPH/DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EMA															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EMA															

Bits 31:0 - **EMA** (W)

Input event mask configuration bitfield update with bitwise AND operation. It allows clearing EMCL[i], EMINTCL[i] or EMSOC[i] if EMA[i]=0b1.

#### 5.2.4.2.12 Hardware task dispatcher cluster core team configuration register. (HW\_DISPATCH\_PUSH\_TEAM\_CONFIG)

Reset value: 0x0000

Host access bus: DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								CT							

Bits 7:0 - **CT** (R/W)

Cluster cores team selection configuration bitfield. It allows to transmit HW\_DISPATCH\_PUSH\_TASK.MSG to cluster core i if CT[i]=0b1.

#### 5.2.4.2.13 Hardware mutex 1 non-blocking put command register. (HW\_MUTEX\_1\_MSG\_PUT)

Reset value: 0x0000

Host access bus: DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MSG															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSG															

Bits 31:0 - **MSG** (W)

Message pushed when releasing hardware mutex 1 configuration bitfield. It is a non-blocking access.

#### 5.2.4.2.14 Hardware mutex 1 blocking get command register. (HW\_MUTEX\_1\_MSG\_GET)

Reset value: 0x0000

Host access bus: DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MSG															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSG															

Bits 31:0 - **MSG** (*R*)

Message popped when taking hardware mutex 1 data bitfiled. It is a blocking access.

#### 5.2.4.2.15 Cluster Software event 1 trigger command register. (SW\_EVENT\_1\_TRIG)

*Reset value: 0x0000*

*Host access bus: PERIPH/DEMUX*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								SW1T							

Bits 7:0 - **SW1T** (*W*)

Triggers software event 1 for cluster core i if SW1T[i]=0b1.

#### 5.2.4.2.16 Cluster Software event 1 trigger and wait command register. (SW\_EVENT\_1\_TRIG\_WAIT)

*Reset value: 0x0000*

*Host access bus: DEMUX*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EBM															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EBM															

Bits 31:0 - **EBM** (*R*)

Triggers software event 1 to all cluster cores targeted in SW\_EVENT\_MASK and gate the issuing cluster core clock until at least one unmasked event occurs. The read content of this bitfield is equivalent to EVENT\_BUFFER\_MASKED.EBM

#### 5.2.4.2.17 Cluster Software event 1 trigger, wait and clear command register. (SW\_EVENT\_1\_TRIG\_WAIT\_CLEAR)

*Reset value: 0x0000*

*Host access bus: DEMUX*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EBM															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EBM															

Bits 31:0 - **EBM** (*R*)

Triggers software event 1 to all cluster cores targeted in SW\_EVENT\_MASK and gate the issuing cluster core clock until at least one unmasked event occurs. In addition, EVENT\_BUFFER.EB[i] bits are cleared after the read if EVT\_MASK[i]=0b1. The read content of this bitfield is equivalent to EVENT\_BUFFER\_MASKED.EBM

#### 5.2.4.2.18 Cluster hardware barrier 0 status register. (HW\_BARRIER\_0\_STATUS)

*Reset value: 0x0000*

*Host access bus: PERIPH/DEMUX*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								HBS							

Bits 7:0 - **HBS** (*R*)

Current status of hardware barrier 0 bitfield. HBS[i]=0b1 means that cluster core i has triggered hardware barrier 0. It is cleared when HBS matches HW\_BARRIER\_0\_TRIG\_MASK.HB0TM.

#### 5.2.4.2.19 Input event mask update command register with bitwise OR operation. (EVT\_MASK\_OR)

*Reset value: 0x0000*

*Host access bus: PERIPH/DEMUX*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EMO															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EMO															

Bits 31:0 - **EMO** (*W*)

Input event mask configuration bitfield update with bitwise OR operation. It allows setting EMCL[i], EMINTCL[i] or EMSOC[i] if EMO[i]=0b1.

#### 5.2.4.2.20 Cluster Software event 2 trigger command register. (SW\_EVENT\_2\_TRIG)

*Reset value: 0x0000*

*Host access bus: PERIPH/DEMUX*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								SW2T							

Bits 7:0 - **SW2T** (*W*)

Triggers software event 2 for cluster core i if SW2T[i]=0b1.

#### 5.2.4.2.21 Cluster Software event 2 trigger and wait command register. (SW\_EVENT\_2\_TRIG\_WAIT)

*Reset value: 0x0000*

*Host access bus: DEMUX*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EBM															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EBM															

Bits 31:0 - **EBM** (*R*)

Triggers software event 2 to all cluster cores targeted in SW\_EVENT\_MASK and gate the issuing cluster core clock until at least one unmasked event occurs. The read content of this bitfield is equivalent to EVENT\_BUFFER\_MASKED.EBM

#### 5.2.4.2.22 Cluster Software event 2 trigger, wait and clear command register. (SW\_EVENT\_2\_TRIG\_WAIT\_CLEAR)

*Reset value: 0x0000*

*Host access bus: DEMUX*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EBM															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EBM															



Bits 31:0 - **EBM** (R)

Triggers software event 2 to all cluster cores targeted in SW\_EVENT\_MASK and gate the issuing cluster core clock until at least one unmasked event occurs. In addition, EVENT\_BUFFER.EB[i] bits are cleared after the read if EVT\_MASK[i]=0b1. The read content of this bitfield is equivalent to EVENT\_BUFFER\_MASKED.EBM

#### 5.2.4.2.23 Cluster hardware barrier summary status register. (HW\_BARRIER\_0\_STATUS\_SUM)

Reset value: 0x0000

Host access bus: PERIPH/DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								HBSS							

Bits 7:0 - **HBSS** (R)

Current status of hardware barrier 0. HBSS[i] represents a summary of the barrier status for core i.

