GAP8 Hardware Reference Manual

Version 1.5.5

This document is preliminary and subject to change

Table of Contents

1 Features

- 1 + 8 high-performance cores: extended RISC-V ISA
  - 1 high performance micro-controller referred to as Fabric Controller or FC (150 MHz @ 1.0V; 250MHz @ 1.2V)
  - 8 cores that execute in parallel for compute intensive tasks referred to as Cluster (87 MHz @ 1.0V; 170MHz @ 1.2V)
  - Ultra low Power : maximum 25mA @ 1.0V
- Memories:
  - A level 2 Memory (512KB) for all the cores
  - A level 1 Memory (64 KB) shared by all the cores in Cluster (0 wait state memory access)
  - A level 1 memory (8 KB) owned by FC (0 wait state memory access)
  - Memory Protection Unit
  - HyperBus Interface to connect external HyperFlash or HyperRAM
- Clock, reset and supply management
  - 1.6V to 3.6V power supply and I/Os
  - Programmable (0.9V to 1.3V, 50mV steps) Internal Regulator
  - 2 independent FLLs, 1 for FC and 1 for Cluster
  - 1 x 32.768kHz RTC
  - Single 32.768kHz crystal for RTC and FLLs
- Low power
  - Sleep, Deep Sleep, Retentive states
- Debug Mode
  - JTAG interface
- DMA
  - A multi-channel 1D/2D cluster-DMA controls the transactions between the L2 Memory and L1 Memory
  - A smart, lightweight and completely autonomous DMA (micro-DMA) capable of handling complex I/O scheme
- I2S: Standard I2S interface for connecting digital audio devices
  - Up to 4 digital microphones (PDM or PCM)
- CPI: Parallel interface for connecting a camera
  - 8 bits interface
  - HSYNC, VSYNC, PCLK
- Up to 5 timers
  - 4 x 32 bit timers each with up to 4 PWM outputs
  - 1 32 bit SysTick timer
- Communication Interfaces
  - Up to 2 I2C
  - 2 x SPI master
  - 1 x SPI slave
  - 1 x UART
- Up to 32 fast I/O ports
  - Different Voltage I/O banks
  - Interrupt on Change
- A HardWare Convolution Engine (HWCE) for Convolutional Neural Networks based applications.
- LVDS interface
  - A 128 Mb/s interface for RF

2 Introduction to the GAP8 IoT application processor

GAP8 is an IoT application processor based on the PULP (Parallel Ultra-Low-Power Processing Platform) open-source platform which itself implements an extended version of the open-source RISC-V instruction set. GAP8 enables cost-effective development, deployment and autonomous operation of intelligent devices that capture, analyze, classify and act on the fusion of rich data sources such as images, sounds or vibrations. In particular, GAP8 is uniquely optimized to execute a large spectrum of image and audio algorithms including convolutional neural network inference with extreme energy efficiency. This allows industrial and consumer product manufacturers to integrate artificial intelligence and advanced classification into new classes of wireless edge devices for IoT applications including image recognition, counting people and objects, machine health monitoring, home security, speech recognition, consumer robotics and smart toys.

3 Device overview

The following figure describes the main functional blocks of GAP8 :

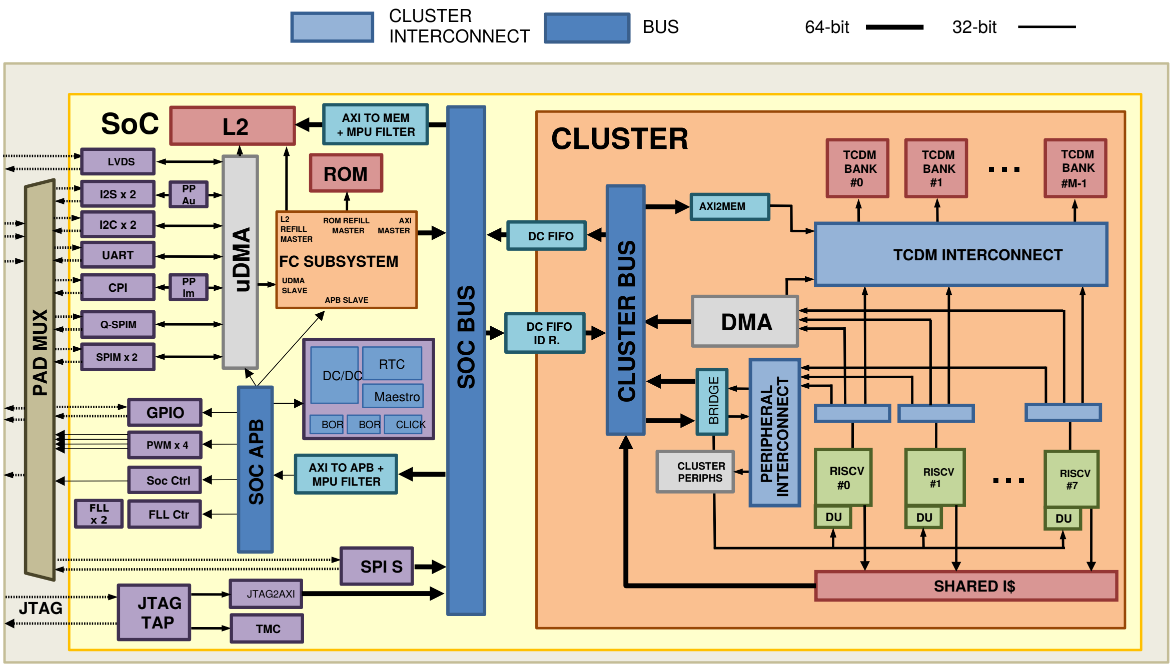

Figure 1. GAP8 Overview

GAP8's hierarchical, demand-driven architecture enables ultra-low-power operation by combining:

- A fabric controller (FC) core for control, communications and security functions. This can be viewed like a classic MCU.
- A cluster of 8 cores with an architecture optimized for the execution of vectorized and parallelized algorithms combined with a specialized Convolutional Neutral Network accelerator (HWCE).

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 GAP8 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 and HWCE share access to an L1 (TCDM) memory area and instruction cache. The FC has its own L1 (TCDM) memory area. Multiple DMA units allow autonomous, fast, low power transfers between cluster L1 and L2 memory and between L2 memory and external peripherals. A memory protection unit is included in the FC block to allow secure execution of applications on GAP8.

3.1 Cores

All 9 cores share the same extended RISC-V instruction set architecture. 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'.

3.2 Memory areas

There are 2 different levels of memory internal to GAP8. A larger level 2 area of 512kB which is accessible by all processors and DMA units and two smaller (Tightly Coupled Device Memory - TCDM) level 1 areas, one for the FC (16kB) and one shared by all the cluster cores and HWCE (64kB). 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.

L2 memory is divided into 4 128kB blocks that can be configured independently as retentive or not when setting GAP8 power mode to retentive sleep state.

Cluster L1 memory supports test-an-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 FC (1kB) and cluster (4kB) will automatically cache instructions as needed. The cluster instruction cache is shared between all the cores in the cluster. 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.

GAP8 can also access external memory areas over the HyperBus (Flash or RAM) or quad-SPI (Flash) interfaces. We refer to RAM accessed over the HyperBus or quad-SPI interfaces as level 3 memory. Since the energy and performance cost of accessing external RAM over external buses is very high compared to the internal memory, generally this should be avoided as much as possible. In consequence, program code is loaded from external flash at boot into the L2 memory area.

3.3 System bus architecture

GAP8 system bus architecture consists of multiple masters and slaves. Possible routings from masters to slaves are represented in below schematic.

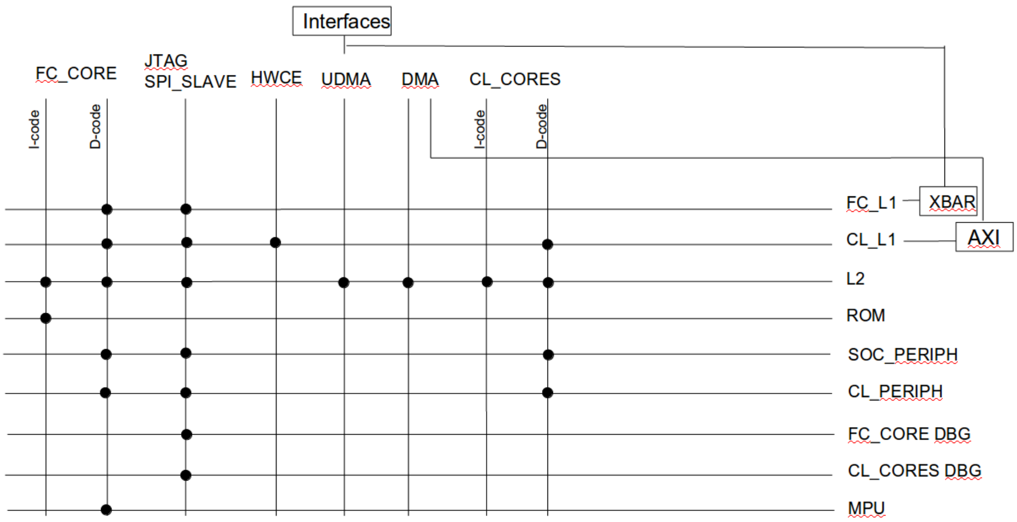

Figure 1. GAP8 system bus architecture

3.4 Clock architecture

3 clock domains exist in GAP8:

- one reference clock at 32,768kHz generated from an external crystal and an internal ultra low power oscillator. It is propagated to be used by the FLLs, the RTC and the timers.
- two FLLs: one for the FC and one for the Cluster. The frequencies must be set up respecting the programmed voltage in the DC/DC regulator. The clock source of the FLLs is the 32,768kHz reference clock.

3.5 Security architecture

A MPU controller in the FC and some address filters placed on specific locations in the bus architecture are instantiated to enable/disable access to specific memory map areas depending on User/Machine privilege mode configured in the RI5CY core of the FC and rules defined in the MPU.

Applications stored in external flash memories can also be protected using AES encryption. The ROM boot-loader in charge of loading applications from external memory into L2 memory has the ability to decrypt the image on-the-fly using an AES-128 key stored in the e-Fuses.

3.6 Power architecture

3 power domains exist in GAP8:

- AO always-on domain containing:
  - JTAG
  - PMU
  - internal DC/DC voltage regulator
  - POR/BOR
  - RTC
- SoC domain containing:
  - FC
  - uDMA and communication interfaces (SPIM, I2S, I2C, UART, CPI)
  - GPIOs
  - Advanced PWM timers
  - SPI Slave
  - L2 memory
  - ROM memory
- Cluster domain containing:
  - 8 cluster RI5CY cores
  - HWCE
  - DMA
  - Cluster L1 TCDM
  - Cluster peripherals

The internal DC/DC voltage regulator can be used to power both the FC and the cluster domains. If not, an external regulator must be used to provide VDD and VDD_EXT_CLUSTER.

The internal DC/DC voltage regulator operates in 3 different modes:

- LDO: the output of the regulator is fixed at 0.8V for retentive mode
- LV: the output can be programmed from 0.9V to 1.1V in 50mV steps
- HV: the output can be programmed from 1.1V to 1.3V in 50mV steps

3.6.1 Low power modes

GAP8 can operate in 6 different power modes. The column REG indicates the operating mode of the internal DC/DC voltage regulator. The column FC indicates if the FC and peripherals are powered (ON) or not (OFF). The column CL indicates if the cluster is powered (ON) or not (OFF). The column RET indicates if the retentive mode is active (ON) or not (OFF). In this mode, only a part of the L2 and few configuration registers stay powered.

Mode	REG	FC	CL	RET	Remark
Deep sleep (DS)	LDO	OFF	OFF	OFF	Only a RTC programmed alarm or a change on GPIO pin event can wake up the chip. When waking up, GAP8 performs a full boot sequence
Retentive (RET)	LDO	OFF	OFF	ON	Only a RTC programmed alarm or a change on GPIO pin event can wake up the chip. When waking up GAP8 starts booting from L2
SOC_LP	LV	ON	OFF	N/A	Only SOC and peripherals are running
SOC_HP	HV	ON	OFF	N/A	Only SOC and peripherals are running
SOC_CL_LP	LV	ON	ON	N/A	Whole chip is running
SOC_CL_HP	HV	ON	ON	N/A	Whole chip is running

Table 1. GAP8 Power Modes

3.6.2 Power-up sequence

3.6.2.1 Cold boot

GAP8 cold boot is sequenced as follows:

- Provide VBAT externally
- Internal DC/DC voltage regulator stabilizes to 1.2V and send power ok signal to PORBOR
- PORBOR internally resets the chip
- Optionally, external chip reset pin should be asserted then de-asserted
- 32kHz oscillator is activated
- SOC domain is turned on

- SOC FLL is turned on
- FC RI5CY core starts executing first stage boot-loader (FSBL) from ROM memory
  - Optionally wait for 32kHz oscillator stabilization
  - Activate PMU cluster bypass mode
  - SOC FLL is locked at 50MHz (unless redefined by e-Fuses)
  - FSBL prepares secondary boot which will be executed from L2 memory starting at address 0x1C000080 which is the address of the reset handler. Multiple boot modes are provided by FSBL to select the location of the application program to be loaded and executed.

3.6.2.2 Warm boot

GAP8 can warm boot after going into deep sleep or retentive sleep state. Warm boot possible sequences are described below:

- Before going into deep sleep or retentive sleep states, save all PMU and FLL configurations needed to wake-up in the retentive SAFE_PMU_SLEEPCTRL and SAFE_PMU_RAR registers of the SoC control unit
- Switch GAP8 power state to deep sleep or retentive sleep mode. Only AO power domain will stay powered by the LDO internal voltage regulator.
- Wake-up can be configured to be raised by RTC or external wake-up event on one pin of the chip.
- When wake-up event arises:
  - if deep sleep mode was chosen, GAP8 chip will continue to boot as described in cold boot sequence
  - if retentive sleep was selected:
    - FLL and PMU configurations are restored automatically from the SAFE_PMU_SLEEPCTRL and SAFE_PMU_RAR registers of the SoC control unit
    - FC RI5CY core starts executing first stage boot-loader (FSBL) from ROM memory. FSBL will directly jump to the reset handler at address 0x1C000080 under the assumption that this part of the L2 memory would have been configured to retentive state before going into retentive sleep mode.

3.7 Boot Modes

GAP8 always boots from the first stage boot-loader contained in ROM memory. Depending on e-Fuses content and retentive state stored in AO power domain, it can boot from:

- JTAG interface
- SPI Slave interface
- external SPI flash
- external HyperFlash
- internal L2 (when waking up from retentive mode)

The security features of GAP8 allows to boot in a secure mode where the code loaded is encrypted with an AES-128 key.

All the boot modes (and the keys used for secure mode boot) are controlled by internal e-fuses or by the APB SoC control unit register JTAGREG. The JTAGREG allows a host connected via the JTAG or SPI slave interface to control which memory device the FC will boot from while GAP8 chip reset is asserted.

3.8 Debug architecture

GAP8 contains debug functionalities to help the developer observing/controlling application code execution.

Debug functionalities are accessible through JTAG or SPI Slave interfaces using a GDB server. They permit access to the FC RI5CY core debug unit and Cluster RI5CY cores debug units to break program execution. A direct connection from JTAG or SPI Slave to GAP8 bus architecture also allows access to all the memory mapped registers of the chip.

3.9 Events and interrupts model

To be completed

3.10 Exceptions model

To be completed

3.11 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.

3.12 e-Fuses

The GAP8 FC subsystem contains a number of write once e-fuses. These are used to control a number of chip functions including boot device, boot mode, FLL configuration and others. A large section of this e-fuse block is available for customer application use.

3.13 MPU (Memory Protection Unit)

The MPU provides features to protect access to L2 memory, FC L1 memory and SOC peripherals memory mapped areas.

3.14 Event Units

Two event units (EU) are available in GAP8. One for the FC and one for the cluster.

The EU allows the RI5CY 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.

3.15 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.

3.16 HWCE (hardware convolution engine)

The Hardware Convolution Engine or HWCE is a special-purpose co-processor designed for accelerated computation of convolution-accumulation kernels. In particular, it is targeted towards acceleration of Convolutional Neural Networks (CNNs). The HWCE assumes input and output pixels and convolution weights are 16-bit, 8-bit or 4-bit fixed point numbers.

The GAP8 overview diagram above shows how the HWCE is integrated within GAP8's cluster domain. Differently from most tightly-coupled accelerators, the HWCE is not tied with a specific core but rather is tightly integrated within the cluster. Memory access is performed directly to logarithmic interconnect of the cluster L1 memory, while control of the accelerator is performed by means of a configuration port that is a target of the cluster peripheral interconnect.

This means that the HWCE directly shares data in cluster L1 with the RI5CY cores, similarly to what cores do with one another. Therefore, from the perspective of the shared L1 memory the HWCE is essentially indistinguishable from a cluster core. The HWCE is, however unable to access L2 memory unlike the other cores.

The Hardware Convolution engine is capable of executing the following operations in a single cycle:

1. a single 5x5 or 4x7 convolution using 16bit weights and 16bit pixels
2. three simultaneous 3x3 convolutions using 16bit weights and 16bit pixels
3. two simultaneous 5x5 or 4x7 convolutions using 8bit weights and 16bit pixels
4. four simultaneous 5x5 or 4x7 convolutions using 4bit weights and 16bit pixels

Moreover, in all considered cases the precision of weights can be reduced to 8bit or 4bit and that of pixels to 8bit; these changes do not come with a performance improvement outside of the 4 cases considered above, but power and memory bandwidth requirements are relaxed.

The HWCE supports a stride of one only.

3.17 Micro DMA

The micro DMA (UDMA) provides direct transfer of data between L2 memory and the different interfaces provided in GAP8 connected to UDMA. It helps in relaxing the execution load of FC RI5CY core. Up to 11 channels can be managed by the UDMA:

- LVDS from/to L2
- Camera to L2
- I2S0 to L2
- I2S1 to L2
- FC L1 from/to L2
- I2C0 from/to L2
- I2C1 from/to L2
- UART from/to L2

- HyperBus from/to L2
- SPIM0 from/to L2
- SPIM1 from/to L2

The width of transfers can be selected between 8, 16 or 32 bits. Up to 128kB can be transferred during a single transaction (8kB for SPIM). In the general case, transactions can be bidirectional but depending on the interface, in some cases only one direction is available.

3.17.1 LVDS (Low-voltage differential signaling)

A bidirectional LVDS link (differential clock and differential data) is available. It supports double data rate (DDR). It is able to communicate at speeds up to 64MHz.

3.17.2 SPI master (serial peripheral interface)

Up to 2 SPI master interfaces are available:

- One Single/Quad SPI (Master) which is able to communicate at speeds up to 50Mbits/s.
- One Single SPI (Master) which is able to communicate at speeds up to 50Mbits/s.

3.17.3 HyperBus

A HyperBus interface is available. It is an 8 bits wide data bus and can communicate at speeds up 50MHz in DDR with HyperFlash or HyperRam devices. Combo chips with HyperFlash and HyperRam are also supported.

Please note that the bootloader and the OS drivers make the assumption that the HyperFlash/RAM chip selects are connected as follows.

- HyperCSN0 CS2# (RAM)
- HyperCSN1 CS0# (Flash)

3.17.4 UART (universal asynchronous receiver-transmitter)

One UART interface is available with up to 1Mbits/s baud rate. No dedicated synchronization (CTS/RTS/DTR/DSR/DCD) signals are provided.

3.17.5 I2C (inter-integrated circuit)

Up to 2 I^2C (Inter-Integrated Circuit) are provided in GAP8. They support multi-master, multi-slave, single-ended modes. I^2C uses only two bidirectional open-drain lines, Serial Data Line (SDA) and Serial Clock Line (SCL), pulled up with resistors.

3.17.6 I2S (digital microphone interface) RX

Up to 2 RX I^2S are available for connecting digital audio devices to GAP8 chip. Up to 4 digital microphones (either PCM or PDM format) can be directly connected to GAP8. These are able to communicate at speeds of up to 10Mbits/s.

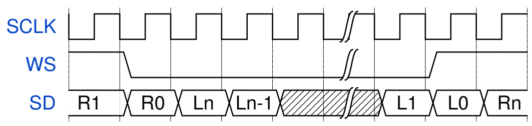

Figure 2. Standard I2S Transfer

The protocol uses 3 wires SCK(clock), WS(word select) and SD(data). Data is always sampled on the rising edge of the clock. WS indicates which channel the following data is part of. While SD is always in a master to slave direction, SCK and WS can be generated by both the master or slave. The GAP8 I^2S interface supports both modes as well as 2 other non standard transfer modes including a DDR (dual data-rate) mode.

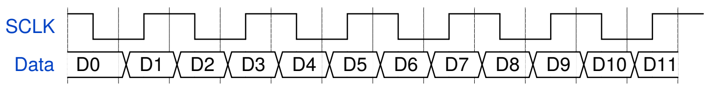

Figure 3. DDR Transfer

More details can be found in the section below.

3.17.7 CPI (camera parallel interface)

The CPI interface is 8 bits wide and can communicate at speeds up to 50MHz. VSYNC, HSYNC and PCLK are provided by the camera.

3.18 GPIOs (general purpose inputs/outputs)

Up to 32 digital general purpose I/Os are available. Each I/O can be configured either as an input or output. Interrupts on event can be generated on the rising or the falling or both edges for all I/Os. I/Os can also be configured to act as an external wake up signal.

3.19 SPI slave

One SPI slave interface directly connected to GAP8 bus architecture is available. It is able to access all memory mapped registers and L2 memory. It can communicate at speeds up to 50Mbits/s.

3.20 Basic timers

2 basic timers are available, one connected to the FC and the other to the cluster. They can be configured either as 2 x 32-bit timers or as a single 64-bit timer. The basic timers can either run continuously or trigger just once. Events can be generated using a compare match.

Clock sources of these timers can be the:

- FLL
- FLL with pre-scaler
- 32.768kHz reference clock

3.21 Advanced PWM Timers

4 advanced PWM timers are available in the SOC domain. Each of them provides 4 output signal channels that can be used for PWM signal generation with multiple configuration possibilities.

3.22 RTC

A real-time clock is available. It provides a set of continuously running counters which can be used with suitable software to provide a clock calendar function. It provides also alarm and a periodic interrupt features. It is clocked by the 32.768 kHz external crystal.

3.23 Performance counters

Each RI5CY 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 Pin-out and Pin description

4.1 Pinout

The following figure describes the pin-out (top view) of GAP8.

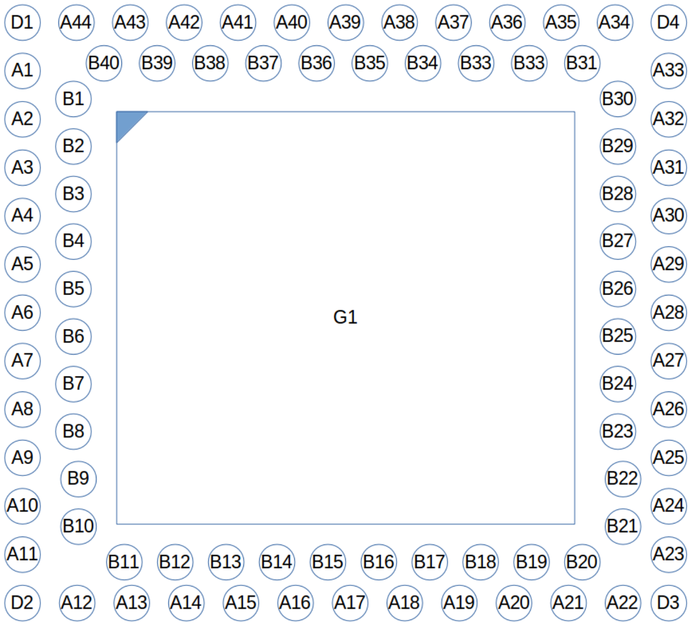

Figure 4. GAP8 Pin-out

4.2 Pin Description

The following tables describe GAP8's chip pins. After power-up, each pin is configured in the default mode. Each pin can be configured by software to one of the alternate functions available.

All pins (except UART_RX, UART_TX and SPIS0_SCK which can only be pulled up) can be configured to be pulled down.

Each pin function is described with an indication of whether it is an Input, Output or Supply pin.

Please see the HyperBus section for details of HyperFlash/RAM chip select assignment.

Position	**Voltage Ref**	**Function** *(Pin type)*			
		Default	**Alternate 1**	**Alternate 2**	**Alternate 3**
D3		VDDIO_LVDS *(Supply)*			
B20	VDDIO_LVDS	RF_TXD_P *(Out)*			
A22	VDDIO_LVDS	RF_TXD_N *(Out)*			
B19	VDDIO_LVDS	RF_TXCLK_P *(Out)*			
A21	VDDIO_LVDS	RF_TXCLK_N *(Out)*			
A19	VDDIO_LVDS	RF_RXD_P *(In)*			
B18	VDDIO_LVDS	RF_RXD_N *(In)*			
A18	VDDIO_LVDS	RF_RXCLK_P *(In)*			
B17	VDDIO_LVDS	RF_RXCLK_N *(In)*			
A1		CAM_VDDIO *(Supply)*			
A39		CAM_VDDIO *(Supply)*			
A4	CAM_VDDIO	SPIM1_MISO *(In)*	GPIOA0 *(In/Out)*		
B3	CAM_VDDIO	SPIM1_MOSI *(Out)*	GPIOA1 *(In/Out)*		
A5	CAM_VDDIO	SPIM1_CS0 *(Out)*	GPIOA2 *(In/Out)*	I2C1_SDA *(In/Out)*	

Position	Voltage Ref	Function *(Pin type)*			
		Default	**Alternate 1**	**Alternate 2**	**Alternate 3**
B4	CAM_VDDIO	SPIM1_SCK *(Out)*	GPIOA3 *(In/Out)*	I2C1_SCL *(In/Out)*	
A3	CAM_VDDIO	ORCA_TXSYNC *(In)*	GPIOA0 *(In/Out)*	SPIM1_CS0 *(Out)*	
B2	CAM_VDDIO	ORCA_RXSYNC *(In)*	GPIOA1 *(In/Out)*	SPIM1_CS1 *(Out)*	
A2	CAM_VDDIO	ORCA_TXI *(Out)*	GPIOA2 *(In/Out)*		
B1	CAM_VDDIO	ORCA_TXQ *(Out)*	GPIOA3 *(In/Out)*		
A44	CAM_VDDIO	ORCA_RXI *(In)*	GPIOA4 *(In/Out)*	SPIS0_SDIO0 *(In/Out)*	SPIS0_SDIO2 *(In/Out)*
B40	CAM_VDDIO	ORCA_RXQ *(In)*	GPIOA5 *(In/Out)*	SPIS0_SDIO1 *(In/Out)*	SPIS0_SDIO3 *(In/Out)*
A43	CAM_VDDIO	CAM_PCLK *(In)*	GPIOA4 *(In/Out)*	TIMER1_CH0 *(Out)*	
A37	CAM_VDDIO	CAM_HSYNC *(In)*	GPIOA5 *(In/Out)*	TIMER1_CH1 *(Out)*	
B39	CAM_VDDIO	CAM_DATA0 *(In)*	GPIOA6 *(In/Out)*	TIMER1_CH2 *(Out)*	
A42	CAM_VDDIO	CAM_DATA1 *(In)*	GPIOA7 *(In/Out)*	TIMER1_CH3 *(Out)*	
B38	CAM_VDDIO	CAM_DATA2 *(In)*	GPIOA8 *(In/Out)*	TIMER2_CH0 *(Out)*	
A41	CAM_VDDIO	CAM_DATA3 *(In)*	GPIOA9 *(In/Out)*	TIMER2_CH1 *(Out)*	
B37	CAM_VDDIO	CAM_DATA4 *(In)*	GPIOA10 *(In/Out)*	TIMER2_CH2 *(Out)*	
A40	CAM_VDDIO	CAM_DATA5 *(In)*	GPIOA11 *(In/Out)*	TIMER2_CH3 *(Out)*	
B36	CAM_VDDIO	CAM_DATA6 *(In)*	GPIOA12 *(In/Out)*	TIMER3_CH0 *(Out)*	
A38	CAM_VDDIO	CAM_DATA7 *(In)*	GPIOA13 *(In/Out)*	TIMER3_CH1 *(Out)*	
A36	CAM_VDDIO	CAM_VSYNC *(In)*	GPIOA14 *(In/Out)*	TIMER3_CH2 *(Out)*	
B34	CAM_VDDIO	I2C1_SDA *(In/Out)*	GPIOA15 *(In/Out)*	TIMER3_CH3 *(Out)*	
D1	CAM_VDDIO	I2C1_SCL *(In/Out)*	GPIOA16 *(In/Out)*	ORCA_CLK *(In)*	
A27		SAFE_VDDIO *(Supply)*			
B22	SAFE_VDDIO	I2C0_SDA *(In/Out)*			
A25	SAFE_VDDIO	I2C0_SCL *(In/Out)*			
A24	SAFE_VDDIO	I2S0_SCK *(In/Out)*			
A26	SAFE_VDDIO	I2S0_WS *(In/Out)*			
B23	SAFE_VDDIO	I2S0_SDI *(In)*			
A23	SAFE_VDDIO	RESET_N *(In)*			
B25	SAFE_VDDIO	JTAG_TCK *(In)*			
A28	SAFE_VDDIO	JTAG_TDI *(In)*			
B26	SAFE_VDDIO	JTAG_TDO *(Out)*			
A29	SAFE_VDDIO	JTAG_TMS *(In)*			
B27	SAFE_VDDIO	JTAG_TRST *(In)*			
B24	SAFE_VDDIO	JTAG_TAPSEL_EN *(In)*			
A12		SPIM_VDDIO *(Supply)*			
B11	SPIM_VDDIO	TIMER0_CH0 *(Out)*	GPIOA17 *(In/Out)*		
A13	SPIM_VDDIO	TIMER0_CH1 *(Out)*	GPIOA18 *(In/Out)*	TIMER1_CH0 *(Out)*	
B12	SPIM_VDDIO	TIMER0_CH2 *(Out)*	GPIOA19 *(In/Out)*	TIMER2_CH0 *(Out)*	
A14	SPIM_VDDIO	TIMER0_CH3 *(Out)*	GPIOA20 *(In/Out)*	TIMER3_CH0 *(Out)*	
B13	SPIM_VDDIO	I2S1_SCK *(In/Out)*	GPIOA21 *(In/Out)*	SPIS0_SCK *(In)*	I2S1_SDI *(In)*
A15	SPIM_VDDIO	I2S1_WS *(In/Out)*	GPIOA22 *(In/Out)*	SPIS0_CS *(In)*	HYPER_CKN *(Out)*
B14	SPIM_VDDIO	I2S1_SDI *(In)*	GPIOA23 *(In/Out)*	SPIS0_SDIO2 *(In/Out)*	HYPER_CK *(Out)*
B6	SPIM_VDDIO	UART_RX *(In)*	GPIOA24 *(In/Out)*		
A7	SPIM_VDDIO	UART_TX *(Out)*	GPIOA25 *(In/Out)*		
D2	SPIM_VDDIO	SPIM0_SDIO0 *(In/Out)*			HYPER_DQ[0] *(In/Out)*
A11	SPIM_VDDIO	SPIM0_SDIO1 *(In/Out)*			HYPER_DQ[1] *(In/Out)*
B10	SPIM_VDDIO	SPIM0_SDIO2 *(In/Out)*	GPIOA26 *(In/Out)*	I2C1_SDA *(In/Out)*	HYPER_DQ[2] *(In/Out)*

Position	Voltage Ref	Function *(Pin type)*			
		Default	**Alternate 1**	**Alternate 2**	**Alternate 3**
A10	SPIM_VDDIO	SPIM0_SDIO3 *(In/Out)*	GPIOA27 *(In/Out)*	I2C1_SCL *(In/Out)*	HYPER_DQ[3] *(In/Out)*
B8	SPIM_VDDIO	SPIM0_CS0 *(Out)*			HYPER_DQ[4] *(In/Out)*
A8	SPIM_VDDIO	SPIM0_CS1 *(Out)*	GPIOA28 *(In/Out)*	SPIS0_SDIO3 *(In/Out)*	HYPER_DQ[5] *(In/Out)*
B7	SPIM_VDDIO	SPIM0_SCK *(Out)*			HYPER_DQ[6] *(In/Out)*
A9	SPIM_VDDIO	SPIS0_CS *(In)*	GPIOA29 *(In/Out)*	SPIM1_CS0 *(Out)*	HYPER_DQ[7] *(In/Out)*
B15	SPIM_VDDIO	SPIS0_SDIO0 *(In/Out)*	GPIOA30 *(In/Out)*	SPIM1_CS1 *(Out)*	HYPER_CSN0 *(Out)*
A16	SPIM_VDDIO	SPIS0_SDIO1 *(In/Out)*	GPIOA31 *(In/Out)*		HYPER_CSN1 *(Out)*
B9	SPIM_VDDIO	SPIS0_SCK *(In)*			HYPER_RWDS *(In/Out)*
A31		XTAL_AVSS *(Supply)*			
B29		XTAL_AVDD *(Supply)*			
A32	XTAL_AVDD,XTAL_AVSS	XTAL_IN *(In)*			
B30	XTAL_AVDD,XTAL_AVSS	XTAL_OUT *(Out)*			
A33		RAR_AVS *(Supply)*			
A34		RAR_AVD *(Supply)*			
A35		RAR_VSENSE *(Supply)*			
B31		RAR_AVDPWR *(Supply)*			
B32		RAR_GNDSENSE *(Supply)*			
D4		RAR_LX *(Supply)*			
B5		VDD_EXT_CLUSTER *(Supply)*			
B16		VDD_EXT_CLUSTER *(Supply)*			
B35		VDD_EXT_CLUSTER *(Supply)*			
A6		VDD_EXT_CLUSTER *(Supply)*			
A17		VDD *(Supply)*			
A30		VDD *(Supply)*			
B33		VDD *(Supply)*			
A20		VDD_LVDS_1P2V *(Supply)*			
B28		VQPS *(Supply)*			
B21		VSS *(Supply)*			

Table 2. Pin description table

5 Memory map

The following table describes GAP8'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 - 0x003FFFFF
	Aliased memory map area		0x00000000 - 0x003FFFFF
Cluster Subsystem			0x10000000 - 0x103FFFFF
	Cluster L1 RAM (64kB)		0x10000000 - 0x1000FFFF
	Cluster L1 memory test and set unit		0x10100000 - 0x1010FFFF
	Cluster control unit		0x10200000 - 0x102003FF
	Cluster timer		0x10200400 - 0x102007FF
	Cluster event unit		0x10200800 - 0x10200FFF
	Cluster instruction cache control unit		0x10201400 - 0x102017FF
	Hardware convolution engine		0x10201800 - 0x10201BFF
	Cluster Core 0 (Debug Unit)		0x10300000 - 0x10307FFF
	Cluster Core 1 (Debug Unit)		0x10308000 - 0x1030FFFF
	Cluster Core 2 (Debug Unit)		0x10310000 - 0x10317FFF
	Cluster Core 3 (Debug Unit)		0x10318000 - 0x1031FFFF
	Cluster Core 4 (Debug Unit)		0x10320000 - 0x10327FFF
	Cluster Core 5 (Debug Unit)		0x10328000 - 0x1032FFFF
	Cluster Core 6 (Debug Unit)		0x10330000 - 0x10337FFF
	Cluster Core 7 (Debug Unit)		0x10338000 - 0x1033FFFF
ROM Memory			0x1A000000 - 0x1A0FFFFF
	ROM (8kB)		0x1A000000 - 0x1A001FFF
SoC Peripherals Subsystem			0x1A100000 - 0x1A10FFFF
	SoC FLL		0x1A100000 - 0x1A1007FF
	Cluster FLL		0x1A100800 - 0x1A100FFF
	GPIO		0x1A101000 - 0x1A101FFF
	SoC control unit		0x1A104000 - 0x1A104FFF
	Advanced timer		0x1A105000 - 0x1A105FFF
	SoC event generator		0x1A106000 - 0x1A106FFF
	PMU DLC bridge		0x1A107000 - 0x1A107FFF
	RealTime Counter		0x1A108000 - 0x1A108FFF
	Efuse		0x1A109000 - 0x1A109FFF
	MicroDMA Subsystem		0x1A102000 - 0x1A102FFF
		uDMA LVDS interface	0x1A102000 - 0x1A10207F
		SPI Master Channel 0	0x1A102080 - 0x1A1020FF
		SPI Master Channel 1	0x1A102100 - 0x1A10217F
		uDMA Hyperbus interface	0x1A102180 - 0x1A1021FF
		uDMA UART interface	0x1A102200 - 0x1A10227F
		I2C Channel 0	0x1A102280 - 0x1A1022FF
		I2C Channel 1	0x1A102300 - 0x1A10237F
		uDMA MEMCPY interface	0x1A102380 - 0x1A1023FF
		uDMA I2S interface	0x1A102400 - 0x1A10247F

			Address range
		uDMA CPI interface	0x1A102480 - 0x1A1024FF
		uDMA control unit	0x1A102780 - 0x1A1027FF
Fabric Controller Subsystem			0x1B000000 - 0x1B3FFFFF
	FC L1 RAM (16kB)		0x1B000000 - 0x1B003FFF
	FC control unit		0x1B200000 - 0x1B2003FF
	FC timer		0x1B200400 - 0x1B2007FF
	FC event unit		0x1B200800 - 0x1B200FFF
	FC instruction cache control unit		0x1B201400 - 0x1B2017FF
	FC Core 0 (Debug Unit)		0x1B300000 - 0x1B307FFF
L2 Memory			0x1C000000 - 0x1FFFFFFF
	L2 RAM (512kB)		0x1C000000 - 0x1C7FFFFF

Table 3. GAP8 memory map table

5.1 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).

5.2 FC aliased address map

Functional unit	Aliased address range
FC L1 RAM (16kB)	0x00000000 - 0x00003FFF
FC control unit	0x00200000 - 0x002003FF
FC timer	0x00200400 - 0x002007FF
FC event unit (FC private)	0x00204000 - 0x002043FF
FC Memory protection unit	0x00204400 - 0x002047FF
FC instruction cache control unit	0x00201400 - 0x002017FF
FC Core 0 (Debug Unit)	0x00300000 - 0x00307FFF

Table 4. Fabric Controller Subsystem aliased memory map table

5.3 Cluster aliased address map

Functional unit	Aliased address range
Cluster L1 RAM (64kB)	0x00000000 - 0x0000FFFF
Cluster L1 memory test and set unit	0x00100000 - 0x0010FFFF
Cluster control unit	0x00200000 - 0x002003FF
Cluster timer	0x00200400 - 0x002007FF
Cluster event unit (Cluster private)	0x00204000 - 0x002043FF
Cluster instruction cache control unit	0x00201400 - 0x002017FF
Hardware convolution engine	0x00201800 - 0x00201BFF
DMA	0x00204400 - 0x002047FF
Cluster Core 0 (Debug Unit)	0x00300000 - 0x00307FFF
Cluster Core 1 (Debug Unit)	0x00308000 - 0x0030FFFF
Cluster Core 2 (Debug Unit)	0x00310000 - 0x00317FFF
Cluster Core 3 (Debug Unit)	0x00318000 - 0x0031FFFF
Cluster Core 4 (Debug Unit)	0x00320000 - 0x00327FFF

Functional unit	Aliased address range
Cluster Core 5 (Debug Unit)	0x00328000 - 0x0032FFFF
Cluster Core 6 (Debug Unit)	0x00330000 - 0x00337FFF
Cluster Core 7 (Debug Unit)	0x00338000 - 0x0033FFFF

Table 5. Cluster Subsystem aliased memory map table

6 Device components description

6.1 RISC-V cores

The FC and cluster cores in GAP8 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. Please note that the RI5CY cores in GAP8 do not contain a floating point unit. For further explanation of the RISC-V instruction format please refer to the RISC-V standards documents.

6.1.1 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]

6.1.1.1 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])>>15 rD[31:16] = (rA[15:0]*rB[31:16] + rA[31:16]*rB[15:0])>>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])>>16 rD[31:16] = (rA[15:0]*rB[31:16] + rA[31:16]*rB[15:0])>>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])>>17 rD[31:16] = (rA[15:0]*rB[31:16] + rA[31:16]*rB[15:0])>>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])>>15 rD[31:16] = (rA[15:0]*rB[15:0] + rA[31:16]*rB[15:0])>>15
pv.cplxmul.s.sci	Vector by I6 Scalar rD[15:0] = (rA[15:0]*ExtS(I6) - rA[31:16]*ExtS(I6))>>15 rD[31:16] = (rA[15:0]*ExtS(I6) + rA[31:16]*ExtS(I6))>>15

Table 6. 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 7. Complex multiplication operations format table

6.1.1.2 Subtraction of 2 complexes with post rotation by -j

R = subrotmj(X, Y), R, X, Y complexes.

R.Re = X.Im – Y.Im R.Im = Y.Re – X.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]) >> 1 rD[31:16] = (rB[15:0] - rA[15:0]) >> 1
pv.subrotmj.h.div4 rD, rA, rB	rD[15:0] = (rA[31:16] - rB[31:16]) >> 2 rD[31:16] = (rB[15:0] - rA[15:0]) >> 2

Table 8. 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 9. Subtraction of 2 complexes with post rotation by -j format table

6.1.1.3 Complex conjugate operation

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

Table 10. Complex conjugate operation summary table

35 31	26	25	24 20	19 15	14 12	17 7	6 0
funct5	**F**	**S**	**rs2**	**rs1**	**funct3**	**rD**	**opcode**
01011	*1*	*000000*		src1	*000*	dest	*1010111*

Table 11. Complex conjugate operation format table

6.1.2 Extensions to existing RI5CY vector operations.

6.1.2.1 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]) >> 1 rD[31:16] = (rA[31:16] + rB[31:16]) >> 1
pv.add.b.div2 rD, rA, rB	rD[7:0] = (rA[7:0] + rB[7:0]) >> 1 rD[15:8] = (rA[15:8] + rB[15:8]) >> 1 rD[23:16] = (rA[23:16] + rB[23:16]) >> 1 rD[31:24] = (rA[31:24] + rB[31:24]) >> 1
pv.add.h.div4 rD, rA, rB	rD[15:0] = (rA[15:0] + rB[15:0]) >> 2 rD[31:16] = (rA[31:16] + rB[31:16]) >> 2

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

Table 12. 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 13. Addition of vector of half words with post right shift format table

6.1.2.2 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]) >> 1 rD[31:16] = (rA[31:16] - rB[31:16]) >> 1
pv.sub.b.div2 rD, rA, rB	rD[7:0] = (rA[7:0] - rB[7:0]) >> 1 rD[15:8] = (rA[15:8] - rB[15:8]) >> 1 rD[23:16] = (rA[23:16] - rB[23:16]) >> 1 rD[31:24] = (rA[31:24] - rB[31:24]) >> 1
pv.sub.h.div4 rD, rA, rB	rD[15:0] = (rA[15:0] - rB[15:0]) >> 2 rD[31:16] = (rA[31:16] - rB[31:16]) >> 2
pv.sub.b.div4 rD, rA, rB	rD[7:0] = (rA[7:0] - rB[7:0]) >> 2 rD[15:8] = (rA[15:8] - rB[15:8]) >> 2 rD[23:16] = (rA[23:16] - rB[23:16]) >> 2 rD[31:24] = (rA[31:24] - rB[31:24]) >> 2

Table 14. 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 15. Subtraction of vectors of half words with post right shift format table

6.1.3 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] <= rB[31:16]) $VF1 = (rA[15:0] <= rB[15:0])
pv.vitop.sel rD, rA, rB	rD[31:16] = ((rA[31:16]<<1)&!$VF0)|(rB[31:16]<<1 | $VF0) rD[15:0] = ((rA[15:0]<<1)&!$VF1)|(rB[15:0]<<1 | $VF1)

Table 16. 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 17. Viterbi specific instructions format table

6.1.4 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 18. 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 19. Enhanced shuffling format table

6.2 Cluster Subsystem

6.2.1 Cluster Subsystem Events

Event number	Event name	IP instance name	Direction	Description
0	SW_EVT_0	Cluster RISCY cores	Input	Software event 0 from one of the CL_COREs
1	SW_EVT_1	Cluster RISCY cores	Input	Software event 1 from one of the CL_COREs
2	SW_EVT_2	Cluster RISCY cores	Input	Software event 2 from one of the CL_COREs
3	SW_EVT_3	Cluster RISCY cores	Input	Software event 3 from one of the CL_COREs
4	SW_EVT_4	Cluster RISCY cores	Input	Software event 4 from one of the CL_COREs
5	SW_EVT_5	Cluster RISCY cores	Input	Software event 5 from one of the CL_COREs
6	SW_EVT_6	Cluster RISCY cores	Input	Software event 6 from one of the CL_COREs
7	SW_EVT_7	Cluster RISCY cores	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	CL_TIMER_UNIT	Input	Cluster basic timer low event
11	CL_TIMER_HI_EVT	CL_TIMER_UNIT	Input	Cluster basic timer high event
12	HWCE_EVT_0	HWCE	Input	Hardware convolution engine event 0
13	HWCE_EVT_1	HWCE	Input	Hardware convolution engine event 1
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	Reserved			Reserved
25	Reserved			Reserved
26	Reserved			Reserved
27	SOC_PERIPH_EVT	SOC_EVENT_GENERATOR	Input	SoC peripheral event
28	Reserved			Reserved
29	Reserved			Reserved
30	Reserved			Reserved
31	Reserved			Reserved

Table 20. Cluster Subsystem Events table

6.2.2 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

6.2.2.1 Cluster control unit registers

Name	Address	Aliased address	Size	Type	Access	Default	Description
EOC	0x10200000	0x00200000	32	Status	R/W	*0x0000*	End Of Computation status register.
FETCH_EN	0x10200008	0x00200008	32	Config	R/W	*0x0000*	Cluster cores fetch enable configuration register.
CLOCK_GATE	0x10200020	0x00200020	32	Config	R/W	*0x0000*	Cluster clock gate configuration register.
DBG_RESUME	0x10200028	0x00200028	32	Config	W	*0x0000*	Cluster cores debug resume register.
DBG_HALT_STATUS	0x10200028	0x00200028	32	Config	R	*0x0000*	Cluster cores debug halt status register.
DBG_HALT_MASK	0x10200038	0x00200038	32	Config	R/W	*0x0000*	Cluster cores debug halt mask configuration register.
BOOT_ADDR0	0x10200040	0x00200040	32	Config	R/W	*0x0000*	Cluster core 0 boot address configuration register.
TCDM_ARB_POLICY_CH0	0x10200080	0x00200080	32	Config	R/W	*0x0000*	TCDM arbitration policy ch0 for cluster cores configuration register.
TCDM_ARB_POLICY_CH1	0x10200088	0x00200088	32	Config	R/W	*0x0000*	TCDM arbitration policy ch1 for DMA/HWCE configuration register.
TCDM_ARB_POLICY_CH0_REP	0x102000C0	0x002000C0	32	Config	R/W	*0x0000*	Read only duplicate of TCDM_ARB_POLICY_CH0 register
TCDM_ARB_POLICY_CH1_REP	0x102000C8	0x002000C8	32	Config	R/W	*0x0000*	Read only duplicate of TCDM_ARB_POLICY_CH1 register

Table 21. Cluster control unit registers table

6.2.2.2 Cluster control unit registers details

6.2.2.2.1 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

6.2.2.2.2 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

6.2.2.2.3 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

6.2.2.2.4 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

6.2.2.2.5 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

6.2.2.2.6 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.