#### 5.2.4.2.24 Interrupt request mask configuration register. (IRQ\_MASK)

Reset value: 0x0000

Host access bus: PERIPH/DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IMSOC	IMINTCL	IMCL													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IMCL															

Bit 31 - **IMSOC** (R/W)

Soc peripheral interrupt request mask configuration bitfield:

- bit[i]=0b0: Interrupt request i is masked
- bit[i]=0b1: Interrupt request i is not masked

Bit 30 - **IMINTCL** (R/W)

Inter-cluster interrupt request mask configuration bitfield:

- bit[i]=0b0: Interrupt request i is masked
- bit[i]=0b1: Interrupt request i is not masked

Bits 29:0 - **IMCL** (R/W)

Cluster internal interrupt request mask configuration bitfield:

- bit[i]=0b0: Interrupt request i is masked
- bit[i]=0b1: Interrupt request i is not masked

#### 5.2.4.2.25 Cluster Software event 3 trigger command register. (SW\_EVENT\_3\_TRIG)

Reset value: 0x0000

Host access bus: PERIPH/DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								SW3T							

Bits 7:0 - **SW3T** (*W*)

Triggers software event 3 for cluster core *i* if SW3T[i]=0b1.

#### 5.2.4.2.26 Cluster Software event 3 trigger and wait command register. (SW\_EVENT\_3\_TRIG\_WAIT)

Reset value: 0x0000

Host access bus: DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EBM															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EBM															

Bits 31:0 - **EBM** (*R*)

Triggers software event 3 to all cluster cores targeted in SW\_EVENT\_MASK and gate the issuing cluster core clock until at least one unmasked event occurs. The read content of this bitfield is equivalent to EVENT\_BUFFER\_MASKED.EBM

#### 5.2.4.2.27 Cluster Software event 3 trigger, wait and clear command register. (SW\_EVENT\_3\_TRIG\_WAIT\_CLEAR)

Reset value: 0x0000

Host access bus: DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EBM															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EBM															

Bits 31:0 - **EBM** (*R*)

Triggers software event 3 to all cluster cores targeted in SW\_EVENT\_MASK and gate the issuing cluster core clock until at least one unmasked event occurs. In addition, EVENT\_BUFFER.EB[i] bits are cleared after the read if EVT\_MASK[i]=0b1. The read content of this bitfield is equivalent to EVENT\_BUFFER\_MASKED.EBM

#### 5.2.4.2.28 Cluster hardware barrier 0 target mask configuration register. (HW\_BARRIER\_0\_TARGET\_MASK)

Reset value: 0x0000

Host access bus: PERIPH/DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								HBTAM							

Bits 7:0 - **HBTAM** (*R/W*)

Cluster hardware barrier 0 target mask configuration bitfield. HBTAM[i]=0b1 means that cluster core *i* will receive hardware barrier 0 event when HW\_BARRIER\_0\_STATUS will match HW\_BARRIER\_0\_TRIG\_MASK.

#### 5.2.4.2.29 Interrupt request mask update command register with bitwise AND operation. (IRQ\_MASK\_AND)

Reset value: 0x0000

Host access bus: PERIPH/DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IMA															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IMA															

Bits 31:0 - **IMA** (*W*)

Interrupt request mask configuration bitfield update with bitwise AND operation. It allows clearing IMCL[i], IMINTCL[i] or IMSOC[i] if IMA[i]=0b1.

#### 5.2.4.2.30 Cluster Software event 4 trigger command register. (SW\_EVENT\_4\_TRIG)

*Reset value: 0x0000*

*Host access bus: PERIPH/DEMUX*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								SW4T							

Bits 7:0 - **SW4T** (*W*)

Triggers software event 4 for cluster core i if SW4T[i]=0b1.

#### 5.2.4.2.31 Cluster Software event 4 trigger and wait command register. (SW\_EVENT\_4\_TRIG\_WAIT)

*Reset value: 0x0000*

*Host access bus: DEMUX*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EBM															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EBM															

Bits 31:0 - **EBM** (*R*)

Triggers software event 4 to all cluster cores targeted in SW\_EVENT\_MASK and gate the issuing cluster core clock until at least one unmasked event occurs. The read content of this bitfield is equivalent to EVENT\_BUFFER\_MASKED.EBM

#### 5.2.4.2.32 Cluster Software event 4 trigger, wait and clear command register. (SW\_EVENT\_4\_TRIG\_WAIT\_CLEAR)

*Reset value: 0x0000*

*Host access bus: DEMUX*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EBM															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EBM															

Bits 31:0 - **EBM** (*R*)

Triggers software event 4 to all cluster cores targeted in SW\_EVENT\_MASK and gate the issuing cluster core clock until at least one unmasked event occurs. In addition, EVENT\_BUFFER.EB[i] bits are cleared after the read if EVT\_MASK[i]=0b1. The read content of this bitfield is equivalent to EVENT\_BUFFER\_MASKED.EBM

#### 5.2.4.2.33 Cluster hardware barrier 0 trigger command register. (HW\_BARRIER\_0\_TRIG)

*Reset value: 0x0000*

*Host access bus: PERIPH/DEMUX*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								T							

Bits 7:0 - **T** (*W*)

Sets HW\_BARRIER\_0\_STATUS.HBS[i] to 0b1 when T[i]=0b1.

#### 5.2.4.2.34 Interrupt request mask update command register with bitwise OR operation. (IRQ\_MASK\_OR)

Reset value: 0x0000

Host access bus: PERIPH/DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IMO															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IMO															

Bits 31:0 - **IMO** (*W*)

Interrupt request mask configuration bitfield update with bitwise OR operation. It allows setting IMCL[i], IMINTCL[i] or IMSOC[i] if IMO[i]=0b1.

#### 5.2.4.2.35 Cluster Software event 5 trigger command register. (SW\_EVENT\_5\_TRIG)

Reset value: 0x0000

Host access bus: PERIPH/DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								SW5T							

Bits 7:0 - **SW5T** (*W*)

Triggers software event 5 for cluster core i if SW5T[i]=0b1.

#### 5.2.4.2.36 Cluster Software event 5 trigger and wait command register. (SW\_EVENT\_5\_TRIG\_WAIT)

Reset value: 0x0000

Host access bus: DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EBM															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EBM															

Bits 31:0 - **EBM** (*R*)

Triggers software event 5 to all cluster cores targeted in SW\_EVENT\_MASK and gate the issuing cluster core clock until at least one unmasked event occurs. The read content of this bitfield is equivalent to EVENT\_BUFFER\_MASKED.EBM

#### 5.2.4.2.37 Cluster Software event 5 trigger, wait and clear command register. (SW\_EVENT\_5\_TRIG\_WAIT\_CLEAR)

Reset value: 0x0000

Host access bus: DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EBM															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EBM															

Bits 31:0 - **EBM** (R)

Triggers software event 5 to all cluster cores targeted in SW\_EVENT\_MASK and gate the issuing cluster core clock until at least one unmasked event occurs. In addition, EVENT\_BUFFER.EB[i] bits are cleared after the read if EVT\_MASK[i]=0b1. The read content of this bitfield is equivalent to EVENT\_BUFFER\_MASKED.EBM

#### 5.2.4.2.38 Cluster hardware barrier 0 self trigger command register. (HW\_BARRIER\_0\_SELF\_TRIG)

Reset value: 0x0000

Host access bus: DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
T															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T															

Bits 31:0 - **T** (R)

Sets HW\_BARRIER\_0\_STATUS.HBS[i] to 0b1 when issued by cluster core i.

#### 5.2.4.2.39 Cluster cores clock status register. (CLOCK\_STATUS)

Reset value: 0x0000

Host access bus: PERIPH/DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															CS

Bit 0 - **CS** (R)

Cluster core clock status bitfield:

- 0b0: Cluster core clocked is gated
- 0b1: Cluster core clocked is running

#### 5.2.4.2.40 Cluster Software event 6 trigger command register. (SW\_EVENT\_6\_TRIG)

Reset value: 0x0000

Host access bus: PERIPH/DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								SW6T							

Bits 7:0 - **SW6T** (W)

Triggers software event 6 for cluster core i if SW6T[i]=0b1.

#### 5.2.4.2.41 Cluster Software event 6 trigger and wait command register. (SW\_EVENT\_6\_TRIG\_WAIT)

Reset value: 0x0000

Host access bus: DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EBM															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EBM															

Bits 31:0 - **EBM** (R)

Triggers software event 6 to all cluster cores targeted in SW\_EVENT\_MASK and gate the issuing cluster core clock until at least one unmasked event occurs. The read content of this bitfield is equivalent to EVENT\_BUFFER\_MASKED.EBM

#### 5.2.4.2.42 Cluster Software event 6 trigger, wait and clear command register. (SW\_EVENT\_6\_TRIG\_WAIT\_CLEAR)

Reset value: 0x0000

Host access bus: DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EBM															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EBM															

Bits 31:0 - **EBM** (R)

Triggers software event 6 to all cluster cores targeted in SW\_EVENT\_MASK and gate the issuing cluster core clock until at least one unmasked event occurs. In addition, EVENT\_BUFFER.EB[i] bits are cleared after the read if EVT\_MASK[i]=0b1. The read content of this bitfield is equivalent to EVENT\_BUFFER\_MASKED.EBM

#### 5.2.4.2.43 Cluster hardware barrier 0 trigger and wait command register. (HW\_BARRIER\_0\_TRIG\_WAIT)

Reset value: 0x0000

Host access bus: DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EBM															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EBM															

Bits 31:0 - **EBM** (R)

Set HW\_BARRIER\_0[i] when issued by cluster core i and gate the issuing cluster core i clock until HW\_BARRIER\_0 is released. The read content of this bitfield is equivalent to EVENT\_BUFFER\_MASKED.EBM

#### 5.2.4.2.44 Pending input events status register. (EVENT\_BUFFER)

Reset value: 0x0000

Host access bus: PERIPH/DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EB															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EB															

Bits 31:0 - **EB** (R)

Pending input events status bitfield.

EB[i]=0b1: one or more input event i request are pending.

**5.2.4.2.45 Cluster Software event 7 trigger command register. (SW\_EVENT\_7\_TRIG)***Reset value: 0x0000**Host access bus: PERIPH/DEMUX*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								SW7T							

Bits 7:0 - **SW7T** (*W*)

Triggers software event 7 for cluster core i if SW7T[i]=0b1.

**5.2.4.2.46 Cluster Software event 7 trigger and wait command register. (SW\_EVENT\_7\_TRIG\_WAIT)***Reset value: 0x0000**Host access bus: DEMUX*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EBM															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EBM															

Bits 31:0 - **EBM** (*R*)

Triggers software event 7 to all cluster cores targeted in SW\_EVENT\_MASK and gate the issuing cluster core clock until at least one unmasked event occurs. The read content of this bitfield is equivalent to EVENT\_BUFFER\_MASKED.EBM

**5.2.4.2.47 Cluster Software event 7 trigger, wait and clear command register. (SW\_EVENT\_7\_TRIG\_WAIT\_CLEAR)***Reset value: 0x0000**Host access bus: DEMUX*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EBM															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EBM															

Bits 31:0 - **EBM** (*R*)

Triggers software event 7 to all cluster cores targeted in SW\_EVENT\_MASK and gate the issuing cluster core clock until at least one unmasked event occurs. In addition, EVENT\_BUFFER.EB[i] bits are cleared after the read if EVT\_MASK[i]=0b1. The read content of this bitfield is equivalent to EVENT\_BUFFER\_MASKED.EBM

**5.2.4.2.48 Cluster hardware barrier 0 trigger, wait and clear command register. (HW\_BARRIER\_0\_TRIG\_WAIT\_CLEAR)***Reset value: 0x0000**Host access bus: DEMUX*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EBM															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EBM															

Bits 31:0 - **EBM** (*R*)

Set HW\_BARRIER\_0[i] when issued by cluster core i and gate the issuing cluster core i clock until HW\_BARRIER\_0 is released. In addition, EVENT\_BUFFER.EB[i] bits are cleared after the read if EVT\_MASK[i]=0b1. The read content of this bitfield is equivalent to EVENT\_BUFFER\_MASKED.EBM

**5.2.4.2.49 Pending input events status register with EVT\_MASK applied. (EVENT\_BUFFER\_MASKED)***Reset value: 0x0000**Host access bus: PERIPH/DEMUX*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EBM															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EBM															

Bits 31:0 - **EBM** (R)

Pending input events status bitfield with EVT\_MASK applied.

EBM[i]=0b1: one or more input event i request are pending.