6.2.2.2.7 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.

6.2.2.2.8 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

6.2.2.2.9 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

6.2.2.2.10 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

6.2.2.2.11 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

6.2.3 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

6.2.3.1 Cluster timer registers

Name	Address	Aliased address	Size	Type	Access	Default	Description
CFG_LO	0x10200400	0x00200400	32	Config	R/W	*0x0000*	Timer Low Configuration register.
CFG_HI	0x10200404	0x00200404	32	Config	R/W	*0x0000*	Timer High Configuration register.
CNT_LO	0x10200408	0x00200408	32	Data	R/W	*0x0000*	Timer Low counter value register.
CNT_HI	0x1020040C	0x0020040C	32	Data	R/W	*0x0000*	Timer High counter value register.
CMP_LO	0x10200410	0x00200410	32	Config	R/W	*0x0000*	Timer Low comparator value register.
CMP_HI	0x10200414	0x00200414	32	Config	R/W	*0x0000*	Timer High comparator value register.
START_LO	0x10200418	0x00200418	32	Config	R/W	*0x0000*	Start Timer Low counting register.
START_HI	0x1020041C	0x0020041C	32	Config	R/W	*0x0000*	Start Timer High counting register.
RESET_LO	0x10200420	0x00200420	32	Config	R/W	*0x0000*	Reset Timer Low counter register.
RESET_HI	0x10200424	0x00200424	32	Config	R/W	*0x0000*	Reset Timer High counter register.

Table 22. Cluster timer registers table

6.2.3.2 Cluster timer registers details

6.2.3.2.1 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	Reserve d	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. Ftimer = Fclk / (1 + 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

6.2.3.2.2 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

6.2.3.2.3 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.

6.2.3.2.4 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.

6.2.3.2.5 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.

6.2.3.2.6 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.

6.2.3.2.7 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.

6.2.3.2.8 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.

6.2.3.2.9 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.

6.2.3.2.10 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.

6.2.4 Cluster event units

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.

6.2.4.1 Cluster event unit registers

Name	Address	Size	Type	Access	Default	Description
EVT_MASK_CORE0	0x10200800	32	Config	R/W	*0x0000*	Input event mask configuration register.
EVT_MASK_AND_CORE0	0x10200804	32	Config	W	*0x0000*	Input event mask update command register with bitwise AND operation.
EVT_MASK_OR_CORE0	0x10200808	32	Config	W	*0x0000*	Input event mask update command register with bitwise OR operation.
IRQ_MASK_CORE0	0x1020080C	32	Config	R/W	*0x0000*	Interrupt request mask configuration register.

Name	Address	Size	Type	Access	Default	Description
IRQ_MASK_AND_CORE0	0x10200810	32	Config	W	*0x0000*	Interrupt request mask update command register with bitwise AND operation.
IRQ_MASK_OR_CORE0	0x10200814	32	Config	W	*0x0000*	Interrupt request mask update command register with bitwise OR operation.
CLOCK_STATUS_CORE0	0x10200818	32	Config	R	*0x0000*	Cluster cores clock status register.
EVENT_BUFFER_CORE0	0x1020081C	32	Config	R	*0x0000*	Pending input events status register.
EVENT_BUFFER_MASKED_CORE0	0x10200820	32	Config	R	*0x0000*	Pending input events status register with EVT_MASK applied.
EVENT_BUFFER_IRQ_MASKED_CORE0	0x10200824	32	Config	R	*0x0000*	Pending input events status register with IRQ_MASK applied.
EVENT_BUFFER_CLEAR_CORE0	0x10200828	32	Config	W	*0x0000*	Pending input events status clear command register.
SW_EVENT_MASK_CORE0	0x1020082C	32	Config	R/W	*0x0000*	Software events cluster cores destination mask configuration register.
SW_EVENT_MASK_AND_CORE0	0x10200830	32	Config	W	*0x0000*	Software events cluster cores destination mask update command register with bitwise AND operation.
SW_EVENT_MASK_OR_CORE0	0x10200834	32	Config	W	*0x0000*	Software events cluster cores destination mask update command register with bitwise OR operation.
EVT_MASK_CORE1	0x10200840	32	Config	R/W	*0x0000*	Input event mask configuration register.
EVT_MASK_AND_CORE1	0x10200844	32	Config	W	*0x0000*	Input event mask update command register with bitwise AND operation.
EVT_MASK_OR_CORE1	0x10200848	32	Config	W	*0x0000*	Input event mask update command register with bitwise OR operation.
IRQ_MASK_CORE1	0x1020084C	32	Config	R/W	*0x0000*	Interrupt request mask configuration register.
IRQ_MASK_AND_CORE1	0x10200850	32	Config	W	*0x0000*	Interrupt request mask update command register with bitwise AND operation.
IRQ_MASK_OR_CORE1	0x10200854	32	Config	W	*0x0000*	Interrupt request mask update command register with bitwise OR operation.
CLOCK_STATUS_CORE1	0x10200858	32	Config	R	*0x0000*	Cluster cores clock status register.
EVENT_BUFFER_CORE1	0x1020085C	32	Config	R	*0x0000*	Pending input events status register.
EVENT_BUFFER_MASKED_CORE1	0x10200860	32	Config	R	*0x0000*	Pending input events status register with EVT_MASK applied.
EVENT_BUFFER_IRQ_MASKED_CORE1	0x10200864	32	Config	R	*0x0000*	Pending input events status register with IRQ_MASK applied.
EVENT_BUFFER_CLEAR_CORE1	0x10200868	32	Config	W	*0x0000*	Pending input events status clear command register.
SW_EVENT_MASK_CORE1	0x1020086C	32	Config	R/W	*0x0000*	Software events cluster cores destination mask configuration register.
SW_EVENT_MASK_AND_CORE1	0x10200870	32	Config	W	*0x0000*	Software events cluster cores destination mask update command register with bitwise AND operation.
SW_EVENT_MASK_OR_CORE1	0x10200874	32	Config	W	*0x0000*	Software events cluster cores destination mask update command register with bitwise OR operation.

Name	Address	Size	Type	Access	Default	Description
EVT_MASK_CORE2	0x10200880	32	Config	R/W	*0x0000*	Input event mask configuration register.
EVT_MASK_AND_CORE2	0x10200884	32	Config	W	*0x0000*	Input event mask update command register with bitwise AND operation.
EVT_MASK_OR_CORE2	0x10200888	32	Config	W	*0x0000*	Input event mask update command register with bitwise OR operation.
IRQ_MASK_CORE2	0x1020088C	32	Config	R/W	*0x0000*	Interrupt request mask configuration register.
IRQ_MASK_AND_CORE2	0x10200890	32	Config	W	*0x0000*	Interrupt request mask update command register with bitwise AND operation.
IRQ_MASK_OR_CORE2	0x10200894	32	Config	W	*0x0000*	Interrupt request mask update command register with bitwise OR operation.
CLOCK_STATUS_CORE2	0x10200898	32	Config	R	*0x0000*	Cluster cores clock status register.
EVENT_BUFFER_CORE2	0x1020089C	32	Config	R	*0x0000*	Pending input events status register.
EVENT_BUFFER_MASKED_CORE2	0x102008A0	32	Config	R	*0x0000*	Pending input events status register with EVT_MASK applied.
EVENT_BUFFER_IRQ_MASKED_CORE2	0x102008A4	32	Config	R	*0x0000*	Pending input events status register with IRQ_MASK applied.
EVENT_BUFFER_CLEAR_CORE2	0x102008A8	32	Config	W	*0x0000*	Pending input events status clear command register.
SW_EVENT_MASK_CORE2	0x102008AC	32	Config	R/W	*0x0000*	Software events cluster cores destination mask configuration register.
SW_EVENT_MASK_AND_CORE2	0x102008B0	32	Config	W	*0x0000*	Software events cluster cores destination mask update command register with bitwise AND operation.
SW_EVENT_MASK_OR_CORE2	0x102008B4	32	Config	W	*0x0000*	Software events cluster cores destination mask update command register with bitwise OR operation.
EVT_MASK_CORE3	0x102008C0	32	Config	R/W	*0x0000*	Input event mask configuration register.
EVT_MASK_AND_CORE3	0x102008C4	32	Config	W	*0x0000*	Input event mask update command register with bitwise AND operation.
EVT_MASK_OR_CORE3	0x102008C8	32	Config	W	*0x0000*	Input event mask update command register with bitwise OR operation.
IRQ_MASK_CORE3	0x102008CC	32	Config	R/W	*0x0000*	Interrupt request mask configuration register.
IRQ_MASK_AND_CORE3	0x102008D0	32	Config	W	*0x0000*	Interrupt request mask update command register with bitwise AND operation.
IRQ_MASK_OR_CORE3	0x102008D4	32	Config	W	*0x0000*	Interrupt request mask update command register with bitwise OR operation.
CLOCK_STATUS_CORE3	0x102008D8	32	Config	R	*0x0000*	Cluster cores clock status register.
EVENT_BUFFER_CORE3	0x102008DC	32	Config	R	*0x0000*	Pending input events status register.
EVENT_BUFFER_MASKED_CORE3	0x102008E0	32	Config	R	*0x0000*	Pending input events status register with EVT_MASK applied.
EVENT_BUFFER_IRQ_MASKED_CORE3	0x102008E4	32	Config	R	*0x0000*	Pending input events status register with IRQ_MASK applied.
EVENT_BUFFER_CLEAR_CORE3	0x102008E8	32	Config	W	*0x0000*	Pending input events status clear command register.

Name	Address	Size	Type	Access	Default	Description
SW_EVENT_MASK_CORE3	0x102008EC	32	Config	R/W	*0x0000*	Software events cluster cores destination mask configuration register.
SW_EVENT_MASK_AND_CORE3	0x102008F0	32	Config	W	*0x0000*	Software events cluster cores destination mask update command register with bitwise AND operation.
SW_EVENT_MASK_OR_CORE3	0x102008F4	32	Config	W	*0x0000*	Software events cluster cores destination mask update command register with bitwise OR operation.
EVT_MASK_CORE4	0x10200900	32	Config	R/W	*0x0000*	Input event mask configuration register.
EVT_MASK_AND_CORE4	0x10200904	32	Config	W	*0x0000*	Input event mask update command register with bitwise AND operation.
EVT_MASK_OR_CORE4	0x10200908	32	Config	W	*0x0000*	Input event mask update command register with bitwise OR operation.
IRQ_MASK_CORE4	0x1020090C	32	Config	R/W	*0x0000*	Interrupt request mask configuration register.
IRQ_MASK_AND_CORE4	0x10200910	32	Config	W	*0x0000*	Interrupt request mask update command register with bitwise AND operation.
IRQ_MASK_OR_CORE4	0x10200914	32	Config	W	*0x0000*	Interrupt request mask update command register with bitwise OR operation.
CLOCK_STATUS_CORE4	0x10200918	32	Config	R	*0x0000*	Cluster cores clock status register.
EVENT_BUFFER_CORE4	0x1020091C	32	Config	R	*0x0000*	Pending input events status register.
EVENT_BUFFER_MASKED_CORE4	0x10200920	32	Config	R	*0x0000*	Pending input events status register with EVT_MASK applied.
EVENT_BUFFER_IRQ_MASKED_CORE4	0x10200924	32	Config	R	*0x0000*	Pending input events status register with IRQ_MASK applied.
EVENT_BUFFER_CLEAR_CORE4	0x10200928	32	Config	W	*0x0000*	Pending input events status clear command register.
SW_EVENT_MASK_CORE4	0x1020092C	32	Config	R/W	*0x0000*	Software events cluster cores destination mask configuration register.
SW_EVENT_MASK_AND_CORE4	0x10200930	32	Config	W	*0x0000*	Software events cluster cores destination mask update command register with bitwise AND operation.
SW_EVENT_MASK_OR_CORE4	0x10200934	32	Config	W	*0x0000*	Software events cluster cores destination mask update command register with bitwise OR operation.
EVT_MASK_CORE5	0x10200940	32	Config	R/W	*0x0000*	Input event mask configuration register.
EVT_MASK_AND_CORE5	0x10200944	32	Config	W	*0x0000*	Input event mask update command register with bitwise AND operation.
EVT_MASK_OR_CORE5	0x10200948	32	Config	W	*0x0000*	Input event mask update command register with bitwise OR operation.
IRQ_MASK_CORE5	0x1020094C	32	Config	R/W	*0x0000*	Interrupt request mask configuration register.
IRQ_MASK_AND_CORE5	0x10200950	32	Config	W	*0x0000*	Interrupt request mask update command register with bitwise AND operation.
IRQ_MASK_OR_CORE5	0x10200954	32	Config	W	*0x0000*	Interrupt request mask update command register with bitwise OR operation.

Name	Address	Size	Type	Access	Default	Description
CLOCK_STATUS_CORE5	0x10200958	32	Config	R	*0x0000*	Cluster cores clock status register.
EVENT_BUFFER_CORE5	0x1020095C	32	Config	R	*0x0000*	Pending input events status register.
EVENT_BUFFER_MASKED_CORE5	0x10200960	32	Config	R	*0x0000*	Pending input events status register with EVT_MASK applied.
EVENT_BUFFER_IRQ_MASKED_CORE5	0x10200964	32	Config	R	*0x0000*	Pending input events status register with IRQ_MASK applied.
EVENT_BUFFER_CLEAR_CORE5	0x10200968	32	Config	W	*0x0000*	Pending input events status clear command register.
SW_EVENT_MASK_CORE5	0x1020096C	32	Config	R/W	*0x0000*	Software events cluster cores destination mask configuration register.
SW_EVENT_MASK_AND_CORE5	0x10200970	32	Config	W	*0x0000*	Software events cluster cores destination mask update command register with bitwise AND operation.
SW_EVENT_MASK_OR_CORE5	0x10200974	32	Config	W	*0x0000*	Software events cluster cores destination mask update command register with bitwise OR operation.
EVT_MASK_CORE6	0x10200980	32	Config	R/W	*0x0000*	Input event mask configuration register.
EVT_MASK_AND_CORE6	0x10200984	32	Config	W	*0x0000*	Input event mask update command register with bitwise AND operation.
EVT_MASK_OR_CORE6	0x10200988	32	Config	W	*0x0000*	Input event mask update command register with bitwise OR operation.
IRQ_MASK_CORE6	0x1020098C	32	Config	R/W	*0x0000*	Interrupt request mask configuration register.
IRQ_MASK_AND_CORE6	0x10200990	32	Config	W	*0x0000*	Interrupt request mask update command register with bitwise AND operation.
IRQ_MASK_OR_CORE6	0x10200994	32	Config	W	*0x0000*	Interrupt request mask update command register with bitwise OR operation.
CLOCK_STATUS_CORE6	0x10200998	32	Config	R	*0x0000*	Cluster cores clock status register.
EVENT_BUFFER_CORE6	0x1020099C	32	Config	R	*0x0000*	Pending input events status register.
EVENT_BUFFER_MASKED_CORE6	0x102009A0	32	Config	R	*0x0000*	Pending input events status register with EVT_MASK applied.
EVENT_BUFFER_IRQ_MASKED_CORE6	0x102009A4	32	Config	R	*0x0000*	Pending input events status register with IRQ_MASK applied.
EVENT_BUFFER_CLEAR_CORE6	0x102009A8	32	Config	W	*0x0000*	Pending input events status clear command register.
SW_EVENT_MASK_CORE6	0x102009AC	32	Config	R/W	*0x0000*	Software events cluster cores destination mask configuration register.
SW_EVENT_MASK_AND_CORE6	0x102009B0	32	Config	W	*0x0000*	Software events cluster cores destination mask update command register with bitwise AND operation.
SW_EVENT_MASK_OR_CORE6	0x102009B4	32	Config	W	*0x0000*	Software events cluster cores destination mask update command register with bitwise OR operation.
EVT_MASK_CORE7	0x102009C0	32	Config	R/W	*0x0000*	Input event mask configuration register.
EVT_MASK_AND_CORE7	0x102009C4	32	Config	W	*0x0000*	Input event mask update command register with bitwise AND operation.
EVT_MASK_OR_CORE7	0x102009C8	32	Config	W	*0x0000*	Input event mask update command register with bitwise OR operation.

Name	Address	Size	Type	Access	Default	Description
IRQ_MASK_CORE7	0x102009CC	32	Config	R/W	*0x0000*	Interrupt request mask configuration register.
IRQ_MASK_AND_CORE7	0x102009D0	32	Config	W	*0x0000*	Interrupt request mask update command register with bitwise AND operation.
IRQ_MASK_OR_CORE7	0x102009D4	32	Config	W	*0x0000*	Interrupt request mask update command register with bitwise OR operation.
CLOCK_STATUS_CORE7	0x102009D8	32	Config	R	*0x0000*	Cluster cores clock status register.
EVENT_BUFFER_CORE7	0x102009DC	32	Config	R	*0x0000*	Pending input events status register.
EVENT_BUFFER_MASKED_CORE7	0x102009E0	32	Config	R	*0x0000*	Pending input events status register with EVT_MASK applied.
EVENT_BUFFER_IRQ_MASKED_CORE7	0x102009E4	32	Config	R	*0x0000*	Pending input events status register with IRQ_MASK applied.
EVENT_BUFFER_CLEAR_CORE7	0x102009E8	32	Config	W	*0x0000*	Pending input events status clear command register.
SW_EVENT_MASK_CORE7	0x102009EC	32	Config	R/W	*0x0000*	Software events cluster cores destination mask configuration register.
SW_EVENT_MASK_AND_CORE7	0x102009F0	32	Config	W	*0x0000*	Software events cluster cores destination mask update command register with bitwise AND operation.
SW_EVENT_MASK_OR_CORE7	0x102009F4	32	Config	W	*0x0000*	Software events cluster cores destination mask update command register with bitwise OR operation.
HW_BARRIER_0_TRIG_MASK	0x10200C00	32	Config	R/W	*0x0000*	Cluster hardware barrier 0 trigger mask configuration register.
HW_BARRIER_0_STATUS	0x10200C04	32	Status	R	*0x0000*	Cluster hardware barrier 0 status register.
HW_BARRIER_0_STATUS_SUM	0x10200C08	32	Status	R	*0x0000*	Cluster hardware barrier summary status register.
HW_BARRIER_0_TARGET_MASK	0x10200C0C	32	Config	R/W	*0x0000*	Cluster hardware barrier 0 target mask configuration register.
HW_BARRIER_0_TRIG	0x10200C10	32	Config	W	*0x0000*	Cluster hardware barrier 0 trigger command register.
HW_BARRIER_1_TRIG_MASK	0x10200C20	32	Config	R/W	*0x0000*	Cluster hardware barrier 1 trigger mask configuration register.
HW_BARRIER_1_STATUS	0x10200C24	32	Status	R	*0x0000*	Cluster hardware barrier 1 status register.
HW_BARRIER_1_STATUS_SUM	0x10200C28	32	Status	R	*0x0000*	Cluster hardware barrier summary status register.
HW_BARRIER_1_TARGET_MASK	0x10200C2C	32	Config	R/W	*0x0000*	Cluster hardware barrier 1 target mask configuration register.
HW_BARRIER_1_TRIG	0x10200C30	32	Config	W	*0x0000*	Cluster hardware barrier 1 trigger command register.
HW_BARRIER_2_TRIG_MASK	0x10200C40	32	Config	R/W	*0x0000*	Cluster hardware barrier 2 trigger mask configuration register.
HW_BARRIER_2_STATUS	0x10200C44	32	Status	R	*0x0000*	Cluster hardware barrier 2 status register.
HW_BARRIER_2_STATUS_SUM	0x10200C48	32	Status	R	*0x0000*	Cluster hardware barrier summary status register.
HW_BARRIER_2_TARGET_MASK	0x10200C4C	32	Config	R/W	*0x0000*	Cluster hardware barrier 2 target mask configuration register.

Name	Address	Size	Type	Access	Default	Description
HW_BARRIER_2_TRIG	0x10200C50	32	Config	W	*0x0000*	Cluster hardware barrier 2 trigger command register.
HW_BARRIER_3_TRIG_MASK	0x10200C60	32	Config	R/W	*0x0000*	Cluster hardware barrier 3 trigger mask configuration register.
HW_BARRIER_3_STATUS	0x10200C64	32	Status	R	*0x0000*	Cluster hardware barrier 3 status register.
HW_BARRIER_3_STATUS_SUM	0x10200C68	32	Status	R	*0x0000*	Cluster hardware barrier summary status register.
HW_BARRIER_3_TARGET_MASK	0x10200C6C	32	Config	R/W	*0x0000*	Cluster hardware barrier 3 target mask configuration register.
HW_BARRIER_3_TRIG	0x10200C70	32	Config	W	*0x0000*	Cluster hardware barrier 3 trigger command register.
HW_BARRIER_4_TRIG_MASK	0x10200C80	32	Config	R/W	*0x0000*	Cluster hardware barrier 4 trigger mask configuration register.
HW_BARRIER_4_STATUS	0x10200C84	32	Status	R	*0x0000*	Cluster hardware barrier 4 status register.
HW_BARRIER_4_STATUS_SUM	0x10200C88	32	Status	R	*0x0000*	Cluster hardware barrier summary status register.
HW_BARRIER_4_TARGET_MASK	0x10200C8C	32	Config	R/W	*0x0000*	Cluster hardware barrier 4 target mask configuration register.
HW_BARRIER_4_TRIG	0x10200C90	32	Config	W	*0x0000*	Cluster hardware barrier 4 trigger command register.
HW_BARRIER_5_TRIG_MASK	0x10200CA0	32	Config	R/W	*0x0000*	Cluster hardware barrier 5 trigger mask configuration register.
HW_BARRIER_5_STATUS	0x10200CA4	32	Status	R	*0x0000*	Cluster hardware barrier 5 status register.
HW_BARRIER_5_STATUS_SUM	0x10200CA8	32	Status	R	*0x0000*	Cluster hardware barrier summary status register.
HW_BARRIER_5_TARGET_MASK	0x10200CAC	32	Config	R/W	*0x0000*	Cluster hardware barrier 5 target mask configuration register.
HW_BARRIER_5_TRIG	0x10200CB0	32	Config	W	*0x0000*	Cluster hardware barrier 5 trigger command register.
HW_BARRIER_6_TRIG_MASK	0x10200CC0	32	Config	R/W	*0x0000*	Cluster hardware barrier 6 trigger mask configuration register.
HW_BARRIER_6_STATUS	0x10200CC4	32	Status	R	*0x0000*	Cluster hardware barrier 6 status register.
HW_BARRIER_6_STATUS_SUM	0x10200CC8	32	Status	R	*0x0000*	Cluster hardware barrier summary status register.
HW_BARRIER_6_TARGET_MASK	0x10200CCC	32	Config	R/W	*0x0000*	Cluster hardware barrier 6 target mask configuration register.
HW_BARRIER_6_TRIG	0x10200CD0	32	Config	W	*0x0000*	Cluster hardware barrier 6 trigger command register.
HW_BARRIER_7_TRIG_MASK	0x10200CE0	32	Config	R/W	*0x0000*	Cluster hardware barrier 7 trigger mask configuration register.
HW_BARRIER_7_STATUS	0x10200CE4	32	Status	R	*0x0000*	Cluster hardware barrier 7 status register.
HW_BARRIER_7_STATUS_SUM	0x10200CE8	32	Status	R	*0x0000*	Cluster hardware barrier summary status register.
HW_BARRIER_7_TARGET_MASK	0x10200CEC	32	Config	R/W	*0x0000*	Cluster hardware barrier 7 target mask configuration register.
HW_BARRIER_7_TRIG	0x10200CF0	32	Config	W	*0x0000*	Cluster hardware barrier 7 trigger command register.
SW_EVENT_0_TRIG	0x10200E00	32	Config	W	*0x0000*	Cluster Software event 0 trigger command register.

Name	Address	Size	Type	Access	Default	Description
SW_EVENT_1_TRIG	0x10200E04	32	Config	W	*0x0000*	Cluster Software event 1 trigger command register.
SW_EVENT_2_TRIG	0x10200E08	32	Config	W	*0x0000*	Cluster Software event 2 trigger command register.
SW_EVENT_3_TRIG	0x10200E0C	32	Config	W	*0x0000*	Cluster Software event 3 trigger command register.
SW_EVENT_4_TRIG	0x10200E10	32	Config	W	*0x0000*	Cluster Software event 4 trigger command register.
SW_EVENT_5_TRIG	0x10200E14	32	Config	W	*0x0000*	Cluster Software event 5 trigger command register.
SW_EVENT_6_TRIG	0x10200E18	32	Config	W	*0x0000*	Cluster Software event 6 trigger command register.
SW_EVENT_7_TRIG	0x10200E1C	32	Config	W	*0x0000*	Cluster Software event 7 trigger command register.
SOC_PERIPH_EVENT_ID	0x10200F00	32	Status	R	*0x0000*	Cluster SoC peripheral event ID status register.

Table 23. Cluster event unit registers table

6.2.4.2 Cluster event unit (Cluster private) registers

Name	Aliased address	Size	Type	Access	Default	Description
EVT_MASK	0x00204000	32	Config	R/W	*0x0000*	Input event mask configuration register.
HW_DISPATCH_PUSH_TASK	0x00204080	32	Config	W	*0x0000*	Hardware task dispatcher push command register.
HW_DISPATCH_POP_TASK	0x00204080	32	Config	R	*0x0000*	Hardware task dispatcher pop command register.
HW_MUTEX_0_MSG_PUT	0x002040C0	32	Config	W	*0x0000*	Hardware mutex 0 non-blocking put command register.
HW_MUTEX_0_MSG_GET	0x002040C0	32	Config	R	*0x0000*	Hardware mutex 0 blocking get command register.
SW_EVENT_0_TRIG	0x00204100	32	Config	W	*0x0000*	Cluster Software event 0 trigger command register.
SW_EVENT_0_TRIG_WAIT	0x00204140	32	Config	R	*0x0000*	Cluster Software event 0 trigger and wait command register.
SW_EVENT_0_TRIG_WAIT_CLEAR	0x00204180	32	Config	R	*0x0000*	Cluster Software event 0 trigger, wait and clear command register.
HW_BARRIER_0_TRIG_MASK	0x00204200	32	Config	R/W	*0x0000*	Cluster hardware barrier 0 trigger mask configuration register.
EVT_MASK_AND	0x00204004	32	Config	W	*0x0000*	Input event mask update command register with bitwise AND operation.
HW_DISPATCH_PUSH_TEAM_CONFIG	0x00204084	32	Config	W	*0x0000*	Hardware task dispatcher cluster core team configuration register.
HW_MUTEX_1_MSG_PUT	0x002040C4	32	Config	W	*0x0000*	Hardware mutex 1 non-blocking put command register.
HW_MUTEX_1_MSG_GET	0x002040C4	32	Config	R	*0x0000*	Hardware mutex 1 blocking get command register.
SW_EVENT_1_TRIG	0x00204104	32	Config	W	*0x0000*	Cluster Software event 1 trigger command register.
SW_EVENT_1_TRIG_WAIT	0x00204144	32	Config	R	*0x0000*	Cluster Software event 1 trigger and wait command register.

Name	Aliased address	Size	Type	Access	Default	Description
SW_EVENT_1_TRIG_WAIT_CLEAR	0x00204184	32	Config	R	*0x0000*	Cluster Software event 1 trigger, wait and clear command register.
HW_BARRIER_0_STATUS	0x00204204	32	Status	R	*0x0000*	Cluster hardware barrier 0 status register.
EVT_MASK_OR	0x00204008	32	Config	W	*0x0000*	Input event mask update command register with bitwise OR operation.
SW_EVENT_2_TRIG	0x00204108	32	Config	W	*0x0000*	Cluster Software event 2 trigger command register.
SW_EVENT_2_TRIG_WAIT	0x00204148	32	Config	R	*0x0000*	Cluster Software event 2 trigger and wait command register.
SW_EVENT_2_TRIG_WAIT_CLEAR	0x00204188	32	Config	R	*0x0000*	Cluster Software event 2 trigger, wait and clear command register.
HW_BARRIER_0_STATUS_SUM	0x00204208	32	Status	R	*0x0000*	Cluster hardware barrier summary status register.
IRQ_MASK	0x0020400C	32	Config	R/W	*0x0000*	Interrupt request mask configuration register.
SW_EVENT_3_TRIG	0x0020410C	32	Config	W	*0x0000*	Cluster Software event 3 trigger command register.
SW_EVENT_3_TRIG_WAIT	0x0020414C	32	Config	R	*0x0000*	Cluster Software event 3 trigger and wait command register.
SW_EVENT_3_TRIG_WAIT_CLEAR	0x0020418C	32	Config	R	*0x0000*	Cluster Software event 3 trigger, wait and clear command register.
HW_BARRIER_0_TARGET_MASK	0x0020420C	32	Config	R/W	*0x0000*	Cluster hardware barrier 0 target mask configuration register.
IRQ_MASK_AND	0x00204010	32	Config	W	*0x0000*	Interrupt request mask update command register with bitwise AND operation.
SW_EVENT_4_TRIG	0x00204110	32	Config	W	*0x0000*	Cluster Software event 4 trigger command register.
SW_EVENT_4_TRIG_WAIT	0x00204150	32	Config	R	*0x0000*	Cluster Software event 4 trigger and wait command register.
SW_EVENT_4_TRIG_WAIT_CLEAR	0x00204190	32	Config	R	*0x0000*	Cluster Software event 4 trigger, wait and clear command register.
HW_BARRIER_0_TRIG	0x00204210	32	Config	W	*0x0000*	Cluster hardware barrier 0 trigger command register.
IRQ_MASK_OR	0x00204014	32	Config	W	*0x0000*	Interrupt request mask update command register with bitwise OR operation.
SW_EVENT_5_TRIG	0x00204114	32	Config	W	*0x0000*	Cluster Software event 5 trigger command register.
SW_EVENT_5_TRIG_WAIT	0x00204154	32	Config	R	*0x0000*	Cluster Software event 5 trigger and wait command register.
SW_EVENT_5_TRIG_WAIT_CLEAR	0x00204194	32	Config	R	*0x0000*	Cluster Software event 5 trigger, wait and clear command register.
HW_BARRIER_0_SELF_TRIG	0x00204214	32	Config	R	*0x0000*	Cluster hardware barrier 0 self trigger command register.
CLOCK_STATUS	0x00204018	32	Config	R	*0x0000*	Cluster cores clock status register.
SW_EVENT_6_TRIG	0x00204118	32	Config	W	*0x0000*	Cluster Software event 6 trigger command register.
SW_EVENT_6_TRIG_WAIT	0x00204158	32	Config	R	*0x0000*	Cluster Software event 6 trigger and wait command register.
SW_EVENT_6_TRIG_WAIT_CLEAR	0x00204198	32	Config	R	*0x0000*	Cluster Software event 6 trigger, wait and clear command register.

Name	Aliased address	Size	Type	Access	Default	Description
HW_BARRIER_0_TRIG_WAIT	0x00204218	32	Config	R	*0x0000*	Cluster hardware barrier 0 trigger and wait command register.
EVENT_BUFFER	0x0020401C	32	Config	R	*0x0000*	Pending input events status register.
SW_EVENT_7_TRIG	0x0020411C	32	Config	W	*0x0000*	Cluster Software event 7 trigger command register.
SW_EVENT_7_TRIG_WAIT	0x0020415C	32	Config	R	*0x0000*	Cluster Software event 7 trigger and wait command register.
SW_EVENT_7_TRIG_WAIT_CLEAR	0x0020419C	32	Config	R	*0x0000*	Cluster Software event 7 trigger, wait and clear command register.
HW_BARRIER_0_TRIG_WAIT_CLEAR	0x0020421C	32	Config	R	*0x0000*	Cluster hardware barrier 0 trigger, wait and clear command register.
EVENT_BUFFER_MASKED	0x00204020	32	Config	R	*0x0000*	Pending input events status register with EVT_MASK applied.
HW_BARRIER_1_TRIG_MASK	0x00204220	32	Config	R/W	*0x0000*	Cluster hardware barrier 1 trigger mask configuration register.
EVENT_BUFFER_IRQ_MASKED	0x00204024	32	Config	R	*0x0000*	Pending input events status register with IRQ_MASK applied.
HW_BARRIER_1_STATUS	0x00204224	32	Status	R	*0x0000*	Cluster hardware barrier 1 status register.
EVENT_BUFFER_CLEAR	0x00204028	32	Config	W	*0x0000*	Pending input events status clear command register.
HW_BARRIER_1_STATUS_SUM	0x00204228	32	Status	R	*0x0000*	Cluster hardware barrier summary status register.
SW_EVENT_MASK	0x0020402C	32	Config	R/W	*0x0000*	Software events cluster cores destination mask configuration register.
HW_BARRIER_1_TARGET_MASK	0x0020422C	32	Config	R/W	*0x0000*	Cluster hardware barrier 1 target mask configuration register.
SW_EVENT_MASK_AND	0x00204030	32	Config	W	*0x0000*	Software events cluster cores destination mask update command register with bitwise AND operation.
HW_BARRIER_1_TRIG	0x00204230	32	Config	W	*0x0000*	Cluster hardware barrier 1 trigger command register.
SW_EVENT_MASK_OR	0x00204034	32	Config	W	*0x0000*	Software events cluster cores destination mask update command register with bitwise OR operation.
HW_BARRIER_1_SELF_TRIG	0x00204234	32	Config	R	*0x0000*	Cluster hardware barrier 1 self trigger command register.
EVENT_WAIT	0x00204038	32	Config	R	*0x0000*	Input event wait command register.
HW_BARRIER_1_TRIG_WAIT	0x00204238	32	Config	R	*0x0000*	Cluster hardware barrier 1 trigger and wait command register.
EVENT_WAIT_CLEAR	0x0020403C	32	Config	R	*0x0000*	Input event wait and clear command register.
HW_BARRIER_1_TRIG_WAIT_CLEAR	0x0020423C	32	Config	R	*0x0000*	Cluster hardware barrier 1 trigger, wait and clear command register.
HW_BARRIER_2_TRIG_MASK	0x00204240	32	Config	R/W	*0x0000*	Cluster hardware barrier 2 trigger mask configuration register.
HW_BARRIER_2_STATUS	0x00204244	32	Status	R	*0x0000*	Cluster hardware barrier 2 status register.
HW_BARRIER_2_STATUS_SUM	0x00204248	32	Status	R	*0x0000*	Cluster hardware barrier summary status register.
HW_BARRIER_2_TARGET_MASK	0x0020424C	32	Config	R/W	*0x0000*	Cluster hardware barrier 2 target mask configuration register.

Name	Aliased address	Size	Type	Access	Default	Description
HW_BARRIER_2_TRIG	0x00204250	32	Config	W	*0x0000*	Cluster hardware barrier 2 trigger command register.
HW_BARRIER_2_SELF_TRIG	0x00204254	32	Config	R	*0x0000*	Cluster hardware barrier 2 self trigger command register.
HW_BARRIER_2_TRIG_WAIT	0x00204258	32	Config	R	*0x0000*	Cluster hardware barrier 2 trigger and wait command register.
HW_BARRIER_2_TRIG_WAIT_CLEAR	0x0020425C	32	Config	R	*0x0000*	Cluster hardware barrier 2 trigger, wait and clear command register.
HW_BARRIER_3_TRIG_MASK	0x00204260	32	Config	R/W	*0x0000*	Cluster hardware barrier 3 trigger mask configuration register.
HW_BARRIER_3_STATUS	0x00204264	32	Status	R	*0x0000*	Cluster hardware barrier 3 status register.
HW_BARRIER_3_STATUS_SUM	0x00204268	32	Status	R	*0x0000*	Cluster hardware barrier summary status register.
HW_BARRIER_3_TARGET_MASK	0x0020426C	32	Config	R/W	*0x0000*	Cluster hardware barrier 3 target mask configuration register.
HW_BARRIER_3_TRIG	0x00204270	32	Config	W	*0x0000*	Cluster hardware barrier 3 trigger command register.
HW_BARRIER_3_SELF_TRIG	0x00204274	32	Config	R	*0x0000*	Cluster hardware barrier 3 self trigger command register.
HW_BARRIER_3_TRIG_WAIT	0x00204278	32	Config	R	*0x0000*	Cluster hardware barrier 3 trigger and wait command register.
HW_BARRIER_3_TRIG_WAIT_CLEAR	0x0020427C	32	Config	R	*0x0000*	Cluster hardware barrier 3 trigger, wait and clear command register.
HW_BARRIER_4_TRIG_MASK	0x00204280	32	Config	R/W	*0x0000*	Cluster hardware barrier 4 trigger mask configuration register.
HW_BARRIER_4_STATUS	0x00204284	32	Status	R	*0x0000*	Cluster hardware barrier 4 status register.
HW_BARRIER_4_STATUS_SUM	0x00204288	32	Status	R	*0x0000*	Cluster hardware barrier summary status register.
HW_BARRIER_4_TARGET_MASK	0x0020428C	32	Config	R/W	*0x0000*	Cluster hardware barrier 4 target mask configuration register.
HW_BARRIER_4_TRIG	0x00204290	32	Config	W	*0x0000*	Cluster hardware barrier 4 trigger command register.
HW_BARRIER_4_SELF_TRIG	0x00204294	32	Config	R	*0x0000*	Cluster hardware barrier 4 self trigger command register.
HW_BARRIER_4_TRIG_WAIT	0x00204298	32	Config	R	*0x0000*	Cluster hardware barrier 4 trigger and wait command register.
HW_BARRIER_4_TRIG_WAIT_CLEAR	0x0020429C	32	Config	R	*0x0000*	Cluster hardware barrier 4 trigger, wait and clear command register.
HW_BARRIER_5_TRIG_MASK	0x002042A0	32	Config	R/W	*0x0000*	Cluster hardware barrier 5 trigger mask configuration register.
HW_BARRIER_5_STATUS	0x002042A4	32	Status	R	*0x0000*	Cluster hardware barrier 5 status register.
HW_BARRIER_5_STATUS_SUM	0x002042A8	32	Status	R	*0x0000*	Cluster hardware barrier summary status register.
HW_BARRIER_5_TARGET_MASK	0x002042AC	32	Config	R/W	*0x0000*	Cluster hardware barrier 5 target mask configuration register.
HW_BARRIER_5_TRIG	0x002042B0	32	Config	W	*0x0000*	Cluster hardware barrier 5 trigger command register.
HW_BARRIER_5_SELF_TRIG	0x002042B4	32	Config	R	*0x0000*	Cluster hardware barrier 5 self trigger command register.

Name	Aliased address	Size	Type	Access	Default	Description
HW_BARRIER_5_TRIG_WAIT	0x002042B8	32	Config	R	*0x0000*	Cluster hardware barrier 5 trigger and wait command register.
HW_BARRIER_5_TRIG_WAIT_CLEAR	0x002042BC	32	Config	R	*0x0000*	Cluster hardware barrier 5 trigger, wait and clear command register.
HW_BARRIER_6_TRIG_MASK	0x002042C0	32	Config	R/W	*0x0000*	Cluster hardware barrier 6 trigger mask configuration register.
HW_BARRIER_6_STATUS	0x002042C4	32	Status	R	*0x0000*	Cluster hardware barrier 6 status register.
HW_BARRIER_6_STATUS_SUM	0x002042C8	32	Status	R	*0x0000*	Cluster hardware barrier summary status register.
HW_BARRIER_6_TARGET_MASK	0x002042CC	32	Config	R/W	*0x0000*	Cluster hardware barrier 6 target mask configuration register.
HW_BARRIER_6_TRIG	0x002042D0	32	Config	W	*0x0000*	Cluster hardware barrier 6 trigger command register.
HW_BARRIER_6_SELF_TRIG	0x002042D4	32	Config	R	*0x0000*	Cluster hardware barrier 6 self trigger command register.
HW_BARRIER_6_TRIG_WAIT	0x002042D8	32	Config	R	*0x0000*	Cluster hardware barrier 6 trigger and wait command register.
HW_BARRIER_6_TRIG_WAIT_CLEAR	0x002042DC	32	Config	R	*0x0000*	Cluster hardware barrier 6 trigger, wait and clear command register.
HW_BARRIER_7_TRIG_MASK	0x002042E0	32	Config	R/W	*0x0000*	Cluster hardware barrier 7 trigger mask configuration register.
HW_BARRIER_7_STATUS	0x002042E4	32	Status	R	*0x0000*	Cluster hardware barrier 7 status register.
HW_BARRIER_7_STATUS_SUM	0x002042E8	32	Status	R	*0x0000*	Cluster hardware barrier summary status register.
HW_BARRIER_7_TARGET_MASK	0x002042EC	32	Config	R/W	*0x0000*	Cluster hardware barrier 7 target mask configuration register.
HW_BARRIER_7_TRIG	0x002042F0	32	Config	W	*0x0000*	Cluster hardware barrier 7 trigger command register.
HW_BARRIER_7_SELF_TRIG	0x002042F4	32	Config	R	*0x0000*	Cluster hardware barrier 7 self trigger command register.
HW_BARRIER_7_TRIG_WAIT	0x002042F8	32	Config	R	*0x0000*	Cluster hardware barrier 7 trigger and wait command register.
HW_BARRIER_7_TRIG_WAIT_CLEAR	0x002042FC	32	Config	R	*0x0000*	Cluster hardware barrier 7 trigger, wait and clear command register.

Table 24. Cluster event unit (Cluster private) registers table

6.2.4.3 Cluster event unit registers details

6.2.4.3.1 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

6.2.4.3.2 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.

6.2.4.3.3 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.

6.2.4.3.4 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.

6.2.4.3.5 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.

6.2.4.3.6 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*.

6.2.4.3.7 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

6.2.4.3.8 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

6.2.4.3.9 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.

6.2.4.3.10 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								HB0TM							

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.

6.2.4.3.11 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*.

6.2.4.3.12 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*.

6.2.4.3.13 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 bitfiled. It is a non-blocking access.

6.2.4.3.14 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.

6.2.4.3.15 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*.

6.2.4.3.16 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

6.2.4.3.17 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

6.2.4.3.18 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.

6.2.4.3.19 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*.

6.2.4.3.20 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*.

6.2.4.3.21 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

6.2.4.3.22 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

6.2.4.3.23 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.

6.2.4.3.24 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

6.2.4.3.25 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*.

6.2.4.3.26 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

6.2.4.3.27 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

6.2.4.3.28 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. HBATM[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.

6.2.4.3.29 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*.

6.2.4.3.30 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*.

6.2.4.3.31 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

6.2.4.3.32 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

6.2.4.3.33 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*.

6.2.4.3.34 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*.

6.2.4.3.35 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*.

6.2.4.3.36 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

6.2.4.3.37 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

6.2.4.3.38 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.

6.2.4.3.39 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

6.2.4.3.40 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*.

6.2.4.3.41 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

6.2.4.3.42 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

6.2.4.3.43 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

6.2.4.3.44 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.

6.2.4.3.45 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*.

6.2.4.3.46 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

6.2.4.3.47 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

6.2.4.3.48 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

6.2.4.3.49 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.

6.2.4.3.50 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.

6.2.4.3.51 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.

6.2.4.3.52 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.

6.2.4.3.53 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*.

6.2.4.3.54 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.

6.2.4.3.55 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]

6.2.4.3.56 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. HBATM[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.

6.2.4.3.57 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*.

6.2.4.3.58 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*.

6.2.4.3.59 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*.

6.2.4.3.60 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.

6.2.4.3.61 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

6.2.4.3.62 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

6.2.4.3.63 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

6.2.4.3.64 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

6.2.4.3.65 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.

6.2.4.3.66 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.

6.2.4.3.67 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.

6.2.4.3.68 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 - **HBTAM** *(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.

6.2.4.3.69 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*.

6.2.4.3.70 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.

6.2.4.3.71 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

6.2.4.3.72 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

6.2.4.3.73 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.

6.2.4.3.74 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.

6.2.4.3.75 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.

6.2.4.3.76 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 - **HBTAM** *(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.

6.2.4.3.77 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*.

6.2.4.3.78 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.

6.2.4.3.79 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

6.2.4.3.80 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

6.2.4.3.81 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.

6.2.4.3.82 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.

6.2.4.3.83 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.

6.2.4.3.84 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 - **HBTAM** *(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.

6.2.4.3.85 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*.

6.2.4.3.86 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.

6.2.4.3.87 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

6.2.4.3.88 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

6.2.4.3.89 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.

6.2.4.3.90 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.

6.2.4.3.91 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.

6.2.4.3.92 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 - **HBTAM** *(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.

6.2.4.3.93 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*.

6.2.4.3.94 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.

6.2.4.3.95 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

6.2.4.3.96 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

6.2.4.3.97 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.

6.2.4.3.98 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.

6.2.4.3.99 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.

6.2.4.3.100 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. HBATM[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.

6.2.4.3.101 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*.

6.2.4.3.102 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.

6.2.4.3.103 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

6.2.4.3.104 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

6.2.4.3.105 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.

6.2.4.3.106 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.

6.2.4.3.107 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.

6.2.4.3.108 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 - **HBTAM** *(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.

6.2.4.3.109 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*.

6.2.4.3.110 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.

6.2.4.3.111 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

6.2.4.3.112 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

6.2.5 Cluster instruction cache control unit

CL_ICACHE_CTRL component manages the following features:

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

6.2.5.1 Cluster instruction cache control unit registers

Name	Address	Aliased address	Size	Type	Access	Default	Description
ENABLE	0x10201400	0x00201400	32	Config	W	*0x0000*	Cluster instruction cache unit enable configuration register.
FLUSH	0x10201404	0x00201404	32	Config	W	*0x0000*	Cluster instruction cache unit flush command register.
SEL_FLUSH	0x1020140C	0x0020140C	32	Config	W	*0x0000*	Cluster instruction cache unit selective flush command register.