**5.2.4.2.50 Cluster hardware barrier 1 trigger mask configuration register. (HW\_BARRIER\_1\_TRIG\_MASK)***Reset value: 0x0000**Host access bus: PERIPH/DEMUX*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								HB1TM							

Bits 7:0 - **HB1TM** (R/W)

Trigger mask for hardware barrier 1 bitfield. Hardware barrier 1 will be triggered only if for all HB1TM[i] = 0b1, HW\_BARRIER\_1\_STATUS.HB1S[i]=0b1. HB1TM=0 means that hardware barrier 1 is disabled.

**5.2.4.2.51 Pending input events status register with IRQ\_MASK applied. (EVENT\_BUFFER\_IRQ\_MASKED)***Reset value: 0x0000**Host access bus: PERIPH/DEMUX*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IBM															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IBM															

Bits 31:0 - **IBM** (R)

Pending input events status bitfield with IRQ\_MASK applied.

IBM[i]=0b1: one or more input events i are pending.

**5.2.4.2.52 Cluster hardware barrier 1 status register. (HW\_BARRIER\_1\_STATUS)***Reset value: 0x0000**Host access bus: PERIPH/DEMUX*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								HBS							



Bits 7:0 - **HBS** (*R*)

Current status of hardware barrier 1 bitfield. HBS[i]=0b1 means that cluster core i has triggered hardware barrier 1. It is cleared when HBS matches HW\_BARRIER\_1\_TRIG\_MASK.HB1TM.

#### 5.2.4.2.53 Pending input events status clear command register. (EVENT\_BUFFER\_CLEAR)

Reset value: 0x0000

Host access bus: PERIPH/DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EBC															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EBC															

Bits 31:0 - **EBC** (*W*)

Pending input events status clear command bitfield. It allows clearing EB[i] if EBC[i]=0b1.

#### 5.2.4.2.54 Cluster hardware barrier summary status register. (HW\_BARRIER\_1\_STATUS\_SUM)

Reset value: 0x0000

Host access bus: PERIPH/DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								HBSS							

Bits 7:0 - **HBSS** (*R*)

Current status of hardware barrier 1. HBSS[i] represents a summary of the barrier status for core i.

#### 5.2.4.2.55 Software events cluster cores destination mask configuration register. (SW\_EVENT\_MASK)

Reset value: 0x0000

Host access bus: PERIPH/DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								SWEM							

Bits 7:0 - **SWEM** (*R/W*)

Software events mask configuration bitfield:

- bit[i]=0b0: software events are masked for CL\_CORE[i]
- bit[i]=0b1: software events are not masked for CL\_CORE[i]

#### 5.2.4.2.56 Cluster hardware barrier 1 target mask configuration register. (HW\_BARRIER\_1\_TARGET\_MASK)

Reset value: 0x0000

Host access bus: PERIPH/DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								HBTAM							

Bits 7:0 - **HBTAM** (R/W)

Cluster hardware barrier 1 target mask configuration bitfield. HBTAM[i]=0b1 means that cluster core i will receive hardware barrier 1 event when HW\_BARRIER\_1\_STATUS will match HW\_BARRIER\_1\_TRIG\_MASK.

#### 5.2.4.2.57 Software events cluster cores destination mask update command register with bitwise AND operation. (SW\_EVENT\_MASK\_AND)

Reset value: 0x0000

Host access bus: PERIPH/DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								SWEMA							

Bits 7:0 - **SWEMA** (W)

Software event mask configuration bitfield update with bitwise AND operation. It allows clearing SWEM[i] if SWEMA[i]=0b1.

#### 5.2.4.2.58 Cluster hardware barrier 1 trigger command register. (HW\_BARRIER\_1\_TRIG)

Reset value: 0x0000

Host access bus: PERIPH/DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								T							

Bits 7:0 - **T** (W)

Sets HW\_BARRIER\_1\_STATUS.HBS[i] to 0b1 when T[i]=0b1.

#### 5.2.4.2.59 Software events cluster cores destination mask update command register with bitwise OR operation. (SW\_EVENT\_MASK\_OR)

Reset value: 0x0000

Host access bus: PERIPH/DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								SWEMO							

Bits 7:0 - **SWEMO** (W)

Software event mask configuration bitfield update with bitwise OR operation. It allows setting SWEM[i] if SWEMO[i]=0b1.

#### 5.2.4.2.60 Cluster hardware barrier 1 self trigger command register. (HW\_BARRIER\_1\_SELF\_TRIG)

Reset value: 0x0000

Host access bus: DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
T															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T															

Bits 31:0 - **T** (R)

Sets HW\_BARRIER\_1\_STATUS.HBS[i] to 0b1 when issued by cluster core i.

#### 5.2.4.2.61 Input event wait command register. (EVENT\_WAIT)

Reset value: 0x0000

Host access bus: DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EBM															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EBM															

Bits 31:0 - **EBM** (R)

Reading this register will gate the Cluster core clock until at least one unmasked event occurs. The read content of this bitfield is equivalent to EVENT\_BUFFER\_MASKED.EBM

#### 5.2.4.2.62 Cluster hardware barrier 1 trigger and wait command register. (HW\_BARRIER\_1\_TRIG\_WAIT)

Reset value: 0x0000

Host access bus: DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EBM															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EBM															

Bits 31:0 - **EBM** (R)

Set HW\_BARRIER\_1[i] when issued by cluster core i and gate the issuing cluster core i clock until HW\_BARRIER\_1 is released. The read content of this bitfield is equivalent to EVENT\_BUFFER\_MASKED.EBM

#### 5.2.4.2.63 Input event wait and clear command register. (EVENT\_WAIT\_CLEAR)

Reset value: 0x0000

Host access bus: DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EBM															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EBM															

Bits 31:0 - **EBM** (R)

Reading this register has the same effect as reading EVENT\_WAIT.EBM. In addition, EVENT\_BUFFER.EB[i] bits are cleared if EVT\_MASK[i]=0b1. The read content of this bitfield is equivalent to EVENT\_BUFFER\_MASKED.EBM

#### 5.2.4.2.64 Cluster hardware barrier 1 trigger, wait and clear command register. (HW\_BARRIER\_1\_TRIG\_WAIT\_CLEAR)

Reset value: 0x0000

Host access bus: DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EBM															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EBM															

Bits 31:0 - **EBM** (R)

Set HW\_BARRIER\_1[i] when issued by cluster core i and gate the issuing cluster core i clock until HW\_BARRIER\_1 is released. In addition, EVENT\_BUFFER.EB[i] bits are cleared after the read if EVT\_MASK[i]=0b1. The read content of this bitfield is equivalent to EVENT\_BUFFER\_MASKED.EBM

#### 5.2.4.2.65 Cluster hardware barrier 2 trigger mask configuration register. (HW\_BARRIER\_2\_TRIG\_MASK)

Reset value: 0x0000

Host access bus: PERIPH/DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								HB2TM							

Bits 7:0 - **HB2TM** (R/W)

Trigger mask for hardware barrier 2 bitfield. Hardware barrier 2 will be triggered only if for all HB2TM[i] = 0b1, HW\_BARRIER\_2\_STATUS.HB2S[i]=0b1. HB2TM=0 means that hardware barrier 2 is disabled.

#### 5.2.4.2.66 Cluster hardware barrier 2 status register. (HW\_BARRIER\_2\_STATUS)

Reset value: 0x0000

Host access bus: PERIPH/DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								HBS							

Bits 7:0 - **HBS** (R)

Current status of hardware barrier 2 bitfield. HBS[i]=0b1 means that cluster core i has triggered hardware barrier 2. It is cleared when HBS matches HW\_BARRIER\_2\_TRIG\_MASK.HB2TM.

#### 5.2.4.2.67 Cluster hardware barrier summary status register. (HW\_BARRIER\_2\_STATUS\_SUM)

Reset value: 0x0000

Host access bus: PERIPH/DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								HBSS							

Bits 7:0 - **HBSS** (R)

Current status of hardware barrier 2. HBSS[i] represents a summary of the barrier status for core i.

#### 5.2.4.2.68 Cluster hardware barrier 2 target mask configuration register. (HW\_BARRIER\_2\_TARGET\_MASK)

Reset value: 0x0000

Host access bus: PERIPH/DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								HBTAM							

Bits 7:0 - **HBATM** (R/W)

Cluster hardware barrier 2 target mask configuration bitfield. HBATM[i]=0b1 means that cluster core i will receive hardware barrier 2 event when HW\_BARRIER\_2\_STATUS will match HW\_BARRIER\_2\_TRIG\_MASK.

#### 5.2.4.2.69 Cluster hardware barrier 2 trigger command register. (HW\_BARRIER\_2\_TRIG)

Reset value: 0x0000

Host access bus: PERIPH/DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								T							

Bits 7:0 - **T** (W)

Sets HW\_BARRIER\_2\_STATUS.HBS[i] to 0b1 when T[i]=0b1.

#### 5.2.4.2.70 Cluster hardware barrier 2 self trigger command register. (HW\_BARRIER\_2\_SELF\_TRIG)

Reset value: 0x0000

Host access bus: DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
T															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T															

Bits 31:0 - **T** (R)

Sets HW\_BARRIER\_2\_STATUS.HBS[i] to 0b1 when issued by cluster core i.

#### 5.2.4.2.71 Cluster hardware barrier 2 trigger and wait command register. (HW\_BARRIER\_2\_TRIG\_WAIT)

Reset value: 0x0000

Host access bus: DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EBM															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EBM															

Bits 31:0 - **EBM** (R)

Set HW\_BARRIER\_2[i] when issued by cluster core i and gate the issuing cluster core i clock until HW\_BARRIER\_2 is released. The read content of this bitfield is equivalent to EVENT\_BUFFER\_MASKED.EBM

#### 5.2.4.2.72 Cluster hardware barrier 2 trigger, wait and clear command register. (HW\_BARRIER\_2\_TRIG\_WAIT\_CLEAR)

Reset value: 0x0000

Host access bus: DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EBM															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EBM															

Bits 31:0 - **EBM** (R)

Set HW\_BARRIER\_2[i] when issued by cluster core i and gate the issuing cluster core i clock until HW\_BARRIER\_2 is released. In addition, EVENT\_BUFFER.EB[i] bits are cleared after the read if EVT\_MASK[i]=0b1. The read content of this bitfield is equivalent to EVENT\_BUFFER\_MASKED.EBM

#### 5.2.4.2.73 Cluster hardware barrier 3 trigger mask configuration register. (HW\_BARRIER\_3\_TRIG\_MASK)

Reset value: 0x0000

Host access bus: PERIPH/DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								HB3TM							

Bits 7:0 - **HB3TM** (R/W)

Trigger mask for hardware barrier 3 bitfield. Hardware barrier 3 will be triggered only if for all HB3TM[i] = 0b1, HW\_BARRIER\_3\_STATUS.HB3S[i]=0b1. HB3TM=0 means that hardware barrier 3 is disabled.

#### 5.2.4.2.74 Cluster hardware barrier 3 status register. (HW\_BARRIER\_3\_STATUS)

Reset value: 0x0000

Host access bus: PERIPH/DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								HBS							

Bits 7:0 - **HBS** (R)

Current status of hardware barrier 3 bitfield. HBS[i]=0b1 means that cluster core i has triggered hardware barrier 3. It is cleared when HBS matches HW\_BARRIER\_3\_TRIG\_MASK.HB3TM.

#### 5.2.4.2.75 Cluster hardware barrier summary status register. (HW\_BARRIER\_3\_STATUS\_SUM)

Reset value: 0x0000

Host access bus: PERIPH/DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								HBSS							

Bits 7:0 - **HBSS** (R)

Current status of hardware barrier 3. HBSS[i] represents a summary of the barrier status for core i.

#### 5.2.4.2.76 Cluster hardware barrier 3 target mask configuration register. (HW\_BARRIER\_3\_TARGET\_MASK)

Reset value: 0x0000

Host access bus: PERIPH/DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								HBTAM							

Bits 7:0 - **HBATM** (R/W)

Cluster hardware barrier 3 target mask configuration bitfield. HBATM[i]=0b1 means that cluster core i will receive hardware barrier 3 event when HW\_BARRIER\_3\_STATUS will match HW\_BARRIER\_3\_TRIG\_MASK.