Table 25. Cluster instruction cache control unit registers table

6.2.5.2 Cluster instruction cache control unit registers details

6.2.5.2.1 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

6.2.5.2.2 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.

6.2.5.2.3 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.

6.2.6 Hardware convolution engine

The Hardware Convolution Engine or HWCE is a special-purpose coprocessor designed for accelerated computation of convolution-accumulation kernels within GAP. In particular, it is targeted towards acceleration of Convolutional Neural Networks (CNNs). The

HWCE assumes input and output pixels and convolution weights are fixed point numbers. Various bit widths of pixels and weights are available.

Differently from most tightly-coupled accelerator, the HWCE is not tied with a specific core but rather it is tightly integrated within the cluster. Memory access is performed through the ports that are directly plugged into the logarithmic interconnect, while control of the accelerator can be performed by means of a configuration port available via the cluster peripheral interconnect.

This means that the HWCE directly shares data in the L1 with the RI5CY cores in the GAP8 cluster, similarly to the way that cores do with each another. Therefore, from the perspective of the shared memory the HWCE is essentially undistinguishable from a cluster core.

While behaving as a core externally, internally the HWCE is a dataflow engine that performs the convolution-accumulation operation over streams of data.

6.2.6.1 Hardware convolution engine registers

Name	Address	Aliased address	Size	Type	Access	Default	Description
TRIGGER	0x10201800	0x00201800	32	Config	W	*0x0000*	Trigger the execution of an offloaded job
ACQUIRE	0x10201804	0x00201804	32	Config	R	*0x0000*	Acquire the lock to offload job
FINISHED_JOBS	0x10201808	0x00201808	32	Status	R	*0x0000*	Number of concluded jobs since last read
STATUS	0x1020180C	0x0020180C	32	Status	R	*0x0000*	Status of the HWCE
RUNNING_JOB	0x10201810	0x00201810	32	Status	R	*0x0000*	ID of the currently running job
SOFT_CLEAR	0x10201814	0x00201814	32	Config	W	*0x0000*	Reset HWCE to known idle state
GEN_CONFIG0	0x10201820	0x00201820	32	Config	R/W	*0x0000*	Generic configuration register 0
GEN_CONFIG1	0x10201824	0x00201824	32	Config	R/W	*0x0000*	Generic configuration register 1
Y_TRANS_SIZE_ALIASED	0x10201840	0x00201840	32	Config	R/W	*0x0000*	Total number of words to be read for y_{in} and y_{out}
Y_LINE_STRIDE_LENGTH_ALIASED	0x10201844	0x00201844	32	Config	R/W	*0x0000*	Line stride and length for y_{in} and y_{out}
Y_FEAT_STRIDE_LENGTH_ALIASED	0x10201848	0x00201848	32	Config	R/W	*0x0000*	Feature (block) stride and length for y_{in} and y_{out}
Y_OUT_3_BASE_ADDR_ALIASED	0x1020184C	0x0020184C	32	Config	R/W	*0x0000*	Base address of $y_{out}[3]$
Y_OUT_2_BASE_ADDR_ALIASED	0x10201850	0x00201850	32	Config	R/W	*0x0000*	Base address of $y_{out}[2]$
Y_OUT_1_BASE_ADDR_ALIASED	0x10201854	0x00201854	32	Config	R/W	*0x0000*	Base address of $y_{out}[1]$
Y_OUT_0_BASE_ADDR_ALIASED	0x10201858	0x00201858	32	Config	R/W	*0x0000*	Base address of $y_{out}[0]$
Y_IN_3_BASE_ADDR_ALIASED	0x1020185C	0x0020185C	32	Config	R/W	*0x0000*	Base address of $y_{in}[3]$
Y_IN_2_BASE_ADDR_ALIASED	0x10201860	0x00201860	32	Config	R/W	*0x0000*	Base address of $y_{in}[2]$
Y_IN_1_BASE_ADDR_ALIASED	0x10201864	0x00201864	32	Config	R/W	*0x0000*	Base address of $y_{in}[1]$
Y_IN_0_BASE_ADDR_ALIASED	0x10201868	0x00201868	32	Config	R/W	*0x0000*	Base address of $y_{in}[0]$
X_TRANS_SIZE_ALIASED	0x1020186C	0x0020186C	32	Config	R/W	*0x0000*	Total number of words to be read for x_{in}
X_LINE_STRIDE_LENGTH_ALIASED	0x10201870	0x00201870	32	Config	R/W	*0x0000*	Line stride and length for x_{in}
X_FEAT_STRIDE_LENGTH_ALIASED	0x10201874	0x00201874	32	Config	R/W	*0x0000*	Feature (block) stride and length for x_{in}
X_IN_BASE_ADDR_ALIASED	0x10201878	0x00201878	32	Config	R/W	*0x0000*	Base address of x_{in}

Name	Address	Aliased address	Size	Type	Access	Default	Description
W_BASE_ADDR_ALIASED	0x1020187C	0x0020187C	32	Config	R/W	*0x0000*	Base address of W
JOB_CONFIG0_ALIASED	0x10201880	0x00201880	32	Config	R/W	*0x0000*	Job configuration register 0
JOB_CONFIG1_ALIASED	0x10201884	0x00201884	32	Config	R/W	*0x0000*	Job configuration register 1
Y_TRANS_SIZE_CTX0	0x10201940	0x00201940	32	Config	R	*0x0000*	Total number of words to be read for y_{in} and y_{out}
Y_LINE_STRIDE_LENGTH_CTX0	0x10201944	0x00201944	32	Config	R	*0x0000*	Line stride and length for y_{in} and y_{out}
Y_FEAT_STRIDE_LENGTH_CTX0	0x10201948	0x00201948	32	Config	R	*0x0000*	Feature (block) stride and length for y_{in} and y_{out}
Y_OUT_3_BASE_ADDR_CTX0	0x1020194C	0x0020194C	32	Config	R	*0x0000*	Base address of $y_{out}[3]$
Y_OUT_2_BASE_ADDR_CTX0	0x10201950	0x00201950	32	Config	R	*0x0000*	Base address of $y_{out}[2]$
Y_OUT_1_BASE_ADDR_CTX0	0x10201954	0x00201954	32	Config	R	*0x0000*	Base address of $y_{out}[1]$
Y_OUT_0_BASE_ADDR_CTX0	0x10201958	0x00201958	32	Config	R	*0x0000*	Base address of $y_{out}[0]$
Y_IN_3_BASE_ADDR_CTX0	0x1020195C	0x0020195C	32	Config	R	*0x0000*	Base address of $y_{in}[3]$
Y_IN_2_BASE_ADDR_CTX0	0x10201960	0x00201960	32	Config	R	*0x0000*	Base address of $y_{in}[2]$
Y_IN_1_BASE_ADDR_CTX0	0x10201964	0x00201964	32	Config	R	*0x0000*	Base address of $y_{in}[1]$
Y_IN_0_BASE_ADDR_CTX0	0x10201968	0x00201968	32	Config	R	*0x0000*	Base address of $y_{in}[0]$
X_TRANS_SIZE_CTX0	0x1020196C	0x0020196C	32	Config	R	*0x0000*	Total number of words to be read for x_{in}
X_LINE_STRIDE_LENGTH_CTX0	0x10201970	0x00201970	32	Config	R	*0x0000*	Line stride and length for x_{in}
X_FEAT_STRIDE_LENGTH_CTX0	0x10201974	0x00201974	32	Config	R	*0x0000*	Feature (block) stride and length for x_{in}
X_IN_BASE_ADDR_CTX0	0x10201978	0x00201978	32	Config	R	*0x0000*	Base address of x_{in}
W_BASE_ADDR_CTX0	0x1020197C	0x0020197C	32	Config	R	*0x0000*	Base address of W
JOB_CONFIG0_CTX0	0x10201980	0x00201980	32	Config	R	*0x0000*	Job configuration register 0
JOB_CONFIG1_CTX0	0x10201984	0x00201984	32	Config	R	*0x0000*	Job configuration register 1
Y_TRANS_SIZE_CTX1	0x10201A40	0x00201A40	32	Config	R	*0x0000*	Total number of words to be read for y_{in} and y_{out}
Y_LINE_STRIDE_LENGTH_CTX1	0x10201A44	0x00201A44	32	Config	R	*0x0000*	Line stride and length for y_{in} and y_{out}
Y_FEAT_STRIDE_LENGTH_CTX1	0x10201A48	0x00201A48	32	Config	R	*0x0000*	Feature (block) stride and length for y_{in} and y_{out}
Y_OUT_3_BASE_ADDR_CTX1	0x10201A4C	0x00201A4C	32	Config	R	*0x0000*	Base address of $y_{out}[3]$
Y_OUT_2_BASE_ADDR_CTX1	0x10201A50	0x00201A50	32	Config	R	*0x0000*	Base address of $y_{out}[2]$
Y_OUT_1_BASE_ADDR_CTX1	0x10201A54	0x00201A54	32	Config	R	*0x0000*	Base address of $y_{out}[1]$
Y_OUT_0_BASE_ADDR_CTX1	0x10201A58	0x00201A58	32	Config	R	*0x0000*	Base address of $y_{out}[0]$
Y_IN_3_BASE_ADDR_CTX1	0x10201A5C	0x00201A5C	32	Config	R	*0x0000*	Base address of $y_{in}[3]$
Y_IN_2_BASE_ADDR_CTX1	0x10201A60	0x00201A60	32	Config	R	*0x0000*	Base address of $y_{in}[2]$
Y_IN_1_BASE_ADDR_CTX1	0x10201A64	0x00201A64	32	Config	R	*0x0000*	Base address of $y_{in}[1]$
Y_IN_0_BASE_ADDR_CTX1	0x10201A68	0x00201A68	32	Config	R	*0x0000*	Base address of $y_{in}[0]$
X_TRANS_SIZE_CTX1	0x10201A6C	0x00201A6C	32	Config	R	*0x0000*	Total number of words to be read for x_{in}

Name	Address	Aliased address	Size	Type	Access	Default	Description
X_LINE_STRIDE_LENGTH_CTX1	0x10201A70	0x00201A70	32	Config	R	*0x0000*	Line stride and length for x_{in}
X_FEAT_STRIDE_LENGTH_CTX1	0x10201A74	0x00201A74	32	Config	R	*0x0000*	Feature (block) stride and length for x_{in}
X_IN_BASE_ADDR_CTX1	0x10201A78	0x00201A78	32	Config	R	*0x0000*	Base address of x_{in}
W_BASE_ADDR_CTX1	0x10201A7C	0x00201A7C	32	Config	R	*0x0000*	Base address of W
JOB_CONFIG0_CTX1	0x10201A80	0x00201A80	32	Config	R	*0x0000*	Job configuration register 0
JOB_CONFIG1_CTX1	0x10201A84	0x00201A84	32	Config	R	*0x0000*	Job configuration register 1

Table 26. Hardware convolution engine registers table

6.2.6.2 Hardware convolution engine registers details

6.2.6.2.1 Trigger the execution of an offloaded job (TRIGGER)

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ANY															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ANY															

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

Write of any value will close the current offload phase by releasing the job offload lock and inserting the currently offloaded job in the control queue.

6.2.6.2.2 Acquire the lock to offload job (ACQUIRE)

Reset value: 0x0000

Any read from this register has the “side effect” of initiating an offload sequence by acquiring the job offload lock. Until the offloading core releases the lock by writing to the TRIGGER register, no other core can start a job offload.
A read to the ACQUIRE register has the further side of effect of copying the full status of the previous context inside the new context to be offloaded, an operation that is performed in 18 cycles.
Any write to this register is ignored.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ERR															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERR								ID_ERR							

Bits 31:8 - **ERR** *(R)*

An error code if one of the following conditions apply:

1. if the context copy is going on, it will answer 0xfffffffd (-3)
2. else, if the job offload lock has been established, it will answer 0xfffffffe (-2)
3. else, if the job queue is full, it will answer 0xffffffff (-1)

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

If ERR is 0 then the ID of the offloaded job. Otherwise, part of the error code

6.2.6.2.3 Number of concluded jobs since last read (FINISHED_JOBS)

Reset value: 0x0000

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

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

The number of jobs that the HWCE executed and finished since the last time the same FINISHED_JOBS register was accessed. A read to FINISHED_JOBS returns:

- 0x0 if no job was completed since the last access
- 0x1 if a single job was completed since the last access
- 0x2 if two or more jobs were completed since the last access

6.2.6.2.4 Status of the HWCE (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															ST

Bit 0 - **ST** *(R)*

Status of the HWCE

- *0b0* Not running a job
- *0b1* Running a job

6.2.6.2.5 ID of the currently running job (RUNNING_JOB)

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								ID							

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

ID of the currently running job

6.2.6.2.6 Reset HWCE to known idle state (SOFT_CLEAR)

Reset value: 0x0000

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

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

A write of any value to this register will reset the HWCE to its idle state.

6.2.6.2.7 Generic configuration register 0 (GEN_CONFIG0)

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WSTRIDE															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		NCP	CONV		VECT		UNS	NY	NF	QF					

Bits 31:16 - **WSTRIDE** *(R/W)*

Stride between one FILTER_SIZExFILTER_SIZE filter and the next.

Bit 13 - **NCP** *(R/W)*

No job copy:

- *0b0* - do job copy
- *0b1* - don't do job copy

Bits 12:11 - **CONV** *(R/W)*

Convolution mode:

- *0b00* - mode is 5x5.
- *0b01* - mode is 3x3.
- *0b10* - mode is 4x7.

Bits 10:9 - **VECT** *(R/W)*

Vector mode:

- *0b00* - mode is scalar (1 feat/cycle, 1x16-bit weights).
- *0b01* - mode is vectorial 2 (2 feat/cycle, 2x8-bit weights).
- *0b10* - mode is vectorial 4 (4 feat/cycle, 4x4-bit weights).

Bit 8 - **UNS** *(R/W)*

Set unsigned multiplication

- *0b0* - Consider multiplication results as signed fixed-point numbers.
- *0b1* - Consider multiplication results as unsigned fixed-point numbers.

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

No y_in mode:

- *0b0* - Normal operation
- *0b1* - Disable loading of y_in and add a constant set in the CONFIG2 register

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

Operation type:

- *0b0* - Normal convolution
- *0b1* - Does not flip weights (i.e. implements a stencil instead of a mathematical convolution).

Bits 5:0 - **QF** *(R/W)*

Fixed-point format. Pixels will be shifted to the right by QF bits in the

normalization step after the sum-of-products stage.

6.2.6.2.8 Generic configuration register 1 (GEN_CONFIG1)

Reset value: 0x0000

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

Bits 20:16 - **PIXSHIFTR** *(R/W)*

Shift output pixels to the right by this number of positions when PIXMODE is not 16bit.

Bits 9:8 - **PIXMODE** *(R/W)*

Input pixel size

- *0b00* – 16bit
- *0b01* - 8bit
- *0b10* - 4bit
- *0b11* – 16bit bis

Bits 4:0 - **PIXSHIFTL** *(R/W)*

Shift input pixels to the left by this number of positions when PIXMODE is not 16bit.

6.2.6.2.9 Total number of words to be read for y_{in} and y_{out} (Y_TRANS_SIZE)

Reset value: 0x0000

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

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

Contains the total number of words (i.e. double 16bit pixels, quad 8bit pixels, etc.) to be read / written by the streaming source and sink interfaces for y_{in} and y_{out} streams.

6.2.6.2.10 Line stride and length for y_{in} and y_{out} (Y_LINE_STRIDE_LENGTH)

Reset value: 0x0000

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

Bits 31:16 - **STRIDE** *(R/W)*

Distance in bytes between two consecutive lines.

Bits 15:0 - **LENGTH** *(R/W)*

Length of a line in number of words

6.2.6.2.11 Feature (block) stride and length for y_{in} and y_{out} (Y_FEAT_STRIDE_LENGTH)

Reset value: 0x0000

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

Bits 31:16 - **STRIDE** *(R/W)*

Distance in bytes between two consecutive lines.

Bits 15:0 - **LENGTH** *(R/W)*

Length of a line in number of words

6.2.6.2.12 Base address of y_{out}[3] (Y_OUT_3_BASE_ADDR)

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** *(R/W)*

Pointer into cluster L1 memory (4x4 bit mode)

6.2.6.2.13 Base address of y_{out}[2] (Y_OUT_2_BASE_ADDR)

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** *(R/W)*

Pointer into cluster L1 memory (4x4 bit mode)

6.2.6.2.14 Base address of y_{out}[1] (Y_OUT_1_BASE_ADDR)

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** *(R/W)*

Pointer into cluster L1 memory (4x4 bit and 2x8 bit modes)

6.2.6.2.15 Base address of y_{out}[0] (Y_OUT_0_BASE_ADDR)

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** *(R/W)*

Pointer into cluster L1 memory (4x4 bit, 2x8 bit and 1x16 bit modes)

6.2.6.2.16 Base address of y_{in}[3] (Y_IN_3_BASE_ADDR)

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** *(R/W)*

Pointer into cluster L1 memory (4x4 bit mode)

6.2.6.2.17 Base address of y_{in}[2] (Y_IN_2_BASE_ADDR)

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** *(R/W)*

Pointer into cluster L1 memory (4x4 bit mode)

6.2.6.2.18 Base address of y_{in}[1] (Y_IN_1_BASE_ADDR)

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** *(R/W)*

Pointer into cluster L1 memory (4x4 bit and 2x8 bit modes)

6.2.6.2.19 Base address of y_{in}[0] (Y_IN_0_BASE_ADDR)

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** *(R/W)*

Pointer into cluster L1 memory (4x4 bit, 2x8 bit and 1x16 bit modes)

6.2.6.2.20 Total number of words to be read for x_{in} (X_TRANS_SIZE)

Reset value: 0x0000

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

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

Contains the total number of words (i.e. double 16bit pixels, quad 8bit pixels, etc.) to be read / written by the streaming source interface for the x_{in} stream.

6.2.6.2.21 Line stride and length for x_{in} (X_LINE_STRIDE_LENGTH)

Reset value: 0x0000

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

Bits 31:16 - **STRIDE** *(R/W)*

Distance in bytes between two consecutive lines.

Bits 15:0 - **LENGTH** *(R/W)*

Length of a line in number of words

6.2.6.2.22 Feature (block) stride and length for x_{in} (X_FEAT_STRIDE_LENGTH)

Reset value: 0x0000

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

Bits 31:16 - **STRIDE** *(R/W)*

Distance in bytes between two consecutive lines.

Bits 15:0 - **LENGTH** *(R/W)*

Length of a line in number of words

6.2.6.2.23 Base address of x_{in} (X_IN_BASE_ADDR)

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** *(R/W)*

Pointer into cluster L1 memory

6.2.6.2.24 Base address of W (W_BASE_ADDR)

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** *(R/W)*

Pointer into cluster L1 memory

6.2.6.2.25 Job configuration register 0 (JOB_CONFIG0)

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NOYCONST															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						LBUFLEN									

Bits 31:16 - **NOYCONST** *(R/W)*

Constant to sum instead of y_{in} if the NY flag is active in the CONFIG1 register.

Bits 9:0 - **LBUFLEN** *(R/W)*

Linebuffer virtual length. Set to the same number as X_LINE_LENGTH. Acceptable LBUFLEN values range between 2 and LINEBUF_LENGTH.

6.2.6.2.26 Job configuration register 1 (JOB_CONFIG1)

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						LN	LO	Reserved		WIF_PARAM					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		WOF_PARAM						Reserved				VECT_DISABLE_MASK			

Bit 25 - **LN** *(R/W)*

Looping mechanism:

- *0b0* - both counters work as inner loops
- *0b1* - outer loop feature address is updated only when the feature counter reaches the wrap parameter, inner loop feature address is updated when the counter is less than the wrap parameter and reset when it is reached

Bit 24 - **LO** *(R/W)*

Loop order:

- *0b0* - output features (OF) are the outer loop
- *0b1* - input features (IF) are the outer loop

Bits 21:16 - **WIF_PARAM** *(R/W)*

Input feature (IF) counter wrap parameter. If both WIF and WOF are *0b0*, the looping mechanism is disabled

Bits 13:8 - **WOF_PARAM** *(R/W)*

Output feature (OF) counter wrap parameter. If both WIF and WOF are *0b0*, the looping mechanism is disabled

Bits 3:0 - **VECT_DISABLE_MASK** *(R/W)*

Vector mode mask. Defaults to 0x0, which means that all vectors are enabled. Can be used to disable unused vector routes when using approximate vector or 3x3 mode. The bits are reversed in order, so bit 3 indicates vector 0, bit 2 vector 1, etc.

6.2.7 DMA

Cluster DMA component manages the following features:

- 8 RX/TX full-duplex channels
- Up to 16 outstanding transfers between L1 and L2 memories
- Linear or 2D transfers modes

6.2.7.1 DMA registers

Name	Aliased address	Size	Type	Access	Default	Description
CMD	0x00204400	32	Config	R/W	*0x0000*	Cluster DMA configuration register.
STATUS	0x00204404	32	Status	R/W	*0x0000*	Cluster DMA status register.

Table 27. DMA registers table

6.2.7.2 DMA registers details

6.2.7.2.1 Cluster DMA configuration register. (CMD)

Reset value: 0x0000

Host access bus: DEMUX

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

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

Format is operation dependent. See below.

6.2.7.2.2 Cluster DMA status register. (STATUS)

Reset value: 0x0000

Host access bus: DEMUX

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

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

Format is operation dependent. See below.

6.2.7.3 DMA states

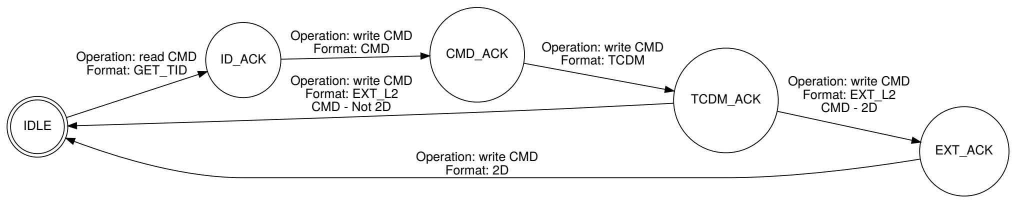

Figure 5. Queue transaction with ID

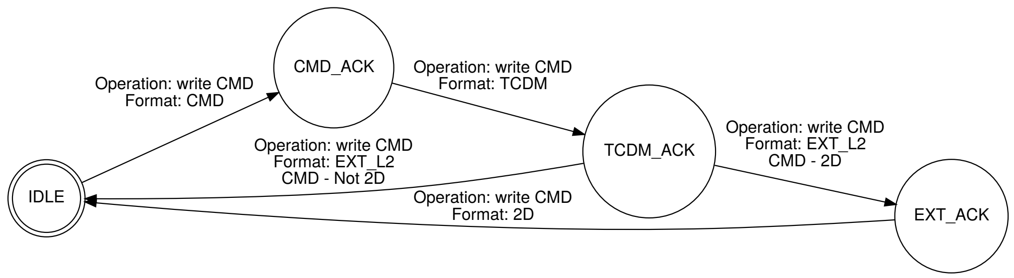

Figure 6. Queue transaction without ID

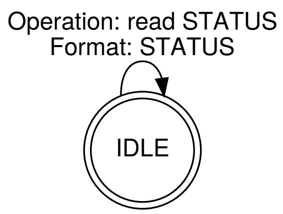

Figure 7. Get DMA status

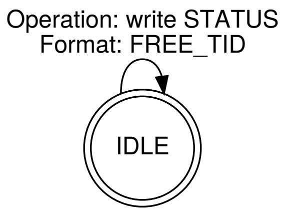

Figure 8. Free DMA transfer

6.2.7.4 DMA state command formats

Format Name	**Register**	**Size**	**Access type**	**Description**
STATUS	STATUS	32	R	Cluster DMA transfer free command format.

Format Name	Register	Size	Access type	Description
TCDM	CMD	32	W	Cluster DMA L1 base address configuration format.
FREE_TID	STATUS	32	W	Cluster DMA transfer status format.
GET_TID	CMD	32	R	Cluster DMA transfer identifier format.
CMD	CMD	32	W	Cluster DMA transfer configuration format.
EXT_L2	CMD	32	W	Cluster DMA L2 base address configuration format.
2D	CMD	32	W	Cluster DMA 2D transfer configuration format.

Table 28. DMA command format table

6.2.7.4.1 Cluster DMA transfer free command format. (STATUS)

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

Bits 31:16 - **TID_ALLOC** *(R)*

Transfer status bitfield:

- TID_TR[i]=*0b0* means that transfer allocator with TID i-16 is free.
- TID_TR[i]=*0b1* means that transfer allocator with TID i-16 is reserved.

Bits 15:0 - **TID_TR** *(R)*

Transfer status bitfield:

TID_TR[i]=*0b1* means that transfer with TID i is active.

6.2.7.4.2 Cluster DMA L1 base address configuration format. (TCDM)

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)*

Transfer L1 base address configuration bitfield.

6.2.7.4.3 Cluster DMA transfer status format. (FREE_TID)

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**
TID_FREE															

Bits 15:0 - **TID_FREE** *(W)*

Transfer canceller configuration bitfield. Writing a *0b1* in TID_FREE[i] will free transfer with TID i.

6.2.7.4.4 Cluster DMA transfer identifier format. (GET_TID)

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												TID			

Bits 3:0 - **TID** *(R)*

Transfer identifier value bitfield.

6.2.7.4.5 Cluster DMA transfer configuration format. (CMD)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved										BLE	ILE	ELE	2D	INC	TYPE
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
LEN															

Bit 21 - **BLE** *(W)*

Transfer event or interrupt broadcast configuration bitfield:

- *0b0*: event or interrupt is routed to the cluster core who initiated the transfer
- *0b1*: event or interrupt are broadcasted to all cluster cores

Bit 20 - **ILE** *(W)*

Transfer interrupt generation configuration bitfield:

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

Bit 19 - **ELE** *(W)*

Transfer event generation configuration bitfield:

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

Bit 18 - **2D** *(W)*

Transfer type configuration bitfield:

- *0b0*: linear transfer
- *0b1*: 2D transfer

Bit 17 - **INC** *(W)*

Transfer incremental configuration bitfield:

- *0b0*: non incremental
- *0b1*: incremental

Bit 16 - **TYPE** *(W)*

Transfer direction configuration bitfield:

- *0b0*: L1 to L2
- *0b1*: L2 to L1

Bits 15:0 - **LEN** *(W)*

Transfer length in bytes configuration bitfield.

6.2.7.4.6 Cluster DMA L2 base address configuration format. (EXT_L2)

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)*

Transfer L2 base address configuration bitfield.

6.2.7.4.7 Cluster DMA 2D transfer configuration format. (2D)

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

Bits 31:16 - **STRIDE** *(W)*

2D transfer stride value configuration bitfield.

Bits 15:0 - **LEN** *(W)*

2D transfer length value configuration bitfield.

6.2.8 Cluster RISCY cores

Cluster RI5CY Debug component manages the following features:

- controls break and single step RI5CY core execution modes
- configurable execution behavior on RI5CY core exception occurrence
- access to PC, GPR and CSR RI5CY core registers
- no HW breakpoint are provided

6.2.8.1 Cluster Core 0 (Debug Unit) registers

Name	Address	Aliased address	Size	Type	Access	Default	Description
CTRL	0x10300000	0x00300000	32	Config	R/W	*0x0000*	Debug control configuration register.
HIT	0x10300004	0x00300004	32	Config	R/W	*0x0000*	Debug hit status register.
IE	0x10300008	0x00300008	32	Config	R/W	*0x0000*	Debug exception trap enable configuration register.
CAUSE	0x1030000C	0x0030000C	32	Config	R	*0x0000*	Debug trap cause status register.
GPR0	0x10300400	0x00300400	32	Config	R/W	*0x0000*	Core general purpose register 0 value register.
GPR1	0x10300404	0x00300404	32	Config	R/W	*0x0000*	Core general purpose register 1 value register.
GPR2	0x10300408	0x00300408	32	Config	R/W	*0x0000*	Core general purpose register 2 value register.
GPR3	0x1030040C	0x0030040C	32	Config	R/W	*0x0000*	Core general purpose register 3 value register.
GPR4	0x10300410	0x00300410	32	Config	R/W	*0x0000*	Core general purpose register 4 value register.
GPR5	0x10300414	0x00300414	32	Config	R/W	*0x0000*	Core general purpose register 5 value register.
GPR6	0x10300418	0x00300418	32	Config	R/W	*0x0000*	Core general purpose register 6 value register.
GPR7	0x1030041C	0x0030041C	32	Config	R/W	*0x0000*	Core general purpose register 7 value register.
GPR8	0x10300420	0x00300420	32	Config	R/W	*0x0000*	Core general purpose register 8 value register.
GPR9	0x10300424	0x00300424	32	Config	R/W	*0x0000*	Core general purpose register 9 value register.
GPR10	0x10300428	0x00300428	32	Config	R/W	*0x0000*	Core general purpose register 10 value register.

Name	Address	Aliased address	Size	Type	Access	Default	Description
GPR11	0x1030042C	0x0030042C	32	Config	R/W	*0x0000*	Core general purpose register 11 value register.
GPR12	0x10300430	0x00300430	32	Config	R/W	*0x0000*	Core general purpose register 12 value register.
GPR13	0x10300434	0x00300434	32	Config	R/W	*0x0000*	Core general purpose register 13 value register.
GPR14	0x10300438	0x00300438	32	Config	R/W	*0x0000*	Core general purpose register 14 value register.
GPR15	0x1030043C	0x0030043C	32	Config	R/W	*0x0000*	Core general purpose register 15 value register.
GPR16	0x10300440	0x00300440	32	Config	R/W	*0x0000*	Core general purpose register 16 value register.
GPR17	0x10300444	0x00300444	32	Config	R/W	*0x0000*	Core general purpose register 17 value register.
GPR18	0x10300448	0x00300448	32	Config	R/W	*0x0000*	Core general purpose register 18 value register.
GPR19	0x1030044C	0x0030044C	32	Config	R/W	*0x0000*	Core general purpose register 19 value register.
GPR20	0x10300450	0x00300450	32	Config	R/W	*0x0000*	Core general purpose register 20 value register.
GPR21	0x10300454	0x00300454	32	Config	R/W	*0x0000*	Core general purpose register 21 value register.
GPR22	0x10300458	0x00300458	32	Config	R/W	*0x0000*	Core general purpose register 22 value register.
GPR23	0x1030045C	0x0030045C	32	Config	R/W	*0x0000*	Core general purpose register 23 value register.
GPR24	0x10300460	0x00300460	32	Config	R/W	*0x0000*	Core general purpose register 24 value register.
GPR25	0x10300464	0x00300464	32	Config	R/W	*0x0000*	Core general purpose register 25 value register.
GPR26	0x10300468	0x00300468	32	Config	R/W	*0x0000*	Core general purpose register 26 value register.
GPR27	0x1030046C	0x0030046C	32	Config	R/W	*0x0000*	Core general purpose register 27 value register.
GPR28	0x10300470	0x00300470	32	Config	R/W	*0x0000*	Core general purpose register 28 value register.
GPR29	0x10300474	0x00300474	32	Config	R/W	*0x0000*	Core general purpose register 29 value register.
GPR30	0x10300478	0x00300478	32	Config	R/W	*0x0000*	Core general purpose register 30 value register.
GPR31	0x1030047C	0x0030047C	32	Config	R/W	*0x0000*	Core general purpose register 31 value register.
NPC	0x10302000	0x00302000	32	Config	R/W	*0x0000*	Debug next program counter value register.
PPC	0x10302004	0x00302004	32	Config	R	*0x0000*	Debug previous program counter value register.
CSR_UHARTID	0x10304050	0x00304050	32	Config	R	*0x0000*	Core CSR user privilege mode hardware thread ID status register.
CSR_MSTATUS	0x10304C00	0x00304C00	32	Config	R/W	*0x0000*	Core CSR machine status value register.
CSR_MTVEC	0x10304C14	0x00304C14	32	Config	R/W	*0x0000*	Core CSR machine vector-trap base address value register.
CSR_MEPC	0x10304D04	0x00304D04	32	Config	R/W	*0x0000*	Core CSR machine exception program counter value register.
CSR_MCAUSE	0x10304D08	0x00304D08	32	Config	R/W	*0x0000*	Core CSR machine trap cause value register.
CSR_PCCR	0x10305E00	0x00305E00	32	Config	R/W	*0x0000*	Core CSR performance counter counter register.
CSR_PCER	0x10305E80	0x00305E80	32	Config	R/W	*0x0000*	Core CSR performance counter enable configuration register.
CSR_PCMR	0x10305E84	0x00305E84	32	Config	R/W	*0x0000*	Core CSR performance counter mode configuration register.
CSR_HWLP0S	0x10305EC0	0x00305EC0	32	Config	R/W	*0x0000*	Core CSR hardware loop 0 start configuration register.
CSR_HWLP0E	0x10305EC4	0x00305EC4	32	Config	R/W	*0x0000*	Core CSR hardware loop 0 end configuration register.
CSR_HWLP0C	0x10305EC8	0x00305EC8	32	Config	R/W	*0x0000*	Core CSR hardware loop 0 counter configuration register.
CSR_HWLP1S	0x10305ED0	0x00305ED0	32	Config	R/W	*0x0000*	Core CSR hardware loop 1 start configuration register.
CSR_HWLP1E	0x10305ED4	0x00305ED4	32	Config	R/W	*0x0000*	Core CSR hardware loop 1 end configuration register.

Name	Address	Aliased address	Size	Type	Access	Default	Description
CSR_HWLP1C	0x10305ED8	0x00305ED8	32	Config	R/W	*0x0000*	Core CSR hardware loop 1 counter configuration register.
CSR_PRIVLV	0x10307040	0x00307040	32	Config	R	*0x0000*	Cose CSR privilege level status register.
CSR_MHARTID	0x10307C50	0x00307C50	32	Config	R	*0x0000*	Core CSR machine privilege mode hardware thread ID status register.

Table 29. Cluster Core 0 (Debug Unit) registers table

6.2.8.2 Cluster Core 1 (Debug Unit) registers

Name	Address	Aliased address	Size	Type	Access	Default	Description
CTRL	0x10308000	0x00308000	32	Config	R/W	*0x0000*	Debug control configuration register.
HIT	0x10308004	0x00308004	32	Config	R/W	*0x0000*	Debug hit status register.
IE	0x10308008	0x00308008	32	Config	R/W	*0x0000*	Debug exception trap enable configuration register.
CAUSE	0x1030800C	0x0030800C	32	Config	R	*0x0000*	Debug trap cause status register.
GPR0	0x10308400	0x00308400	32	Config	R/W	*0x0000*	Core general purpose register 0 value register.
GPR1	0x10308404	0x00308404	32	Config	R/W	*0x0000*	Core general purpose register 1 value register.
GPR2	0x10308408	0x00308408	32	Config	R/W	*0x0000*	Core general purpose register 2 value register.
GPR3	0x1030840C	0x0030840C	32	Config	R/W	*0x0000*	Core general purpose register 3 value register.
GPR4	0x10308410	0x00308410	32	Config	R/W	*0x0000*	Core general purpose register 4 value register.
GPR5	0x10308414	0x00308414	32	Config	R/W	*0x0000*	Core general purpose register 5 value register.
GPR6	0x10308418	0x00308418	32	Config	R/W	*0x0000*	Core general purpose register 6 value register.
GPR7	0x1030841C	0x0030841C	32	Config	R/W	*0x0000*	Core general purpose register 7 value register.
GPR8	0x10308420	0x00308420	32	Config	R/W	*0x0000*	Core general purpose register 8 value register.
GPR9	0x10308424	0x00308424	32	Config	R/W	*0x0000*	Core general purpose register 9 value register.
GPR10	0x10308428	0x00308428	32	Config	R/W	*0x0000*	Core general purpose register 10 value register.
GPR11	0x1030842C	0x0030842C	32	Config	R/W	*0x0000*	Core general purpose register 11 value register.
GPR12	0x10308430	0x00308430	32	Config	R/W	*0x0000*	Core general purpose register 12 value register.
GPR13	0x10308434	0x00308434	32	Config	R/W	*0x0000*	Core general purpose register 13 value register.
GPR14	0x10308438	0x00308438	32	Config	R/W	*0x0000*	Core general purpose register 14 value register.
GPR15	0x1030843C	0x0030843C	32	Config	R/W	*0x0000*	Core general purpose register 15 value register.
GPR16	0x10308440	0x00308440	32	Config	R/W	*0x0000*	Core general purpose register 16 value register.
GPR17	0x10308444	0x00308444	32	Config	R/W	*0x0000*	Core general purpose register 17 value register.
GPR18	0x10308448	0x00308448	32	Config	R/W	*0x0000*	Core general purpose register 18 value register.
GPR19	0x1030844C	0x0030844C	32	Config	R/W	*0x0000*	Core general purpose register 19 value register.
GPR20	0x10308450	0x00308450	32	Config	R/W	*0x0000*	Core general purpose register 20 value register.
GPR21	0x10308454	0x00308454	32	Config	R/W	*0x0000*	Core general purpose register 21 value register.
GPR22	0x10308458	0x00308458	32	Config	R/W	*0x0000*	Core general purpose register 22 value register.
GPR23	0x1030845C	0x0030845C	32	Config	R/W	*0x0000*	Core general purpose register 23 value register.
GPR24	0x10308460	0x00308460	32	Config	R/W	*0x0000*	Core general purpose register 24 value register.
GPR25	0x10308464	0x00308464	32	Config	R/W	*0x0000*	Core general purpose register 25 value register.
GPR26	0x10308468	0x00308468	32	Config	R/W	*0x0000*	Core general purpose register 26 value register.
GPR27	0x1030846C	0x0030846C	32	Config	R/W	*0x0000*	Core general purpose register 27 value register.
GPR28	0x10308470	0x00308470	32	Config	R/W	*0x0000*	Core general purpose register 28 value register.

Name	Address	Aliased address	Size	Type	Access	Default	Description
GPR29	0x10308474	0x00308474	32	Config	R/W	*0x0000*	Core general purpose register 29 value register.
GPR30	0x10308478	0x00308478	32	Config	R/W	*0x0000*	Core general purpose register 30 value register.
GPR31	0x1030847C	0x0030847C	32	Config	R/W	*0x0000*	Core general purpose register 31 value register.
NPC	0x1030A000	0x0030A000	32	Config	R/W	*0x0000*	Debug next program counter value register.
PPC	0x1030A004	0x0030A004	32	Config	R	*0x0000*	Debug previous program counter value register.
CSR_UHARTID	0x1030C050	0x0030C050	32	Config	R	*0x0000*	Core CSR user privilege mode hardware thread ID status register.
CSR_MSTATUS	0x1030CC00	0x0030CC00	32	Config	R/W	*0x0000*	Core CSR machine status value register.
CSR_MTVEC	0x1030CC14	0x0030CC14	32	Config	R/W	*0x0000*	Core CSR machine vector-trap base address value register.
CSR_MEPC	0x1030CD04	0x0030CD04	32	Config	R/W	*0x0000*	Core CSR machine exception program counter value register.
CSR_MCAUSE	0x1030CD08	0x0030CD08	32	Config	R/W	*0x0000*	Core CSR machine trap cause value register.
CSR_PCCR	0x1030DE00	0x0030DE00	32	Config	R/W	*0x0000*	Core CSR performance counter counter register.
CSR_PCER	0x1030DE80	0x0030DE80	32	Config	R/W	*0x0000*	Core CSR performance counter enable configuration register.
CSR_PCMR	0x1030DE84	0x0030DE84	32	Config	R/W	*0x0000*	Core CSR performance counter mode configuration register.
CSR_HWLP0S	0x1030DEC0	0x0030DEC0	32	Config	R/W	*0x0000*	Core CSR hardware loop 0 start configuration register.
CSR_HWLP0E	0x1030DEC4	0x0030DEC4	32	Config	R/W	*0x0000*	Core CSR hardware loop 0 end configuration register.
CSR_HWLP0C	0x1030DEC8	0x0030DEC8	32	Config	R/W	*0x0000*	Core CSR hardware loop 0 counter configuration register.
CSR_HWLP1S	0x1030DED0	0x0030DED0	32	Config	R/W	*0x0000*	Core CSR hardware loop 1 start configuration register.
CSR_HWLP1E	0x1030DED4	0x0030DED4	32	Config	R/W	*0x0000*	Core CSR hardware loop 1 end configuration register.
CSR_HWLP1C	0x1030DED8	0x0030DED8	32	Config	R/W	*0x0000*	Core CSR hardware loop 1 counter configuration register.
CSR_PRIVLV	0x1030F040	0x0030F040	32	Config	R	*0x0000*	Cose CSR privilege level status register.
CSR_MHARTID	0x1030FC50	0x0030FC50	32	Config	R	*0x0000*	Core CSR machine privilege mode hardware thread ID status register.

Table 30. Cluster Core 1 (Debug Unit) registers table

6.2.8.3 Cluster Core 2 (Debug Unit) registers

Name	Address	Aliased address	Size	Type	Access	Default	Description
CTRL	0x10310000	0x00310000	32	Config	R/W	*0x0000*	Debug control configuration register.
HIT	0x10310004	0x00310004	32	Config	R/W	*0x0000*	Debug hit status register.
IE	0x10310008	0x00310008	32	Config	R/W	*0x0000*	Debug exception trap enable configuration register.
CAUSE	0x1031000C	0x0031000C	32	Config	R	*0x0000*	Debug trap cause status register.
GPR0	0x10310400	0x00310400	32	Config	R/W	*0x0000*	Core general purpose register 0 value register.
GPR1	0x10310404	0x00310404	32	Config	R/W	*0x0000*	Core general purpose register 1 value register.
GPR2	0x10310408	0x00310408	32	Config	R/W	*0x0000*	Core general purpose register 2 value register.

Name	Address	Aliased address	Size	Type	Access	Default	Description
GPR3	0x1031040C	0x0031040C	32	Config	R/W	*0x0000*	Core general purpose register 3 value register.
GPR4	0x10310410	0x00310410	32	Config	R/W	*0x0000*	Core general purpose register 4 value register.
GPR5	0x10310414	0x00310414	32	Config	R/W	*0x0000*	Core general purpose register 5 value register.
GPR6	0x10310418	0x00310418	32	Config	R/W	*0x0000*	Core general purpose register 6 value register.
GPR7	0x1031041C	0x0031041C	32	Config	R/W	*0x0000*	Core general purpose register 7 value register.
GPR8	0x10310420	0x00310420	32	Config	R/W	*0x0000*	Core general purpose register 8 value register.
GPR9	0x10310424	0x00310424	32	Config	R/W	*0x0000*	Core general purpose register 9 value register.
GPR10	0x10310428	0x00310428	32	Config	R/W	*0x0000*	Core general purpose register 10 value register.
GPR11	0x1031042C	0x0031042C	32	Config	R/W	*0x0000*	Core general purpose register 11 value register.
GPR12	0x10310430	0x00310430	32	Config	R/W	*0x0000*	Core general purpose register 12 value register.
GPR13	0x10310434	0x00310434	32	Config	R/W	*0x0000*	Core general purpose register 13 value register.
GPR14	0x10310438	0x00310438	32	Config	R/W	*0x0000*	Core general purpose register 14 value register.
GPR15	0x1031043C	0x0031043C	32	Config	R/W	*0x0000*	Core general purpose register 15 value register.
GPR16	0x10310440	0x00310440	32	Config	R/W	*0x0000*	Core general purpose register 16 value register.
GPR17	0x10310444	0x00310444	32	Config	R/W	*0x0000*	Core general purpose register 17 value register.
GPR18	0x10310448	0x00310448	32	Config	R/W	*0x0000*	Core general purpose register 18 value register.
GPR19	0x1031044C	0x0031044C	32	Config	R/W	*0x0000*	Core general purpose register 19 value register.
GPR20	0x10310450	0x00310450	32	Config	R/W	*0x0000*	Core general purpose register 20 value register.
GPR21	0x10310454	0x00310454	32	Config	R/W	*0x0000*	Core general purpose register 21 value register.
GPR22	0x10310458	0x00310458	32	Config	R/W	*0x0000*	Core general purpose register 22 value register.
GPR23	0x1031045C	0x0031045C	32	Config	R/W	*0x0000*	Core general purpose register 23 value register.
GPR24	0x10310460	0x00310460	32	Config	R/W	*0x0000*	Core general purpose register 24 value register.
GPR25	0x10310464	0x00310464	32	Config	R/W	*0x0000*	Core general purpose register 25 value register.
GPR26	0x10310468	0x00310468	32	Config	R/W	*0x0000*	Core general purpose register 26 value register.
GPR27	0x1031046C	0x0031046C	32	Config	R/W	*0x0000*	Core general purpose register 27 value register.
GPR28	0x10310470	0x00310470	32	Config	R/W	*0x0000*	Core general purpose register 28 value register.
GPR29	0x10310474	0x00310474	32	Config	R/W	*0x0000*	Core general purpose register 29 value register.
GPR30	0x10310478	0x00310478	32	Config	R/W	*0x0000*	Core general purpose register 30 value register.
GPR31	0x1031047C	0x0031047C	32	Config	R/W	*0x0000*	Core general purpose register 31 value register.
NPC	0x10312000	0x00312000	32	Config	R/W	*0x0000*	Debug next program counter value register.
PPC	0x10312004	0x00312004	32	Config	R	*0x0000*	Debug previous program counter value register.
CSR_UHARTID	0x10314050	0x00314050	32	Config	R	*0x0000*	Core CSR user privilege mode hardware thread ID status register.
CSR_MSTATUS	0x10314C00	0x00314C00	32	Config	R/W	*0x0000*	Core CSR machine status value register.
CSR_MTVEC	0x10314C14	0x00314C14	32	Config	R/W	*0x0000*	Core CSR machine vector-trap base address value register.
CSR_MEPC	0x10314D04	0x00314D04	32	Config	R/W	*0x0000*	Core CSR machine exception program counter value register.
CSR_MCAUSE	0x10314D08	0x00314D08	32	Config	R/W	*0x0000*	Core CSR machine trap cause value register.
CSR_PCCR	0x10315E00	0x00315E00	32	Config	R/W	*0x0000*	Core CSR performance counter counter register.
CSR_PCER	0x10315E80	0x00315E80	32	Config	R/W	*0x0000*	Core CSR performance counter enable configuration register.
CSR_PCMR	0x10315E84	0x00315E84	32	Config	R/W	*0x0000*	Core CSR performance counter mode configuration register.