#### 5.2.4.2.77 Cluster hardware barrier 3 trigger command register. (HW\_BARRIER\_3\_TRIG)

Reset value: 0x0000

Host access bus: PERIPH/DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								T							

Bits 7:0 - **T** (W)

Sets HW\_BARRIER\_3\_STATUS.HBS[i] to 0b1 when T[i]=0b1.

#### 5.2.4.2.78 Cluster hardware barrier 3 self trigger command register. (HW\_BARRIER\_3\_SELF\_TRIG)

Reset value: 0x0000

Host access bus: DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
T															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T															

Bits 31:0 - **T** (R)

Sets HW\_BARRIER\_3\_STATUS.HBS[i] to 0b1 when issued by cluster core i.

#### 5.2.4.2.79 Cluster hardware barrier 3 trigger and wait command register. (HW\_BARRIER\_3\_TRIG\_WAIT)

Reset value: 0x0000

Host access bus: DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EBM															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EBM															

Bits 31:0 - **EBM** (R)

Set HW\_BARRIER\_3[i] when issued by cluster core i and gate the issuing cluster core i clock until HW\_BARRIER\_3 is released. The read content of this bitfield is equivalent to EVENT\_BUFFER\_MASKED.EBM

#### 5.2.4.2.80 Cluster hardware barrier 3 trigger, wait and clear command register. (HW\_BARRIER\_3\_TRIG\_WAIT\_CLEAR)

Reset value: 0x0000

Host access bus: DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EBM															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EBM															

Bits 31:0 - **EBM** (R)

Set HW\_BARRIER\_3[i] when issued by cluster core i and gate the issuing cluster core i clock until HW\_BARRIER\_3 is released. In addition, EVENT\_BUFFER.EB[i] bits are cleared after the read if EVT\_MASK[i]=0b1. The read content of this bitfield is equivalent to EVENT\_BUFFER\_MASKED.EBM

#### 5.2.4.2.81 Cluster hardware barrier 4 trigger mask configuration register. (HW\_BARRIER\_4\_TRIG\_MASK)

Reset value: 0x0000

Host access bus: PERIPH/DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								HB4TM							

Bits 7:0 - **HB4TM** (R/W)

Trigger mask for hardware barrier 4 bitfield. Hardware barrier 4 will be triggered only if for all HB4TM[i] = 0b1, HW\_BARRIER\_4\_STATUS.HB4S[i]=0b1. HB4TM=0 means that hardware barrier 4 is disabled.

#### 5.2.4.2.82 Cluster hardware barrier 4 status register. (HW\_BARRIER\_4\_STATUS)

Reset value: 0x0000

Host access bus: PERIPH/DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								HBS							

Bits 7:0 - **HBS** (R)

Current status of hardware barrier 4 bitfield. HBS[i]=0b1 means that cluster core i has triggered hardware barrier 4. It is cleared when HBS matches HW\_BARRIER\_4\_TRIG\_MASK.HB4TM.

#### 5.2.4.2.83 Cluster hardware barrier summary status register. (HW\_BARRIER\_4\_STATUS\_SUM)

Reset value: 0x0000

Host access bus: PERIPH/DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								HBSS							

Bits 7:0 - **HBSS** (R)

Current status of hardware barrier 4. HBSS[i] represents a summary of the barrier status for core i.

#### 5.2.4.2.84 Cluster hardware barrier 4 target mask configuration register. (HW\_BARRIER\_4\_TARGET\_MASK)

Reset value: 0x0000

Host access bus: PERIPH/DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								HBTAM							



Bits 7:0 - **HBATM** (R/W)

Cluster hardware barrier 4 target mask configuration bitfield. HBATM[i]=0b1 means that cluster core i will receive hardware barrier 4 event when HW\_BARRIER\_4\_STATUS will match HW\_BARRIER\_4\_TRIG\_MASK.

#### 5.2.4.2.85 Cluster hardware barrier 4 trigger command register. (HW\_BARRIER\_4\_TRIG)

Reset value: 0x0000

Host access bus: PERIPH/DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								T							

Bits 7:0 - **T** (W)

Sets HW\_BARRIER\_4\_STATUS.HBS[i] to 0b1 when T[i]=0b1.

#### 5.2.4.2.86 Cluster hardware barrier 4 self trigger command register. (HW\_BARRIER\_4\_SELF\_TRIG)

Reset value: 0x0000

Host access bus: DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
T															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T															

Bits 31:0 - **T** (R)

Sets HW\_BARRIER\_4\_STATUS.HBS[i] to 0b1 when issued by cluster core i.

#### 5.2.4.2.87 Cluster hardware barrier 4 trigger and wait command register. (HW\_BARRIER\_4\_TRIG\_WAIT)

Reset value: 0x0000

Host access bus: DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EBM															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EBM															

Bits 31:0 - **EBM** (R)

Set HW\_BARRIER\_4[i] when issued by cluster core i and gate the issuing cluster core i clock until HW\_BARRIER\_4 is released. The read content of this bitfield is equivalent to EVENT\_BUFFER\_MASKED.EBM

#### 5.2.4.2.88 Cluster hardware barrier 4 trigger, wait and clear command register. (HW\_BARRIER\_4\_TRIG\_WAIT\_CLEAR)

Reset value: 0x0000

Host access bus: DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EBM															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EBM															

Bits 31:0 - **EBM** (R)

Set HW\_BARRIER\_4[i] when issued by cluster core i and gate the issuing cluster core i clock until HW\_BARRIER\_4 is released. In addition, EVENT\_BUFFER.EB[i] bits are cleared after the read if EVT\_MASK[i]=0b1. The read content of this bitfield is equivalent to EVENT\_BUFFER\_MASKED.EBM

#### 5.2.4.2.89 Cluster hardware barrier 5 trigger mask configuration register. (HW\_BARRIER\_5\_TRIG\_MASK)

Reset value: 0x0000

Host access bus: PERIPH/DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								HB5TM							

Bits 7:0 - **HB5TM** (R/W)

Trigger mask for hardware barrier 5 bitfield. Hardware barrier 5 will be triggered only if for all HB5TM[i] = 0b1, HW\_BARRIER\_5\_STATUS.HB5S[i]=0b1. HB5TM=0 means that hardware barrier 5 is disabled.