Name	Address	Aliased address	Size	Type	Access	Default	Description
CSR_HWLP0S	0x10315EC0	0x00315EC0	32	Config	R/W	*0x0000*	Core CSR hardware loop 0 start configuration register.
CSR_HWLP0E	0x10315EC4	0x00315EC4	32	Config	R/W	*0x0000*	Core CSR hardware loop 0 end configuration register.
CSR_HWLP0C	0x10315EC8	0x00315EC8	32	Config	R/W	*0x0000*	Core CSR hardware loop 0 counter configuration register.
CSR_HWLP1S	0x10315ED0	0x00315ED0	32	Config	R/W	*0x0000*	Core CSR hardware loop 1 start configuration register.
CSR_HWLP1E	0x10315ED4	0x00315ED4	32	Config	R/W	*0x0000*	Core CSR hardware loop 1 end configuration register.
CSR_HWLP1C	0x10315ED8	0x00315ED8	32	Config	R/W	*0x0000*	Core CSR hardware loop 1 counter configuration register.
CSR_PRIVLV	0x10317040	0x00317040	32	Config	R	*0x0000*	Cose CSR privilege level status register.
CSR_MHARTID	0x10317C50	0x00317C50	32	Config	R	*0x0000*	Core CSR machine privilege mode hardware thread ID status register.

Table 31. Cluster Core 2 (Debug Unit) registers table

6.2.8.4 Cluster Core 3 (Debug Unit) registers

Name	Address	Aliased address	Size	Type	Access	Default	Description
CTRL	0x10318000	0x00318000	32	Config	R/W	*0x0000*	Debug control configuration register.
HIT	0x10318004	0x00318004	32	Config	R/W	*0x0000*	Debug hit status register.
IE	0x10318008	0x00318008	32	Config	R/W	*0x0000*	Debug exception trap enable configuration register.
CAUSE	0x1031800C	0x0031800C	32	Config	R	*0x0000*	Debug trap cause status register.
GPR0	0x10318400	0x00318400	32	Config	R/W	*0x0000*	Core general purpose register 0 value register.
GPR1	0x10318404	0x00318404	32	Config	R/W	*0x0000*	Core general purpose register 1 value register.
GPR2	0x10318408	0x00318408	32	Config	R/W	*0x0000*	Core general purpose register 2 value register.
GPR3	0x1031840C	0x0031840C	32	Config	R/W	*0x0000*	Core general purpose register 3 value register.
GPR4	0x10318410	0x00318410	32	Config	R/W	*0x0000*	Core general purpose register 4 value register.
GPR5	0x10318414	0x00318414	32	Config	R/W	*0x0000*	Core general purpose register 5 value register.
GPR6	0x10318418	0x00318418	32	Config	R/W	*0x0000*	Core general purpose register 6 value register.
GPR7	0x1031841C	0x0031841C	32	Config	R/W	*0x0000*	Core general purpose register 7 value register.
GPR8	0x10318420	0x00318420	32	Config	R/W	*0x0000*	Core general purpose register 8 value register.
GPR9	0x10318424	0x00318424	32	Config	R/W	*0x0000*	Core general purpose register 9 value register.
GPR10	0x10318428	0x00318428	32	Config	R/W	*0x0000*	Core general purpose register 10 value register.
GPR11	0x1031842C	0x0031842C	32	Config	R/W	*0x0000*	Core general purpose register 11 value register.
GPR12	0x10318430	0x00318430	32	Config	R/W	*0x0000*	Core general purpose register 12 value register.
GPR13	0x10318434	0x00318434	32	Config	R/W	*0x0000*	Core general purpose register 13 value register.
GPR14	0x10318438	0x00318438	32	Config	R/W	*0x0000*	Core general purpose register 14 value register.
GPR15	0x1031843C	0x0031843C	32	Config	R/W	*0x0000*	Core general purpose register 15 value register.
GPR16	0x10318440	0x00318440	32	Config	R/W	*0x0000*	Core general purpose register 16 value register.
GPR17	0x10318444	0x00318444	32	Config	R/W	*0x0000*	Core general purpose register 17 value register.
GPR18	0x10318448	0x00318448	32	Config	R/W	*0x0000*	Core general purpose register 18 value register.
GPR19	0x1031844C	0x0031844C	32	Config	R/W	*0x0000*	Core general purpose register 19 value register.

Name	Address	Aliased address	Size	Type	Access	Default	Description
GPR20	0x10318450	0x00318450	32	Config	R/W	*0x0000*	Core general purpose register 20 value register.
GPR21	0x10318454	0x00318454	32	Config	R/W	*0x0000*	Core general purpose register 21 value register.
GPR22	0x10318458	0x00318458	32	Config	R/W	*0x0000*	Core general purpose register 22 value register.
GPR23	0x1031845C	0x0031845C	32	Config	R/W	*0x0000*	Core general purpose register 23 value register.
GPR24	0x10318460	0x00318460	32	Config	R/W	*0x0000*	Core general purpose register 24 value register.
GPR25	0x10318464	0x00318464	32	Config	R/W	*0x0000*	Core general purpose register 25 value register.
GPR26	0x10318468	0x00318468	32	Config	R/W	*0x0000*	Core general purpose register 26 value register.
GPR27	0x1031846C	0x0031846C	32	Config	R/W	*0x0000*	Core general purpose register 27 value register.
GPR28	0x10318470	0x00318470	32	Config	R/W	*0x0000*	Core general purpose register 28 value register.
GPR29	0x10318474	0x00318474	32	Config	R/W	*0x0000*	Core general purpose register 29 value register.
GPR30	0x10318478	0x00318478	32	Config	R/W	*0x0000*	Core general purpose register 30 value register.
GPR31	0x1031847C	0x0031847C	32	Config	R/W	*0x0000*	Core general purpose register 31 value register.
NPC	0x1031A000	0x0031A000	32	Config	R/W	*0x0000*	Debug next program counter value register.
PPC	0x1031A004	0x0031A004	32	Config	R	*0x0000*	Debug previous program counter value register.
CSR_UHARTID	0x1031C050	0x0031C050	32	Config	R	*0x0000*	Core CSR user privilege mode hardware thread ID status register.
CSR_MSTATUS	0x1031CC00	0x0031CC00	32	Config	R/W	*0x0000*	Core CSR machine status value register.
CSR_MTVEC	0x1031CC14	0x0031CC14	32	Config	R/W	*0x0000*	Core CSR machine vector-trap base address value register.
CSR_MEPC	0x1031CD04	0x0031CD04	32	Config	R/W	*0x0000*	Core CSR machine exception program counter value register.
CSR_MCAUSE	0x1031CD08	0x0031CD08	32	Config	R/W	*0x0000*	Core CSR machine trap cause value register.
CSR_PCCR	0x1031DE00	0x0031DE00	32	Config	R/W	*0x0000*	Core CSR performance counter counter register.
CSR_PCER	0x1031DE80	0x0031DE80	32	Config	R/W	*0x0000*	Core CSR performance counter enable configuration register.
CSR_PCMR	0x1031DE84	0x0031DE84	32	Config	R/W	*0x0000*	Core CSR performance counter mode configuration register.
CSR_HWLP0S	0x1031DEC0	0x0031DEC0	32	Config	R/W	*0x0000*	Core CSR hardware loop 0 start configuration register.
CSR_HWLP0E	0x1031DEC4	0x0031DEC4	32	Config	R/W	*0x0000*	Core CSR hardware loop 0 end configuration register.
CSR_HWLP0C	0x1031DEC8	0x0031DEC8	32	Config	R/W	*0x0000*	Core CSR hardware loop 0 counter configuration register.
CSR_HWLP1S	0x1031DED0	0x0031DED0	32	Config	R/W	*0x0000*	Core CSR hardware loop 1 start configuration register.
CSR_HWLP1E	0x1031DED4	0x0031DED4	32	Config	R/W	*0x0000*	Core CSR hardware loop 1 end configuration register.
CSR_HWLP1C	0x1031DED8	0x0031DED8	32	Config	R/W	*0x0000*	Core CSR hardware loop 1 counter configuration register.
CSR_PRIVLV	0x1031F040	0x0031F040	32	Config	R	*0x0000*	Cose CSR privilege level status register.
CSR_MHARTID	0x1031FC50	0x0031FC50	32	Config	R	*0x0000*	Core CSR machine privilege mode hardware thread ID status register.

Table 32. Cluster Core 3 (Debug Unit) registers table

6.2.8.5 Cluster Core 4 (Debug Unit) registers

Name	Address	Aliased address	Size	Type	Access	Default	Description
CTRL	0x10320000	0x00320000	32	Config	R/W	*0x0000*	Debug control configuration register.
HIT	0x10320004	0x00320004	32	Config	R/W	*0x0000*	Debug hit status register.
IE	0x10320008	0x00320008	32	Config	R/W	*0x0000*	Debug exception trap enable configuration register.
CAUSE	0x1032000C	0x0032000C	32	Config	R	*0x0000*	Debug trap cause status register.
GPR0	0x10320400	0x00320400	32	Config	R/W	*0x0000*	Core general purpose register 0 value register.
GPR1	0x10320404	0x00320404	32	Config	R/W	*0x0000*	Core general purpose register 1 value register.
GPR2	0x10320408	0x00320408	32	Config	R/W	*0x0000*	Core general purpose register 2 value register.
GPR3	0x1032040C	0x0032040C	32	Config	R/W	*0x0000*	Core general purpose register 3 value register.
GPR4	0x10320410	0x00320410	32	Config	R/W	*0x0000*	Core general purpose register 4 value register.
GPR5	0x10320414	0x00320414	32	Config	R/W	*0x0000*	Core general purpose register 5 value register.
GPR6	0x10320418	0x00320418	32	Config	R/W	*0x0000*	Core general purpose register 6 value register.
GPR7	0x1032041C	0x0032041C	32	Config	R/W	*0x0000*	Core general purpose register 7 value register.
GPR8	0x10320420	0x00320420	32	Config	R/W	*0x0000*	Core general purpose register 8 value register.
GPR9	0x10320424	0x00320424	32	Config	R/W	*0x0000*	Core general purpose register 9 value register.
GPR10	0x10320428	0x00320428	32	Config	R/W	*0x0000*	Core general purpose register 10 value register.
GPR11	0x1032042C	0x0032042C	32	Config	R/W	*0x0000*	Core general purpose register 11 value register.
GPR12	0x10320430	0x00320430	32	Config	R/W	*0x0000*	Core general purpose register 12 value register.
GPR13	0x10320434	0x00320434	32	Config	R/W	*0x0000*	Core general purpose register 13 value register.
GPR14	0x10320438	0x00320438	32	Config	R/W	*0x0000*	Core general purpose register 14 value register.
GPR15	0x1032043C	0x0032043C	32	Config	R/W	*0x0000*	Core general purpose register 15 value register.
GPR16	0x10320440	0x00320440	32	Config	R/W	*0x0000*	Core general purpose register 16 value register.
GPR17	0x10320444	0x00320444	32	Config	R/W	*0x0000*	Core general purpose register 17 value register.
GPR18	0x10320448	0x00320448	32	Config	R/W	*0x0000*	Core general purpose register 18 value register.
GPR19	0x1032044C	0x0032044C	32	Config	R/W	*0x0000*	Core general purpose register 19 value register.
GPR20	0x10320450	0x00320450	32	Config	R/W	*0x0000*	Core general purpose register 20 value register.
GPR21	0x10320454	0x00320454	32	Config	R/W	*0x0000*	Core general purpose register 21 value register.
GPR22	0x10320458	0x00320458	32	Config	R/W	*0x0000*	Core general purpose register 22 value register.
GPR23	0x1032045C	0x0032045C	32	Config	R/W	*0x0000*	Core general purpose register 23 value register.
GPR24	0x10320460	0x00320460	32	Config	R/W	*0x0000*	Core general purpose register 24 value register.
GPR25	0x10320464	0x00320464	32	Config	R/W	*0x0000*	Core general purpose register 25 value register.
GPR26	0x10320468	0x00320468	32	Config	R/W	*0x0000*	Core general purpose register 26 value register.
GPR27	0x1032046C	0x0032046C	32	Config	R/W	*0x0000*	Core general purpose register 27 value register.
GPR28	0x10320470	0x00320470	32	Config	R/W	*0x0000*	Core general purpose register 28 value register.
GPR29	0x10320474	0x00320474	32	Config	R/W	*0x0000*	Core general purpose register 29 value register.
GPR30	0x10320478	0x00320478	32	Config	R/W	*0x0000*	Core general purpose register 30 value register.
GPR31	0x1032047C	0x0032047C	32	Config	R/W	*0x0000*	Core general purpose register 31 value register.
NPC	0x10322000	0x00322000	32	Config	R/W	*0x0000*	Debug next program counter value register.
PPC	0x10322004	0x00322004	32	Config	R	*0x0000*	Debug previous program counter value register.
CSR_UHARTID	0x10324050	0x00324050	32	Config	R	*0x0000*	Core CSR user privilege mode hardware thread ID status register.
CSR_MSTATUS	0x10324C00	0x00324C00	32	Config	R/W	*0x0000*	Core CSR machine status value register.
CSR_MTVEC	0x10324C14	0x00324C14	32	Config	R/W	*0x0000*	Core CSR machine vector-trap base address value register.

Name	Address	Aliased address	Size	Type	Access	Default	Description
CSR_MEPC	0x10324D04	0x00324D04	32	Config	R/W	*0x0000*	Core CSR machine exception program counter value register.
CSR_MCAUSE	0x10324D08	0x00324D08	32	Config	R/W	*0x0000*	Core CSR machine trap cause value register.
CSR_PCCR	0x10325E00	0x00325E00	32	Config	R/W	*0x0000*	Core CSR performance counter counter register.
CSR_PCER	0x10325E80	0x00325E80	32	Config	R/W	*0x0000*	Core CSR performance counter enable configuration register.
CSR_PCMR	0x10325E84	0x00325E84	32	Config	R/W	*0x0000*	Core CSR performance counter mode configuration register.
CSR_HWLP0S	0x10325EC0	0x00325EC0	32	Config	R/W	*0x0000*	Core CSR hardware loop 0 start configuration register.
CSR_HWLP0E	0x10325EC4	0x00325EC4	32	Config	R/W	*0x0000*	Core CSR hardware loop 0 end configuration register.
CSR_HWLP0C	0x10325EC8	0x00325EC8	32	Config	R/W	*0x0000*	Core CSR hardware loop 0 counter configuration register.
CSR_HWLP1S	0x10325ED0	0x00325ED0	32	Config	R/W	*0x0000*	Core CSR hardware loop 1 start configuration register.
CSR_HWLP1E	0x10325ED4	0x00325ED4	32	Config	R/W	*0x0000*	Core CSR hardware loop 1 end configuration register.
CSR_HWLP1C	0x10325ED8	0x00325ED8	32	Config	R/W	*0x0000*	Core CSR hardware loop 1 counter configuration register.
CSR_PRIVLV	0x10327040	0x00327040	32	Config	R	*0x0000*	Cose CSR privilege level status register.
CSR_MHARTID	0x10327C50	0x00327C50	32	Config	R	*0x0000*	Core CSR machine privilege mode hardware thread ID status register.

Table 33. Cluster Core 4 (Debug Unit) registers table

6.2.8.6 Cluster Core 5 (Debug Unit) registers

Name	Address	Aliased address	Size	Type	Access	Default	Description
CTRL	0x10328000	0x00328000	32	Config	R/W	*0x0000*	Debug control configuration register.
HIT	0x10328004	0x00328004	32	Config	R/W	*0x0000*	Debug hit status register.
IE	0x10328008	0x00328008	32	Config	R/W	*0x0000*	Debug exception trap enable configuration register.
CAUSE	0x1032800C	0x0032800C	32	Config	R	*0x0000*	Debug trap cause status register.
GPR0	0x10328400	0x00328400	32	Config	R/W	*0x0000*	Core general purpose register 0 value register.
GPR1	0x10328404	0x00328404	32	Config	R/W	*0x0000*	Core general purpose register 1 value register.
GPR2	0x10328408	0x00328408	32	Config	R/W	*0x0000*	Core general purpose register 2 value register.
GPR3	0x1032840C	0x0032840C	32	Config	R/W	*0x0000*	Core general purpose register 3 value register.
GPR4	0x10328410	0x00328410	32	Config	R/W	*0x0000*	Core general purpose register 4 value register.
GPR5	0x10328414	0x00328414	32	Config	R/W	*0x0000*	Core general purpose register 5 value register.
GPR6	0x10328418	0x00328418	32	Config	R/W	*0x0000*	Core general purpose register 6 value register.
GPR7	0x1032841C	0x0032841C	32	Config	R/W	*0x0000*	Core general purpose register 7 value register.
GPR8	0x10328420	0x00328420	32	Config	R/W	*0x0000*	Core general purpose register 8 value register.
GPR9	0x10328424	0x00328424	32	Config	R/W	*0x0000*	Core general purpose register 9 value register.
GPR10	0x10328428	0x00328428	32	Config	R/W	*0x0000*	Core general purpose register 10 value register.
GPR11	0x1032842C	0x0032842C	32	Config	R/W	*0x0000*	Core general purpose register 11 value register.
GPR12	0x10328430	0x00328430	32	Config	R/W	*0x0000*	Core general purpose register 12 value register.

Name	Address	Aliased address	Size	Type	Access	Default	Description
GPR13	0x10328434	0x00328434	32	Config	R/W	*0x0000*	Core general purpose register 13 value register.
GPR14	0x10328438	0x00328438	32	Config	R/W	*0x0000*	Core general purpose register 14 value register.
GPR15	0x1032843C	0x0032843C	32	Config	R/W	*0x0000*	Core general purpose register 15 value register.
GPR16	0x10328440	0x00328440	32	Config	R/W	*0x0000*	Core general purpose register 16 value register.
GPR17	0x10328444	0x00328444	32	Config	R/W	*0x0000*	Core general purpose register 17 value register.
GPR18	0x10328448	0x00328448	32	Config	R/W	*0x0000*	Core general purpose register 18 value register.
GPR19	0x1032844C	0x0032844C	32	Config	R/W	*0x0000*	Core general purpose register 19 value register.
GPR20	0x10328450	0x00328450	32	Config	R/W	*0x0000*	Core general purpose register 20 value register.
GPR21	0x10328454	0x00328454	32	Config	R/W	*0x0000*	Core general purpose register 21 value register.
GPR22	0x10328458	0x00328458	32	Config	R/W	*0x0000*	Core general purpose register 22 value register.
GPR23	0x1032845C	0x0032845C	32	Config	R/W	*0x0000*	Core general purpose register 23 value register.
GPR24	0x10328460	0x00328460	32	Config	R/W	*0x0000*	Core general purpose register 24 value register.
GPR25	0x10328464	0x00328464	32	Config	R/W	*0x0000*	Core general purpose register 25 value register.
GPR26	0x10328468	0x00328468	32	Config	R/W	*0x0000*	Core general purpose register 26 value register.
GPR27	0x1032846C	0x0032846C	32	Config	R/W	*0x0000*	Core general purpose register 27 value register.
GPR28	0x10328470	0x00328470	32	Config	R/W	*0x0000*	Core general purpose register 28 value register.
GPR29	0x10328474	0x00328474	32	Config	R/W	*0x0000*	Core general purpose register 29 value register.
GPR30	0x10328478	0x00328478	32	Config	R/W	*0x0000*	Core general purpose register 30 value register.
GPR31	0x1032847C	0x0032847C	32	Config	R/W	*0x0000*	Core general purpose register 31 value register.
NPC	0x1032A000	0x0032A000	32	Config	R/W	*0x0000*	Debug next program counter value register.
PPC	0x1032A004	0x0032A004	32	Config	R	*0x0000*	Debug previous program counter value register.
CSR_UHARTID	0x1032C050	0x0032C050	32	Config	R	*0x0000*	Core CSR user privilege mode hardware thread ID status register.
CSR_MSTATUS	0x1032CC00	0x0032CC00	32	Config	R/W	*0x0000*	Core CSR machine status value register.
CSR_MTVEC	0x1032CC14	0x0032CC14	32	Config	R/W	*0x0000*	Core CSR machine vector-trap base address value register.
CSR_MEPC	0x1032CD04	0x0032CD04	32	Config	R/W	*0x0000*	Core CSR machine exception program counter value register.
CSR_MCAUSE	0x1032CD08	0x0032CD08	32	Config	R/W	*0x0000*	Core CSR machine trap cause value register.
CSR_PCCR	0x1032DE00	0x0032DE00	32	Config	R/W	*0x0000*	Core CSR performance counter counter register.
CSR_PCER	0x1032DE80	0x0032DE80	32	Config	R/W	*0x0000*	Core CSR performance counter enable configuration register.
CSR_PCMR	0x1032DE84	0x0032DE84	32	Config	R/W	*0x0000*	Core CSR performance counter mode configuration register.
CSR_HWLP0S	0x1032DEC0	0x0032DEC0	32	Config	R/W	*0x0000*	Core CSR hardware loop 0 start configuration register.
CSR_HWLP0E	0x1032DEC4	0x0032DEC4	32	Config	R/W	*0x0000*	Core CSR hardware loop 0 end configuration register.
CSR_HWLP0C	0x1032DEC8	0x0032DEC8	32	Config	R/W	*0x0000*	Core CSR hardware loop 0 counter configuration register.
CSR_HWLP1S	0x1032DED0	0x0032DED0	32	Config	R/W	*0x0000*	Core CSR hardware loop 1 start configuration register.
CSR_HWLP1E	0x1032DED4	0x0032DED4	32	Config	R/W	*0x0000*	Core CSR hardware loop 1 end configuration register.
CSR_HWLP1C	0x1032DED8	0x0032DED8	32	Config	R/W	*0x0000*	Core CSR hardware loop 1 counter configuration register.

Name	Address	Aliased address	Size	Type	Access	Default	Description
CSR_PRIVLV	0x1032F040	0x0032F040	32	Config	R	*0x0000*	Cose CSR privilege level status register.
CSR_MHARTID	0x1032FC50	0x0032FC50	32	Config	R	*0x0000*	Core CSR machine privilege mode hardware thread ID status register.

Table 34. Cluster Core 5 (Debug Unit) registers table

6.2.8.7 Cluster Core 6 (Debug Unit) registers

Name	Address	Aliased address	Size	Type	Access	Default	Description
CTRL	0x10330000	0x00330000	32	Config	R/W	*0x0000*	Debug control configuration register.
HIT	0x10330004	0x00330004	32	Config	R/W	*0x0000*	Debug hit status register.
IE	0x10330008	0x00330008	32	Config	R/W	*0x0000*	Debug exception trap enable configuration register.
CAUSE	0x1033000C	0x0033000C	32	Config	R	*0x0000*	Debug trap cause status register.
GPR0	0x10330400	0x00330400	32	Config	R/W	*0x0000*	Core general purpose register 0 value register.
GPR1	0x10330404	0x00330404	32	Config	R/W	*0x0000*	Core general purpose register 1 value register.
GPR2	0x10330408	0x00330408	32	Config	R/W	*0x0000*	Core general purpose register 2 value register.
GPR3	0x1033040C	0x0033040C	32	Config	R/W	*0x0000*	Core general purpose register 3 value register.
GPR4	0x10330410	0x00330410	32	Config	R/W	*0x0000*	Core general purpose register 4 value register.
GPR5	0x10330414	0x00330414	32	Config	R/W	*0x0000*	Core general purpose register 5 value register.
GPR6	0x10330418	0x00330418	32	Config	R/W	*0x0000*	Core general purpose register 6 value register.
GPR7	0x1033041C	0x0033041C	32	Config	R/W	*0x0000*	Core general purpose register 7 value register.
GPR8	0x10330420	0x00330420	32	Config	R/W	*0x0000*	Core general purpose register 8 value register.
GPR9	0x10330424	0x00330424	32	Config	R/W	*0x0000*	Core general purpose register 9 value register.
GPR10	0x10330428	0x00330428	32	Config	R/W	*0x0000*	Core general purpose register 10 value register.
GPR11	0x1033042C	0x0033042C	32	Config	R/W	*0x0000*	Core general purpose register 11 value register.
GPR12	0x10330430	0x00330430	32	Config	R/W	*0x0000*	Core general purpose register 12 value register.
GPR13	0x10330434	0x00330434	32	Config	R/W	*0x0000*	Core general purpose register 13 value register.
GPR14	0x10330438	0x00330438	32	Config	R/W	*0x0000*	Core general purpose register 14 value register.
GPR15	0x1033043C	0x0033043C	32	Config	R/W	*0x0000*	Core general purpose register 15 value register.
GPR16	0x10330440	0x00330440	32	Config	R/W	*0x0000*	Core general purpose register 16 value register.
GPR17	0x10330444	0x00330444	32	Config	R/W	*0x0000*	Core general purpose register 17 value register.
GPR18	0x10330448	0x00330448	32	Config	R/W	*0x0000*	Core general purpose register 18 value register.
GPR19	0x1033044C	0x0033044C	32	Config	R/W	*0x0000*	Core general purpose register 19 value register.
GPR20	0x10330450	0x00330450	32	Config	R/W	*0x0000*	Core general purpose register 20 value register.
GPR21	0x10330454	0x00330454	32	Config	R/W	*0x0000*	Core general purpose register 21 value register.
GPR22	0x10330458	0x00330458	32	Config	R/W	*0x0000*	Core general purpose register 22 value register.
GPR23	0x1033045C	0x0033045C	32	Config	R/W	*0x0000*	Core general purpose register 23 value register.
GPR24	0x10330460	0x00330460	32	Config	R/W	*0x0000*	Core general purpose register 24 value register.
GPR25	0x10330464	0x00330464	32	Config	R/W	*0x0000*	Core general purpose register 25 value register.
GPR26	0x10330468	0x00330468	32	Config	R/W	*0x0000*	Core general purpose register 26 value register.
GPR27	0x1033046C	0x0033046C	32	Config	R/W	*0x0000*	Core general purpose register 27 value register.
GPR28	0x10330470	0x00330470	32	Config	R/W	*0x0000*	Core general purpose register 28 value register.
GPR29	0x10330474	0x00330474	32	Config	R/W	*0x0000*	Core general purpose register 29 value register.

Name	Address	Aliased address	Size	Type	Access	Default	Description
GPR30	0x10330478	0x00330478	32	Config	R/W	*0x0000*	Core general purpose register 30 value register.
GPR31	0x1033047C	0x0033047C	32	Config	R/W	*0x0000*	Core general purpose register 31 value register.
NPC	0x10332000	0x00332000	32	Config	R/W	*0x0000*	Debug next program counter value register.
PPC	0x10332004	0x00332004	32	Config	R	*0x0000*	Debug previous program counter value register.
CSR_UHARTID	0x10334050	0x00334050	32	Config	R	*0x0000*	Core CSR user privilege mode hardware thread ID status register.
CSR_MSTATUS	0x10334C00	0x00334C00	32	Config	R/W	*0x0000*	Core CSR machine status value register.
CSR_MTVEC	0x10334C14	0x00334C14	32	Config	R/W	*0x0000*	Core CSR machine vector-trap base address value register.
CSR_MEPC	0x10334D04	0x00334D04	32	Config	R/W	*0x0000*	Core CSR machine exception program counter value register.
CSR_MCAUSE	0x10334D08	0x00334D08	32	Config	R/W	*0x0000*	Core CSR machine trap cause value register.
CSR_PCCR	0x10335E00	0x00335E00	32	Config	R/W	*0x0000*	Core CSR performance counter counter register.
CSR_PCER	0x10335E80	0x00335E80	32	Config	R/W	*0x0000*	Core CSR performance counter enable configuration register.
CSR_PCMR	0x10335E84	0x00335E84	32	Config	R/W	*0x0000*	Core CSR performance counter mode configuration register.
CSR_HWLP0S	0x10335EC0	0x00335EC0	32	Config	R/W	*0x0000*	Core CSR hardware loop 0 start configuration register.
CSR_HWLP0E	0x10335EC4	0x00335EC4	32	Config	R/W	*0x0000*	Core CSR hardware loop 0 end configuration register.
CSR_HWLP0C	0x10335EC8	0x00335EC8	32	Config	R/W	*0x0000*	Core CSR hardware loop 0 counter configuration register.
CSR_HWLP1S	0x10335ED0	0x00335ED0	32	Config	R/W	*0x0000*	Core CSR hardware loop 1 start configuration register.
CSR_HWLP1E	0x10335ED4	0x00335ED4	32	Config	R/W	*0x0000*	Core CSR hardware loop 1 end configuration register.
CSR_HWLP1C	0x10335ED8	0x00335ED8	32	Config	R/W	*0x0000*	Core CSR hardware loop 1 counter configuration register.
CSR_PRIVLV	0x10337040	0x00337040	32	Config	R	*0x0000*	Cose CSR privilege level status register.
CSR_MHARTID	0x10337C50	0x00337C50	32	Config	R	*0x0000*	Core CSR machine privilege mode hardware thread ID status register.

Table 35. Cluster Core 6 (Debug Unit) registers table

6.2.8.8 Cluster Core 7 (Debug Unit) registers

Name	Address	Aliased address	Size	Type	Access	Default	Description
CTRL	0x10338000	0x00338000	32	Config	R/W	*0x0000*	Debug control configuration register.
HIT	0x10338004	0x00338004	32	Config	R/W	*0x0000*	Debug hit status register.
IE	0x10338008	0x00338008	32	Config	R/W	*0x0000*	Debug exception trap enable configuration register.
CAUSE	0x1033800C	0x0033800C	32	Config	R	*0x0000*	Debug trap cause status register.
GPR0	0x10338400	0x00338400	32	Config	R/W	*0x0000*	Core general purpose register 0 value register.
GPR1	0x10338404	0x00338404	32	Config	R/W	*0x0000*	Core general purpose register 1 value register.
GPR2	0x10338408	0x00338408	32	Config	R/W	*0x0000*	Core general purpose register 2 value register.
GPR3	0x1033840C	0x0033840C	32	Config	R/W	*0x0000*	Core general purpose register 3 value register.

Name	Address	Aliased address	Size	Type	Access	Default	Description
GPR4	0x10338410	0x00338410	32	Config	R/W	*0x0000*	Core general purpose register 4 value register.
GPR5	0x10338414	0x00338414	32	Config	R/W	*0x0000*	Core general purpose register 5 value register.
GPR6	0x10338418	0x00338418	32	Config	R/W	*0x0000*	Core general purpose register 6 value register.
GPR7	0x1033841C	0x0033841C	32	Config	R/W	*0x0000*	Core general purpose register 7 value register.
GPR8	0x10338420	0x00338420	32	Config	R/W	*0x0000*	Core general purpose register 8 value register.
GPR9	0x10338424	0x00338424	32	Config	R/W	*0x0000*	Core general purpose register 9 value register.
GPR10	0x10338428	0x00338428	32	Config	R/W	*0x0000*	Core general purpose register 10 value register.
GPR11	0x1033842C	0x0033842C	32	Config	R/W	*0x0000*	Core general purpose register 11 value register.
GPR12	0x10338430	0x00338430	32	Config	R/W	*0x0000*	Core general purpose register 12 value register.
GPR13	0x10338434	0x00338434	32	Config	R/W	*0x0000*	Core general purpose register 13 value register.
GPR14	0x10338438	0x00338438	32	Config	R/W	*0x0000*	Core general purpose register 14 value register.
GPR15	0x1033843C	0x0033843C	32	Config	R/W	*0x0000*	Core general purpose register 15 value register.
GPR16	0x10338440	0x00338440	32	Config	R/W	*0x0000*	Core general purpose register 16 value register.
GPR17	0x10338444	0x00338444	32	Config	R/W	*0x0000*	Core general purpose register 17 value register.
GPR18	0x10338448	0x00338448	32	Config	R/W	*0x0000*	Core general purpose register 18 value register.
GPR19	0x1033844C	0x0033844C	32	Config	R/W	*0x0000*	Core general purpose register 19 value register.
GPR20	0x10338450	0x00338450	32	Config	R/W	*0x0000*	Core general purpose register 20 value register.
GPR21	0x10338454	0x00338454	32	Config	R/W	*0x0000*	Core general purpose register 21 value register.
GPR22	0x10338458	0x00338458	32	Config	R/W	*0x0000*	Core general purpose register 22 value register.
GPR23	0x1033845C	0x0033845C	32	Config	R/W	*0x0000*	Core general purpose register 23 value register.
GPR24	0x10338460	0x00338460	32	Config	R/W	*0x0000*	Core general purpose register 24 value register.
GPR25	0x10338464	0x00338464	32	Config	R/W	*0x0000*	Core general purpose register 25 value register.
GPR26	0x10338468	0x00338468	32	Config	R/W	*0x0000*	Core general purpose register 26 value register.
GPR27	0x1033846C	0x0033846C	32	Config	R/W	*0x0000*	Core general purpose register 27 value register.
GPR28	0x10338470	0x00338470	32	Config	R/W	*0x0000*	Core general purpose register 28 value register.
GPR29	0x10338474	0x00338474	32	Config	R/W	*0x0000*	Core general purpose register 29 value register.
GPR30	0x10338478	0x00338478	32	Config	R/W	*0x0000*	Core general purpose register 30 value register.
GPR31	0x1033847C	0x0033847C	32	Config	R/W	*0x0000*	Core general purpose register 31 value register.
NPC	0x1033A000	0x0033A000	32	Config	R/W	*0x0000*	Debug next program counter value register.
PPC	0x1033A004	0x0033A004	32	Config	R	*0x0000*	Debug previous program counter value register.
CSR_UHARTID	0x1033C050	0x0033C050	32	Config	R	*0x0000*	Core CSR user privilege mode hardware thread ID status register.
CSR_MSTATUS	0x1033CC00	0x0033CC00	32	Config	R/W	*0x0000*	Core CSR machine status value register.
CSR_MTVEC	0x1033CC14	0x0033CC14	32	Config	R/W	*0x0000*	Core CSR machine vector-trap base address value register.
CSR_MEPC	0x1033CD04	0x0033CD04	32	Config	R/W	*0x0000*	Core CSR machine exception program counter value register.
CSR_MCAUSE	0x1033CD08	0x0033CD08	32	Config	R/W	*0x0000*	Core CSR machine trap cause value register.
CSR_PCCR	0x1033DE00	0x0033DE00	32	Config	R/W	*0x0000*	Core CSR performance counter counter register.
CSR_PCER	0x1033DE80	0x0033DE80	32	Config	R/W	*0x0000*	Core CSR performance counter enable configuration register.
CSR_PCMR	0x1033DE84	0x0033DE84	32	Config	R/W	*0x0000*	Core CSR performance counter mode configuration register.

Name	Address	Aliased address	Size	Type	Access	Default	Description
CSR_HWLP0S	0x1033DEC0	0x0033DEC0	32	Config	R/W	*0x0000*	Core CSR hardware loop 0 start configuration register.
CSR_HWLP0E	0x1033DEC4	0x0033DEC4	32	Config	R/W	*0x0000*	Core CSR hardware loop 0 end configuration register.
CSR_HWLP0C	0x1033DEC8	0x0033DEC8	32	Config	R/W	*0x0000*	Core CSR hardware loop 0 counter configuration register.
CSR_HWLP1S	0x1033DED0	0x0033DED0	32	Config	R/W	*0x0000*	Core CSR hardware loop 1 start configuration register.
CSR_HWLP1E	0x1033DED4	0x0033DED4	32	Config	R/W	*0x0000*	Core CSR hardware loop 1 end configuration register.
CSR_HWLP1C	0x1033DED8	0x0033DED8	32	Config	R/W	*0x0000*	Core CSR hardware loop 1 counter configuration register.
CSR_PRIVLV	0x1033F040	0x0033F040	32	Config	R	*0x0000*	Cose CSR privilege level status register.
CSR_MHARTID	0x1033FC50	0x0033FC50	32	Config	R	*0x0000*	Core CSR machine privilege mode hardware thread ID status register.

Table 36. Cluster Core 7 (Debug Unit) registers table

6.2.8.9 Cluster RISCY core registers details

6.2.8.9.1 Debug control configuration register. (CTRL)

Reset value: 0x0000

Always accessible, even when the RI5CY core is running.

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

Bit 16 - **HALT** *(W)*

Debug mode configuration bitfield:

- *0b0*: exits debug mode
- *0b1*: enters debug mode breaking code execution

Bit 16 - **HALT_STATUS** *(R)*

Debug mode status bitfield:

- *0b0*: running mode
- *0b1*: debug mode

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

Single step mode configuration bitfield:

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

6.2.8.9.2 Debug hit status register. (HIT)

Reset value: 0x0000

Always accessible, even when the RI5CY core is running.

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

Bit 16 - **SLEEP** *(R)*

Sleep mode status bitfield:

- *0b0*: running - core is in running state
- *0b1*: sleeping – core is in sleeping state and waits for en event to wake up

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

Single step hit status bitfield:

- *0b0*: disabled - single step mode disabled
- *0b1*: enabled – single step mode enabled

Sticky bit that must be cleared by external debugger.

6.2.8.9.3 Debug exception trap enable configuration register. (IE)

Reset value: 0x0000

Always accessible, even when the RI5CY core is running.

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				ECALL	Reserved			ELSU_DUP	Reserved	ELSU	Reserved	EBRK	EILL	Reserved	

Bit 11 - **ECALL** *(R/W)*

Environment call trap configuration bitfield:

- *0b0*: normal - normal exception behavior mode
- *0b1*: cause trap - exception causes trap and core switch into debug mode

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

Load/store access fault trap configuration bitfield:

- *0b0*: normal - normal exception behavior mode
- *0b1*: cause trap – exception causes trap and core switch into debug mode

This bitfield is duplicates the ELSU bitfield.

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

Load/store access fault trap configuration bitfield:

- *0b0*: normal - normal exception behavior mode
- *0b1*: cause trap – exception causes trap and core switch into debug mode

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

Environment break trap configuration bitfield:

- *0b0*: normal - normal exception behavior mode
- *0b1*: cause trap – exception causes trap and core switch into debug mode

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

Illegal instruction trap configuration bitfield:

- *0b0*: normal - normal exception behavior mode
- *0b1*: cause trap – exception causes trap and core switch into debug mode

6.2.8.9.4 Debug trap cause status register. (CAUSE)

Reset value: 0x0000

Always accessible, even when the RI5CY core is running.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IRQ	Reserved														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											CAUSE				

Bit 31 - **IRQ** *(R)*

Core in debug mode due to interrupt trap status bitfield:

- *0b0*: false
- *0b1*: true

Bits 4:0 - **CAUSE** *(R)*

Exception ID bitfield. If IRQ is *0b1* contains interrupt number otherwise:

- *0x2*: sigill - Illegal Instruction
- *0x3*: sigtrap - breakpoint
- *0xB*: sigecall - eCall user mode
- *0xB* sigecall - eCall machine mode
- *0x1F* sigstop - core was halted by an external signal

6.2.8.9.5 Core general purpose register 0 value register. (GPR0)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPR0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPR0															

Bits 31:0 - **GPR0** *(R/W)*

General purpose register 0 value bitfield.

6.2.8.9.6 Core general purpose register 1 value register. (GPR1)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPR1															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPR1															

Bits 31:0 - **GPR1** *(R/W)*

General purpose register 1 value bitfield.

6.2.8.9.7 Core general purpose register 2 value register. (GPR2)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPR2															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPR2															

Bits 31:0 - **GPR2** *(R/W)*

General purpose register 2 value bitfield.

6.2.8.9.8 Core general purpose register 3 value register. (GPR3)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPR3															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPR3															

Bits 31:0 - **GPR3** *(R/W)*

General purpose register 3 value bitfield.

6.2.8.9.9 Core general purpose register 4 value register. (GPR4)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPR4															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPR4															

Bits 31:0 - **GPR4** *(R/W)*

General purpose register 4 value bitfield.

6.2.8.9.10 Core general purpose register 5 value register. (GPR5)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPR5															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPR5															

Bits 31:0 - **GPR5** *(R/W)*

General purpose register 5 value bitfield.

6.2.8.9.11 Core general purpose register 6 value register. (GPR6)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPR6															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPR6															

Bits 31:0 - **GPR6** *(R/W)*

General purpose register 6 value bitfield.

6.2.8.9.12 Core general purpose register 7 value register. (GPR7)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPR7															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPR7															

Bits 31:0 - **GPR7** *(R/W)*

General purpose register 7 value bitfield.

6.2.8.9.13 Core general purpose register 8 value register. (GPR8)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPR8															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPR8															

Bits 31:0 - **GPR8** *(R/W)*

General purpose register 8 value bitfield.

6.2.8.9.14 Core general purpose register 9 value register. (GPR9)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPR9															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPR9															

Bits 31:0 - **GPR9** *(R/W)*

General purpose register 9 value bitfield.

6.2.8.9.15 Core general purpose register 10 value register. (GPR10)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPR10															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPR10															

Bits 31:0 - **GPR10** *(R/W)*

General purpose register 10 value bitfield.

6.2.8.9.16 Core general purpose register 11 value register. (GPR11)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPR11															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPR11															

Bits 31:0 - **GPR11** *(R/W)*

General purpose register 11 value bitfield.

6.2.8.9.17 Core general purpose register 12 value register. (GPR12)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPR12															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPR12															

Bits 31:0 - **GPR12** *(R/W)*

General purpose register 12 value bitfield.

6.2.8.9.18 Core general purpose register 13 value register. (GPR13)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPR13															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPR13															

Bits 31:0 - **GPR13** *(R/W)*

General purpose register 13 value bitfield.

6.2.8.9.19 Core general purpose register 14 value register. (GPR14)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPR14															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPR14															

Bits 31:0 - **GPR14** *(R/W)*

General purpose register 14 value bitfield.

6.2.8.9.20 Core general purpose register 15 value register. (GPR15)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPR15															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPR15															

Bits 31:0 - **GPR15** *(R/W)*

General purpose register 15 value bitfield.

6.2.8.9.21 Core general purpose register 16 value register. (GPR16)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPR16															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPR16															

Bits 31:0 - **GPR16** *(R/W)*

General purpose register 16 value bitfield.

6.2.8.9.22 Core general purpose register 17 value register. (GPR17)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPR17															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPR17															

Bits 31:0 - **GPR17** *(R/W)*

General purpose register 17 value bitfield.

6.2.8.9.23 Core general purpose register 18 value register. (GPR18)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPR18															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPR18															

Bits 31:0 - **GPR18** *(R/W)*

General purpose register 18 value bitfield.

6.2.8.9.24 Core general purpose register 19 value register. (GPR19)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPR19															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPR19															

Bits 31:0 - **GPR19** *(R/W)*

General purpose register 19 value bitfield.

6.2.8.9.25 Core general purpose register 20 value register. (GPR20)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPR20															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPR20															

Bits 31:0 - **GPR20** *(R/W)*

General purpose register 20 value bitfield.

6.2.8.9.26 Core general purpose register 21 value register. (GPR21)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPR21															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPR21															

Bits 31:0 - **GPR21** *(R/W)*

General purpose register 21 value bitfield.

6.2.8.9.27 Core general purpose register 22 value register. (GPR22)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPR22															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPR22															

Bits 31:0 - **GPR22** *(R/W)*

General purpose register 22 value bitfield.

6.2.8.9.28 Core general purpose register 23 value register. (GPR23)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPR23															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPR23															

Bits 31:0 - **GPR23** *(R/W)*

General purpose register 23 value bitfield.

6.2.8.9.29 Core general purpose register 24 value register. (GPR24)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPR24															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPR24															

Bits 31:0 - **GPR24** *(R/W)*

General purpose register 24 value bitfield.

6.2.8.9.30 Core general purpose register 25 value register. (GPR25)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPR25															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPR25															

Bits 31:0 - **GPR25** *(R/W)*

General purpose register 25 value bitfield.

6.2.8.9.31 Core general purpose register 26 value register. (GPR26)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPR26															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPR26															

Bits 31:0 - **GPR26** *(R/W)*

General purpose register 26 value bitfield.

6.2.8.9.32 Core general purpose register 27 value register. (GPR27)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPR27															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPR27															

Bits 31:0 - **GPR27** *(R/W)*

General purpose register 27 value bitfield.

6.2.8.9.33 Core general purpose register 28 value register. (GPR28)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPR28															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPR28															

Bits 31:0 - **GPR28** *(R/W)*

General purpose register 28 value bitfield.

6.2.8.9.34 Core general purpose register 29 value register. (GPR29)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPR29															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPR29															

Bits 31:0 - **GPR29** *(R/W)*

General purpose register 29 value bitfield.

6.2.8.9.35 Core general purpose register 30 value register. (GPR30)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPR30															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPR30															

Bits 31:0 - **GPR30** *(R/W)*

General purpose register 30 value bitfield.

6.2.8.9.36 Core general purpose register 31 value register. (GPR31)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPR31															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPR31															

Bits 31:0 - **GPR31** *(R/W)*

General purpose register 31 value bitfield.

6.2.8.9.37 Debug next program counter value register. (NPC)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

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

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

Next program counter value bitfield.

6.2.8.9.38 Debug previous program counter value register. (PPC)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

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

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

Previous program counter value bitfield.

6.2.8.9.39 Core CSR user privilege mode hardware thread ID status register. (CSR_UHARTID)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

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					CLUSTER_ID						Reserved	CORE_ID			

Bits 10:5 - **CLUSTER_ID** *(R)*

Cluster ID value bitfield.

Bits 3:0 - **CORE_ID** *(R)*

RI5CY core ID value bitfield.

6.2.8.9.40 Core CSR machine status value register. (CSR_MSTATUS)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

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			MPP		Reserved			MPIE	Reserved			MIE	Reserved		

Bits 12:11 - **MPP** *(R/W)*

Hardwired to *0b11*.

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

Machine privilege mode previous interrupt enable value bitfield. When an interrupis encountered, MPIE will store the value existing in MIE. When mret instruction is executed, the value of MPIE is restored into MIE.

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

Machine privilege mode interrupt enable configuration bitfield:

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

6.2.8.9.41 Core CSR machine vector-trap base address value register. (CSR_MTVEC)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

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

Bits 31:8 - **MTVEC** *(R/W)*

Machine trap-vector base address value bitfield.

6.2.8.9.42 Core CSR machine exception program counter value register. (CSR_MEPC)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

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

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

Machine exception program counter value bitfield. When an exception is encountered, the current program counter is saved in MEPC, and the core jumps to the exception address. When an mret instruction is executed, the value from MEPC is restored to the current program counter.

6.2.8.9.43 Core CSR machine trap cause value register. (CSR_MCAUSE)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

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

Bit 31 - **IRQ** *(R)*

Core triggered an exception due to interrupt status bitfield:

- *0b0*: false
- *0b1*: true

Bits 4:0 - **CAUSE** *(R)*

Exception ID bitfield. If IRQ is *0b1* contains interrupt number otherwise:

- *0x2*: sigill - Illegal Instruction
- *0x3*: sigtrap - breakpoint
- *0xB*: sigecall - eCall user mode
- *0xB* sigecall - eCall machine mode
- *0x1F* sigstop - core was halted by an external signal

6.2.8.9.44 Core CSR performance counter counter register. (CSR_PCCR)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

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

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

Program counter counter value bitfield.

6.2.8.9.45 Core CSR performance counter enable configuration register. (CSR_PCER)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

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

Bits 20:0 - **PCER** *(R/W)*

See documentation on RI5CY core for details.

6.2.8.9.46 Core CSR performance counter mode configuration register. (CSR_PCMR)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

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														GE	SAT

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

Performance counter activation configuration bitfield:

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

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

Performance counter saturation mode configuration bitfield:

- *0b0*: wrap around - wrap-around mode
- *0b1*: saturation – saturation mode

6.2.8.9.47 Core CSR hardware loop 0 start configuration register. (CSR_HWLP0S)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

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

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

Hardware loop start address configuration bitfield.

6.2.8.9.48 Core CSR hardware loop 0 end configuration register. (CSR_HWLP0E)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

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

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

Hardware loop end address configuration bitfield.

6.2.8.9.49 Core CSR hardware loop 0 counter configuration register. (CSR_HWLP0C)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

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

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

Hardware loop counter configuration bitfield.

6.2.8.9.50 Core CSR hardware loop 1 start configuration register. (CSR_HWLP1S)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

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

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

Hardware loop start address configuration bitfield.

6.2.8.9.51 Core CSR hardware loop 1 end configuration register. (CSR_HWLP1E)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

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

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

Hardware loop end address configuration bitfield.

6.2.8.9.52 Core CSR hardware loop 1 counter configuration register. (CSR_HWLP1C)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

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

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

Hardware loop counter configuration bitfield.

6.2.8.9.53 Cose CSR privilege level status register. (CSR_PRIVLV)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

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														PLEV	

Bits 1:0 - **PLEV** *(R)*

Hardwired to *0b11*.

6.2.8.9.54 Core CSR machine privilege mode hardware thread ID status register. (CSR_MHARTID)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

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					CLUSTER_ID						Reserved	CORE_ID			

Bits 10:5 - **CLUSTER_ID** *(R)*

Cluster ID value bitfield.

Bits 3:0 - **CORE_ID** *(R)*

RI5CY core ID value bitfield.

6.3 SoC Peripherals Subsystem

6.3.1 SoC Peripherals Subsystem Events

Event number	Event name	IP instance name	Direction	Description
0	LVDS_RX_EVT	LVDS	Input	uDMA LVDS RX channel transfer done event
1	LVDS_TX_EVT	LVDS	Input	uDMA LVDS TX channel transfer done event
2	SPIM0_RX_EVT	SPIM0	Input	uDMA SPIM0 RX channel transfer done event
3	SPIM0_TX_EVT	SPIM0	Input	uDMA SPIM0 TX channel transfer done event

Event number	Event name	IP instance name	Direction	Description
4	SPIM1_RX_EVT	SPIM1	Input	uDMA SPIM1 RX channel transfer done event
5	SPIM1_TX_EVT	SPIM1	Input	uDMA SPIM1 TX channel transfer done event
6	HYPER_RX_EVT	HYPER	Input	uDMA HYPER RX channel transfer done event
7	HYPER_TX_EVT	HYPER	Input	uDMA HYPER TX channel transfer done event
8	UART_RX_EVT	UART	Input	uDMA UART RX channel transfer done event
9	UART_TX_EVT	UART	Input	uDMA UART TX channel transfer done event
10	I2C0_RX_EVT	I2C0	Input	uDMA I2C0 RX channel transfer done event
11	I2C0_TX_EVT	I2C0	Input	uDMA I2C0 TX channel transfer done event
12	I2C1_RX_EVT	I2C1	Input	uDMA I2C1 RX channel transfer done event
13	I2C1_TX_EVT	I2C1	Input	uDMA I2C1 TX channel transfer done event
14	MEMCPY_RX_EVT	MEMCPY	Input	uDMA MEMCPY RX channel transfer done event
15	MEMCPY_TX_EVT	MEMCPY	Input	uDMA MEMCPY TX channel transfer done event
16	I2S_CH0_RX_EVT	I2S	Input	uDMA I2S RX channel transfer done event
17	I2S_CH1_RX_EVT	I2S	Input	uDMA I2S TX channel transfer done event
18	CPI_RX_EVT	CPI	Input	uDMA CPI RX channel transfer done event
19	Reserved			Reserved
20	LVDS_HP_0_EVT	LVDS	Input	LVDS High priority event 0. It occurs when LVDS transceiver has received a number of sample corresponding to LVDS.RX_SAMPLE_CNT.CMP_VAL bitfield.
22	SPIM0_EOT_EVT	SPIM0	Input	uDMA SPIM0 End of Transaction event
23	SPIM1_EOT_EVT	SPIM0	Input	uDMA SPIM1 End of Transaction event
24	Reserved			Reserved
25	Reserved			Reserved
26	Reserved			Reserved
27	Reserved			Reserved
28	Reserved			Reserved
29	Reserved			Reserved
30	Reserved			Reserved
31	CL_POK_EVT	DLC_BRIDGE	Input	PMU cluster power ok event
32	Reserved			Reserved
33	Reserved			Reserved
34	Reserved			Reserved
35	CL_CG_EVT	DLC_BRIDGE	Input	PMU cluster clock gated ok event
36	PICL_OK	DLC_BRIDGE	Input	PMU register write done event
37	SCU_OK	DLC_BRIDGE	Input	PMU operating point reached event
38	ADV_TIMER_EVT_0	APB_ADV_TIMER	Input	ADV_TIMER event 0
39	ADV_TIMER_EVT_1	APB_ADV_TIMER	Input	ADV_TIMER event 1
40	ADV_TIMER_EVT_2	APB_ADV_TIMER	Input	ADV_TIMER event 2
41	ADV_TIMER_EVT_3	APB_ADV_TIMER	Input	ADV_TIMER event 3
42	GPIO_EVT	APB_GPIO	Input	GPIO event
43	RTC_APB_EVT	RTC	Input	RTC register R/W access done event
44	RTC_EVT	RTC	Input	RTC 32kHz reference clock rise event
45	Reserved			Reserved
46	Reserved			Reserved
47	Reserved			Reserved

Table 37. SoC Peripherals Subsystem Events table

Table 37. SoC Peripherals Subsystem Events table

Event number	Event name	IP instance name	Direction	Description

6.3.2 FLLs

FLL component manages the following features:

- generation of a clock signal that is locked to a frequency requested
- 2 configurable modes: normal and standalone
- output clock can be gated when unlocked

6.3.2.1 SoC FLL registers

Name	Address	Size	Type	Access	Default	Description
STATUS	0x1A100000	32	Config	R/W	*0x0000*	FLL status register.
CFG1	0x1A100004	32	Config	R/W	*0x0000*	FLL configuration 1 register.
CFG2	0x1A100008	32	Config	R/W	*0x0000*	FLL configuration 2 register.
INTEG	0x1A10000C	32	Config	R/W	*0x0000*	FLL integrator configuration register.

Table 38. SoC FLL registers table

6.3.2.2 Cluster FLL registers

Name	Address	Size	Type	Access	Default	Description
STATUS	0x1A100800	32	Config	R/W	*0x0000*	FLL status register.
CFG1	0x1A100804	32	Config	R/W	*0x0000*	FLL configuration 1 register.
CFG2	0x1A100808	32	Config	R/W	*0x0000*	FLL configuration 2 register.
INTEG	0x1A10080C	32	Config	R/W	*0x0000*	FLL integrator configuration register.

Table 39. Cluster FLL registers table

6.3.2.3 FLL registers details

6.3.2.3.1 FLL status register. (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**
MF															

Bits 15:0 - **MF** *(R)*

Current DCO multiplication factor value bitfield.

6.3.2.3.2 FLL configuration 1 register. (CFG1)

Reset value: 0x0000

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

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

FLL operation mode configuration bitfield:

- *0b0*: standalone
- *0b1*: normal

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

FLL output gated by LOCK signal configuration bitfield:

- *0b0*: not gated
- *0b1*: gated

Bits 29:26 - **CKDIV** *(R/W)*

FLL output clock divider configuration bitfield.

Bits 25:16 - **ICS** *(R/W)*

DCO input code in standalone mode bitfield.

Bits 15:0 - **MFN** *(R/W)*

Target clock multiplication factor in normal mode bitfield.

6.3.2.3.3 FLL configuration 2 register. (CFG2)

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DITH	OL	CKSEL	Reserved	LT											
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SCKL						UCKL						LG			

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

Dithering activation bitfield:

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

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

Open loop when locked activation bitfield:

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

Bit 29 - **CKSEL** *(R/W)*

Configuration clock selection in standalone mode bitfield:

- *0b0*: DCO clock
- *0b1*: Reference clock

Bits 27:16 - **LT** *(R/W)*

Lock tolerance configuration bitfield. It is the margin around multiplication factor within which the output clock is considered stable.

Bits 15:10 - **SCKL** *(R/W)*

Number of stable REFCLK cycles until LOCK assert in normal mode. Upper 6-bit of LOCK assert counter target in standalone mode.

Bits 9:4 - **UCKL** *(R/W)*

Number of unstable REFCLK cycles until LOCK de-assert in normal mode. Lower 6-bit of LOCK assert counter target in standalone mode.

Bits 3:0 - **LG** *(R/W)*

FLL loop gain setting bitfield.

6.3.2.3.4 FLL integrator configuration register. (INTEG)

Reset value: 0x0000

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

Bits 25:16 - **INTEG** *(R/W)*

integer part of integrator state bitfield. It corresponds to DCO unit bits.

Bits 15:6 - **FRAC** *(R/W)*

Fractional part of integrator state bitfield. It corresponds to dither unit input.

6.3.3 GPIO

GPIO component manages the following features:

- 32 general purpose input/ouput pads
- configurable pull and drive strength on each GPIOs
- configurable event trigger on GPIO event

6.3.3.1 GPIO registers

Name	Address	Size	Type	Access	Default	Description
PADDIR	0x1A101000	32	Config	R/W	*0x0000*	GPIO pad direction configuration register.
PADIN	0x1A101004	32	Config	R	*0x0000*	GPIO pad input value register.
PADOUT	0x1A101008	32	Config	R/W	*0x0000*	GPIO pad output value register.
INTEN	0x1A10100C	32	Config	R/W	*0x0000*	GPIO pad interrupt enable configuration register.
INTTYPE0	0x1A101010	32	Config	R/W	*0x0000*	GPIO pad interrupt type bit 0 configuration register.
INTTYPE1	0x1A101014	32	Config	R/W	*0x0000*	GPIO pad interrupt type bit 1 configuration register.
INTSTATUS	0x1A101018	32	Status	R	*0x0000*	GPIO pad interrupt status register.
GPIOEN	0x1A10101C	32	Config	R/W	*0x0000*	GPIO pad enable configuration register.
PADCFG0	0x1A101020	32	Config	R/W	*0x0000*	GPIO pad pin 0 to 3 configuration register.
PADCFG1	0x1A101024	32	Config	R/W	*0x0000*	GPIO pad pin 4 to 7 configuration register.
PADCFG2	0x1A101028	32	Config	R/W	*0x0000*	GPIO pad pin 8 to 11 configuration register.
PADCFG3	0x1A10102C	32	Config	R/W	*0x0000*	GPIO pad pin 12 to 15 configuration register.
PADCFG4	0x1A101030	32	Config	R/W	*0x0000*	GPIO pad pin 16 to 19 configuration register.

Name	Address	Size	Type	Access	Default	Description
PADCFG5	0x1A101034	32	Config	R/W	*0x0000*	GPIO pad pin 20 to 23 configuration register.
PADCFG6	0x1A101038	32	Config	R/W	*0x0000*	GPIO pad pin 24 to 27 configuration register.
PADCFG7	0x1A10103C	32	Config	R/W	*0x0000*	GPIO pad pin 28 to 31 configuration register.

Table 40. GPIO registers table

6.3.3.2 GPIO registers details

6.3.3.2.1 GPIO pad direction configuration register. (PADDIR)

Reset value: 0x0000

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

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

GPIO direction configuration bitfield:

- bit[i]=*0b0*: Input mode for GPIO[i]
- bit[i]=*0b1*: Output mode for GPIO[i]

6.3.3.2.2 GPIO pad input value register. (PADIN)

Reset value: 0x0000

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

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

GPIO input data read bitfield. DATA_IN[i] corresponds to input data of GPIO[i].

6.3.3.2.3 GPIO pad output value register. (PADOUT)

Reset value: 0x0000

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

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

GPIO output data read bitfield. DATA_OUT[i] corresponds to output data set on GPIO[i].

6.3.3.2.4 GPIO pad interrupt enable configuration register. (INTEN)

Reset value: 0x0000

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

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

GPIO interrupt enable configuration bitfield:

- bit[i]=*0b0*: disable interrupt for GPIO[i]
- bit[i]=*0b1*: enable interrupt for GPIO[i]

6.3.3.2.5 GPIO pad interrupt type bit 0 configuration register. (INTTYPE0)

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INTTYPE0															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
INTTYPE0															

Bits 31:0 - **INTTYPE0** *(R/W)*

GPIO[15:0] interrupt type configuration bitfield:

- bit[2*i+1:2*i]=*0b00*: interrupt on falling edge for GPIO[i]
- bit[2*i+1:2*i]=*0b01*: interrupt on rising edge for GPIO[i]
- bit[2*i+1:2*i]=*0b10*: interrupt on rising and falling edge for GPIO[i]
- bit[2*i+1:2*i]=*0b11*: RFU

6.3.3.2.6 GPIO pad interrupt type bit 1 configuration register. (INTTYPE1)

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INTTYPE1															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
INTTYPE1															

Bits 31:0 - **INTTYPE1** *(R/W)*

GPIO[31:16] interrupt type configuration bitfield:

- bit[2*i+1:2*i]=*0b00*: interrupt on falling edge for GPIO[16+i]
- bit[2*i+1:2*i]=*0b01*: interrupt on rising edge for GPIO[16+i]
- bit[2*i+1:2*i]=*0b10*: interrupt on rising and falling edge for GPIO[16+i]
- bit[2*i+1:2*i]=*0b11*: RFU

6.3.3.2.7 GPIO pad interrupt status register. (INTSTATUS)

Reset value: 0x0000

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

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

GPIO Interrupt status flags bitfield. INTSTATUS[i]=1 when interrupt received on GPIO[i]. INTSTATUS is cleared when it is red. GPIO interrupt line is also cleared when INTSTATUS register is red.

6.3.3.2.8 GPIO pad enable configuration register. (GPIOEN)

Reset value: 0x0000

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

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

GPIO clock enable configuration bitfield:

- bit[i]=*0b0*: disable clock for GPIO[i]
- bit[i]=*0b1*: enable clock for GPIO[i]

GPIOs are gathered by groups of 4. The clock gating of one group is done only if all 4 GPIOs are disabled.

Clock must be enabled for a GPIO if it's direction is configured in input mode.

6.3.3.2.9 GPIO pad pin 0 to 3 configuration register. (PADCFG0)

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						GPIO3_DS	GPIO3_PE	Reserved						GPIO2_DS	GPIO2_PE
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						GPIO1_DS	GPIO1_PE	Reserved						GPIO0_DS	GPIO0_PE

Bit 25 - **GPIO3_DS** *(R/W)*

GPIO[3] drive strength configuration bitfield:

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 24 - **GPIO3_PE** *(R/W)*

GPIO[3] pull activation configuration bitfield:

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

Bit 17 - **GPIO2_DS** *(R/W)*

GPIO[2] drive strength configuration bitfield:

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 16 - **GPIO2_PE** *(R/W)*

GPIO[2] pull activation configuration bitfield:

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

Bit 9 - **GPIO1_DS** *(R/W)*

GPIO[1] drive strength configuration bitfield:

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 8 - **GPIO1_PE** *(R/W)*

GPIO[1] pull activation configuration bitfield:

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

Bit 1 - **GPIO0_DS** *(R/W)*

GPIO[0] drive strength configuration bitfield:

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 0 - **GPIO0_PE** *(R/W)*

GPIO[0] pull activation configuration bitfield:

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

6.3.3.2.10 GPIO pad pin 4 to 7 configuration register. (PADCFG1)

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						GPIO7_DS	GPIO7_PE	Reserved						GPIO6_DS	GPIO6_PE

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						GPIO5_DS	GPIO5_PE	Reserved						GPIO4_DS	GPIO4_PE

Bit 25 - **GPIO7_DS** *(R/W)*

GPIO[7] drive strength configuration bitfield:

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 24 - **GPIO7_PE** *(R/W)*

GPIO[7] pull activation configuration bitfield:

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

Bit 17 - **GPIO6_DS** *(R/W)*

GPIO[6] drive strength configuration bitfield:

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 16 - **GPIO6_PE** *(R/W)*

GPIO[6] pull activation configuration bitfield:

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

Bit 9 - **GPIO5_DS** *(R/W)*

GPIO[5] drive strength configuration bitfield:

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 8 - **GPIO5_PE** *(R/W)*

GPIO[5] pull activation configuration bitfield:

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

Bit 1 - **GPIO4_DS** *(R/W)*

GPIO[4] drive strength configuration bitfield:

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 0 - **GPIO4_PE** *(R/W)*

GPIO[4] pull activation configuration bitfield:

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

6.3.3.2.11 GPIO pad pin 8 to 11 configuration register. (PADCFG2)

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						GPIO11_DS	GPIO11_PE	Reserved						GPIO10_DS	GPIO10_PE
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						GPIO9_DS	GPIO9_PE	Reserved						GPIO8_DS	GPIO8_PE

Bit 25 - **GPIO11_DS** *(R/W)*

GPIO[11] drive strength configuration bitfield:

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 24 - **GPIO11_PE** *(R/W)*

GPIO[11] pull activation configuration bitfield:

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

Bit 17 - **GPIO10_DS** *(R/W)*

GPIO[10] drive strength configuration bitfield:

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 16 - **GPIO10_PE** *(R/W)*

GPIO[10] pull activation configuration bitfield:

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

Bit 9 - **GPIO9_DS** *(R/W)*

GPIO[9] drive strength configuration bitfield:

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 8 - **GPIO9_PE** *(R/W)*

GPIO[9] pull activation configuration bitfield:

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

Bit 1 - **GPIO8_DS** *(R/W)*

GPIO[8] drive strength configuration bitfield:

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 0 - **GPIO8_PE** *(R/W)*

GPIO[8] pull activation configuration bitfield:

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

6.3.3.2.12 GPIO pad pin 12 to 15 configuration register. (PADCFG3)

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						GPIO15_DS	GPIO15_PE	Reserved						GPIO14_DS	GPIO14_PE
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						GPIO13_DS	GPIO13_PE	Reserved						GPIO12_DS	GPIO12_PE

Bit 25 - **GPIO15_DS** *(R/W)*

GPIO[15] drive strength configuration bitfield:

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 24 - **GPIO15_PE** *(R/W)*

GPIO[15] pull activation configuration bitfield:

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

Bit 17 - **GPIO14_DS** *(R/W)*

GPIO[14] drive strength configuration bitfield:

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 16 - **GPIO14_PE** *(R/W)*

GPIO[14] pull activation configuration bitfield:

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

Bit 9 - **GPIO13_DS** *(R/W)*

GPIO[13] drive strength configuration bitfield:

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 8 - **GPIO13_PE** *(R/W)*

GPIO[13] pull activation configuration bitfield:

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

Bit 1 - **GPIO12_DS** *(R/W)*

GPIO[12] drive strength configuration bitfield:

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 0 - **GPIO12_PE** *(R/W)*

GPIO[12] pull activation configuration bitfield:

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

6.3.3.2.13 GPIO pad pin 16 to 19 configuration register. (PADCFG4)

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						GPIO19_DS	GPIO19_PE	Reserved						GPIO18_DS	GPIO18_PE
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						GPIO17_DS	GPIO17_PE	Reserved						GPIO16_DS	GPIO16_PE

Bit 25 - **GPIO19_DS** *(R/W)*

GPIO[19] drive strength configuration bitfield:

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 24 - **GPIO19_PE** *(R/W)*

GPIO[19] pull activation configuration bitfield:

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

Bit 17 - **GPIO18_DS** *(R/W)*

GPIO[18] drive strength configuration bitfield:

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 16 - **GPIO18_PE** *(R/W)*

GPIO[18] pull activation configuration bitfield:

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

Bit 9 - **GPIO17_DS** *(R/W)*

GPIO[17] drive strength configuration bitfield:

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 8 - **GPIO17_PE** *(R/W)*

GPIO[17] pull activation configuration bitfield:

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

Bit 1 - **GPIO16_DS** *(R/W)*

GPIO[16] drive strength configuration bitfield:

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 0 - **GPIO16_PE** *(R/W)*

GPIO[16] pull activation configuration bitfield:

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

6.3.3.2.14 GPIO pad pin 20 to 23 configuration register. (PADCFG5)

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						GPIO23_DS	GPIO23_PE	Reserved						GPIO22_DS	GPIO22_PE
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						GPIO21_DS	GPIO21_PE	Reserved						GPIO20_DS	GPIO20_PE

Bit 25 - **GPIO23_DS** *(R/W)*

GPIO[23] drive strength configuration bitfield:

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 24 - **GPIO23_PE** *(R/W)*

GPIO[23] pull activation configuration bitfield:

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

Bit 17 - **GPIO22_DS** *(R/W)*

GPIO[22] drive strength configuration bitfield:

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 16 - **GPIO22_PE** *(R/W)*

GPIO[22] pull activation configuration bitfield:

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

Bit 9 - **GPIO21_DS** *(R/W)*

GPIO[21] drive strength configuration bitfield:

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 8 - **GPIO21_PE** *(R/W)*

GPIO[21] pull activation configuration bitfield:

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

Bit 1 - **GPIO20_DS** *(R/W)*

GPIO[20] drive strength configuration bitfield:

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 0 - **GPIO20_PE** *(R/W)*

GPIO[20] pull activation configuration bitfield:

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

6.3.3.2.15 GPIO pad pin 24 to 27 configuration register. (PADCFG6)

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						GPIO27_DS	GPIO27_PE	Reserved						GPIO26_DS	GPIO26_PE
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						GPIO25_DS	GPIO25_PE	Reserved						GPIO24_DS	GPIO24_PE

Bit 25 - **GPIO27_DS** *(R/W)*

GPIO[27] drive strength configuration bitfield:

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 24 - **GPIO27_PE** *(R/W)*

GPIO[27] pull activation configuration bitfield:

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

Bit 17 - **GPIO26_DS** *(R/W)*

GPIO[26] drive strength configuration bitfield:

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 16 - **GPIO26_PE** *(R/W)*

GPIO[26] pull activation configuration bitfield:

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

Bit 9 - **GPIO25_DS** *(R/W)*

GPIO[25] drive strength configuration bitfield:

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 8 - **GPIO25_PE** *(R/W)*

GPIO[25] pull activation configuration bitfield:

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

Bit 1 - **GPIO24_DS** *(R/W)*

GPIO[24] drive strength configuration bitfield:

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 0 - **GPIO24_PE** *(R/W)*

GPIO[24] pull activation configuration bitfield:

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

6.3.3.2.16 GPIO pad pin 28 to 31 configuration register. (PADCFG7)

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						GPIO31_DS	GPIO31_PE	Reserved						GPIO30_DS	GPIO30_PE
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						GPIO29_DS	GPIO29_PE	Reserved						GPIO28_DS	GPIO28_PE

Bit 25 - **GPIO31_DS** *(R/W)*

GPIO[31] drive strength configuration bitfield:

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 24 - **GPIO31_PE** *(R/W)*

GPIO[31] pull activation configuration bitfield:

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

Bit 17 - **GPIO30_DS** *(R/W)*

GPIO[30] drive strength configuration bitfield:

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 16 - **GPIO30_PE** *(R/W)*

GPIO[30] pull activation configuration bitfield:

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

Bit 9 - **GPIO29_DS** *(R/W)*

GPIO[29] drive strength configuration bitfield:

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 8 - **GPIO29_PE** *(R/W)*

GPIO[29] pull activation configuration bitfield:

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

Bit 1 - **GPIO28_DS** *(R/W)*

GPIO[28] drive strength configuration bitfield:

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 0 - **GPIO28_PE** *(R/W)*

GPIO[28] pull activation configuration bitfield:

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

6.3.4 SoC control unit

The APB SoC Controller allows control of various SoC features such as:

- PAD mux configuration
- JTAG boot mode
- Sleep modes
- L2 retentive state configuration
- DC/DC voltage regulator configuration in bypass mode

6.3.4.1 SoC control unit registers

Name	Address	Size	Type	Access	Default	Description
INFO	0x1A104000	32	Status	R	*0x0018*	Core information register
CL_ISOLATE	0x1A10400C	32	Config	R/W	*0x0001*	Isolate cluster register
CL_BUSY	0x1A10406C	32	Status	R	*0x0000*	Cluster busy register
CL_BYPASS	0x1A104070	32	Config	R/W	*0x0400*	PMU bypass configuration register
JTAGREG	0x1A104074	32	Config	R/W	*0x0000*	JTAG external register
L2_SLEEP	0x1A104078	32	Config	R/W	*0x0001*	L2 sleep configuration register
SLEEP_CTRL	0x1A10407C	32	Status	R	*0x0000*	Alias for SAFE_PMU_SLEEPCTRL
CORESTATUS	0x1A1040A0	32	Status	R/W	*0x0000*	EOC and chip status register
CORESTATUS_RO	0x1A1040C0	32	Status	R	*0x0000*	EOC and chip status register read mirror
SAFE_PMU_RAR	0x1A104100	32	Config	R/W	*0x2A52D*	DC/DC configuration register
SAFE_PMU_SLEEPCTRL	0x1A104104	32	Config	R/W	*0x0000*	Sleep modes configuration register
SAFE_PMU_FORCE	0x1A104108	32	Config	R/W	*0x0000*	L2 rententive state configuration
SAFE_PADFUN0	0x1A104140	32	Config	R/W	*0x0000*	Mux config register (pad 0–15)
SAFE_PADFUN1	0x1A104144	32	Config	R/W	*0x0000*	Mux config register (pad 16–31)
SAFE_PADFUN2	0x1A104148	32	Config	R/W	*0x0000*	Mux config register (pad 32–47)
SAFE_SLEEPPADCFG0	0x1A104150	32	Config	R/W	*0x0000*	Sleep config register (pad 0–15)
SAFE_SLEEPPADCFG1	0x1A104154	32	Config	R/W	*0x0000*	Mux config register (pad 16–31)

Name	Address	Size	Type	Access	Default	Description
SAFE_SLEEPPADCFG2	0x1A104158	32	Config	R/W	*0x0000*	Mux config register (pad 32–47)
SAFE_PADSLEEP	0x1A104160	32	Config	R/W	*0x0000*	Enable Sleep mode for pads
SAFE_PADCFG0	0x1A104180	32	Config	R/W	*0x0000*	Function register (pad 0 to 3)
SAFE_PADCFG1	0x1A104184	32	Config	R/W	*0x0000*	Function register (pad 4 to 7)
SAFE_PADCFG2	0x1A104188	32	Config	R/W	*0x0000*	Function register (pad 8 to 11)
SAFE_PADCFG3	0x1A10418C	32	Config	R/W	*0x0000*	Function register (pad 12 to 15)
SAFE_PADCFG4	0x1A104190	32	Config	R/W	*0x0000*	Function register (pad 16 to 19)
SAFE_PADCFG5	0x1A104194	32	Config	R/W	*0x0000*	Function register (pad 20 to 23)
SAFE_PADCFG6	0x1A104198	32	Config	R/W	*0x0000*	Function register (pad 24 to 27)
SAFE_PADCFG7	0x1A10419C	32	Config	R/W	*0x0000*	Function register (pad 28 to 31)
SAFE_PADCFG8	0x1A1041A0	32	Config	R/W	*0x0000*	Function register (pad 32 to 35)
SAFE_PADCFG9	0x1A1041A4	32	Config	R/W	*0x0000*	Function register (pad 36 to 39)
SAFE_PADCFG10	0x1A1041A8	32	Config	R/W	*0x0000*	Function register (pad 40 to 43)
SAFE_PADCFG11	0x1A1041AC	32	Config	R/W	*0x0000*	Function register (pad 44 to 47)
REG_GPIO_ISO	0x1A1041C0	32	Config	R/W	*0x0000*	GPIO power domain pad input isolation register
REG_CAM_ISO	0x1A1041C4	32	Config	R/W	*0x0000*	CAM power domain pad input isolation register
REG_LVDS_ISO	0x1A1041C8	32	Config	R/W	*0x0000*	LVDS power domain pad input isolation register

Table 41. SoC control unit registers table

6.3.4.2 SoC control unit registers details

6.3.4.2.1 Core information register (INFO)

Reset value: 0x0018

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

Bits 31:16 - **NB_CORES** *(R)*

Reset value: 0x00

Number of cores

Bits 15:0 - **NB_CL** *(R)*

Reset value: 0x18

Number of clusters

6.3.4.2.2 Isolate cluster register (CL_ISOLATE)

Reset value: 0x0001

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)*

Reset value: 0b1

Isolate cluster. Inhibits AXI transactions from cluster to SoC:

- *0b0*: Disable
- *0b1*: Enable

6.3.4.2.3 Cluster busy register (CL_BUSY)

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															BUSY

Bit 0 - **BUSY** *(R)*

Cluster busy flag (i.e. It's 1 if there is at least 1 active block in the cluster)

6.3.4.2.4 PMU bypass configuration register (CL_BYPASS)

Reset value: 0x0400

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved														PMUPOWDOWN	TRCPOWOK
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		RST	FLL_RET	FLL_PWD	CG	BYP_CLK	PROG_DEL		CURRSET			CL_STATE	Reserved	BYP_CFG	BYP_POW

Bit 17 - **PMUPOWDOWN** *(R/W)*

Reset value: 0b0

Cluster power down from Maestro PMU status bitfield.

Bit 16 - **TRCPOWOK** *(R/W)*

Reset value: 0b0

Cluster power ok from cluster TRC status bitfield

Bit 13 - **RST** *(R/W)*

Reset value: 0b0

Cluster reset configuration bitfield:

- 1'b0: nothing
- 1'b1: reset the cluster

Bit 12 - **FLL_RET** *(R/W)*

Reset value: 0b0

Cluster FLL retentive configuration bitfield:

- 1'b0: FLL on
- 1'b1: FLL retentive mode

Bit 11 - **FLL_PWD** *(R/W)*

Reset value: 0b0

Cluster FLL shutdown configuration bitfield:

- 1’b0: FLL on
- 1’b1: FLL shutdown mode

Bit 10 - **CG** *(R/W)*

Reset value: 0b1

Cluster clock gate configuration bitfield:

- 1’b0: disabled
- 1’b1: enabled

It should always be used before switching cluster FLL to shutdown or retentive mode.

Bit 9 - **BYP_CLK** *(R/W)*

Reset value: 0b0

Bypass cluster clock and reset control by Maestro PMU configuration bitfield:

- 1’b0: disabled
- 1’b1: enabled

Bits 8:7 - **PROG_DEL** *(R/W)*

Reset value: 0b00

Number of REFCLK 32kHz after cluster power ok has arised to release TR isolation configuration bitfield.

Bits 6:4 - **CURRSET** *(R/W)*

Reset value: 0b000

Max current allowed on cluster TRC configuration bitfield.

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

Reset value: 0b0

Cluster state configuration and status bitfield:

- 1’b0: off
- 1’b1: on

Status information is correct only when bypass mode is enabled.

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

Reset value: 0b0

Bypass Maestro PMU configuration selection configuration bitfield:

- 1’b0: use default
- 1’b1: use user configuration (bitfields from bits 3 to 15 of CL_BYPASS register)

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

Reset value: 0b0

Bypass Maestro PMU controller configuration bitfield:

- 1'b0: disabled
- 1'b1: enabled

6.3.4.2.5 JTAG external register (JTAGREG)

Reset value: 0x0000

This register is used for synchronisation and boot mode configuration from an external debugger.

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				EXT_BT_MD			EXT_SYNC	Reserved				INT_BT_MD			INT_SYNC

Bits 11:9 - **EXT_BT_MD** *(R)*

JTAG external register used for selecting boot mode configuration from external debugger

Bit 8 - **EXT_SYNC** *(R)*

JTAG external register used for synchronisation from external debugger

Bits 3:1 - **INT_BT_MD** *(R/W)*

JTAG internal register used for selecting boot mode configuration from external debugger

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

JTAG internal register used for synchronisation from external debugger

6.3.4.2.6 L2 sleep configuration register (L2_SLEEP)

Reset value: 0x0001

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															L2_SLEEP

Bit 0 - **L2_SLEEP** *(R/W)*

Reset value: 0b1

L2 memory sleep configuration

6.3.4.2.7 Alias for SAFE_PMU_SLEEPCTRL (SLEEP_CTRL)

Reset value: 0x0000

This register will be accessible in 1 clock cycle

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

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

Alias for SAFE_PMU_SLEEPCTRL(i.e. will be accessible in 1 clock cycle)

6.3.4.2.8 EOC and chip status register (CORESTATUS)

Reset value: 0x0000

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

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

EOC and chip status register

6.3.4.2.9 EOC and chip status register read mirror (CORESTATUS_RO)

Reset value: 0x0000

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

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

EOC and chip status register

6.3.4.2.10 DC/DC configuration register (SAFE_PMU_RAR)

Reset value: 0x2A52D

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

Bits 28:24 - **RV_VOLT** *(R/W)*
Reset value: 0x0

DC/DC Retentive Voltage setting

Bits 20:16 - **LV_VOLT** *(R/W)*
Reset value: 0x2

DC/DC Low Voltage setting

Bits 12:8 - **MV_VOLT** *(R/W)*
Reset value: 0x5

DC/DC Medium Voltage setting (not used)

Bits 4:0 - **NV_VOLT** *(R/W)*
Reset value: 0xD

DC/DC Nominal Voltage setting

6.3.4.2.11 Sleep modes configuration register (SAFE_PMU_SLEEPCTRL)

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved											CL_WAKE	BTTYPE		EXTINT	BTDEV

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	WAKESTATE	EXTWAKE_EN	EXTWAKE_TYPE		EXTWAKE_SRC					CL_FLL	SOC_FLL	L2_R3	L2_R2	L2_R1	L2_R0

Bit 20 - **CL_WAKE** *(R/W)*

Cluster state to restore after warm boot:

- 1'b0: off
- 1'b1: on

Bits 19:18 - **BTTYPE** *(R/W)*

Select boot type:

- *0b00*: cold boot
- *0b01*: deep sleep
- *0b10*: retentive deep sleep

Bit 17 - **EXTINT** *(R)*

External wake-up interrupt status (automatically resetted after read)

- *0b0*: wake-up triggered by RTC
- *0b1*: wake-up triggered by external event

Bit 16 - **BTDEV** *(R/W)*

Warm bootmode:

- *0b0*: Boot from ROM
- *0b1*: Boot from L2

Bit 14 - **WAKESTATE** *(R/W)*

Power state to restore after warm boot

- *0b0*: SoC_LV
- *0b1*: SoC_HV

Bit 13 - **EXTWAKE_EN** *(R/W)*

Enable external wake-up;

- *0b0*; external wake-up disabled
- *0b1*: external wake-up enabled

Bits 12:11 - **EXTWAKE_TYPE** *(R/W)*

Select external wake-up mode:

- *0b00*: rise event
- *0b01*: fall event
- *0b10*: high level
- *0b11*: low level

Bits 10:6 - **EXTWAKE_SRC** *(R/W)*

Select external wake-up source (GPIO ID):

- *0b00000*: GPIO 0
- *0b00001*: GPIO 1
- *0b00010*: GPIO 2
- *0b00011*: GPIO 3
- *0b00100*: GPIO 4
- *0b00101*: GPIO 5
- *0b00110*: GPIO 6
- *0b00111*: GPIO 7
- *0b01000*: GPIO 8
- *0b01001*: GPIO 9
- *0b01010*: GPIO 10
- *0b01011*: GPIO 11
- *0b01100*: GPIO 12
- *0b01101*: GPIO 13
- *0b01110*: GPIO 14
- *0b01111*: GPIO 15
- *0b10000*: GPIO 16
- *0b10001*: GPIO 17
- *0b10010*: GPIO 18
- *0b10011*: GPIO 19
- *0b10100*: GPIO 20
- *0b10101*: GPIO 21
- *0b10110*: GPIO 22
- *0b10111*: GPIO 23
- *0b11000*: GPIO 24
- *0b11001*: GPIO 25
- *0b11010*: GPIO 26
- *0b11011*: GPIO 27
- *0b11100*: GPIO 28
- *0b11101*: GPIO 29
- *0b11110*: GPIO 30
- *0b11111*: GPIO 31

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

Configure retention mode for cluster FLL:

- *0b0*: Non retentive
- *0b1*: Retentive

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

Configure retention mode for SoC FLL:

- *0b0*: Non retentive
- *0b1*: Retentive

Bit 3 - **L2_R3** *(R/W)*

Configure retention mode for region 3 of L2 memory:

- *0b0*: Non retentive
- *0b1*: Retentive

Bit 2 - **L2_R2** *(R/W)*

Configure retention mode for region 2 of L2 memory:

- *0b0*: Non retentive
- *0b1*: Retentive

Bit 1 - **L2_R1** *(R/W)*

Configure retention mode for region 1 of L2 memory:

- *0b0*: Non retentive
- *0b1*: Retentive

Bit 0 - **L2_R0** *(R/W)*

Configure retention mode for region 0 of L2 memory:

- *0b0*: Non retentive
- *0b1*: Retentive

6.3.4.2.12 L2 rententive state configuration (SAFE_PMU_FORCE)

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								PD_L2_R3	PD_L2_R2	PD_L2_R1	PD_L2_R0	RET_L2_R3	RET_L2_R2	RET_L2_R1	RET_L2_R0

Bit 7 - **PD_L2_R3** *(R/W)*

Force power down on region 3 of L2 memory:

0b0: power up

0b1: power down

Bit 6 - **PD_L2_R2** *(R/W)*

Force power down on region 2 of L2 memory:

0b0: power up

0b1: power down

Bit 5 - **PD_L2_R1** *(R/W)*

Force power down on region 1 of L2 memory:

0b0: power up

0b1: power down

Bit 4 - **PD_L2_R0** *(R/W)*

Force power down on region 0 of L2 memory:

0b0: power up

0b1: power down

Bit 3 - **RET_L2_R3** *(R/W)*

Force retentive state on region 3 of L2 memory:

0b0: not state retentive

0b1: state retentive

Bit 2 - **RET_L2_R2** *(R/W)*

Force retentive state on region 2 of L2 memory:

0b0: not state retentive

0b1: state retentive

Bit 1 - **RET_L2_R1** *(R/W)*

Force retentive state on region 1 of L2 memory:

0b0: not state retentive

0b1: state retentive

Bit 0 - **RET_L2_R0** *(R/W)*

Force retentive state on region 0 of L2 memory:

0b0: not state retentive

0b1: state retentive

6.3.4.2.13 Mux config register (pad 0–15) (SAFE_PADFUN0)

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
A41		B38		A42		B39		A37		A43		B40		A44	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
B1		A2		B2		A3		B4		A5		B3		A4	

Bits 31:30 - **A41** *(R/W)*

Configure A41 functionality:

- *0b00*: Alternate 0 - CAM_DATA3 (default)
- *0b01*: Alternate 1 - GPIOA9
- *0b10*: Alternate 2 - TIMER2_CH1
- *0b11*: Alternate 3 -

Bits 29:28 - **B38** *(R/W)*

Configure B38 functionality:

- *0b00*: Alternate 0 - CAM_DATA2 (default)
- *0b01*: Alternate 1 - GPIOA8
- *0b10*: Alternate 2 - TIMER2_CH0
- *0b11*: Alternate 3 -

Bits 27:26 - **A42** *(R/W)*

Configure A42 functionality:

- *0b00*: Alternate 0 - CAM_DATA1 (default)
- *0b01*: Alternate 1 - GPIOA7
- *0b10*: Alternate 2 - TIMER1_CH3
- *0b11*: Alternate 3 -

Bits 25:24 - **B39** *(R/W)*

Configure B39 functionality:

- *0b00*: Alternate 0 - CAM_DATA0 (default)
- *0b01*: Alternate 1 - GPIOA6
- *0b10*: Alternate 2 - TIMER1_CH2
- *0b11*: Alternate 3 -

Bits 23:22 - **A37** *(R/W)*

Configure A37 functionality:

- *0b00*: Alternate 0 - CAM_HSYNC (default)
- *0b01*: Alternate 1 - GPIOA5
- *0b10*: Alternate 2 - TIMER1_CH1
- *0b11*: Alternate 3 -

Bits 21:20 - **A43** *(R/W)*

Configure A43 functionality:

- *0b00*: Alternate 0 - CAM_PCLK (default)
- *0b01*: Alternate 1 - GPIOA4
- *0b10*: Alternate 2 - TIMER1_CH0
- *0b11*: Alternate 3 -

Bits 19:18 - **B40** *(R/W)*

Configure B40 functionality:

- *0b00*: Alternate 0 - ORCA_RXQ (default)
- *0b01*: Alternate 1 - GPIOA5
- *0b10*: Alternate 2 - SPIS0_SDIO1
- *0b11*: Alternate 3 - SPIS0_SDIO3

Bits 17:16 - **A44** *(R/W)*

Configure A44 functionality:

- *0b00*: Alternate 0 - ORCA_RXI (default)
- *0b01*: Alternate 1 - GPIOA4
- *0b10*: Alternate 2 - SPIS0_SDIO0
- *0b11*: Alternate 3 - SPIS0_SDIO2

Bits 15:14 - **B1** *(R/W)*

Configure B1 functionality:

- *0b00*: Alternate 0 - ORCA_TXQ (default)
- *0b01*: Alternate 1 - GPIOA3
- *0b10*: Alternate 2 -
- *0b11*: Alternate 3 -

Bits 13:12 - **A2** *(R/W)*

Configure A2 functionality:

- *0b00*: Alternate 0 - ORCA_TXI (default)
- *0b01*: Alternate 1 - GPIOA2
- *0b10*: Alternate 2 -
- *0b11*: Alternate 3 -

Bits 11:10 - **B2** *(R/W)*

Configure B2 functionality:

- *0b00*: Alternate 0 - ORCA_RXSYNC (default)
- *0b01*: Alternate 1 - GPIOA1
- *0b10*: Alternate 2 - SPIM1_CS1
- *0b11*: Alternate 3 -

Bits 9:8 - **A3** *(R/W)*

Configure A3 functionality:

- *0b00*: Alternate 0 - ORCA_TXSYNC (default)
- *0b01*: Alternate 1 - GPIOA0
- *0b10*: Alternate 2 - SPIM1_CS0
- *0b11*: Alternate 3 -

Bits 7:6 - **B4** *(R/W)*

Configure B4 functionality:

- *0b00*: Alternate 0 - SPIM1_SCK (default)
- *0b01*: Alternate 1 - GPIOA3
- *0b10*: Alternate 2 - I2C1_SCL
- *0b11*: Alternate 3 -

Bits 5:4 - **A5** *(R/W)*

Configure A5 functionality:

- *0b00*: Alternate 0 - SPIM1_CS0 (default)
- *0b01*: Alternate 1 - GPIOA2
- *0b10*: Alternate 2 - I2C1_SDA
- *0b11*: Alternate 3 -

Bits 3:2 - **B3** *(R/W)*

Configure B3 functionality:

- *0b00*: Alternate 0 - SPIM1_MOSI (default)
- *0b01*: Alternate 1 - GPIOA1
- *0b10*: Alternate 2 -
- *0b11*: Alternate 3 -

Bits 1:0 - **A4** *(R/W)*

Configure A4 functionality:

- *0b00*: Alternate 0 - SPIM1_MISO (default)
- *0b01*: Alternate 1 - GPIOA0
- *0b10*: Alternate 2 -
- *0b11*: Alternate 3 -

6.3.4.2.14 Mux config register (pad 16–31) (SAFE_PADFUN1)

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
A7		B6		B14		A15		B13		A14		B12		A13	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
B11		D1		B34		A36		A38		B36		A40		B37	

Bits 31:30 - **A7** *(R/W)*

Configure A7 functionality:

- *0b00*: Alternate 0 - UART_TX (default)
- *0b01*: Alternate 1 - GPIOA25
- *0b10*: Alternate 2 -
- *0b11*: Alternate 3 -

Bits 29:28 - **B6** *(R/W)*

Configure B6 functionality:

- *0b00*: Alternate 0 - UART_RX (default)
- *0b01*: Alternate 1 - GPIOA24
- *0b10*: Alternate 2 -
- *0b11*: Alternate 3 -

Bits 27:26 - **B14** *(R/W)*

Configure B14 functionality:

- *0b00*: Alternate 0 - I2S1_SDI (default)
- *0b01*: Alternate 1 - GPIOA23
- *0b10*: Alternate 2 - SPIS0_SDIO2
- *0b11*: Alternate 3 - HYPER_CK

Bits 25:24 - **A15** *(R/W)*

Configure A15 functionality:

- *0b00*: Alternate 0 - I2S1_WS (default)
- *0b01*: Alternate 1 - GPIOA22
- *0b10*: Alternate 2 - SPIS0_CS
- *0b11*: Alternate 3 - HYPER_CKN

Bits 23:22 - **B13** *(R/W)*

Configure B13 functionality:

- *0b00*: Alternate 0 - I2S1_SCK (default)
- *0b01*: Alternate 1 - GPIOA21
- *0b10*: Alternate 2 - SPIS0_SCK
- *0b11*: Alternate 3 - I2S1_SDI

Bits 21:20 - **A14** *(R/W)*

Configure A14 functionality:

- *0b00*: Alternate 0 - TIMER0_CH3 (default)
- *0b01*: Alternate 1 - GPIOA20
- *0b10*: Alternate 2 - TIMER3_CH0
- *0b11*: Alternate 3 -

Bits 19:18 - **B12** *(R/W)*

Configure B12 functionality:

- *0b00*: Alternate 0 - TIMER0_CH2 (default)
- *0b01*: Alternate 1 - GPIOA19
- *0b10*: Alternate 2 - TIMER2_CH0
- *0b11*: Alternate 3 -

Bits 17:16 - **A13** *(R/W)*

Configure A13 functionality:

- *0b00*: Alternate 0 - TIMER0_CH1 (default)
- *0b01*: Alternate 1 - GPIOA18
- *0b10*: Alternate 2 - TIMER1_CH0
- *0b11*: Alternate 3 -

Bits 15:14 - **B11** *(R/W)*

Configure B11 functionality:

- *0b00*: Alternate 0 - TIMER0_CH0 (default)
- *0b01*: Alternate 1 - GPIOA17
- *0b10*: Alternate 2 -
- *0b11*: Alternate 3 -

Bits 13:12 - **D1** *(R/W)*

Configure D1 functionality:

- *0b00*: Alternate 0 - I2C1_SCL (default)
- *0b01*: Alternate 1 - GPIOA16
- *0b10*: Alternate 2 - ORCA_CLK
- *0b11*: Alternate 3 -

Bits 11:10 - **B34** *(R/W)*

Configure B34 functionality:

- *0b00*: Alternate 0 - I2C1_SDA (default)
- *0b01*: Alternate 1 - GPIOA15
- *0b10*: Alternate 2 - TIMER3_CH3
- *0b11*: Alternate 3 -

Bits 9:8 - **A36** *(R/W)*

Configure A36 functionality:

- *0b00*: Alternate 0 - CAM_VSYNC (default)
- *0b01*: Alternate 1 - GPIOA14
- *0b10*: Alternate 2 - TIMER3_CH2
- *0b11*: Alternate 3 -

Bits 7:6 - **A38** *(R/W)*

Configure A38 functionality:

- *0b00*: Alternate 0 - CAM_DATA7 (default)
- *0b01*: Alternate 1 - GPIOA13
- *0b10*: Alternate 2 - TIMER3_CH1
- *0b11*: Alternate 3 -

Bits 5:4 - **B36** *(R/W)*

Configure B36 functionality:

- *0b00*: Alternate 0 - CAM_DATA6 (default)
- *0b01*: Alternate 1 - GPIOA12
- *0b10*: Alternate 2 - TIMER3_CH0
- *0b11*: Alternate 3 -

Bits 3:2 - **A40** *(R/W)*

Configure A40 functionality:

- *0b00*: Alternate 0 - CAM_DATA5 (default)
- *0b01*: Alternate 1 - GPIOA11
- *0b10*: Alternate 2 - TIMER2_CH3
- *0b11*: Alternate 3 -

Bits 1:0 - **B37** *(R/W)*

Configure B37 functionality:

- *0b00*: Alternate 0 - CAM_DATA4 (default)
- *0b01*: Alternate 1 - GPIOA10
- *0b10*: Alternate 2 - TIMER2_CH2
- *0b11*: Alternate 3 -

6.3.4.2.15 Mux config register (pad 32–47) (SAFE_PADFUN2)

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
B23		A26		A24		A25		B22		B9		A16		B15	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A9		B7		A8		B8		A10		B10		A11		D2	

Bits 31:30 - **B23** *(R/W)*

Configure B23 functionality:

- *0b00*: Alternate 0 - I2S0_SDI (default)
- *0b01*: Alternate 1 -
- *0b10*: Alternate 2 -
- *0b11*: Alternate 3 -

Bits 29:28 - **A26** *(R/W)*

Configure A26 functionality:

- *0b00*: Alternate 0 - I2S0_WS (default)
- *0b01*: Alternate 1 -
- *0b10*: Alternate 2 -
- *0b11*: Alternate 3 -

Bits 27:26 - **A24** *(R/W)*

Configure A24 functionality:

- *0b00*: Alternate 0 - I2S0_SCK (default)
- *0b01*: Alternate 1 -
- *0b10*: Alternate 2 -
- *0b11*: Alternate 3 -

Bits 25:24 - **A25** *(R/W)*

Configure A25 functionality:

- *0b00*: Alternate 0 - I2C0_SCL (default)
- *0b01*: Alternate 1 -
- *0b10*: Alternate 2 -
- *0b11*: Alternate 3 -

Bits 23:22 - **B22** *(R/W)*

Configure B22 functionality:

- *0b00*: Alternate 0 - I2C0_SDA (default)
- *0b01*: Alternate 1 -
- *0b10*: Alternate 2 -
- *0b11*: Alternate 3 -

Bits 21:20 - **B9** *(R/W)*

Configure B9 functionality:

- *0b00*: Alternate 0 - SPIS0_SCK (default)
- *0b01*: Alternate 1 -
- *0b10*: Alternate 2 -
- *0b11*: Alternate 3 - HYPER_RWDS

Bits 19:18 - **A16** *(R/W)*

Configure A16 functionality:

- *0b00*: Alternate 0 - SPIS0_SDIO1 (default)
- *0b01*: Alternate 1 - GPIOA31
- *0b10*: Alternate 2 -
- *0b11*: Alternate 3 - HYPER_CSN1

Bits 17:16 - **B15** *(R/W)*

Configure B15 functionality:

- *0b00*: Alternate 0 - SPIS0_SDIO0 (default)
- *0b01*: Alternate 1 - GPIOA30
- *0b10*: Alternate 2 - SPIM1_CS1
- *0b11*: Alternate 3 - HYPER_CSN0

Bits 15:14 - **A9** *(R/W)*

Configure A9 functionality:

- *0b00*: Alternate 0 - SPIS0_CS (default)
- *0b01*: Alternate 1 - GPIOA29
- *0b10*: Alternate 2 - SPIM1_CS0
- *0b11*: Alternate 3 - HYPER_DQ[7]

Bits 13:12 - **B7** *(R/W)*

Configure B7 functionality:

- *0b00*: Alternate 0 - SPIM0_SCK (default)
- *0b01*: Alternate 1 -
- *0b10*: Alternate 2 -
- *0b11*: Alternate 3 - HYPER_DQ[6]

Bits 11:10 - **A8** *(R/W)*

Configure A8 functionality:

- *0b00*: Alternate 0 - SPIM0_CS1 (default)
- *0b01*: Alternate 1 - GPIOA28
- *0b10*: Alternate 2 - SPIS0_SDIO3
- *0b11*: Alternate 3 - HYPER_DQ[5]

Bits 9:8 - **B8** *(R/W)*

Configure B8 functionality:

- *0b00*: Alternate 0 - SPIM0_CS0 (default)
- *0b01*: Alternate 1 -
- *0b10*: Alternate 2 -
- *0b11*: Alternate 3 - HYPER_DQ[4]

Bits 7:6 - **A10** *(R/W)*

Configure A10 functionality:

- *0b00*: Alternate 0 - SPIM0_SDIO3 (default)
- *0b01*: Alternate 1 - GPIOA27
- *0b10*: Alternate 2 - I2C1_SCL
- *0b11*: Alternate 3 - HYPER_DQ[3]

Bits 5:4 - **B10** *(R/W)*

Configure B10 functionality:

- *0b00*: Alternate 0 - SPIM0_SDIO2 (default)
- *0b01*: Alternate 1 - GPIOA26
- *0b10*: Alternate 2 - I2C1_SDA
- *0b11*: Alternate 3 - HYPER_DQ[2]

Bits 3:2 - **A11** *(R/W)*

Configure A11 functionality:

- *0b00*: Alternate 0 - SPIM0_SDIO1 (default)
- *0b01*: Alternate 1 -
- *0b10*: Alternate 2 -
- *0b11*: Alternate 3 - HYPER_DQ[1]

Bits 1:0 - **D2** *(R/W)*

Configure D2 functionality:

- *0b00*: Alternate 0 - SPIM0_SDIO0 (default)
- *0b01*: Alternate 1 -
- *0b10*: Alternate 2 -
- *0b11*: Alternate 3 - HYPER_DQ[0]

6.3.4.2.16 Sleep config register (pad 0–15) (SAFE_SLEEPPADCFG0)

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
VAL_A41	DIR_A41	VAL_B38	DIR_B38	VAL_A42	DIR_A42	VAL_B39	DIR_B39	VAL_A37	DIR_A37	VAL_A43	DIR_A43	VAL_B40	DIR_B40	VAL_A44	DIR_A44
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
VAL_B1	DIR_B1	VAL_A2	DIR_A2	VAL_B2	DIR_B2	VAL_A3	DIR_A3	VAL_B4	DIR_B4	VAL_A5	DIR_A5	VAL_B3	DIR_B3	VAL_A4	DIR_A4

Bit 31 - **VAL_A41** *(R/W)*

Select A41 sleep state

- *0b0*: Input or output low
- *0b1*: Input or output high

Bit 30 - **DIR_A41** *(R/W)*

Select A41 sleep direction

- *0b0*: Input
- *0b1*: Output

Bit 29 - **VAL_B38** *(R/W)*

Select B38 sleep state

- *0b0*: Input or output low
- *0b1*: Input or output high

Bit 28 - **DIR_B38** *(R/W)*

Select B38 sleep direction

- *0b0*: Input
- *0b1*: Output

Bit 27 - **VAL_A42** *(R/W)*

Select A42 sleep state

- *0b0*: Input or output low
- *0b1*: Input or output high

Bit 26 - **DIR_A42** *(R/W)*

Select A42 sleep direction

- *0b0*: Input
- *0b1*: Output

Bit 25 - **VAL_B39** *(R/W)*

Select B39 sleep state

- *0b0*: Input or output low
- *0b1*: Input or output high

Bit 24 - **DIR_B39** *(R/W)*

Select B39 sleep direction

- *0b0*: Input
- *0b1*: Output

Bit 23 - **VAL_A37** *(R/W)*

Select A37 sleep state

- *0b0*: Input or output low
- *0b1*: Input or output high

Bit 22 - **DIR_A37** *(R/W)*

Select A37 sleep direction

- *0b0*: Input
- *0b1*: Output

Bit 21 - **VAL_A43** *(R/W)*

Select A43 sleep state

- *0b0*: Input or output low
- *0b1*: Input or output high

Bit 20 - **DIR_A43** *(R/W)*

Select A43 sleep direction

- *0b0*: Input
- *0b1*: Output

Bit 19 - **VAL_B40** *(R/W)*

Select B40 sleep state

- *0b0*: Input or output low
- *0b1*: Input or output high

Bit 18 - **DIR_B40** *(R/W)*

Select B40 sleep direction

- *0b0*: Input
- *0b1*: Output

Bit 17 - **VAL_A44** *(R/W)*

Select A44 sleep state

- *0b0*: Input or output low
- *0b1*: Input or output high

Bit 16 - **DIR_A44** *(R/W)*

Select A44 sleep direction

- *0b0*: Input
- *0b1*: Output

Bit 15 - **VAL_B1** *(R/W)*

Select B1 sleep state

- *0b0*: Input or output low
- *0b1*: Input or output high

Bit 14 - **DIR_B1** *(R/W)*

Select B1 sleep direction

- *0b0*: Input
- *0b1*: Output

Bit 13 - **VAL_A2** *(R/W)*

Select A2 sleep state

- *0b0*: Input or output low
- *0b1*: Input or output high

Bit 12 - **DIR_A2** *(R/W)*

Select A2 sleep direction

- *0b0*: Input
- *0b1*: Output

Bit 11 - **VAL_B2** *(R/W)*

Select B2 sleep state

- *0b0*: Input or output low
- *0b1*: Input or output high

Bit 10 - **DIR_B2** *(R/W)*

Select B2 sleep direction

- *0b0*: Input
- *0b1*: Output

Bit 9 - **VAL_A3** *(R/W)*

Select A3 sleep state

- *0b0*: Input or output low
- *0b1*: Input or output high

Bit 8 - **DIR_A3** *(R/W)*

Select A3 sleep direction

- *0b0*: Input
- *0b1*: Output

Bit 7 - **VAL_B4** *(R/W)*

Select B4 sleep state

- *0b0*: Input or output low
- *0b1*: Input or output high

Bit 6 - **DIR_B4** *(R/W)*

Select B4 sleep direction

- *0b0*: Input
- *0b1*: Output

Bit 5 - **VAL_A5** *(R/W)*

Select A5 sleep state

- *0b0*: Input or output low
- *0b1*: Input or output high

Bit 4 - **DIR_A5** *(R/W)*

Select A5 sleep direction

- *0b0*: Input
- *0b1*: Output

Bit 3 - **VAL_B3** *(R/W)*

Select B3 sleep state

- *0b0*: Input or output low
- *0b1*: Input or output high

Bit 2 - **DIR_B3** *(R/W)*

Select B3 sleep direction

- *0b0*: Input
- *0b1*: Output

Bit 1 - **VAL_A4** *(R/W)*

Select A4 sleep state

- *0b0*: Input or output low
- *0b1*: Input or output high

Bit 0 - **DIR_A4** *(R/W)*

Select A4 sleep direction

- *0b0*: Input
- *0b1*: Output

6.3.4.2.17 Mux config register (pad 16–31) (SAFE_SLEEPPADCFG1)

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
VAL_A7	DIR_A7	VAL_B6	DIR_B6	VAL_B14	DIR_B14	VAL_A15	DIR_A15	VAL_B13	DIR_B13	VAL_A14	DIR_A14	VAL_B12	DIR_B12	VAL_A13	DIR_A13
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
VAL_B11	DIR_B11	VAL_D1	DIR_D1	VAL_B34	DIR_B34	VAL_A36	DIR_A36	VAL_A38	DIR_A38	VAL_B36	DIR_B36	VAL_A40	DIR_A40	VAL_B37	DIR_B37

Bit 31 - **VAL_A7** *(R/W)*

Select A7 sleep state

- *0b0*: Input or output low
- *0b1*: Input or output high

Bit 30 - **DIR_A7** *(R/W)*

Select A7 sleep direction

- *0b0*: Input
- *0b1*: Output

Bit 29 - **VAL_B6** *(R/W)*

Select B6 sleep state

- *0b0*: Input or output low
- *0b1*: Input or output high

Bit 28 - **DIR_B6** *(R/W)*

Select B6 sleep direction

- *0b0*: Input
- *0b1*: Output

Bit 27 - **VAL_B14** *(R/W)*

Select B14 sleep state

- *0b0*: Input or output low
- *0b1*: Input or output high

Bit 26 - **DIR_B14** *(R/W)*

Select B14 sleep direction

- *0b0*: Input
- *0b1*: Output

Bit 25 - **VAL_A15** *(R/W)*

Select A15 sleep state

- *0b0*: Input or output low
- *0b1*: Input or output high

Bit 24 - **DIR_A15** *(R/W)*

Select A15 sleep direction

- *0b0*: Input
- *0b1*: Output

Bit 23 - **VAL_B13** *(R/W)*

Select B13 sleep state

- *0b0*: Input or output low
- *0b1*: Input or output high

Bit 22 - **DIR_B13** *(R/W)*

Select B13 sleep direction

- *0b0*: Input
- *0b1*: Output

Bit 21 - **VAL_A14** *(R/W)*

Select A14 sleep state

- *0b0*: Input or output low
- *0b1*: Input or output high

Bit 20 - **DIR_A14** *(R/W)*

Select A14 sleep direction

- *0b0*: Input
- *0b1*: Output

Bit 19 - **VAL_B12** *(R/W)*

Select B12 sleep state

- *0b0*: Input or output low
- *0b1*: Input or output high

Bit 18 - **DIR_B12** *(R/W)*

Select B12 sleep direction

- *0b0*: Input
- *0b1*: Output

Bit 17 - **VAL_A13** *(R/W)*

Select A13 sleep state

- *0b0*: Input or output low
- *0b1*: Input or output high

Bit 16 - **DIR_A13** *(R/W)*

Select A13 sleep direction

- *0b0*: Input
- *0b1*: Output

Bit 15 - **VAL_B11** *(R/W)*

Select B11 sleep state

- *0b0*: Input or output low
- *0b1*: Input or output high

Bit 14 - **DIR_B11** *(R/W)*

Select B11 sleep direction

- *0b0*: Input
- *0b1*: Output

Bit 13 - **VAL_D1** *(R/W)*

Select D1 sleep state

- *0b0*: Input or output low
- *0b1*: Input or output high

Bit 12 - **DIR_D1** *(R/W)*

Select D1 sleep direction

- *0b0*: Input
- *0b1*: Output

Bit 11 - **VAL_B34** *(R/W)*

Select B34 sleep state

- *0b0*: Input or output low
- *0b1*: Input or output high

Bit 10 - **DIR_B34** *(R/W)*

Select B34 sleep direction

- *0b0*: Input
- *0b1*: Output

Bit 9 - **VAL_A36** *(R/W)*

Select A36 sleep state

- *0b0*: Input or output low
- *0b1*: Input or output high

Bit 8 - **DIR_A36** *(R/W)*

Select A36 sleep direction

- *0b0*: Input
- *0b1*: Output

Bit 7 - **VAL_A38** *(R/W)*

Select A38 sleep state

- *0b0*: Input or output low
- *0b1*: Input or output high

Bit 6 - **DIR_A38** *(R/W)*

Select A38 sleep direction

- *0b0*: Input
- *0b1*: Output

Bit 5 - **VAL_B36** *(R/W)*

Select B36 sleep state

- *0b0*: Input or output low
- *0b1*: Input or output high

Bit 4 - **DIR_B36** *(R/W)*

Select B36 sleep direction

- *0b0*: Input
- *0b1*: Output

Bit 3 - **VAL_A40** *(R/W)*

Select A40 sleep state

- *0b0*: Input or output low
- *0b1*: Input or output high

Bit 2 - **DIR_A40** *(R/W)*

Select A40 sleep direction

- *0b0*: Input
- *0b1*: Output

Bit 1 - **VAL_B37** *(R/W)*

Select B37 sleep state

- *0b0*: Input or output low
- *0b1*: Input or output high

Bit 0 - **DIR_B37** *(R/W)*

Select B37 sleep direction

- *0b0*: Input
- *0b1*: Output

6.3.4.2.18 Mux config register (pad 32–47) (SAFE_SLEEPPADCFG2)

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
VAL_B23	DIR_B23	VAL_A26	DIR_A26	VAL_A24	DIR_A24	VAL_A25	DIR_A25	VAL_B22	DIR_B22	VAL_B9	DIR_B9	VAL_A16	DIR_A16	VAL_B15	DIR_B15
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
VAL_A9	DIR_A9	VAL_B7	DIR_B7	VAL_A8	DIR_A8	VAL_B8	DIR_B8	VAL_A10	DIR_A10	VAL_B10	DIR_B10	VAL_A11	DIR_A11	VAL_D2	DIR_D2

Bit 31 - **VAL_B23** *(R/W)*

Select B23 sleep state

- *0b0*: Input or output low
- *0b1*: Input or output high

Bit 30 - **DIR_B23** *(R/W)*

Select B23 sleep direction

- *0b0*: Input
- *0b1*: Output

Bit 29 - **VAL_A26** *(R/W)*

Select A26 sleep state

- *0b0*: Input or output low
- *0b1*: Input or output high

Bit 28 - **DIR_A26** *(R/W)*

Select A26 sleep direction

- *0b0*: Input
- *0b1*: Output

Bit 27 - **VAL_A24** *(R/W)*

Select A24 sleep state

- *0b0*: Input or output low
- *0b1*: Input or output high

Bit 26 - **DIR_A24** *(R/W)*

Select A24 sleep direction

- *0b0*: Input
- *0b1*: Output

Bit 25 - **VAL_A25** *(R/W)*

Select A25 sleep state

- *0b0*: Input or output low
- *0b1*: Input or output high

Bit 24 - **DIR_A25** *(R/W)*

Select A25 sleep direction

- *0b0*: Input
- *0b1*: Output

Bit 23 - **VAL_B22** *(R/W)*

Select B22 sleep state

- *0b0*: Input or output low
- *0b1*: Input or output high

Bit 22 - **DIR_B22** *(R/W)*

Select B22 sleep direction

- *0b0*: Input
- *0b1*: Output

Bit 21 - **VAL_B9** *(R/W)*

Select B9 sleep state

- *0b0*: Input or output low
- *0b1*: Input or output high

Bit 20 - **DIR_B9** *(R/W)*

Select B9 sleep direction

- *0b0*: Input
- *0b1*: Output

Bit 19 - **VAL_A16** *(R/W)*

Select A16 sleep state

- *0b0*: Input or output low
- *0b1*: Input or output high

Bit 18 - **DIR_A16** *(R/W)*

Select A16 sleep direction

- *0b0*: Input
- *0b1*: Output

Bit 17 - **VAL_B15** *(R/W)*

Select B15 sleep state

- *0b0*: Input or output low
- *0b1*: Input or output high

Bit 16 - **DIR_B15** *(R/W)*

Select B15 sleep direction

- *0b0*: Input
- *0b1*: Output

Bit 15 - **VAL_A9** *(R/W)*

Select A9 sleep state

- *0b0*: Input or output low
- *0b1*: Input or output high

Bit 14 - **DIR_A9** *(R/W)*

Select A9 sleep direction

- *0b0*: Input
- *0b1*: Output

Bit 13 - **VAL_B7** *(R/W)*

Select B7 sleep state

- *0b0*: Input or output low
- *0b1*: Input or output high

Bit 12 - **DIR_B7** *(R/W)*

Select B7 sleep direction

- *0b0*: Input
- *0b1*: Output

Bit 11 - **VAL_A8** *(R/W)*

Select A8 sleep state

- *0b0*: Input or output low
- *0b1*: Input or output high

Bit 10 - **DIR_A8** *(R/W)*

Select A8 sleep direction

- *0b0*: Input
- *0b1*: Output

Bit 9 - **VAL_B8** *(R/W)*

Select B8 sleep state

- *0b0*: Input or output low
- *0b1*: Input or output high

Bit 8 - **DIR_B8** *(R/W)*

Select B8 sleep direction

- *0b0*: Input
- *0b1*: Output

Bit 7 - **VAL_A10** *(R/W)*

Select A10 sleep state

- *0b0*: Input or output low
- *0b1*: Input or output high

Bit 6 - **DIR_A10** *(R/W)*

Select A10 sleep direction

- *0b0*: Input
- *0b1*: Output

Bit 5 - **VAL_B10** *(R/W)*

Select B10 sleep state

- *0b0*: Input or output low
- *0b1*: Input or output high

Bit 4 - **DIR_B10** *(R/W)*

Select B10 sleep direction

- *0b0*: Input
- *0b1*: Output

Bit 3 - **VAL_A11** *(R/W)*

Select A11 sleep state

- *0b0*: Input or output low
- *0b1*: Input or output high

Bit 2 - **DIR_A11** *(R/W)*

Select A11 sleep direction

- *0b0*: Input
- *0b1*: Output

Bit 1 - **VAL_D2** *(R/W)*

Select D2 sleep state

- *0b0*: Input or output low
- *0b1*: Input or output high

Bit 0 - **DIR_D2** *(R/W)*

Select D2 sleep direction

- *0b0*: Input
- *0b1*: Output

6.3.4.2.19 Enable Sleep mode for pads (SAFE_PADSLEEP)

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)*

Enable pad sleep mode:

0b0: disable

0b1: enable

6.3.4.2.20 Function register (pad 0 to 3) (SAFE_PADCFG0)

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						DR_B4	PULL_B4	Reserved						DR_A5	PULL_A5
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						DR_B3	PULL_B3	Reserved						DR_A4	PULL_A4

Bit 25 - **DR_B4** *(R/W)*

Select B4 drive strength

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 24 - **PULL_B4** *(R/W)*

Select B4 pull activation

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

Bit 17 - **DR_A5** *(R/W)*

Select A5 drive strength

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 16 - **PULL_A5** *(R/W)*

Select A5 pull activation

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

Bit 9 - **DR_B3** *(R/W)*

Select B3 drive strength

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 8 - **PULL_B3** *(R/W)*

Select B3 pull activation

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

Bit 1 - **DR_A4** *(R/W)*

Select A4 drive strength

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 0 - **PULL_A4** *(R/W)*

Select A4 pull activation

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

6.3.4.2.21 Function register (pad 4 to 7) (SAFE_PADCFG1)

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						DR_B1	PULL_B1	Reserved						DR_A2	PULL_A2
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						DR_B2	PULL_B2	Reserved						DR_A3	PULL_A3

Bit 25 - **DR_B1** *(R/W)*

Select B1 drive strength

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 24 - **PULL_B1** *(R/W)*

Select B1 pull activation

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

Bit 17 - **DR_A2** *(R/W)*

Select A2 drive strength

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 16 - **PULL_A2** *(R/W)*

Select A2 pull activation

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

Bit 9 - **DR_B2** *(R/W)*

Select B2 drive strength

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 8 - **PULL_B2** *(R/W)*

Select B2 pull activation

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

Bit 1 - **DR_A3** *(R/W)*

Select A3 drive strength

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 0 - **PULL_A3** *(R/W)*

Select A3 pull activation

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

6.3.4.2.22 Function register (pad 8 to 11) (SAFE_PADCFG2)

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						DR_A37	PULL_A 37	Reserved						DR_A43	PULL_A 43

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						DR_B40	PULL_B 40	Reserved						DR_A44	PULL_A 44

Bit 25 - **DR_A37** *(R/W)*

Select A37 drive strength

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 24 - **PULL_A37** *(R/W)*

Select A37 pull activation

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

Bit 17 - **DR_A43** *(R/W)*

Select A43 drive strength

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 16 - **PULL_A43** *(R/W)*

Select A43 pull activation

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

Bit 9 - **DR_B40** *(R/W)*

Select B40 drive strength

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 8 - **PULL_B40** *(R/W)*

Select B40 pull activation

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

Bit 1 - **DR_A44** *(R/W)*

Select A44 drive strength

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 0 - **PULL_A44** *(R/W)*

Select A44 pull activation

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

6.3.4.2.23 Function register (pad 12 to 15) (SAFE_PADCFG3)

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						DR_A41	PULL_A 41	Reserved						DR_B38	PULL_B 38
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						DR_A42	PULL_A 42	Reserved						DR_B39	PULL_B 39

Bit 25 - **DR_A41** *(R/W)*

Select A41 drive strength

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 24 - **PULL_A41** *(R/W)*

Select A41 pull activation

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

Bit 17 - **DR_B38** *(R/W)*

Select B38 drive strength

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 16 - **PULL_B38** *(R/W)*

Select B38 pull activation

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

Bit 9 - **DR_A42** *(R/W)*

Select A42 drive strength

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 8 - **PULL_A42** *(R/W)*

Select A42 pull activation

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

Bit 1 - **DR_B39** *(R/W)*

Select B39 drive strength

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 0 - **PULL_B39** *(R/W)*

Select B39 pull activation

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

6.3.4.2.24 Function register (pad 16 to 19) (SAFE_PADCFG4)

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						DR_A38	PULL_A 38	Reserved						DR_B36	PULL_B 36
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						DR_A40	PULL_A 40	Reserved						DR_B37	PULL_B 37

Bit 25 - **DR_A38** *(R/W)*

Select A38 drive strength

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 24 - **PULL_A38** *(R/W)*

Select A38 pull activation

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

Bit 17 - **DR_B36** *(R/W)*

Select B36 drive strength

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 16 - **PULL_B36** *(R/W)*

Select B36 pull activation

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

Bit 9 - **DR_A40** *(R/W)*

Select A40 drive strength

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 8 - **PULL_A40** *(R/W)*

Select A40 pull activation

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

Bit 1 - **DR_B37** *(R/W)*

Select B37 drive strength

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 0 - **PULL_B37** *(R/W)*

Select B37 pull activation

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

6.3.4.2.25 Function register (pad 20 to 23) (SAFE_PADCFG5)

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						DR_B11	PULL_B11	Reserved						DR_D1	PULL_D1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						DR_B34	PULL_B34	Reserved						DR_A36	PULL_A36

Bit 25 - **DR_B11** *(R/W)*

Select B11 drive strength

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 24 - **PULL_B11** *(R/W)*

Select B11 pull activation

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

Bit 17 - **DR_D1** *(R/W)*

Select D1 drive strength

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 16 - **PULL_D1** *(R/W)*

Select D1 pull activation

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

Bit 9 - **DR_B34** *(R/W)*

Select B34 drive strength

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 8 - **PULL_B34** *(R/W)*

Select B34 pull activation

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

Bit 1 - **DR_A36** *(R/W)*

Select A36 drive strength

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 0 - **PULL_A36** *(R/W)*

Select A36 pull activation

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

6.3.4.2.26 Function register (pad 24 to 27) (SAFE_PADCFG6)

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						DR_B13	PULL_B13	Reserved						DR_A14	PULL_A14
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						DR_B12	PULL_B12	Reserved						DR_A13	PULL_A13

Bit 25 - **DR_B13** *(R/W)*

Select B13 drive strength

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 24 - **PULL_B13** *(R/W)*

Select B13 pull activation

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

Bit 17 - **DR_A14** *(R/W)*

Select A14 drive strength

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 16 - **PULL_A14** *(R/W)*

Select A14 pull activation

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

Bit 9 - **DR_B12** *(R/W)*

Select B12 drive strength

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 8 - **PULL_B12** *(R/W)*

Select B12 pull activation

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

Bit 1 - **DR_A13** *(R/W)*

Select A13 drive strength

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 0 - **PULL_A13** *(R/W)*

Select A13 pull activation

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

6.3.4.2.27 Function register (pad 28 to 31) (SAFE_PADCFG7)

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	DR_A7	PULL_A7	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	DR_B6	PULL_B6

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	DR_B14	PULL_B14	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	DR_A15	PULL_A15

Bit 25 - **DR_A7** *(R/W)*

Select A7 drive strength

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 24 - **PULL_A7** *(R/W)*

Select A7 pull activation

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

Bit 17 - **DR_B6** *(R/W)*

Select B6 drive strength

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 16 - **PULL_B6** *(R/W)*

Select B6 pull activation

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

Bit 9 - **DR_B14** *(R/W)*

Select B14 drive strength

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 8 - **PULL_B14** *(R/W)*

Select B14 pull activation

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

Bit 1 - **DR_A15** *(R/W)*

Select A15 drive strength

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 0 - **PULL_A15** *(R/W)*

Select A15 pull activation

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

6.3.4.2.28 Function register (pad 32 to 35) (SAFE_PADCFG8)

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						DR_A10	PULL_A 10	Reserved						DR_B10	PULL_B 10
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						DR_A11	PULL_A 11	Reserved						DR_D2	PULL_D 2

Bit 25 - **DR_A10** *(R/W)*

Select A10 drive strength

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 24 - **PULL_A10** *(R/W)*

Select A10 pull activation

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

Bit 17 - **DR_B10** *(R/W)*

Select B10 drive strength

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 16 - **PULL_B10** *(R/W)*

Select B10 pull activation

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

Bit 9 - **DR_A11** *(R/W)*

Select A11 drive strength

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 8 - **PULL_A11** *(R/W)*

Select A11 pull activation

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

Bit 1 - **DR_D2** *(R/W)*

Select D2 drive strength

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 0 - **PULL_D2** *(R/W)*

Select D2 pull activation

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

6.3.4.2.29 Function register (pad 36 to 39) (SAFE_PADCFG9)

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						DR_A9	PULL_A9	Reserved						DR_B7	PULL_B7

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						DR_A8	PULL_A8	Reserved						DR_B8	PULL_B8

Bit 25 - **DR_A9** *(R/W)*

Select A9 drive strength

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 24 - **PULL_A9** *(R/W)*

Select A9 pull activation

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

Bit 17 - **DR_B7** *(R/W)*

Select B7 drive strength

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 16 - **PULL_B7** *(R/W)*

Select B7 pull activation

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

Bit 9 - **DR_A8** *(R/W)*

Select A8 drive strength

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 8 - **PULL_A8** *(R/W)*

Select A8 pull activation

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

Bit 1 - **DR_B8** *(R/W)*

Select B8 drive strength

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 0 - **PULL_B8** *(R/W)*

Select B8 pull activation

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

6.3.4.2.30 Function register (pad 40 to 43) (SAFE_PADCFG10)

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						DR_B22	PULL_B 22	Reserved						DR_B9	PULL_B 9
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						DR_A16	PULL_A 16	Reserved						DR_B15	PULL_B 15

Bit 25 - **DR_B22** *(R/W)*

Select B22 drive strength

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 24 - **PULL_B22** *(R/W)*

Select B22 pull activation

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

Bit 17 - **DR_B9** *(R/W)*

Select B9 drive strength

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 16 - **PULL_B9** *(R/W)*

Select B9 pull activation

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

Bit 9 - **DR_A16** *(R/W)*

Select A16 drive strength

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 8 - **PULL_A16** *(R/W)*

Select A16 pull activation

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

Bit 1 - **DR_B15** *(R/W)*

Select B15 drive strength

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 0 - **PULL_B15** *(R/W)*

Select B15 pull activation

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

6.3.4.2.31 Function register (pad 44 to 47) (SAFE_PADCFG11)

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						DR_B23	PULL_B 23	Reserved						DR_A26	PULL_A 26
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						DR_A24	PULL_A 24	Reserved						DR_A25	PULL_A 25

Bit 25 - **DR_B23** *(R/W)*

Select B23 drive strength

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 24 - **PULL_B23** *(R/W)*

Select B23 pull activation

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

Bit 17 - **DR_A26** *(R/W)*

Select A26 drive strength

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 16 - **PULL_A26** *(R/W)*

Select A26 pull activation

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

Bit 9 - **DR_A24** *(R/W)*

Select A24 drive strength

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 8 - **PULL_A24** *(R/W)*

Select A24 pull activation

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

Bit 1 - **DR_A25** *(R/W)*

Select A25 drive strength

- *0b0*: low drive strength
- *0b1*: high drive strength

Bit 0 - **PULL_A25** *(R/W)*

Select A25 pull activation

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

6.3.4.2.32 GPIO power domain pad input isolation register (REG_GPIO_ISO)

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															ISO

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

Configuration of GPIO domain pads isolation:

- *0b0*: not isolated
- *0b1*: isolated

6.3.4.2.33 CAM power domain pad input isolation register (REG_CAM_ISO)

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															ISO

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

Configuration of CAM domain pads isolation:

- *0b0*: not isolated
- *0b1*: isolated

6.3.4.2.34 LVDS power domain pad input isolation register (REG_LVDS_ISO)

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															ISO

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

Configuration of LVDS domain pads isolation:

- *0b0*: not isolated
- *0b1*: isolated

6.3.5 Advanced timer

ADV_ TIMER component manages the following features:

- 4 advanced timers with 4 output signal channels each. Provides PWM generation functionality
- multiple trigger input sources:
  - output signal channels of all timers
  - 32 GPIOs
  - reference clock at 32kHz
  - SoC FLL clock
- configurable input trigger modes
- configurable prescaler for each timer
- configurable counting mode for each timer
- configurable channel threshold action for each timer
- 4 configurable output events
- configurable clock gating of each timer

6.3.5.1 Advanced timer registers

Name	Address	Size	Type	Access	Default	Description
T0_CMD	0x1A105000	32	Config	R/W	*0x0000*	ADV_TIMER0 command register.
T0_CONFIG	0x1A105004	32	Config	R/W	*0x0000*	ADV_TIMER0 configuration register.

Name	Address	Size	Type	Access	Default	Description
T0_THRESHOLD	0x1A105008	32	Config	R/W	*0x0000*	ADV_TIMER0 threshold configuration register.
T0_TH_CHANNEL0	0x1A10500C	32	Config	R/W	*0x0000*	ADV_TIMER0 channel 0 threshold configuration register.
T0_TH_CHANNEL1	0x1A105010	32	Config	R/W	*0x0000*	ADV_TIMER0 channel 1 threshold configuration register.
T0_TH_CHANNEL2	0x1A105014	32	Config	R/W	*0x0000*	ADV_TIMER0 channel 2 threshold configuration register.
T0_TH_CHANNEL3	0x1A105018	32	Config	R/W	*0x0000*	ADV_TIMER0 channel 3 threshold configuration register.
T0_COUNTER	0x1A10502C	32	Status	R	*0x0000*	ADV_TIMER0 counter register.
T1_CMD	0x1A105040	32	Config	R/W	*0x0000*	ADV_TIMER1 command register.
T1_CONFIG	0x1A105044	32	Config	R/W	*0x0000*	ADV_TIMER1 configuration register.
T1_THRESHOLD	0x1A105048	32	Config	R/W	*0x0000*	ADV_TIMER1 threshold configuration register.
T1_TH_CHANNEL0	0x1A10504C	32	Config	R/W	*0x0000*	ADV_TIMER1 channel 0 threshold configuration register.
T1_TH_CHANNEL1	0x1A105050	32	Config	R/W	*0x0000*	ADV_TIMER1 channel 1 threshold configuration register.
T1_TH_CHANNEL2	0x1A105054	32	Config	R/W	*0x0000*	ADV_TIMER1 channel 2 threshold configuration register.
T1_TH_CHANNEL3	0x1A105058	32	Config	R/W	*0x0000*	ADV_TIMER1 channel 3 threshold configuration register.
T1_COUNTER	0x1A10506C	32	Status	R	*0x0000*	ADV_TIMER1 counter register.
T2_CMD	0x1A105080	32	Config	R/W	*0x0000*	ADV_TIMER2 command register.
T2_CONFIG	0x1A105084	32	Config	R/W	*0x0000*	ADV_TIMER2 configuration register.
T2_THRESHOLD	0x1A105088	32	Config	R/W	*0x0000*	ADV_TIMER2 threshold configuration register.
T2_TH_CHANNEL0	0x1A10508C	32	Config	R/W	*0x0000*	ADV_TIMER2 channel 0 threshold configuration register.
T2_TH_CHANNEL1	0x1A105090	32	Config	R/W	*0x0000*	ADV_TIMER2 channel 1 threshold configuration register.
T2_TH_CHANNEL2	0x1A105094	32	Config	R/W	*0x0000*	ADV_TIMER2 channel 2 threshold configuration register.
T2_TH_CHANNEL3	0x1A105098	32	Config	R/W	*0x0000*	ADV_TIMER2 channel 3 threshold configuration register.
T2_COUNTER	0x1A1050AC	32	Status	R	*0x0000*	ADV_TIMER2 counter register.
T3_CMD	0x1A1050C0	32	Config	R/W	*0x0000*	ADV_TIMER3 command register.
T3_CONFIG	0x1A1050C4	32	Config	R/W	*0x0000*	ADV_TIMER3 configuration register.
T3_THRESHOLD	0x1A1050C8	32	Config	R/W	*0x0000*	ADV_TIMER3 threshold configuration register.
T3_TH_CHANNEL0	0x1A1050CC	32	Config	R/W	*0x0000*	ADV_TIMER3 channel 0 threshold configuration register.
T3_TH_CHANNEL1	0x1A1050D0	32	Config	R/W	*0x0000*	ADV_TIMER3 channel 1 threshold configuration register.
T3_TH_CHANNEL2	0x1A1050D4	32	Config	R/W	*0x0000*	ADV_TIMER3 channel 2 threshold configuration register.
T3_TH_CHANNEL3	0x1A1050D8	32	Config	R/W	*0x0000*	ADV_TIMER3 channel 3 threshold configuration register.
T3_COUNTER	0x1A1050EC	32	Status	R	*0x0000*	ADV_TIMER3 counter register.
EVENT_CFG	0x1A105100	32	Config	R/W	*0x0000*	ADV_TIMERS events configuration register.
CG	0x1A105104	32	Config	R/W	*0x0000*	ADV_TIMERS channels clock gating configuration register.

Table 42. Advanced timer registers table

6.3.5.2 Advanced timer registers details

6.3.5.2.1 ADV_TIMER0 command register. (T0_CMD)

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											ARM	RESET	UPDATE	STOP	START

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

ADV_TIMER0 arm command bitfield.

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

ADV_TIMER0 reset command bitfield.

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

ADV_TIMER0 update command bitfield.

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

ADV_TIMER0 stop command bitfield.

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

ADV_TIMER0 start command bitfield.

6.3.5.2.2 ADV_TIMER0 configuration register. (T0_CONFIG)

Reset value: 0x0000

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

Bits 23:16 - **PRESC** *(R/W)*

ADV_TIMER0 prescaler value configuration bitfield.

Bit 12 - **UPDOWNSEL** *(R/W)*

ADV_TIMER0 center-aligned mode configuration bitfield:

- *0b0*: The counter counts up and down alternatively.
- *0b1*: The counter counts up and resets to 0 when reach threshold.

Bit 11 - **CLKSEL** *(R/W)*

ADV_TIMER0 clock source configuration bitfield:

- *0b0*: FLL
- *0b1*: reference clock at 32kHz

Bits 10:8 - **MODE** *(R/W)*

ADV_TIMER0 trigger mode configuration bitfield:

- *0b000*: trigger event at each clock cycle.
- *0b001*: trigger event if input source is 0
- *0b010*: trigger event if input source is 1
- *0b011*: trigger event on input source rising edge
- *0b100*: trigger event on input source falling edge
- *0b101*: trigger event on input source falling or rising edge
- *0b110*: trigger event on input source rising edge when armed
- *0b111*: trigger event on input source falling edge when armed

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

ADV_TIMER0 input source configuration bitfield:

- 0–31: GPIO[0] to GPIO[31]
- 32–35: Channel 0 to 3 of ADV_TIMER0
- 36–39: Channel 0 to 3 of ADV_TIMER1
- 40–43: Channel 0 to 3 of ADV_TIMER2
- 44–47: Channel 0 to 3 of ADV_TIMER3

6.3.5.2.3 ADV_TIMER0 threshold configuration register. (T0_THRESHOLD)

Reset value: 0x0000

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

Bits 31:16 - **TH_HI** *(R/W)*

ADV_TIMER0 threshold high part configuration bitfield. It defines end counter value.

Bits 15:0 - **TH_LO** *(R/W)*

ADV_TIMER0 threshold low part configuration bitfield. It defines start counter value.

6.3.5.2.4 ADV_TIMER0 channel 0 threshold configuration register. (T0_TH_CHANNEL0)

Reset value: 0x0000

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

Bits 18:16 - **MODE** *(R/W)*

ADV_TIMER0 channel 0 threshold match action on channel output signal configuration bitfield:

- *0b000*: set.
- *0b001*: toggle then next threshold match action is clear.
- *0b010*: set then next threshold match action is clear.
- *0b011*: toggle.
- *0b100*: clear.
- *0b101*: toggle then next threshold match action is set.
- *0b110*: clear then next threshold match action is set.

Bits 15:0 - **TH** *(R/W)*

ADV_TIMER0 channel 0 threshold configuration bitfield.

6.3.5.2.5 ADV_TIMER0 channel 1 threshold configuration register. (T0_TH_CHANNEL1)

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved													MODE		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TH															

Bits 18:16 - **MODE** *(R/W)*

ADV_TIMER0 channel 1 threshold match action on channel output signal configuration bitfield:

- *0b000*: set.
- *0b001*: toggle then next threshold match action is clear.
- *0b010*: set then next threshold match action is clear.
- *0b011*: toggle.
- *0b100*: clear.
- *0b101*: toggle then next threshold match action is set.
- *0b110*: clear then next threshold match action is set.

Bits 15:0 - **TH** *(R/W)*

ADV_TIMER0 channel 1 threshold configuration bitfield.

6.3.5.2.6 ADV_TIMER0 channel 2 threshold configuration register. (T0_TH_CHANNEL2)

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved													MODE		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TH															

Bits 18:16 - **MODE** *(R/W)*

ADV_TIMER0 channel 2 threshold match action on channel output signal configuration bitfield:

- *0b000*: set.
- *0b001*: toggle then next threshold match action is clear.
- *0b010*: set then next threshold match action is clear.
- *0b011*: toggle.
- *0b100*: clear.
- *0b101*: toggle then next threshold match action is set.
- *0b110*: clear then next threshold match action is set.

Bits 15:0 - **TH** *(R/W)*

ADV_TIMER0 channel 2 threshold configuration bitfield.

6.3.5.2.7 ADV_TIMER0 channel 3 threshold configuration register. (T0_TH_CHANNEL3)

Reset value: 0x0000

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

Bits 18:16 - **MODE** *(R/W)*

ADV_TIMER0 channel 3 threshold match action on channel output signal configuration bitfield:

- *0b000*: set.
- *0b001*: toggle then next threshold match action is clear.
- *0b010*: set then next threshold match action is clear.
- *0b011*: toggle.
- *0b100*: clear.
- *0b101*: toggle then next threshold match action is set.
- *0b110*: clear then next threshold match action is set.

Bits 15:0 - **TH** *(R/W)*

ADV_TIMER0 channel 3 threshold configuration bitfield.

6.3.5.2.8 ADV_TIMER1 command register. (T1_CMD)

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											ARM	RESET	UPDATE	STOP	START

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

ADV_TIMER1 arm command bitfield.

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

ADV_TIMER1 reset command bitfield.

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

ADV_TIMER1 update command bitfield.

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

ADV_TIMER1 stop command bitfield.

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

ADV_TIMER1 start command bitfield.

6.3.5.2.9 ADV_TIMER1 configuration register. (T1_CONFIG)

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								PRESC							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			UPDOW NSEL	CLKSEL	MODE			INSEL							

Bits 23:16 - **PRESC** *(R/W)*

ADV_TIMER1 prescaler value configuration bitfield.

Bit 12 - **UPDOWNSEL** *(R/W)*

ADV_TIMER1 center-aligned mode configuration bitfield:

- *0b0*: The counter counts up and down alternatively.
- *0b1*: The counter counts up and resets to 0 when reach threshold.

Bit 11 - **CLKSEL** *(R/W)*

ADV_TIMER1 clock source configuration bitfield:

- *0b0*: FLL
- *0b1*: reference clock at 32kHz

Bits 10:8 - **MODE** *(R/W)*

ADV_TIMER1 trigger mode configuration bitfield:

- *0b000*: trigger event at each clock cycle.
- *0b001*: trigger event if input source is 0
- *0b010*: trigger event if input source is 1
- *0b011*: trigger event on input source rising edge
- *0b100*: trigger event on input source falling edge
- *0b101*: trigger event on input source falling or rising edge
- *0b110*: trigger event on input source rising edge when armed
- *0b111*: trigger event on input source falling edge when armed

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

ADV_TIMER1 input source configuration bitfield:

- 0–31: GPIO[0] to GPIO[31]
- 32–35: Channel 0 to 3 of ADV_TIMER0
- 36–39: Channel 0 to 3 of ADV_TIMER1
- 40–43: Channel 0 to 3 of ADV_TIMER2
- 44–47: Channel 0 to 3 of ADV_TIMER3

6.3.5.2.10 ADV_TIMER1 threshold configuration register. (T1_THRESHOLD)

Reset value: 0x0000

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

Bits 31:16 - **TH_HI** *(R/W)*

ADV_TIMER1 threshold high part configuration bitfield. It defines end counter value.