#### 5.2.4.2.90 Cluster hardware barrier 5 status register. (HW\_BARRIER\_5\_STATUS)

Reset value: 0x0000

Host access bus: PERIPH/DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								HBS							

Bits 7:0 - **HBS** (R)

Current status of hardware barrier 5 bitfield. HBS[i]=0b1 means that cluster core i has triggered hardware barrier 5. It is cleared when HBS matches HW\_BARRIER\_5\_TRIG\_MASK.HB5TM.

#### 5.2.4.2.91 Cluster hardware barrier summary status register. (HW\_BARRIER\_5\_STATUS\_SUM)

Reset value: 0x0000

Host access bus: PERIPH/DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								HBSS							

Bits 7:0 - **HBSS** (R)

Current status of hardware barrier 5. HBSS[i] represents a summary of the barrier status for core i.

#### 5.2.4.2.92 Cluster hardware barrier 5 target mask configuration register. (HW\_BARRIER\_5\_TARGET\_MASK)

Reset value: 0x0000

Host access bus: PERIPH/DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								HBTAM							

Bits 7:0 - **HBATM** (R/W)

Cluster hardware barrier 5 target mask configuration bitfield. HBATM[i]=0b1 means that cluster core i will receive hardware barrier 5 event when HW\_BARRIER\_5\_STATUS will match HW\_BARRIER\_5\_TRIG\_MASK.

#### 5.2.4.2.93 Cluster hardware barrier 5 trigger command register. (HW\_BARRIER\_5\_TRIG)

Reset value: 0x0000

Host access bus: PERIPH/DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								T							

Bits 7:0 - **T** (W)

Sets HW\_BARRIER\_5\_STATUS.HBS[i] to 0b1 when T[i]=0b1.

#### 5.2.4.2.94 Cluster hardware barrier 5 self trigger command register. (HW\_BARRIER\_5\_SELF\_TRIG)

Reset value: 0x0000

Host access bus: DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
T															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T															

Bits 31:0 - **T** (R)

Sets HW\_BARRIER\_5\_STATUS.HBS[i] to 0b1 when issued by cluster core i.

#### 5.2.4.2.95 Cluster hardware barrier 5 trigger and wait command register. (HW\_BARRIER\_5\_TRIG\_WAIT)

Reset value: 0x0000

Host access bus: DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EBM															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EBM															

Bits 31:0 - **EBM** (R)

Set HW\_BARRIER\_5[i] when issued by cluster core i and gate the issuing cluster core i clock until HW\_BARRIER\_5 is released. The read content of this bitfield is equivalent to EVENT\_BUFFER\_MASKED.EBM

#### 5.2.4.2.96 Cluster hardware barrier 5 trigger, wait and clear command register. (HW\_BARRIER\_5\_TRIG\_WAIT\_CLEAR)

Reset value: 0x0000

Host access bus: DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EBM															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EBM															

Bits 31:0 - **EBM** (R)

Set HW\_BARRIER\_5[i] when issued by cluster core i and gate the issuing cluster core i clock until HW\_BARRIER\_5 is released. In addition, EVENT\_BUFFER.EB[i] bits are cleared after the read if EVT\_MASK[i]=0b1. The read content of this bitfield is equivalent to EVENT\_BUFFER\_MASKED.EBM

#### 5.2.4.2.97 Cluster hardware barrier 6 trigger mask configuration register. (HW\_BARRIER\_6\_TRIG\_MASK)

Reset value: 0x0000

Host access bus: PERIPH/DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								HB6TM							

Bits 7:0 - **HB6TM** (R/W)

Trigger mask for hardware barrier 6 bitfield. Hardware barrier 6 will be triggered only if for all HB6TM[i] = 0b1, HW\_BARRIER\_6\_STATUS.HB6S[i]=0b1. HB6TM=0 means that hardware barrier 6 is disabled.

#### 5.2.4.2.98 Cluster hardware barrier 6 status register. (HW\_BARRIER\_6\_STATUS)

Reset value: 0x0000

Host access bus: PERIPH/DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								HBS							

Bits 7:0 - **HBS** (R)

Current status of hardware barrier 6 bitfield. HBS[i]=0b1 means that cluster core i has triggered hardware barrier 6. It is cleared when HBS matches HW\_BARRIER\_6\_TRIG\_MASK.HB6TM.

#### 5.2.4.2.99 Cluster hardware barrier summary status register. (HW\_BARRIER\_6\_STATUS\_SUM)

Reset value: 0x0000

Host access bus: PERIPH/DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								HBSS							

Bits 7:0 - **HBSS** (R)

Current status of hardware barrier 6. HBSS[i] represents a summary of the barrier status for core i.

#### 5.2.4.2.100 Cluster hardware barrier 6 target mask configuration register. (HW\_BARRIER\_6\_TARGET\_MASK)

Reset value: 0x0000

Host access bus: PERIPH/DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								HBTAM							

Bits 7:0 - **HBTAM** (R/W)

Cluster hardware barrier 6 target mask configuration bitfield. HBTAM[i]=0b1 means that cluster core i will receive hardware barrier 6 event when HW\_BARRIER\_6\_STATUS will match HW\_BARRIER\_6\_TRIG\_MASK.

#### 5.2.4.2.101 Cluster hardware barrier 6 trigger command register. (HW\_BARRIER\_6\_TRIG)

Reset value: 0x0000

Host access bus: PERIPH/DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								T							

Bits 7:0 - **T** (W)

Sets HW\_BARRIER\_6\_STATUS.HBS[i] to 0b1 when T[i]=0b1.

#### 5.2.4.2.102 Cluster hardware barrier 6 self trigger command register. (HW\_BARRIER\_6\_SELF\_TRIG)

Reset value: 0x0000

Host access bus: DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
T															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T															

Bits 31:0 - **T** (R)

Sets HW\_BARRIER\_6\_STATUS.HBS[i] to 0b1 when issued by cluster core i.