Bits 15:0 - **TH_LO** *(R/W)*

ADV_TIMER1 threshold low part configuration bitfield. It defines start counter value.

6.3.5.2.11 ADV_TIMER1 channel 0 threshold configuration register. (T1_TH_CHANNEL0)

Reset value: 0x0000

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

Bits 18:16 - **MODE** *(R/W)*

ADV_TIMER1 channel 0 threshold match action on channel output signal configuration bitfield:

- *0b000*: set.
- *0b001*: toggle then next threshold match action is clear.
- *0b010*: set then next threshold match action is clear.
- *0b011*: toggle.
- *0b100*: clear.
- *0b101*: toggle then next threshold match action is set.
- *0b110*: clear then next threshold match action is set.

Bits 15:0 - **TH** *(R/W)*

ADV_TIMER1 channel 0 threshold configuration bitfield.

6.3.5.2.12 ADV_TIMER1 channel 1 threshold configuration register. (T1_TH_CHANNEL1)

Reset value: 0x0000

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

Bits 18:16 - **MODE** *(R/W)*

ADV_TIMER1 channel 1 threshold match action on channel output signal configuration bitfield:

- *0b000*: set.
- *0b001*: toggle then next threshold match action is clear.
- *0b010*: set then next threshold match action is clear.
- *0b011*: toggle.
- *0b100*: clear.
- *0b101*: toggle then next threshold match action is set.
- *0b110*: clear then next threshold match action is set.

Bits 15:0 - **TH** *(R/W)*

ADV_TIMER1 channel 1 threshold configuration bitfield.

6.3.5.2.13 ADV_TIMER1 channel 2 threshold configuration register. (T1_TH_CHANNEL2)

Reset value: 0x0000

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

Bits 18:16 - **MODE** *(R/W)*

ADV_TIMER1 channel 2 threshold match action on channel output signal configuration bitfield:

- *0b000*: set.
- *0b001*: toggle then next threshold match action is clear.
- *0b010*: set then next threshold match action is clear.
- *0b011*: toggle.
- *0b100*: clear.
- *0b101*: toggle then next threshold match action is set.
- *0b110*: clear then next threshold match action is set.

Bits 15:0 - **TH** *(R/W)*

ADV_TIMER1 channel 2 threshold configuration bitfield.

6.3.5.2.14 ADV_TIMER1 channel 3 threshold configuration register. (T1_TH_CHANNEL3)

Reset value: 0x0000

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

Bits 18:16 - **MODE** *(R/W)*

ADV_TIMER1 channel 3 threshold match action on channel output signal configuration bitfield:

- *0b000*: set.
- *0b001*: toggle then next threshold match action is clear.
- *0b010*: set then next threshold match action is clear.
- *0b011*: toggle.
- *0b100*: clear.
- *0b101*: toggle then next threshold match action is set.
- *0b110*: clear then next threshold match action is set.

Bits 15:0 - **TH** *(R/W)*

ADV_TIMER1 channel 3 threshold configuration bitfield.

6.3.5.2.15 ADV_TIMER2 command register. (T2_CMD)

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											ARM	RESET	UPDATE	STOP	START

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

ADV_TIMER2 arm command bitfield.

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

ADV_TIMER2 reset command bitfield.

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

ADV_TIMER2 update command bitfield.

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

ADV_TIMER2 stop command bitfield.

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

ADV_TIMER2 start command bitfield.

6.3.5.2.16 ADV_TIMER2 configuration register. (T2_CONFIG)

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								PRESC							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			UPDOWNSEL	CLKSEL	MODE			INSEL							

Bits 23:16 - **PRESC** *(R/W)*

ADV_TIMER2 prescaler value configuration bitfield.

Bit 12 - **UPDOWNSEL** *(R/W)*

ADV_TIMER2 center-aligned mode configuration bitfield:

- *0b0*: The counter counts up and down alternatively.
- *0b1*: The counter counts up and resets to 0 when reach threshold.

Bit 11 - **CLKSEL** *(R/W)*

ADV_TIMER2 clock source configuration bitfield:

- *0b0*: FLL
- *0b1*: reference clock at 32kHz

Bits 10:8 - **MODE** *(R/W)*

ADV_TIMER2 trigger mode configuration bitfield:

- *0b000*: trigger event at each clock cycle.
- *0b001*: trigger event if input source is 0
- *0b010*: trigger event if input source is 1
- *0b011*: trigger event on input source rising edge
- *0b100*: trigger event on input source falling edge
- *0b101*: trigger event on input source falling or rising edge
- *0b110*: trigger event on input source rising edge when armed
- *0b111*: trigger event on input source falling edge when armed

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

ADV_TIMER2 input source configuration bitfield:

- 0–31: GPIO[0] to GPIO[31]
- 32–35: Channel 0 to 3 of ADV_TIMER0
- 36–39: Channel 0 to 3 of ADV_TIMER1
- 40–43: Channel 0 to 3 of ADV_TIMER2
- 44–47: Channel 0 to 3 of ADV_TIMER3

6.3.5.2.17 ADV_TIMER2 threshold configuration register. (T2_THRESHOLD)

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TH_HI															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TH_LO															

Bits 31:16 - **TH_HI** *(R/W)*

ADV_TIMER2 threshold high part configuration bitfield. It defines end counter value.

Bits 15:0 - **TH_LO** *(R/W)*

ADV_TIMER2 threshold low part configuration bitfield. It defines start counter value.

6.3.5.2.18 ADV_TIMER2 channel 0 threshold configuration register. (T2_TH_CHANNEL0)

Reset value: 0x0000

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

Bits 18:16 - **MODE** *(R/W)*

ADV_TIMER2 channel 0 threshold match action on channel output signal configuration bitfield:

- *0b000*: set.
- *0b001*: toggle then next threshold match action is clear.
- *0b010*: set then next threshold match action is clear.
- *0b011*: toggle.
- *0b100*: clear.
- *0b101*: toggle then next threshold match action is set.
- *0b110*: clear then next threshold match action is set.

Bits 15:0 - **TH** *(R/W)*

ADV_TIMER2 channel 0 threshold configuration bitfield.

6.3.5.2.19 ADV_TIMER2 channel 1 threshold configuration register. (T2_TH_CHANNEL1)

Reset value: 0x0000

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

Bits 18:16 - **MODE** *(R/W)*

ADV_TIMER2 channel 1 threshold match action on channel output signal configuration bitfield:

- *0b000*: set.
- *0b001*: toggle then next threshold match action is clear.
- *0b010*: set then next threshold match action is clear.
- *0b011*: toggle.
- *0b100*: clear.
- *0b101*: toggle then next threshold match action is set.
- *0b110*: clear then next threshold match action is set.

Bits 15:0 - **TH** *(R/W)*

ADV_TIMER2 channel 1 threshold configuration bitfield.

6.3.5.2.20 ADV_TIMER2 channel 2 threshold configuration register. (T2_TH_CHANNEL2)

Reset value: 0x0000

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

Bits 18:16 - **MODE** *(R/W)*

ADV_TIMER2 channel 2 threshold match action on channel output signal configuration bitfield:

- *0b000*: set.
- *0b001*: toggle then next threshold match action is clear.
- *0b010*: set then next threshold match action is clear.
- *0b011*: toggle.
- *0b100*: clear.
- *0b101*: toggle then next threshold match action is set.
- *0b110*: clear then next threshold match action is set.

Bits 15:0 - **TH** *(R/W)*

ADV_TIMER2 channel 2 threshold configuration bitfield.

6.3.5.2.21 ADV_TIMER2 channel 3 threshold configuration register. (T2_TH_CHANNEL3)

Reset value: 0x0000

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

Bits 18:16 - **MODE** *(R/W)*

ADV_TIMER2 channel 3 threshold match action on channel output signal configuration bitfield:

- *0b000*: set.
- *0b001*: toggle then next threshold match action is clear.
- *0b010*: set then next threshold match action is clear.
- *0b011*: toggle.
- *0b100*: clear.
- *0b101*: toggle then next threshold match action is set.
- *0b110*: clear then next threshold match action is set.

Bits 15:0 - **TH** *(R/W)*

ADV_TIMER2 channel 3 threshold configuration bitfield.

6.3.5.2.22 ADV_TIMER3 command register. (T3_CMD)

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											ARM	RESET	UPDATE	STOP	START

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

ADV_TIMER3 arm command bitfield.

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

ADV_TIMER3 reset command bitfield.

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

ADV_TIMER3 update command bitfield.

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

ADV_TIMER3 stop command bitfield.

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

ADV_TIMER3 start command bitfield.

6.3.5.2.23 ADV_TIMER3 configuration register. (T3_CONFIG)

Reset value: 0x0000

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

Bits 23:16 - **PRESC** *(R/W)*

ADV_TIMER3 prescaler value configuration bitfield.

Bit 12 - **UPDOWNSEL** *(R/W)*

ADV_TIMER3 center-aligned mode configuration bitfield:

- *0b0*: The counter counts up and down alternatively.
- *0b1*: The counter counts up and resets to 0 when reach threshold.

Bit 11 - **CLKSEL** *(R/W)*

ADV_TIMER3 clock source configuration bitfield:

- *0b0*: FLL
- *0b1*: reference clock at 32kHz

Bits 10:8 - **MODE** *(R/W)*

ADV_TIMER3 trigger mode configuration bitfield:

- *0b000*: trigger event at each clock cycle.
- *0b001*: trigger event if input source is 0
- *0b010*: trigger event if input source is 1
- *0b011*: trigger event on input source rising edge
- *0b100*: trigger event on input source falling edge
- *0b101*: trigger event on input source falling or rising edge
- *0b110*: trigger event on input source rising edge when armed
- *0b111*: trigger event on input source falling edge when armed

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

ADV_TIMER3 input source configuration bitfield:

- 0–31: GPIO[0] to GPIO[31]
- 32–35: Channel 0 to 3 of ADV_TIMER0
- 36–39: Channel 0 to 3 of ADV_TIMER1
- 40–43: Channel 0 to 3 of ADV_TIMER2
- 44–47: Channel 0 to 3 of ADV_TIMER3

6.3.5.2.24 ADV_TIMER3 threshold configuration register. (T3_THRESHOLD)

Reset value: 0x0000

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

Bits 31:16 - **TH_HI** *(R/W)*

ADV_TIMER3 threshold high part configuration bitfield. It defines end counter value.

Bits 15:0 - **TH_LO** *(R/W)*

ADV_TIMER3 threshold low part configuration bitfield. It defines start counter value.

6.3.5.2.25 ADV_TIMER3 channel 0 threshold configuration register. (T3_TH_CHANNEL0)

Reset value: 0x0000

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

Bits 18:16 - **MODE** *(R/W)*

ADV_TIMER3 channel 0 threshold match action on channel output signal configuration bitfield:

- *0b000*: set.
- *0b001*: toggle then next threshold match action is clear.
- *0b010*: set then next threshold match action is clear.
- *0b011*: toggle.
- *0b100*: clear.
- *0b101*: toggle then next threshold match action is set.
- *0b110*: clear then next threshold match action is set.

Bits 15:0 - **TH** *(R/W)*

ADV_TIMER3 channel 0 threshold configuration bitfield.

6.3.5.2.26 ADV_TIMER3 channel 1 threshold configuration register. (T3_TH_CHANNEL1)

Reset value: 0x0000

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

Bits 18:16 - **MODE** *(R/W)*

ADV_TIMER3 channel 1 threshold match action on channel output signal configuration bitfield:

- *0b000*: set.
- *0b001*: toggle then next threshold match action is clear.
- *0b010*: set then next threshold match action is clear.
- *0b011*: toggle.
- *0b100*: clear.
- *0b101*: toggle then next threshold match action is set.
- *0b110*: clear then next threshold match action is set.

Bits 15:0 - **TH** *(R/W)*

ADV_TIMER3 channel 1 threshold configuration bitfield.

6.3.5.2.27 ADV_TIMER3 channel 2 threshold configuration register. (T3_TH_CHANNEL2)

Reset value: 0x0000

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

Bits 18:16 - **MODE** *(R/W)*

ADV_TIMER3 channel 2 threshold match action on channel output signal configuration bitfield:

- *0b000*: set.
- *0b001*: toggle then next threshold match action is clear.
- *0b010*: set then next threshold match action is clear.
- *0b011*: toggle.
- *0b100*: clear.
- *0b101*: toggle then next threshold match action is set.
- *0b110*: clear then next threshold match action is set.

Bits 15:0 - **TH** *(R/W)*

ADV_TIMER3 channel 2 threshold configuration bitfield.

6.3.5.2.28 ADV_TIMER3 channel 3 threshold configuration register. (T3_TH_CHANNEL3)

Reset value: 0x0000

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

Bits 18:16 - **MODE** *(R/W)*

ADV_TIMER3 channel 3 threshold match action on channel output signal configuration bitfield:

- *0b000*: set.
- *0b001*: toggle then next threshold match action is clear.
- *0b010*: set then next threshold match action is clear.
- *0b011*: toggle.
- *0b100*: clear.
- *0b101*: toggle then next threshold match action is set.
- *0b110*: clear then next threshold match action is set.

Bits 15:0 - **TH** *(R/W)*

ADV_TIMER3 channel 3 threshold configuration bitfield.

6.3.5.2.29 ADV_TIMERS events configuration register. (EVENT_CFG)

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved												ENA			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SEL3				SEL2				SEL1				SEL0			

Bits 19:16 - **ENA** *(R/W)*

ADV_TIMER output event enable configuration bitfield. ENA[i]=1 enables output event i generation.

Bits 15:12 - **SEL3** *(R/W)*

ADV_TIMER output event 3 source configuration bitfiled:

- *0b0000*: ADV_TIMER0 channel 0.
- *0b0001*: ADV_TIMER0 channel 1.
- *0b0010*: ADV_TIMER0 channel 2.
- *0b0011*: ADV_TIMER0 channel 3.
- *0b0100*: ADV_TIMER1 channel 0.
- *0b0101*: ADV_TIMER1 channel 1.
- *0b0110*: ADV_TIMER1 channel 2.
- *0b0111*: ADV_TIMER1 channel 3.
- *0b1000*: ADV_TIMER2 channel 0.
- *0b1001*: ADV_TIMER2 channel 1.
- *0b1010*: ADV_TIMER2 channel 2.
- *0b1011*: ADV_TIMER2 channel 3.
- *0b1100*: ADV_TIMER3 channel 0.
- *0b1101*: ADV_TIMER3 channel 1.
- *0b1110*: ADV_TIMER3 channel 2.
- *0b1111*: ADV_TIMER3 channel 3.

Bits 11:8 - **SEL2** *(R/W)*

ADV_TIMER output event 2 source configuration bitfiled:

- *0b0000*: ADV_TIMER0 channel 0.
- *0b0001*: ADV_TIMER0 channel 1.
- *0b0010*: ADV_TIMER0 channel 2.
- *0b0011*: ADV_TIMER0 channel 3.
- *0b0100*: ADV_TIMER1 channel 0.
- *0b0101*: ADV_TIMER1 channel 1.
- *0b0110*: ADV_TIMER1 channel 2.
- *0b0111*: ADV_TIMER1 channel 3.
- *0b1000*: ADV_TIMER2 channel 0.
- *0b1001*: ADV_TIMER2 channel 1.
- *0b1010*: ADV_TIMER2 channel 2.
- *0b1011*: ADV_TIMER2 channel 3.
- *0b1100*: ADV_TIMER3 channel 0.
- *0b1101*: ADV_TIMER3 channel 1.
- *0b1110*: ADV_TIMER3 channel 2.
- *0b1111*: ADV_TIMER3 channel 3.

Bits 7:4 - **SEL1** *(R/W)*

ADV_TIMER output event 1 source configuration bitfiled:

- *0b0000*: ADV_TIMER0 channel 0.
- *0b0001*: ADV_TIMER0 channel 1.
- *0b0010*: ADV_TIMER0 channel 2.
- *0b0011*: ADV_TIMER0 channel 3.
- *0b0100*: ADV_TIMER1 channel 0.
- *0b0101*: ADV_TIMER1 channel 1.
- *0b0110*: ADV_TIMER1 channel 2.
- *0b0111*: ADV_TIMER1 channel 3.
- *0b1000*: ADV_TIMER2 channel 0.
- *0b1001*: ADV_TIMER2 channel 1.
- *0b1010*: ADV_TIMER2 channel 2.
- *0b1011*: ADV_TIMER2 channel 3.
- *0b1100*: ADV_TIMER3 channel 0.
- *0b1101*: ADV_TIMER3 channel 1.
- *0b1110*: ADV_TIMER3 channel 2.
- *0b1111*: ADV_TIMER3 channel 3.

Bits 3:0 - **SEL0** *(R/W)*

ADV_TIMER output event 0 source configuration bitfiled:

- *0b0000*: ADV_TIMER0 channel 0.
- *0b0001*: ADV_TIMER0 channel 1.
- *0b0010*: ADV_TIMER0 channel 2.
- *0b0011*: ADV_TIMER0 channel 3.
- *0b0100*: ADV_TIMER1 channel 0.
- *0b0101*: ADV_TIMER1 channel 1.
- *0b0110*: ADV_TIMER1 channel 2.
- *0b0111*: ADV_TIMER1 channel 3.
- *0b1000*: ADV_TIMER2 channel 0.
- *0b1001*: ADV_TIMER2 channel 1.
- *0b1010*: ADV_TIMER2 channel 2.
- *0b1011*: ADV_TIMER2 channel 3.
- *0b1100*: ADV_TIMER3 channel 0.
- *0b1101*: ADV_TIMER3 channel 1.
- *0b1110*: ADV_TIMER3 channel 2.
- *0b1111*: ADV_TIMER3 channel 3.

6.3.5.2.30 ADV_TIMERS channels clock gating configuration register. (CG)

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**
ENA															

Bits 15:0 - **ENA** *(R/W)*

ADV_TIMER clock gating configuration bitfield.

- ENA[i]=0: clock gate ADV_TIMERi.
- ENA[i]=1: enable ADV_TIMERi.

6.3.6 SoC event generator

SoC event generator component manages the following features:

- Soc events dispatching to FC event unit, Cluster event unit and uDMA
- FC High and Low timers input trigger events configuration
- SoC software event generation
- Event queue of width 2 for each event line with overflow error event generation
- 2 high priority events generation

Input events managed by SoC event generator are:

- 32kHz reference clock
- 48 SoC peripherals events inciming from uDMA interfaces, PMU, Advanced timers, GPIOs and RTC
- 8 software events

6.3.6.1 SoC event generator registers

Name	Address	Size	Type	Access	Default	Description
SW_EVENT	0x1A106000	32	Config	W	*0x0000*	SoC software events trigger command register.
FC_MASK_MSB	0x1A106004	32	Config	R/W	*0xFFFFFFFF*	MSB FC event unit event dispatch mask configuration register.
FC_MASK_LSB	0x1A106008	32	Config	R/W	*0xFFFFFFFF*	LSB FC event unit event dispatch mask configuration register.
CL_MASK_MSB	0x1A10600C	32	Config	R/W	*0xFFFFFFFF*	MSB Cluster event dispatch mask configuration register.
CL_MASK_LSB	0x1A106010	32	Config	R/W	*0xFFFFFFFF*	LSB Cluster event dispatch mask configuration register.
PR_MASK_MSB	0x1A106014	32	Config	R/W	*0xFFFFFFFF*	MSB uDMA event dispatch mask configuration register.
PR_MASK_LSB	0x1A106018	32	Config	R/W	*0xFFFFFFFF*	LSB uDMA event dispatch mask configuration register.
ERR_MSB	0x1A10601C	32	Status	R	*0x0000*	MSB event queue overflow status register.
ERR_LSB	0x1A106020	32	Status	R	*0x0000*	LSB event queue overflow status register.
TIMER_SEL_HI	0x1A106024	32	Config	R/W	*0x0000*	FC High Timer source event configuration register.
TIMER_SEL_LO	0x1A106028	32	Config	R/W	*0x0000*	FC Low Timer source event configuration register.

Table 43. SoC event generator registers table

6.3.6.2 SoC event generator registers details

6.3.6.2.1 SoC software events trigger command register. (SW_EVENT)

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								EVENT							

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

Writing a one-hot value into EVENT bitfield triggers SoC software event i. 8 software events are provided.

6.3.6.2.2 MSB FC event unit event dispatch mask configuration register. (FC_MASK_MSB)

Reset value: 0xFFFFFFFF

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

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

MSB event mask to enable/disable event dispatch to FC event unit.

- Setting bit[i] to *0b1* disable dispatching event[32+i] to FC event unit.
- Setting bit[i] to *0b0* enable dispatching event[32+i] to FC event unit.

6.3.6.2.3 LSB FC event unit event dispatch mask configuration register. (FC_MASK_LSB)

Reset value: 0xFFFFFFFF

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

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

LSB event mask to enable/disable event dispatch to FC event unit.

- Setting bit[i] to *0b1* disable dispatching event[i] to FC event unit.
- Setting bit[i] to *0b0* enable dispatching event[i] to FC event unit.

6.3.6.2.4 MSB Cluster event dispatch mask configuration register. (CL_MASK_MSB)

Reset value: 0xFFFFFFFF

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

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

MSB event mask to enable/disable event dispatch to Cluster event unit.

- Setting bit[i] to *0b1* disable dispatching event[32+i] to Cluster event unit.
- Setting bit[i] to *0b0* enable dispatching event[32+i] to Cluster event unit.

6.3.6.2.5 LSB Cluster event dispatch mask configuration register. (CL_MASK_LSB)

Reset value: 0xFFFFFFFF

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

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

LSB event mask to enable/disable event dispatch to Cluster event unit.

- Setting bit[i] to *0b1* disable dispatching event[i] to Cluster event unit.
- Setting bit[i] to *0b0* enable dispatching event[i] to Cluster event unit.

6.3.6.2.6 MSB uDMA event dispatch mask configuration register. (PR_MASK_MSB)

Reset value: 0xFFFFFFFF

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

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

MSB event mask to enable/disable event dispatch to UDMA peripherals.

- Setting bit[i] to *0b1* disable dispatching event[32+i] to uDMA.
- Setting bit[i] to *0b0* enable dispatching event[32+i] to uDMA.

6.3.6.2.7 LSB uDMA event dispatch mask configuration register. (PR_MASK_LSB)

Reset value: 0xFFFFFFFF

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

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

LSB event mask to enable/disable event dispatch to UDMA peripherals.

- Setting bit[i] to *0b1* disable dispatching event[i] to uDMA.
- Setting bit[i] to *0b0* enable dispatching event[i] to uDMA.

6.3.6.2.8 MSB event queue overflow status register. (ERR_MSB)

Reset value: 0x0000

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

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

Report MSB event queue overflows. Cleared after read.

Reading a *0b1* at ERR_MSB[i] means that an overflow occurred for SoC event[32+i] FIFO queue.

6.3.6.2.9 LSB event queue overflow status register. (ERR_LSB)

Reset value: 0x0000

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

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

Report LSB event queue overflows. Cleared after read.

Reading a *0b1* at ERR_LSB[i] means that an overflow occurred for SoC event[i] FIFO queue.

6.3.6.2.10 FC High Timer source event configuration register. (TIMER_SEL_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										TIMER_SEL_HI					

Bits 5:0 - **TIMER_SEL_HI** *(R/W)*

Configure which SoC event generator input event is propagated to FC Timer High input trigger event.

6.3.6.2.11 FC Low Timer source event configuration register. (TIMER_SEL_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										TIMER_SEL_LO					

Bits 5:0 - **TIMER_SEL_LO** *(R/W)*

Configure which SoC event generator input event is propagated to FC Timer Low input trigger event.

6.3.7 PMU DLC bridge

Maestro PMU component manages the following features:

- control of the power network of GAP8 chip: start-up, wake-up and low power transition sequences
- control the reset of the power islands

6.3.7.1 PMU DLC bridge registers

Name	Address	Size	Type	Access	Default	Description
DLC_PCTRL	0x1A107000	32	Config	R/W	*0x0000*	DLC PICL control register.
DLC_PRDATA	0x1A107004	32	Config	R	*0x0000*	DLC PICL data read register
DLC_SR	0x1A107008	32	Status	R	*0x0000*	DLC Status register

Name	Address	Size	Type	Access	Default	Description
DLC_IMR	0x1A10700C	32	Config	R/W	*0x0000*	DLC Interrupt mask register
DLC_IFR	0x1A107010	32	Status	R	*0x0000*	DLC Interrupt flag register
DLC_IOIFR	0x1A107014	32	Status	R	*0x0000*	DLC icu_ok interrupt flag register
DLC_IDIFR	0x1A107018	32	Status	R	*0x0000*	DLC icu_delayed interrupt flag register
DLC_IMCIFR	0x1A10701C	32	Status	R	*0x0000*	DLC icu_mode_changed interrupt flag register

Table 44. PMU DLC bridge registers table

6.3.7.2 PMU DLC bridge registers details

6.3.7.2.1 DLC PICL control register. (DLC_PCTRL)

Reset value: 0x0000

Comntains PICL address, data to write and go bit.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PWDATA															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIR	PADDR														START

Bits 31:16 - **PWDATA** *(R/W)*

Data to write on the PICL bus.

Bit 15 - **DIR** *(R/W)*

Direction of the transfer on the PICL bus.

- *0b0*: write operation
- *0b1*: read operation

Bits 14:1 - **PADDR** *(R/W)*

Address of the transfer on the PICL bus.

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

Start of PICL access sequence. A rising edge of the start bit starts a PICL picl transfer. Start bit remains high until the end of the sequence, which means that no new access can be performed if an access is on going.

6.3.7.2.2 DLC PICL data read register (DLC_PRDATA)

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								PRDATA							

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

Data read on the PICL bus. This data is valid after a PICL read operation, when the picl_busy bit of the DLC_SR register becomes low.

6.3.7.2.3 DLC Status register (DLC_SR)

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														SCU_BUSY	PICL_BUSY

Bit 1 - **SCU_BUSY** *(R)*

SCU busy. This bit is set to *0b1* if when a SCU sequence is on going. This bit is cleared at the end of the sequence.

Bit 0 - **PICL_BUSY** *(R)*

PICL busy. This bit is set to *0b1* if when a transfer is ongoing on the PICL bus. This bit is cleared at the end of the transfer.

6.3.7.2.4 DLC Interrupt mask register (DLC_IMR)

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											SCU_OK	PICL_OK	ICU_MD_CHG	ICU_DLYD	ICU_OK

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

Mask of scu_ok interrupt.

- *0b0*: mask cleared
- *0b1*: mask set

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

Mask of picl_ok interrupt.

- *0b0*: mask cleared
- *0b1*: mask set

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

Mask of icu_mode_changed interrupt.

- *0b0*: mask cleared
- *0b1*: mask set

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

Mask of icu_delayed interrupt.

- *0b0*: mask cleared
- *0b1*: mask set

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

Mask of icu_ok interrupt.

- *0b0*: mask cleared
- *0b1*: mask set

6.3.7.2.5 DLC Interrupt flag register (DLC_IFR)

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											SCU_OK	PICL_OK	ICU_MD_CHG	ICU_DLYD	ICU_OK

Bit 4 - **SCU_OK** *(R)*

Set to *0b1* when SCU sequence is finished. Cleared by writing 1 to bit 4 of the DLC_IFR register.

Bit 3 - **PICL_OK** *(R)*

Set to *0b1* when PICL transfer is finish. Cleared by writing 1 to bit 3 of the DLC_IFR register.

Bit 2 - **ICU_MD_CHG** *(R)*

Set to *0b1* when at least one of the bit of the DLC_IMCIFR register is set. Cleared by reading the DCL_IMCIFR register.

Bit 1 - **ICU_DLYD** *(R)*

Set to *0b1* when at least one of the bit of the DLC_IDIFR register is set. Cleared by reading the DCL_IDIFR register.

Bit 0 - **ICU_OK** *(R)*

Set to *0b1* when at least one of the bit of the DLC_IOIFR register is set. Cleared by reading the DCL_IOIFR register.

6.3.7.2.6 DLC icu_ok interrupt flag register (DLC_IOIFR)

Reset value: 0x0000

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

Bits 31:1 - **ICU_OK_FLGS** *(R)*

Flags of the icu_ok interrupts. Bit[x] is set to *0b1* when a rising edge of the signal i_icu_ok_irq[x] occurs and MSP is requester of the change mode order of the ICU[x]. A read of this register clears the register and the bit icu_ok_flag of the DLC_IFR register.

6.3.7.2.7 DLC icu_delayed interrupt flag register (DLC_IDIFR)

Reset value: 0x0000

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

Bits 31:1 - **ICU_DLYD_FLGS** *(R)*

Flags of the icu_delayed interrupts. Bit[x] is set to *0b1* when a rising edge of the signal i_icu_delayed_irq[x] occurs and MSP is requester of the change mode order of the ICU[x]. A read this register clears the register and the bit icu_delayed_flag of the DLC_IFR register.

6.3.7.2.8 DLC icu_mode_changed interrupt flag register (DLC_IMCIFR)

Reset value: 0x0000

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

Bits 31:1 - **ICU_MD_CHNG_FLGS** *(R)*

Flags of the icu_mode_changed interrupts. Bit [x] is set to *0b1* when a rising edge of the signal i_icu_mode_changed_irq[x] occurs. A read this register clears the register and the bit icu_mode_changed_flag of the DLC_IFR register.

6.3.8 RealTime Counter

RTC is an always-operating function used to keep track of “time of the day”, even when the most of the device is off. It can also be used to wake up the device from low power modes on regular intervals.

6.3.8.1 RealTime Counter registers

Name	Address	Size	Type	Access	Default	Description
APB_SR	0x1A108000	32	Status	R	*0x0000*	RTC APB status register.
APB_CR	0x1A108004	32	Config	R/W	*0x0000*	RTC APB control register.
APB_DR	0x1A108008	32	Data	R/W	*0x0000*	RTC APB data register.
APB_ICR	0x1A10800C	32	Config	R/W	*0x0000*	RTC APB interrupt control register.
APB_IMR	0x1A108010	32	Config	R/W	*0x0000*	RTC APB interrupt mask register.
APB_IFR	0x1A108014	32	Status	R/W	*0x0000*	RTC APB interrupt flag register.

Table 45. RealTime Counter registers table

6.3.8.2 RealTime Counter registers details

6.3.8.2.1 RTC APB status register. (APB_SR)

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														INT	

Bits 1:0 - **INT** *(R)*

APB interrupt status bitfield:

- *0b0*: No interruption has been requested
- *0b1*: Interruption requested

6.3.8.2.2 RTC APB control register. (APB_CR)

Reset value: 0x0000

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

Bit 16 - **OP** *(R/W)*

Indirect access operation configuration bitfield:

- *0b0*: APB read operation
- *0b1*: APB write operation

Bits 5:0 - **ADDR** *(R/W)*

Indirect address configuration bitfield.

6.3.8.2.3 RTC APB data register. (APB_DR)

Reset value: 0x0000

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

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

Indirect data bitfield.

6.3.8.2.4 RTC APB interrupt control register. (APB_ICR)

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														MODE	

Bits 1:0 - **MODE** *(R/W)*

APB interrupt signal mode configuration bitfield:

- *0b00*: APB interrupt is a high level signal
- *0b01*: APB interrupt is a low level signal
- *0b10*: APB interrupt is a high level pulse of 1 PCLK duration
- *0b11*: APB interrupt is a low level pulse of 1 PCLK duration

6.3.8.2.5 RTC APB interrupt mask register. (APB_IMR)

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														WM	RM

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

APB write operation interrupt mask bitfield:

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

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

APB read operation interrupt mask bitfield:

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

6.3.8.2.6 RTC APB interrupt flag register. (APB_IFR)

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														WF	RF

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

APB write operation status flag bitfield:

- *0b0*: nothing
- *0b1*: write operation done

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

APB read operation status flag bitfield:

- *0b0*: nothing
- *0b1*: read operation done and requested indirect data is available

6.3.9 Efuse

EFUSE component manages the following features:

- provides 128 bytes one time programmable.
- access in read mode by byte.
- byte program bit per bit operation.
- configurable low power sleep mode.

6.3.9.1 Efuse registers

Name	Address	Size	Type	Access	Default	Description
CMD	0x1A109000	32	Config	R/W	*0x0000*	EFUSE command register.
CFG	0x1A109004	32	Config	R/W	*0x0000*	EFUSE configuration register.
READ	0x1A109200	8	Data	R	*0x0*	EFUSE byte read command register.
WRITE	0x1A109200	8	Data	W	*0x0*	EFUSE byte bit index write command register.

Table 46. Efuse registers table

6.3.9.2 Efuse registers details

6.3.9.2.1 EFUSE command register. (CMD)

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								STATUS/CMD							

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

Status is not currently used:

- *0b0*: Always returns 0

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

EFUSE command configuration bitfield:

- *0b1*: Start EFUSE read mode.
- *0b10*: Start EFUSE program mode.
- *0b100*: EFUSE sleep mode.

6.3.9.2.2 EFUSE configuration register. (CFG)

Reset value: 0x0000

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

Bits 29:20 - **LONG** *(R/W)*

EFUSE long delay configuration bitfield.

Bits 19:10 - **MEDIUM** *(R/W)*

EFUSE medium delay configuration bitfield.

Bits 9:0 - **SHORT** *(R/W)*

EFUSE short delay configuration bitfield.

6.3.9.2.3 EFUSE byte read command register. (READ)

Reset value: 0x0

7	6	5	4	3	2	1	0
RD_DATA							

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

EFUSE byte data read at address (base+RegId*4). RegId goes from 0 to 0x7F.

6.3.9.2.4 EFUSE byte bit index write command register. (WRITE)

Reset value: 0x0

7	6	5	4	3	2	1	0
WR_DATA_BIT_IDX							

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

EFUSE bit index to burn to value 1 of the byte at address (base+RegId*4). RegId goes from 0 to 0x7F.

6.3.10 MicroDMA Subsystem

6.3.10.1 uDMA LVDS interface

LVDS/ORCA component manages the following features:

- controls the LVDS PAD configuration
- enables TX serializer and RX deserializer
- 2 modes are supported:
    - One is with specific HW support to interface with AT86RF215.
    - The other performs acquisition of the serial stream and synchronization is performed by software.
- supports ORCA protocol (documention to be completed)

6.3.10.1.1 uDMA LVDS interface registers

Name	Address	Size	Type	Access	Default	Description
RX_SADDR	0x1A102000	32	Config	R/W	*0x0000*	uDMA RX LVDS/ORCA buffer base address configuration register.
RX_SIZE	0x1A102004	32	Config	R/W	*0x0000*	uDMA RX LVDS/ORCA buffer size configuration register.
RX_CFG	0x1A102008	32	Config	R/W	*0x0000*	uDMA RX LVDS/ORCA stream configuration register.
TX_SADDR	0x1A102010	32	Config	R/W	*0x0000*	uDMA TX LVDS/ORCA buffer base address configuration register.
TX_SIZE	0x1A102014	32	Config	R/W	*0x0000*	uDMA TX LVDS/ORCA buffer size configuration register.
TX_CFG	0x1A102018	32	Config	R/W	*0x0000*	uDMA TX LVDS/ORCA stream configuration register.
RF_CFG	0x1A102020	32	Config	R/W	*0x0013*	uDMA LVDS configuration register.
RF_STATUS	0x1A102028	32	Status	R	*0x0000*	uDMA LVDS status register.
CLKDIV_EN	0x1A102030	32	Config	R/W	*0x0000*	uDMA LVDS clock divider enable configuration register.
CLKDIV_CFG	0x1A102034	32	Config	R/W	*0x0000*	uDMA LVDS clock divider configuration register.
CLKDIV_UPD	0x1A102038	32	Config	W	*0x0000*	uDMA LVDS clock divider update configuration register.
ORCA_CFG	0x1A10203C	32	Config	R/W	*0x0000*	uDMA ORCA configuration register.
RX_SAMPLE_CNT	0x1A102040	32	Status	R/W	*0x0000*	uDMA LVDS RX sample counter configuration register.

Table 47. uDMA LVDS interface registers table

6.3.10.1.2 uDMA LVDS interface registers details

6.3.10.1.2.1 uDMA RX LVDS/ORCA buffer base address configuration register. (RX_SADDR)

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
RX_SADDR															

Bits 15:0 - **RX_SADDR** *(R/W)*

RX buffer base address bitfield:

- Read: returns value of the buffer pointer until transfer is finished. Else returns 0.
- Write: sets RX buffer base address

6.3.10.1.2.2 uDMA RX LVDS/ORCA buffer size configuration register. (RX_SIZE)

Reset value: 0x0000

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

Bits 16:0 - **RX_SIZE** *(R/W)*

RX buffer size bitfield in bytes. (128kBytes maximum)

- Read: returns remaining buffer size to transfer.
- Write: sets buffer size.

6.3.10.1.2.3 uDMA RX LVDS/ORCA stream configuration register. (RX_CFG)

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										CLR/PENDING	EN	Reserved			CONTINOUS

Bit 5 - **CLR** *(W)*

RX channel clear and stop transfer:

- *0b0*: disabled - no action
- *0b1*: stop and clear - stop and clear the on-going transfer

Bit 5 - **PENDING** *(R)*

RX transfer pending in queue status flag:

- *0b0*: no pending - no pending transfer in the queue
- *0b1*: pending - pending transfer in the queue

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

RX channel enable and start transfer bitfield:

- *0b0*: disabled
- *0b1*: start - enable and start the transfer

This signal is used also to queue a transfer if one is already ongoing.

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

RX channel continuous mode bitfield:

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

At the end of the buffer transfer, the uDMA reloads the address / buffer size and starts a new transfer.

6.3.10.1.2.4 uDMA TX LVDS/ORCA buffer base address configuration register. (TX_SADDR)

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**
TX_SADDR															

Bits 15:0 - **TX_SADDR** *(R/W)*

TX buffer base address bitfield:

- Read: returns value of the buffer pointer until transfer is finished. Else returns 0.
- Write: sets buffer base address

6.3.10.1.2.5 uDMA TX LVDS/ORCA buffer size configuration register. (TX_SIZE)

Reset value: 0x0000

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

Bits 16:0 - **TX_SIZE** *(R/W)*

TX buffer size bitfield in bytes. (128kBytes maximum)

- Read: returns remaining buffer size to transfer.
- Write: sets buffer size.

6.3.10.1.2.6 uDMA TX LVDS/ORCA stream configuration register. (TX_CFG)

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										CLR/PENDING	EN	Reserved			CONTINOUS

Bit 5 - **CLR** *(W)*

TX channel clear and stop transfer bitfield:

- *0b0*: disabled
- *0b1*: stop and clear the on-going transfer

Bit 5 - **PENDING** *(R)*

TX transfer pending in queue status flag:

- *0b0*: no pending - no pending transfer in the queue
- *0b1*: pending - pending transfer in the queue

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

TX channel enable and start transfer bitfield:

- *0b0*: disabled
- *0b1*: enable and start the transfer

This signal is used also to queue a transfer if one is already ongoing.

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

TX channel continuous mode bitfield:

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

At the end of the buffer transfer, the uDMA reloads the address / buffer size and starts a new transfer.

6.3.10.1.2.7 uDMA LVDS configuration register. (RF_CFG)

Reset value: 0x0013

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			RF_MODE_RX	RF_MODE_VAL	RF_MODE	RF_CLKSEL	Reserved		RF_SD_TX_EN	RF_SD_RX_EN	RF_RX_ENB	RF_TX_VSEL	RF_TX_MODE	RF_TX_OEB	RF_TX_ENB

Bit 12 - **RF_MODE_RX** *(R/W)*
Reset value: 0b0

RX AT86RF215 synchronization bits detection bitfield:

- *0b0*: enable - synchronization bits detection on the RxLVDS pad. The 2 bits of synchronization are on the 16bits MSB for I and Q of IQ data received.
- *0b1*: disable - potential of 16 bits data versus 13bits for AT86RF215 but synchronization to be managed by software

Bit 11 - **RF_MODE_VAL** *(R/W)*
Reset value: 0b0

TX set data bit0 value bitfield:

- *0b0*: NO_IMMEDIATE_TX - set bit0 at 0, indicate AT86RF215 no immediate TX so TX is triggered by SPI command
- *0b1*: IMMEDIATE_TX - set bit0 at 1, indicate AT86RF215 immediate TX in case its register RF_IQIFC0.EEC(bit0)=1

Bit 10 - **RF_MODE** *(R/W)*
Reset value: 0b0

TX configuration specific to AT86RF215 bitfield:

- *0b0*: enable - add synchronisation bits on bit15–14 and bit0 used for immediate TX, payload is on bits 13 to 1.
- *0b1*: disable - this means SW to manage synchronisation with the 16bits data to TX

Bit 9 - **RF_CLKSEL** *(R/W)*
Reset value: 0b0

CLK selection bitfield:

- *0b0*: EXT_RX_CLK - use the external RX clk received by RX LVDS PAD
- *0b1*: SOC_CLK - use the SoC clk (useful for early functionality tests). For SoC clk availability, RF_CLKDIV_EN as to be 1(enabled) then the minimum clock ratio is divided by 2 if RF_CLKDIV_CFG is 0.

Bit 6 - **RF_SD_TX_EN** *(R/W)*
Reset value: 0b0

Enable TX serializer bitfield:

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

Bit 5 - **RF_SD_RX_EN** *(R/W)*
Reset value: 0b0

Enable RX serdes bitfield:

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

Bit 4 - **RF_RX_ENB** *(R/W)*
Reset value: 0b1

Enable RX LVDS PAD bitfield:

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

Bit 3 - **RF_TX_VSEL** *(R/W)*
Reset value: 0b0

TX LVDS PAD Core voltage selector bitfield:

- *0b0*: VDD_1_0V - if VDD=1.0V
- *0b1*: VDD_1_2V - if VDD=1.2V

Bit 2 - **RF_TX_MODE** *(R/W)*
Reset value: 0b0

TX LVDS PAD Current selector pin bitfield:

- *0b0*: low current - MODE = 0 selects low current
- *0b1*: high current - MODE = 1 selects high current

Bit 1 - **RF_TX_OEB** *(R/W)*
Reset value: 0b1

TX LVDS PAD Output Enable bitfield:

- *0b0*: enable - enable output
- *0b1*: disable - disable output

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

Reset value: 0b1

TX LVDS PAD Active-high power down mode select bitfield:

- *0b0*: enable - enable output
- *0b1*: disable - disable output

6.3.10.1.2.8 uDMA LVDS status register. (RF_STATUS)

Reset value: 0x0000

only bit0: RX serdes synchronization flag with at86rf215

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															RF_SYNC_FLAG

Bit 0 - **RF_SYNC_FLAG** *(R)*

Read only RX Synchronization flag between RX serdes block and the AT86RF215 bitfield:

- *0b0* not synchonized
- *0b1* synchonized - Rx serdes synchronized with external AT86RF215 synchronization

6.3.10.1.2.9 uDMA LVDS clock divider enable configuration register. (CLKDIV_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															RF_CLKDIV_EN

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

Clock divider activation bitfield.

6.3.10.1.2.10 uDMA LVDS clock divider configuration register. (CLKDIV_CFG)

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								RF_CLKDIV_CFG							

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

Clock division ratio increased by power of 2 suite. 0 is divided by 2, 1 by 4, 2 by 8... 7 by 256

6.3.10.1.2.11 uDMA LVDS clock divider update configuration register. (CLKDIV_UPD)

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															RF_CLKDIV_UPD

Bit 0 - **RF_CLKDIV_UPD** *(W)*

Apply effectively CLKDIV_CFG configuration

6.3.10.1.2.12 uDMA ORCA configuration register. (ORCA_CFG)

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							ORCA_EN	ORCA_DELAY				ORCA_SIZE			

Bit 8 - **ORCA_EN** *(R/W)*

ORCA Channel enable and start transfer bitfield:

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

Bits 7:4 - **ORCA_DELAY** *(R/W)*

ORCA delay between first data to TXSYNC signal in cycle for TX bitfield (minimum 1 cycle delay).

Default value is 0x0 for 1 cycle delay. 0X1 for 2 cycle delay etc.

Bits 3:0 - **ORCA_SIZE** *(R/W)*

ORCA data size in bits bitfield (16Bits maximum).

6.3.10.1.2.13 uDMA LVDS RX sample counter configuration register. (RX_SAMPLE_CNT)

Reset value: 0x0000

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

Bit 16 - **ENA** *(R/W)*

LVDS event generation on compare match with CMP_VAL configuration bitfield:

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

Bits 15:0 - **CMP_VAL** *(R/W)*

LVDS RX sample comparison value configuration bitfield.

6.3.10.2 uDMA SPI master interfaces

SPIM component manages the following features:

- Controls all SPI master bus specific sequencing, protocol, arbitration and timing.
- Standard or Quad half-duplex and full-duplex SPI master interface modes.
- Configurable CPOL and CPHA parameters.
- Configurable SPIM clock frequencey related to SoC clock frequency.

SPIM interface uses a stream pre-processing protocol to ease the construction of SPIM transfers combining commands and data stream. A list of the available commands and their encoding is shown in the table below.

6.3.10.2.1 SPI Master Channel 0 registers

Name	Address	Size	Type	Access	Default	Description
RX_SADDR	0x1A102080	32	Config	R/W	*0x0000*	uDMA RX SPIM buffer base address configuration register.
RX_SIZE	0x1A102084	32	Config	R/W	*0x0000*	uDMA RX SPIM buffer size configuration register.
RX_CFG	0x1A102088	32	Config	R/W	*0x0000*	uDMA RX SPIM stream configuration register.
TX_SADDR	0x1A102090	32	Config	R/W	*0x0000*	uDMA TX SPIM buffer base address configuration register.
TX_SIZE	0x1A102094	32	Config	R/W	*0x0000*	uDMA TX SPIM buffer size configuration register.
TX_CFG	0x1A102098	32	Config	R/W	*0x0000*	uDMA TX SPIM stream configuration register.

Table 48. SPI Master Channel 0 registers table

6.3.10.2.2 SPI Master Channel 1 registers

Name	Address	Size	Type	Access	Default	Description
RX_SADDR	0x1A102100	32	Config	R/W	*0x0000*	uDMA RX SPIM buffer base address configuration register.
RX_SIZE	0x1A102104	32	Config	R/W	*0x0000*	uDMA RX SPIM buffer size configuration register.
RX_CFG	0x1A102108	32	Config	R/W	*0x0000*	uDMA RX SPIM stream configuration register.
TX_SADDR	0x1A102110	32	Config	R/W	*0x0000*	uDMA TX SPIM buffer base address configuration register.
TX_SIZE	0x1A102114	32	Config	R/W	*0x0000*	uDMA TX SPIM buffer size configuration register.
TX_CFG	0x1A102118	32	Config	R/W	*0x0000*	uDMA TX SPIM stream configuration register.