#### 5.2.4.2.103 Cluster hardware barrier 6 trigger and wait command register. (HW\_BARRIER\_6\_TRIG\_WAIT)

Reset value: 0x0000

Host access bus: DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EBM															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EBM															

Bits 31:0 - **EBM** (R)

Set HW\_BARRIER\_6[i] when issued by cluster core i and gate the issuing cluster core i clock until HW\_BARRIER\_6 is released. The read content of this bitfield is equivalent to EVENT\_BUFFER\_MASKED.EBM

#### 5.2.4.2.104 Cluster hardware barrier 6 trigger, wait and clear command register. (HW\_BARRIER\_6\_TRIG\_WAIT\_CLEAR)

Reset value: 0x0000

Host access bus: DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EBM															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EBM															

Bits 31:0 - **EBM** (R)

Set HW\_BARRIER\_6[i] when issued by cluster core i and gate the issuing cluster core i clock until HW\_BARRIER\_6 is released. In addition, EVENT\_BUFFER.EB[i] bits are cleared after the read if EVT\_MASK[i]=0b1. The read content of this bitfield is equivalent to EVENT\_BUFFER\_MASKED.EBM

#### 5.2.4.2.105 Cluster hardware barrier 7 trigger mask configuration register. (HW\_BARRIER\_7\_TRIG\_MASK)

Reset value: 0x0000

Host access bus: PERIPH/DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								HB7TM							

Bits 7:0 - **HB7TM** (R/W)

Trigger mask for hardware barrier 7 bitfield. Hardware barrier 7 will be triggered only if for all HB7TM[i] = 0b1, HW\_BARRIER\_7\_STATUS.HB7S[i]=0b1. HB7TM=0 means that hardware barrier 7 is disabled.

#### 5.2.4.2.106 Cluster hardware barrier 7 status register. (HW\_BARRIER\_7\_STATUS)

Reset value: 0x0000

Host access bus: PERIPH/DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								HBS							

Bits 7:0 - **HBS** (R)

Current status of hardware barrier 7 bitfield. HBS[i]=0b1 means that cluster core i has triggered hardware barrier 7. It is cleared when HBS matches HW\_BARRIER\_7\_TRIG\_MASK.HB7TM.

#### 5.2.4.2.107 Cluster hardware barrier summary status register. (HW\_BARRIER\_7\_STATUS\_SUM)

Reset value: 0x0000

Host access bus: PERIPH/DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								HBSS							

Bits 7:0 - **HBSS** (R)

Current status of hardware barrier 7. HBSS[i] represents a summary of the barrier status for core i.

#### 5.2.4.2.108 Cluster hardware barrier 7 target mask configuration register. (HW\_BARRIER\_7\_TARGET\_MASK)

Reset value: 0x0000

Host access bus: PERIPH/DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								HBTAM							

Bits 7:0 - **HBATM** (R/W)

Cluster hardware barrier 7 target mask configuration bitfield. HBATM[i]=0b1 means that cluster core i will receive hardware barrier 7 event when HW\_BARRIER\_7\_STATUS will match HW\_BARRIER\_7\_TRIG\_MASK.

#### 5.2.4.2.109 Cluster hardware barrier 7 trigger command register. (HW\_BARRIER\_7\_TRIG)

Reset value: 0x0000

Host access bus: PERIPH/DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								T							

Bits 7:0 - **T** (W)

Sets HW\_BARRIER\_7\_STATUS.HBS[i] to 0b1 when T[i]=0b1.

#### 5.2.4.2.110 Cluster hardware barrier 7 self trigger command register. (HW\_BARRIER\_7\_SELF\_TRIG)

Reset value: 0x0000

Host access bus: DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
T															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T															

Bits 31:0 - **T** (R)

Sets HW\_BARRIER\_7\_STATUS.HBS[i] to 0b1 when issued by cluster core i.

#### 5.2.4.2.111 Cluster hardware barrier 7 trigger and wait command register. (HW\_BARRIER\_7\_TRIG\_WAIT)

Reset value: 0x0000

Host access bus: DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EBM															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EBM															

Bits 31:0 - **EBM** (R)

Set HW\_BARRIER\_7[i] when issued by cluster core i and gate the issuing cluster core i clock until HW\_BARRIER\_7 is released. The read content of this bitfield is equivalent to EVENT\_BUFFER\_MASKED.EBM

#### 5.2.4.2.112 Cluster hardware barrier 7 trigger, wait and clear command register. (HW\_BARRIER\_7\_TRIG\_WAIT\_CLEAR)

Reset value: 0x0000

Host access bus: DEMUX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EBM															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EBM															

Bits 31:0 - **EBM** (*R*)

Set HW\_BARRIER\_7[i] when issued by cluster core i and gate the issuing cluster core i clock until HW\_BARRIER\_7 is released. In addition, EVENT\_BUFFER.EB[i] bits are cleared after the read if EVT\_MASK[i]=0b1. The read content of this bitfield is equivalent to EVENT\_BUFFER\_MASKED.EBM

## 5.2.5 Cluster instruction cache control unit

CL\_ICACHE\_CTRL component manages the following features:

- Bypassable Cluster instruction cache controller
- Flush and selective flush commands

None

### 5.2.5.1 Cluster instruction cache control unit registers

Name	Address	Aliased address	Size	Type	Access	Default	Description
<a href="#">ENABLE</a>	0x10201400	0x1B201400	32	Config	W	0x0000	Cluster instruction cache unit enable configuration register.
<a href="#">FLUSH</a>	0x10201404	0x1B201404	32	Config	W	0x0000	Cluster instruction cache unit flush command register.
<a href="#">SEL_FLUSH</a>	0x1020140C	0x1B20140C	32	Config	W	0x0000	Cluster instruction cache unit selective flush command register.

Table 21. Cluster instruction cache control unit registers table

### 5.2.5.2 Cluster instruction cache control unit registers details

#### 5.2.5.2.1 Cluster instruction cache unit enable configuration register. (ENABLE)

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															EN

Bit 0 - **EN** (*W*)

Cluster instruction cache enable configuration bitfield:

- 0b0: disabled
- 0b1: enabled

#### 5.2.5.2.2 Cluster instruction cache unit flush command register. (FLUSH)

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															FL

Bit 0 - **FL** (*W*)

Cluster instruction cache full flush command.

#### 5.2.5.2.3 Cluster instruction cache unit selective flush command register. (SEL\_FLUSH)



*Reset value: 0x0000*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR															

Bits 31:0 - **ADDR** (*W*)

Cluster instruction cache selective flush address configuration bitfield.