Table 49. SPI Master Channel 1 registers table

6.3.10.2.3 uDMA SPI master interface registers details

6.3.10.2.3.1 uDMA RX SPIM buffer base address configuration register. (RX_SADDR)

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
RX_SADDR															

Bits 15:0 - **RX_SADDR** *(R/W)*

RX buffer base address bitfield:

- Read: returns value of the buffer pointer until transfer is finished. Else returns 0.
- Write: sets RX buffer base address

6.3.10.2.3.2 uDMA RX SPIM buffer size configuration register. (RX_SIZE)

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															RX_SIZE

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_SIZE															

Bits 16:0 - **RX_SIZE** *(R/W)*

RX buffer size bitfield in bytes. (128kBytes maximum)

- Read: returns remaining buffer size to transfer.
- Write: sets buffer size.

6.3.10.2.3.3 uDMA RX SPIM stream configuration register. (RX_CFG)

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										CLR/PENDING	EN	Reserved	DATASIZE		CONTINOUS

Bit 5 - **CLR** *(W)*

RX channel clear and stop transfer:

- *0b0*: disable
- *0b1*: stop and clear - stop and clear the on-going transfer

Bit 5 - **PENDING** *(R)*

RX transfer pending in queue status flag:

- *0b0*: no pending - no pending transfer in the queue
- *0b1*: pending - pending transfer in the queue

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

RX channel enable and start transfer bitfield:

- *0b0*: disable
- *0b1*: start - enable and start the transfer

This signal is used also to queue a transfer if one is already ongoing.

Bits 2:1 - **DATASIZE** *(R/W)*

RX channel transfer size used to increment uDMA SPIM RX buffer address pointer:

- *0b00*: plus 1 - +1 (8 bits)
- *0b01*: plus 2 - +2 (16 bits)
- *0b10*: plus 4 - +4 (32 bits)
- *0b11*: plus 0 - +0

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

RX channel continuous mode bitfield:

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

At the end of the buffer transfer, the uDMA reloads the address / buffer size and starts a new transfer.

6.3.10.2.3.4 uDMA TX SPIM buffer base address configuration register. (TX_SADDR)

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**
TX_SADDR															

Bits 15:0 - **TX_SADDR** *(R/W)*

TX buffer base address bitfield:

- Read: returns value of the buffer pointer until transfer is finished. Else returns 0.
- Write: sets buffer base address

6.3.10.2.3.5 uDMA TX SPIM buffer size configuration register. (TX_SIZE)

Reset value: 0x0000

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

Bits 16:0 - **TX_SIZE** *(R/W)*

TX buffer size bitfield in bytes. (128kBytes maximum)

- Read: returns remaining buffer size to transfer.
- Write: sets buffer size.

6.3.10.2.3.6 uDMA TX SPIM stream configuration register. (TX_CFG)

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										CLR/PENDING	EN	Reserved			CONTINOUS

Bit 5 - **CLR** *(W)*

TX channel clear and stop transfer bitfield:

- *0b0*: disabled
- *0b1*: stop and clear - stop and clear the on-going transfer

Bit 5 - **PENDING** *(R)*

TX transfer pending in queue status flag:

- *0b0*: no pending - no pending transfer in the queue
- *0b1*: pending - pending transfer in the queue

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

TX channel enable and start transfer bitfield:

- *0b0*: disabled
- *0b1*: start - enable and start the transfer

This signal is used also to queue a transfer if one is already ongoing.

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

TX channel continuous mode bitfield:

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

At the end of the buffer transfer, the uDMA reloads the address / buffer size and starts a new transfer.

6.3.10.2.4 uDMA SPI master interface commands

Name	Command number	Size	Description
SPI_CMD_CFG	0	32	SPIM configuration command.
SPI_CMD_SOT	1	32	SPIM Start of Transfer command.
SPI_CMD_SEND_CMD	2	32	SPIM send command command.
SPI_CMD_SEND_ADDR	3	32	SPIM send address command.
SPI_CMD_DUMMY	4	32	SPIM dummy RX command.
SPI_CMD_WAIT	5	32	SPIM wait uDMA external event command.
SPI_CMD_TX_DATA	6	32	SPIM send data command (max 64kbits).
SPI_CMD_RX_DATA	7	32	SPIM receive data command (max 64kbits).
SPI_CMD_RPT	8	32	SPIM repeat next transfer command.
SPI_CMD_EOT	9	32	SPIM End of Transfer command.
SPI_CMD_RPT_END	10	32	SPIM end of repeat command.
SPI_CMD_RX_CHECK	11	32	SPIM RX check data command.
SPI_CMD_FULL_DUPL	12	32	SPIM full duplex mode command.

Table 50. uDMA SPI master interface commands table

6.3.10.2.5 uDMA SPI master interface commands details

6.3.10.2.5.1 SPIM configuration command. (SPI_CMD_CFG)

Command number: 0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SPI_CMD (0)				Reserved											

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						CPOL	CPHA	CLKDIV							

Bits 31:28 - **SPI_CMD**

SPIM command to be processed is "CFG" in this case.

Bit 9 - **CPOL**

SPIM clock polarity bitfield:

- *0b0*: leading edge is rising edge
- *0b1*: leading edge is falling edge

Bit 8 - **CPHA**

SPIM clock phase bitfield:

- *0b0*: the "out" side changes the data on the trailing edge of the preceding clock cycle, while the "in" side captures the data on the leading edge of the clock cycle.
- *0b1*: the "out" side changes the data on the leading edge of the current clock cycle, while the "in" side captures the data on the trailing edge of the clock cycle.

Bits 7:0 - **CLKDIV**

SPIM clock divider bitfield.

6.3.10.2.5.2 SPIM Start of Transfer command. (SPI_CMD_SOT)

Command number: 1

<table>
<tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th></tr>
<tr><td colspan="4">SPI_CMD (1)</td><td colspan="12">Reserved</td></tr>
<tr><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="14">Reserved</td><td colspan="2">CS</td></tr>
</table>

Bits 31:28 - **SPI_CMD**

SPIM command to be processed is "SOT" in this case.

Bits 1:0 - **CS**

SPIM Chip Select (CS) bitfield:

- *0b00*: select csn0
- *0b01*: select csn1
- *0b10*: select csn2
- *0b11*: select csn3

6.3.10.2.5.3 SPIM send command command. (SPI_CMD_SEND_CMD)

Command number: 2

<table>
<tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th></tr>
<tr><td colspan="4">SPI_CMD (2)</td><td>QPI</td><td colspan="6">Reserved</td><td colspan="5">CMD_SIZE</td></tr>
<tr><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="16">CMD_VALUE</td></tr>
</table>

Bits 31:28 - **SPI_CMD**

SPIM command to be processed is "SEND_CMD" in this case.

Bit 27 - **QPI**

SPIM mode configuration bitfield:

- *0b0*: Standard
- *0b1*: Quad

Bits 20:16 - **CMD_SIZE**

SPIM command to send size in bits bitfield. The value is (num bits – 1).

Bits 15:0 - **CMD_VALUE**

SPIM command to send bitfield. MSB of the command must be left aligned if command size is lower than 16.

6.3.10.2.5.4 SPIM send address command. (SPI_CMD_SEND_ADDR)

Command number: 3

This command is followed by extra parameter bytes that are SPI_CMD_SEND_ADDR.CMD_SIZE bits long. The value following the SPI_CMD_SEND_ADDR command indicates address value.

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

Bits 31:28 - **SPI_CMD**

SPIM command to be processed is “SEND_ADDR” in this case.

Bit 27 - **QPI**

SPIM mode configuration bitfield:

- *0b0*: Standard
- *0b1*: Quad

Bits 20:16 - **CMD_SIZE**

SPIM address to send size in bits bitfield. The value is (num bits – 1).

6.3.10.2.5.5 SPIM dummy RX command. (SPI_CMD_DUMMY)

Command number: 4

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

Bits 31:28 - **SPI_CMD**

SPIM command to be processed is “DUMMY” in this case.

Bits 20:16 - **DUMMY_CYCLE**

SPIM dummy cycles value bitfield.

6.3.10.2.5.6 SPIM wait uDMA external event command. (SPI_CMD_WAIT)

Command number: 5

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SPI_CMD (5)				Reserved											

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														EVENT_ID	

Bits 31:28 - **SPI_CMD**

SPIM command to be processed is “WAIT” in this case.

Bits 1:0 - **EVENT_ID**

SPIM uDMA external event bitfield.

6.3.10.2.5.7 SPIM send data command (max 64kbits). (SPI_CMD_TX_DATA)

Command number: 6

This command is followed by extra parameter bytes that are SPI_CMD_TX_DATA.DATA_SIZE bits long. The value following the SPI_CMD_TX_DATA command indicates data value stream to transmit.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SPI_CMD (6)				QPI	BYTE_ALIGN	Reserved									

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA_SIZE															

Bits 31:28 - **SPI_CMD**

SPIM command to be processed is “TX_DATA” in this case.

Bit 27 - **QPI**

SPIM mode configuration bitfield:

- *0b0*: Standard
- *0b1*: Quad

Bit 26 - **BYTE_ALIGN**

SPIM byte alignment configuration bitfield:

- *0b0*: enable byte alignment
- *0b1*: disable byte alignment

Bits 15:0 - **DATA_SIZE**

SPIM bits size to send bitfield (max 64kbits). The value is (num bits – 1).

6.3.10.2.5.8 SPIM receive data command (max 64kbits). (SPI_CMD_RX_DATA)

Command number: 7

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

Bits 31:28 - **SPI_CMD**

SPIM command to be processed is “RX_DATA” in this case.

Bit 27 - **QPI**

SPIM mode configuration bitfield:

- *0b0*: Standard
- *0b1*: Quad

Bit 26 - **BYTE_ALIGN**

SPIM byte alignment configuration bitfield:

- *0b0*: enable byte alignment
- *0b1*: disable byte alignment

Bits 15:0 - **DATA_SIZE**

SPIM bits size to receive bitfield (max 64kbits). The value is (num bits – 1).

6.3.10.2.5.9 SPIM repeat next transfer command. (SPI_CMD_RPT)

Command number: 8

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

Bits 31:28 - **SPI_CMD**

SPIM command to be processed is “RPT” in this case.

Bits 15:0 - **RPT_CNT**

SPIM transfer repeat count value bitfield (max 64k).

6.3.10.2.5.10 SPIM End of Transfer command. (SPI_CMD_EOT)

Command number: 9

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

Bits 31:28 - **SPI_CMD**

SPIM command to be processed is “EOT” in this case.

Bit 0 - **EVENT_GEN**

SPIM uDMA EOT event generation bitfield:

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

6.3.10.2.5.11 SPIM end of repeat command. (SPI_CMD_RPT_END)

Command number: 10

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

Bits 31:28 - **SPI_CMD**

SPIM command to be processed is “RPT_END” in this case.

6.3.10.2.5.12 SPIM RX check data command. (SPI_CMD_RX_CHECK)

Command number: 11

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SPI_CMD (11)				QPI	BYTE_ALIGN	CHECK_TYPE		Reserved				STATUS_SIZE			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
COMP_DATA															

Bits 31:28 - **SPI_CMD**

SPIM command to be processed is “RX_CHECK” in this case.

Bit 27 - **QPI**

SPIM mode configuration bitfield:

- *0b0*: Standard
- *0b1*: Quad

Bit 26 - **BYTE_ALIGN**

SPIM byte alignment configuration bitfield:

- *0b0*: enable byte alignment
- *0b1*: disable byte alignment

Bits 25:24 - **CHECK_TYPE**

SPIM check mode bitfield:

- *0b00*: compare bit to bit
- *0b01*: compare only ones
- *0b10*: compare ony zeros

Bits 19:16 - **STATUS_SIZE**

SPIM read data size in bits bitfield. The value is (num bits – 1).

Bits 15:0 - **COMP_DATA**

SPIM comparison value bitfield (max 16bits).

6.3.10.2.5.13 SPIM full duplex mode command. (SPI_CMD_FULL_DUPL)

Command number: 12

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

Bits 31:28 - **SPI_CMD**

SPIM command to be processed is “FULL_DUPLEX” in this case.

Bit 26 - **BYTE_ALIGN**

SPIM byte alignment configuration bitfield:

- *0b0*: enable byte alignment
- *0b1*: disable byte alignment

Bits 15:0 - **DATA_SIZE**

SPIM bits size to send bitfield (max 64kbits). The value is (num bits – 1).

6.3.10.3 uDMA Hyperbus interface

HYPERBUS component manages the following features:

- Controls all HYPERBUS bus specific sequencing, protocol, arbitration and timing.
- Supports RAM and Flash memories types

6.3.10.3.1 uDMA Hyperbus interface registers

Name	Address	Size	Type	Access	Default	Description
RX_SADDR	0x1A102180	32	Config	R/W	*0x0000*	uDMA RX HYPERBUS buffer base address configuration register.
RX_SIZE	0x1A102184	32	Config	R/W	*0x0000*	uDMA RX HYPERBUS buffer size configuration register.
RX_CFG	0x1A102188	32	Config	R/W	*0x0000*	uDMA RX HYPERBUS stream configuration register.
TX_SADDR	0x1A102190	32	Config	R/W	*0x0000*	uDMA TX HYPERBUS buffer base address configuration register.
TX_SIZE	0x1A102194	32	Config	R/W	*0x0000*	uDMA TX HYPERBUS buffer size configuration register.
TX_CFG	0x1A102198	32	Config	R/W	*0x0000*	uDMA TX HYPERBUS stream configuration register.
EXT_ADDR	0x1A1021A0	32	Config	R/W	*0x0000*	Memory access address register.
MEM_CFG0	0x1A1021A8	32	Config	R/W	*0x3700*	Memory Control Configuration 0 register.
MEM_CFG1	0x1A1021AC	32	Config	R/W	*0x888888*	Memory Control Configuration 1 register.
MEM_CFG2	0x1A1021B0	32	Config	R/W	*0x0000*	Memory Control Configuration 2 register.
MEM_CFG3	0x1A1021B4	32	Config	R/W	*0x0004*	Memory Control Configuration 3 register.
MEM_CFG4	0x1A1021B8	32	Config	R/W	*0x3001*	Memory Control Configuration 4 register.

Name	Address	Size	Type	Access	Default	Description
MEM_CFG5	0x1A1021BC	32	Config	R/W	*0x888888*	Memory Control Configuration 5 register.
MEM_CFG6	0x1A1021C0	32	Config	R/W	*0x0000*	Memory Control Configuration 6 register.
MEM_CFG7	0x1A1021C4	32	Config	R/W	*0x0004*	Memory Control Configuration 7 register.

Table 51. uDMA Hyperbus interface registers table

6.3.10.3.2 uDMA Hyperbus interface registers details

6.3.10.3.2.1 uDMA RX HYPERBUS buffer base address configuration register. (RX_SADDR)

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**
RX_SADDR															

Bits 15:0 - **RX_SADDR** *(R/W)*

RX buffer base address bitfield:

- Read: returns value of the buffer pointer until transfer is finished. Else returns 0.
- Write: sets RX buffer base address

6.3.10.3.2.2 uDMA RX HYPERBUS buffer size configuration register. (RX_SIZE)

Reset value: 0x0000

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

Bits 16:0 - **RX_SIZE** *(R/W)*

RX buffer size bitfield in bytes. (128kBytes maximum)

- Read: returns remaining buffer size to transfer.
- Write: sets buffer size.

6.3.10.3.2.3 uDMA RX HYPERBUS stream configuration register. (RX_CFG)

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										CLR/PENDING	EN	Reserved			CONTINOUS

Bit 5 - **CLR** *(W)*

RX channel clear and stop transfer:

- *0b0*: disable
- *0b1*: stop and clear - stop and clear the on-going transfer

Bit 5 - **PENDING** *(R)*

RX transfer pending in queue status flag:

- *0b0*: no pending - no pending transfer in the queue
- *0b1*: pending - pending transfer in the queue

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

RX channel enable and start transfer bitfield:

- *0b0*: disable
- *0b1*: start - enable and start the transfer

This signal is used also to queue a transfer if one is already ongoing.

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

RX channel continuous mode bitfield:

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

At the end of the buffer transfer, the uDMA reloads the address / buffer size and starts a new transfer.

6.3.10.3.2.4 uDMA TX HYPERBUS buffer base address configuration register. (TX_SADDR)

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**
TX_SADDR															

Bits 15:0 - **TX_SADDR** *(R/W)*

TX buffer base address bitfield:

- Read: returns value of the buffer pointer until transfer is finished. Else returns 0.
- Write: sets buffer base address

6.3.10.3.2.5 uDMA TX HYPERBUS buffer size configuration register. (TX_SIZE)

Reset value: 0x0000

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

Bits 16:0 - **TX_SIZE** *(R/W)*

TX buffer size bitfield in bytes. (128kBytes maximum)

- Read: returns remaining buffer size to transfer.
- Write: sets buffer size.

6.3.10.3.2.6 uDMA TX HYPERBUS stream configuration register. (TX_CFG)

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										CLR/PENDING	EN	Reserved			CONTINOUS

Bit 5 - **CLR** *(W)*

TX channel clear and stop transfer bitfield:

- *0b0*: disabled
- *0b1*: stop and clear - stop and clear the on-going transfer

Bit 5 - **PENDING** *(R)*

TX transfer pending in queue status flag:

- *0b0*: no pending - no pending transfer in the queue
- *0b1*: pending - pending transfer in the queue

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

TX channel enable and start transfer bitfield:

- *0b0*: disabled
- *0b1*: start - enable and start the transfer

This signal is used also to queue a transfer if one is already ongoing.

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

TX channel continuous mode bitfield:

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

At the end of the buffer transfer, the uDMA reloads the address / buffer size and starts a new transfer.

6.3.10.3.2.7 Memory access address register. (EXT_ADDR)

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SADDR															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SADDR															

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

Memory access address bitfield.

6.3.10.3.2.8 Memory Control Configuration 0 register. (MEM_CFG0)

Reset value: 0x3700

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		WRAP_SIZE0		LATENCY0				MBR0							

Bits 13:12 - **WRAP_SIZE0** *(R/W)*
Reset value: 0b11

Wrapped Burst Size bitfield:

- *0b00*: not used
- *0b01*: 64 bytes
- *0b10*: 16 bytes
- *0b11*: 32 bytes

Bits 11:8 - **LATENCY0** *(R/W)*
Reset value: 0b0111

Latency Cycle value for HyperRAM bitfield. When using HyperRAM memory, this bit should be set to the same value as the read latency in configuration register of HyperRAM memory the read latency in configuration register of HyperRAM memory.

- *0b0000*: 5 CK
- *0b0001*: 6 CK
- *0b1110*: 3 CK
- *0b1111*: 4 CK
- Others: 6 CK

Bits 7:0 - **MBR0** *(R/W)*
Reset value: 0x0

Memory Base Address 0 for RAM bitfield. The base address of addressable region to each memory is set up. Since register can be set in 16M bytes boundary, lower 24 bit is fixed to 0.

6.3.10.3.2.9 Memory Control Configuration 1 register. (MEM_CFG1)

Reset value: 0x888888

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								WR_CSH0				WR_CSS0			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WR_CSHI0				RD_CSH0				RD_CSS0				RD_CSHI0			

Bits 23:20 - **WR_CSH0** *(R/W)*
Reset value: 0b1000

Write Chip Select Hold After CK Falling Edge bitfield = 1 CK * (1 + val)

Bits 19:16 - **WR_CSS0** *(R/W)*

Reset value: 0b1000

Write Chip Select Setup to next CK Rising Edge bitfield = 1 CK * (1 + val)

Bits 15:12 - **WR_CSHI0** *(R/W)*

Reset value: 0b1000

Write Chip Select High Between Operations bitfield = 1.5 CK * (1 + val)

Bits 11:8 - **RD_CSH0** *(R/W)*

Reset value: 0b1000

Read Chip Select Hold After CK Falling Edge bitfield = 1 CK * (1 + val)

Bits 7:4 - **RD_CSS0** *(R/W)*

Reset value: 0b1000

Read Chip Select Setup to next CK Rising Edge bitfield = 1 CK * (1 + val)

Bits 3:0 - **RD_CSHI0** *(R/W)*

Reset value: 0b1000

Read Chip Select High Between Operations bitfield = 1.5 CK * (1 + val)

6.3.10.3.2.10 Memory Control Configuration 2 register. (MEM_CFG2)

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved							WR_MAX_LENGTH0								
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							RD_MAX_LENGTH0								

Bits 24:16 - **WR_MAX_LENGTH0** *(R/W)*

Write Maximum Length bitfield = 2 CK * (1 + val)

Bits 8:0 - **RD_MAX_LENGTH0** *(R/W)*

Read Maximum Length bitfield (Maximum access data length in order to restrict HyperBus CS# assertion time) = 2 CK * (1 + val)

6.3.10.3.2.11 Memory Control Configuration 3 register. (MEM_CFG3)

Reset value: 0x0004

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					RDS_DELAY_ADJ			Reserved		WR_MAX_LEN_EN0	RD_MAX_LEN_EN0	CRT0	DT0	TCO0	ACS0

Bits 10:8 - **RDS_DELAY_ADJ** *(R/W)*

Reset value: 0b000

RDS Delay Configuration bitfield

Bit 5 - **WR_MAX_LEN_EN0** *(R/W)*

Reset value: 0b0

Write Maximum Length Enable bitfield (enables maximum access data length control for restricting HyperBus CS# assertion time by MAXLEN bit):

- *0b0*: Disable maximum access data length control.
- *0b1*: Enable maximum access data length control.

Bit 4 - **RD_MAX_LEN_EN0** *(R/W)*

Reset value: 0b0

Read Maximum Length Enable bitfield (enables maximum access data length control for restricting HyperBus CS# assertion time by MAXLEN bit):

- *0b0*: disable - Disable maximum access data length control.
- *0b1*: enable - Enable maximum access data length control.

Bit 3 - **CRT0** *(R/W)*

Reset value: 0b0

Configuration Register Target bitfield (indicates whether access is to memory space or register space):

- This bit is mapped to CA[46] bit in command/address cycle to HyperRAM device.
- When using HyperFlash memory, this bit should be set to 0.
- *0b0*: memory - Memory space.
- *0b1*: config - Configuration register space.

Bit 2 - **DT0** *(R/W)*

Reset value: 0b1

Device Type bitfield (set as a device type of connected memory):

- *0b0*: HyperFlash
- *0b1*: HyperRAM

Bit 1 - **TCO0** *(R/W)*

Reset value: 0b0

True Continuous Merging Option bitfield (set when the wrap transaction and subsequent continuous transaction can be merged):

- *0b0*: no merge - No merging WRAP and INCR.
- *0b1*: merge - Merging WRAP and INCR.

Bit 0 - **ACS0** *(R/W)*

Reset value: 0b0

Asymmetry Cache System Support bitfield (set when the different wrap size (cache size) is required by multi-core in system):

- *0b0*: no merge - No merging WRAP and INCR.
- *0b1*: merge - Merging WRAP and INCR.

6.3.10.3.2.12 Memory Control Configuration 4 register. (MEM_CFG4)

Reset value: 0x3001

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		WRAP_SIZE1		LATENCY1				MBR1							

Bits 13:12 - **WRAP_SIZE1** *(R/W)*

Reset value: 0b11

Wrapped Burst Size bitfield:

- *0b00*: not used
- *0b01*: 64 bytes
- *0b10*: 16 bytes
- *0b11*: 32 bytes

Bits 11:8 - **LATENCY1** *(R/W)*

Reset value: 0b0000

This bit is ignored when the DEVTYPE in HYPERBUS_CTRL_MCRi register is chosen to the HyperFlash memory (Defult latency is 1 for write, 5 for read).

Bits 7:0 - **MBR1** *(R/W)*

Reset value: 0x1

Memory Base Address1 for FLASH bitfield. HYPERBUS_MBRi+1 register for CSi+1# must be set higher than HYPERBUS_MBRi register for CSi#

6.3.10.3.2.13 Memory Control Configuration 5 register. (MEM_CFG5)

Reset value: 0x888888

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								WR_CSH1				WR_CSS1			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
WR_CSHI1				RD_CSH1				RD_CSS1				RD_CSHI1			

Bits 23:20 - **WR_CSH1** *(R/W)*

Reset value: 0b1000

Write Chip Select Hold After CK Falling Edge bitfield = 1 CK * (1 + val)

Bits 19:16 - **WR_CSS1** *(R/W)*

Reset value: 0b1000

Write Chip Select Setup to next CK Rising Edge bitfield = 1 CK * (1 + val)

Bits 15:12 - **WR_CSHI1** *(R/W)*

Reset value: 0b1000

Write Chip Select High Between Operations bitfield = 1.5 CK * (1 + val)

Bits 11:8 - **RD_CSH1** *(R/W)*

Reset value: 0b1000

Read Chip Select Hold After CK Falling Edge bitfield = 1 CK * (1 + val)

Bits 7:4 - **RD_CSS1** *(R/W)*

Reset value: 0b1000

Read Chip Select Setup to next CK Rising Edge bitfield = 1 CK * (1 + val)

Bits 3:0 - **RD_CSHI1** *(R/W)*

Reset value: 0b1000

Read Chip Select High Between Operations bitfield = 1.5 CK * (1 + val)

6.3.10.3.2.14 Memory Control Configuration 6 register. (MEM_CFG6)

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved							WR_MAX_LENGTH1								
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							RD_MAX_LENGTH1								

Bits 24:16 - **WR_MAX_LENGTH1** *(R/W)*

Write Maximum Length bitfield = 2 CK * (1 + val)

Bits 8:0 - **RD_MAX_LENGTH1** *(R/W)*

Read Maximum Length bitfield (Maximum access data length in order to restrict HyperBus CS# assertion time) = 2 CK * (1 + val)

6.3.10.3.2.15 Memory Control Configuration 7 register. (MEM_CFG7)

Reset value: 0x0004

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										WR_MAX_LEN_EN1	RD_MAX_LEN_EN1	CRT1	DT1	TCO1	ACS1

Bit 5 - **WR_MAX_LEN_EN1** *(R/W)*

Reset value: 0b0

Write Maximum Length Enable bitfield (enables maximum access data length control for restricting HyperBus CS# assertion time by MAXLEN bit):

- *0b0*: disable - Disable maximum access data length control.
- *0b1*: enable - Enable maximum access data length control.

Bit 4 - **RD_MAX_LEN_EN1** *(R/W)*

Reset value: 0b0

Read Maximum Length Enable bitfield (enables maximum access data length control for restricting HyperBus CS# assertion time by MAXLEN bit):

- *0b0*: disable - Disable maximum access data length control.
- *0b1*: enable - Enable maximum access data length control.

Bit 3 - **CRT1** *(R/W)*

Reset value: 0b0

Configuration Register Target bitfield (indicates whether access is to memory space or register space):

- *0b0*: memory - Memory space.
- *0b1*: config - Configuration register space.

This bit is mapped to CA[46] bit in command/address cycle to HyperRAM device.

When using HyperFlash memory, this bit should be set to 0.

Bit 2 - **DT1** *(R/W)*

Reset value: 0b1

Device Type bitfield (set as a device type of connected memory):

- *0b0*: HyperFlash
- *0b1*: HyperRAM

Bit 1 - **TCO1** *(R/W)*

Reset value: 0b0

True Continuous Merging Option bitfield (set when the wrap transaction and subsequent continuous transaction can be merged):

- *0b0*: no merge - No merging WRAP and INCR.
- *0b1*: merge - Merging WRAP and INCR.

Bit 0 - **ACS1** *(R/W)*

Reset value: 0b0

Asymmetry Cache System Support bitfield (set when the different wrap size (cache size) is required by multi-core in system):

- *0b0*: no merge - No merging WRAP and INCR.
- *0b1*: merge - Merging WRAP and INCR.

6.3.10.4 uDMA UART interface

UART component manages the following features:

- Standard full-duplex UART interface
- Configurable baudrate related to SoC domain clock frequency
- Configurable parity bit generation and check
- Configurable stop bit length
- Configurable character length

6.3.10.4.1 uDMA UART interface registers

Name	Address	Size	Type	Access	Default	Description
RX_SADDR	0x1A102200	32	Config	R/W	*0x0000*	uDMA RX UART buffer base address configuration register.
RX_SIZE	0x1A102204	32	Config	R/W	*0x0000*	uDMA RX UART buffer size configuration register.
RX_CFG	0x1A102208	32	Config	R/W	*0x0000*	uDMA RX UART stream configuration register.
TX_SADDR	0x1A102210	32	Config	R/W	*0x0000*	uDMA TX UART buffer base address configuration register.
TX_SIZE	0x1A102214	32	Config	R/W	*0x0000*	uDMA TX UART buffer size configuration register.

Name	Address	Size	Type	Access	Default	Description
TX_CFG	0x1A102218	32	Config	R/W	*0x0000*	uDMA TX UART stream configuration register.
STATUS	0x1A102220	32	Status	R	*0x0000*	uDMA UART status register.
SETUP	0x1A102224	32	Config	R/W	*0x0000*	UDMA UART configuration register.

Table 52. uDMA UART interface registers table

6.3.10.4.2 uDMA UART interface registers details

6.3.10.4.2.1 uDMA RX UART buffer base address configuration register. (RX_SADDR)

Reset value: 0x0000

<table>
<tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th></tr>
<tr><td colspan="16">Reserved</td></tr>
<tr><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="16">RX_SADDR</td></tr>
</table>

Bits 15:0 - **RX_SADDR** *(R/W)*

RX buffer base address bitfield:

- Read: returns value of the buffer pointer until transfer is finished. Else returns 0.
- Write: sets RX buffer base address

6.3.10.4.2.2 uDMA RX UART buffer size configuration register. (RX_SIZE)

Reset value: 0x0000

<table>
<tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th></tr>
<tr><td colspan="15">Reserved</td><td>RX_SIZE</td></tr>
<tr><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="16">RX_SIZE</td></tr>
</table>

Bits 16:0 - **RX_SIZE** *(R/W)*

RX buffer size bitfield in bytes. (128kBytes maximum)

- Read: returns remaining buffer size to transfer.
- Write: sets buffer size.

6.3.10.4.2.3 uDMA RX UART stream configuration register. (RX_CFG)

Reset value: 0x0000

<table>
<tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th></tr>
<tr><td colspan="16">Reserved</td></tr>
<tr><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="10">Reserved</td><td>CLR/PENDING</td><td>EN</td><td colspan="3">Reserved</td><td>CONTINOUS</td></tr>
</table>

Bit 5 - **CLR** *(W)*

RX channel clear and stop transfer:

- *0b0*: disable
- *0b1*: stop and clear the on-going transfer

Bit 5 - **PENDING** *(R)*

RX transfer pending in queue status flag:

- *0b0*: no pending transfer in the queue
- *0b1*: pending transfer in the queue

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

RX channel enable and start transfer bitfield:

- *0b0*: disable
- *0b1*: enable and start the transfer

This signal is used also to queue a transfer if one is already ongoing.

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

RX channel continuous mode bitfield:

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

At the end of the buffer transfer, the uDMA reloads the address / buffer size and starts a new transfer.

6.3.10.4.2.4 uDMA TX UART buffer base address configuration register. (TX_SADDR)

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**
TX_SADDR															

Bits 15:0 - **TX_SADDR** *(R/W)*

TX buffer base address bitfield:

- Read: returns value of the buffer pointer until transfer is finished. Else returns 0.
- Write: sets buffer base address

6.3.10.4.2.5 uDMA TX UART buffer size configuration register. (TX_SIZE)

Reset value: 0x0000

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

Bits 16:0 - **TX_SIZE** *(R/W)*

TX buffer size bitfield in bytes. (128kBytes maximum)

- Read: returns remaining buffer size to transfer.
- Write: sets buffer size.

6.3.10.4.2.6 uDMA TX UART stream configuration register. (TX_CFG)

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										CLR/PENDING	EN	Reserved			CONTINOUS

Bit 5 - **CLR** *(W)*

TX channel clear and stop transfer bitfield:

- *0b0*: disabled
- *0b1*: stop and clear the on-going transfer

Bit 5 - **PENDING** *(R)*

TX transfer pending in queue status flag:

- *0b0*: no pending transfer in the queue
- *0b1*: pending transfer in the queue

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

TX channel enable and start transfer bitfield:

- *0b0*: disabled
- *0b1*: enable and start the transfer

This signal is used also to queue a transfer if one is already ongoing.

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

TX channel continuous mode bitfield:

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

At the end of the buffer transfer, the uDMA reloads the address / buffer size and starts a new transfer.

6.3.10.4.2.7 uDMA UART status register. (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													RX_PE	RX_BUSY	TX_BUSY

Bit 2 - **RX_PE** *(R)*

RX parity error status flag:

- *0b0*: no error
- *0b1*: RX parity error occurred

Bit 1 - **RX_BUSY** *(R)*

RX busy status flag:

- *0b0*: no RX transfer on-going
- *0b1*: RX transfer on-going

Bit 0 - **TX_BUSY** *(R)*

TX busy status flag:

- *0b0*: no TX transfer on-going
- *0b1*: TX transfer on-going

6.3.10.4.2.8 UDMA UART configuration register. (SETUP)

Reset value: 0x0000

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

Bits 31:16 - **CLKDIV** *(R/W)*

UART Clock divider configuration bitfield. The baudrate is equal to SOC_FREQ/CLKDIV.

Bit 9 - **RX_ENA** *(R/W)*

RX transceiver configuration bitfield:

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

Bit 8 - **TX_ENA** *(R/W)*

TX transceiver configuration bitfield:

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

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

Stop bits length bitfield:

- *0b0*: 1 stop bit
- *0b1*: 2 stop bits

Bits 2:1 - **BIT_LENGTH** *(R/W)*

Character length bitfield:

- *0b00*: 5 bits
- *0b01*: 6 bits
- *0b10*: 7 bits
- *0b11*: 8 bits

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

Parity bit generation and check configuration bitfield:

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

6.3.10.5 uDMA I2C interfaces

I2C component manages the following features:

- Controls all I2C bus specific sequencing, protocol, arbitration and timing.
- Configurable I2C clock frequency related to SoC clock frequency.
- Status flags for busy bus and arbitration lost.

I2C interface uses a stream pre-processing protocol to ease the construction of I2C transfers combining commands and data stream. A list of the available commands and their encoding is shown in the table below.

6.3.10.5.1 I2C Channel 0 registers

Name	Address	Size	Type	Access	Default	Description
RX_SADDR	0x1A102280	32	Config	R/W	*0x0000*	uDMA RX I2C buffer base address configuration register.
RX_SIZE	0x1A102284	32	Config	R/W	*0x0000*	uDMA RX I2C buffer size configuration register.
RX_CFG	0x1A102288	32	Config	R/W	*0x0000*	uDMA RX I2C stream configuration register.
TX_SADDR	0x1A102290	32	Config	R/W	*0x0000*	uDMA TX I2C buffer base address configuration register.
TX_SIZE	0x1A102294	32	Config	R/W	*0x0000*	uDMA TX I2C buffer size configuration register.
TX_CFG	0x1A102298	32	Config	R/W	*0x0000*	uDMA TX I2C stream configuration register.
STATUS	0x1A1022A0	32	Status	R/W	*0x0000*	uDMA I2C Status register.
SETUP	0x1A1022A4	32	Config	R/W	*0x0000*	uDMA I2C Configuration register.

Table 53. I2C Channel 0 registers table

6.3.10.5.2 I2C Channel 1 registers

Name	Address	Size	Type	Access	Default	Description
RX_SADDR	0x1A102300	32	Config	R/W	*0x0000*	uDMA RX I2C buffer base address configuration register.
RX_SIZE	0x1A102304	32	Config	R/W	*0x0000*	uDMA RX I2C buffer size configuration register.
RX_CFG	0x1A102308	32	Config	R/W	*0x0000*	uDMA RX I2C stream configuration register.
TX_SADDR	0x1A102310	32	Config	R/W	*0x0000*	uDMA TX I2C buffer base address configuration register.
TX_SIZE	0x1A102314	32	Config	R/W	*0x0000*	uDMA TX I2C buffer size configuration register.
TX_CFG	0x1A102318	32	Config	R/W	*0x0000*	uDMA TX I2C stream configuration register.
STATUS	0x1A102320	32	Status	R/W	*0x0000*	uDMA I2C Status register.

Name	Address	Size	Type	Access	Default	Description
SETUP	0x1A102324	32	Config	R/W	*0x0000*	uDMA I2C Configuration register.

Table 54. I2C Channel 1 registers table

6.3.10.5.3 uDMA I2C interface registers details

6.3.10.5.3.1 uDMA RX I2C buffer base address configuration register. (RX_SADDR)

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**
RX_SADDR															

Bits 15:0 - **RX_SADDR** *(R/W)*

RX buffer base address bitfield:

- Read: returns value of the buffer pointer until transfer is finished. Else returns 0.
- Write: sets RX buffer base address

6.3.10.5.3.2 uDMA RX I2C buffer size configuration register. (RX_SIZE)

Reset value: 0x0000

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

Bits 16:0 - **RX_SIZE** *(R/W)*

RX buffer size bitfield in bytes. (128kBytes maximum)

- Read: returns remaining buffer size to transfer.
- Write: sets buffer size.

6.3.10.5.3.3 uDMA RX I2C stream configuration register. (RX_CFG)

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										CLR/PENDING	EN	Reserved			CONTINOUS

Bit 5 - **CLR** *(W)*

RX channel clear and stop transfer:

- *0b0*: disable
- *0b1*: stop and clear the on-going transfer

Bit 5 - **PENDING** *(R)*

RX transfer pending in queue status flag:

- *0b0*: no pending transfer in the queue
- *0b1*: pending transfer in the queue

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

RX channel enable and start transfer bitfield:

- *0b0*: disable
- *0b1*: enable and start the transfer

This signal is used also to queue a transfer if one is already ongoing.

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

RX channel continuous mode bitfield:

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

At the end of the buffer transfer, the uDMA reloads the address / buffer size and starts a new transfer.

6.3.10.5.3.4 uDMA TX I2C buffer base address configuration register. (TX_SADDR)

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**
TX_SADDR															

Bits 15:0 - **TX_SADDR** *(R/W)*

TX buffer base address bitfield:

- Read: returns value of the buffer pointer until transfer is finished. Else returns 0.
- Write: sets buffer base address

6.3.10.5.3.5 uDMA TX I2C buffer size configuration register. (TX_SIZE)

Reset value: 0x0000

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

Bits 16:0 - **TX_SIZE** *(R/W)*

TX buffer size bitfield in bytes. (128kBytes maximum)

- Read: returns remaining buffer size to transfer.
- Write: sets buffer size.

6.3.10.5.3.6 uDMA TX I2C stream configuration register. (TX_CFG)

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										CLR/PENDING	EN	Reserved			CONTINOUS

Bit 5 - **CLR** *(W)*

TX channel clear and stop transfer bitfield:

- *0b0*: disabled
- *0b1*: stop and clear the on-going transfer

Bit 5 - **PENDING** *(R)*

TX transfer pending in queue status flag:

- *0b0*: no pending transfer in the queue
- *0b1*: pending transfer in the queue

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

TX channel enable and start transfer bitfield:

- *0b0*: disabled
- *0b1*: enable and start the transfer

This signal is used also to queue a transfer if one is already ongoing.

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

TX channel continuous mode bitfield:

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

At the end of the buffer transfer, the uDMA reloads the address / buffer size and starts a new transfer.

6.3.10.5.3.7 uDMA I2C Status register. (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														ARB_LOST	BUSY

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

I2C arbitration lost status flag:

- *0b0*: no error
- *0b1*: arbitration lost error

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

I2C bus busy status flag:

- *0b0*: no transfer on-going
- *0b1*: transfer on-going

6.3.10.5.3.8 uDMA I2C Configuration register. (SETUP)

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															DO_RST

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

Reset command used to abort the on-going transfer and clear busy and arbitration lost status flags.

6.3.10.5.4 uDMA I2C interface commands

Name	Command number	Size	Description
I2C_CMD_START	0	8	I2C Start of Transfer command.
I2C_CMD_WAIT_EV	1	8	I2C wait uDMA external event command.
I2C_CMD_STOP	2	8	I2C End of Transfer command.
I2C_CMD_RD_ACK	4	8	I2C receive data and acknowledge command.
I2C_CMD_RD_NACK	6	8	I2C receive data and not acknowledge command.
I2C_CMD_WR	8	8	I2C send data and wait acknowledge command.
I2C_CMD_WAIT	10	8	I2C wait dummy cycles command.
I2C_CMD_RPT	12	8	I2C next command repeat command.
I2C_CMD_CFG	14	8	I2C configuration command.

Table 55. uDMA I2C interface commands table

6.3.10.5.5 uDMA I2C interface commands details

6.3.10.5.5.1 I2C Start of Transfer command. (I2C_CMD_START)

Command number: 0

7	6	5	4	3	2	1	0
I2C_CMD (0)							

Bits 7:0 - **I2C_CMD**

I2C Start of Transfer command.

6.3.10.5.5.2 I2C wait uDMA external event command. (I2C_CMD_WAIT_EV)

Command number: 1

This command is followed by extra parameter bytes that are 1 byte (bit[1:0] – event_id). long. The value following the

I2C_CMD_WAIT_EV command indicates selected uDMA external event ID.

7	6	5	4	3	2	1	0
I2C_CMD (1)							

Bits 7:0 - **I2C_CMD**

I2C wait uDMA external event command.

6.3.10.5.5.3 I2C End of Transfer command. (I2C_CMD_STOP)

Command number: 2

7	6	5	4	3	2	1	0
I2C_CMD (2)							

Bits 7:0 - **I2C_CMD**

I2C End of Transfer command.

6.3.10.5.5.4 I2C receive data and acknowledge command. (I2C_CMD_RD_ACK)

Command number: 4

7	6	5	4	3	2	1	0
I2C_CMD (4)							

Bits 7:0 - **I2C_CMD**

I2C receive data and acknowledge command.

6.3.10.5.5.5 I2C receive data and not acknowledge command. (I2C_CMD_RD_NACK)

Command number: 6

7	6	5	4	3	2	1	0
I2C_CMD (6)							

Bits 7:0 - **I2C_CMD**

I2C receive data and not acknowledge command.

6.3.10.5.5.6 I2C send data and wait acknowledge command. (I2C_CMD_WR)

Command number: 8

This command is followed by extra parameter bytes that are 1 byte or many bytes if preceded by a I2C_CMD_RPT command. long. The value following the I2C_CMD_WR command indicates byte value to transmit or multiple byte values to transmit if this command is preceded by a I2C_CMD_RPT command.

7	6	5	4	3	2	1	0
I2C_CMD (8)							

Bits 7:0 - **I2C_CMD**

I2C send data and wait acknowledge command.

6.3.10.5.5.7 I2C wait dummy cycles command. (I2C_CMD_WAIT)

Command number: 10

This command is followed by extra parameter bytes that are 1 byte. long. The value following the I2C_CMD_WAIT command indicates I2C dummy clock cycles value.

7	6	5	4	3	2	1	0
I2C_CMD (10)							

Bits 7:0 - **I2C_CMD**

I2C wait dummy cycles command.

6.3.10.5.5.8 I2C next command repeat command. (I2C_CMD_RPT)

Command number: 12

This command is followed by extra parameter bytes that are 1 byte. long. The value following the I2C_CMD_RPT command indicates number of times to repeat next command.

7	6	5	4	3	2	1	0
I2C_CMD (12)							

Bits 7:0 - **I2C_CMD**

I2C next command repeat command.

6.3.10.5.5.9 I2C configuration command. (I2C_CMD_CFG)

Command number: 14

This command is followed by extra parameter bytes that are 2 bytes. long. The value following the I2C_CMD_CFG command indicates I2C clock divider 16bits value related to SoC clock frequency. MSB byte is sent first.

7	6	5	4	3	2	1	0
I2C_CMD (14)							

Bits 7:0 - **I2C_CMD**

I2C configuration command.

6.3.10.6 uDMA MEMCPY interface

MEMCPY component manages the following features:

- memory copy functionality
- one source/destination is always L2 memory
- second source/destination can be FC_TCDM or L2 memory

6.3.10.6.1 uDMA MEMCPY interface registers

Name	Address	Size	Type	Access	Default	Description
RX_SADDR	0x1A102380	32	Config	R/W	*0x0000*	uDMA RX MEMCPY buffer base address configuration register.

Name	Address	Size	Type	Access	Default	Description
RX_SIZE	0x1A102384	32	Config	R/W	*0x0000*	uDMA RX MEMCPY buffer size configuration register.
RX_CFG	0x1A102388	32	Config	R/W	*0x0000*	uDMA RX MEMCPY stream configuration register.
TX_SADDR	0x1A102390	32	Config	R/W	*0x0000*	uDMA TX MEMCPY buffer base address configuration register.
TX_SIZE	0x1A102394	32	Config	R/W	*0x0000*	uDMA TX MEMCPY buffer size configuration register.
TX_CFG	0x1A102398	32	Config	R/W	*0x0000*	uDMA TX MEMCPY stream configuration register.
DST_ADDR	0x1A1023A0	32	Config	R/W	*0x0000*	MEMCPY TX destination address configuration register.
SRC_ADDR	0x1A1023A4	32	Config	R/W	*0x0000*	MEMCPY RX source address configuration register.
MEM_SEL	0x1A1023A8	32	Config	R/W	*0x0000*	MEMCPY memory source/destination select configuration register.

Table 56. uDMA MEMCPY interface registers table

6.3.10.6.2 uDMA MEMCPY interface registers details

6.3.10.6.2.1 uDMA RX MEMCPY buffer base address configuration register. (RX_SADDR)

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**
RX_SADDR															

Bits 15:0 - **RX_SADDR** *(R/W)*

RX buffer base address bitfield:

- Read: returns value of the buffer pointer until transfer is finished. Else returns 0.
- Write: sets RX buffer base address

6.3.10.6.2.2 uDMA RX MEMCPY buffer size configuration register. (RX_SIZE)

Reset value: 0x0000

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

Bits 16:0 - **RX_SIZE** *(R/W)*

RX buffer size bitfield in bytes. (128kBytes maximum)

- Read: returns remaining buffer size to transfer.
- Write: sets buffer size.

6.3.10.6.2.3 uDMA RX MEMCPY stream configuration register. (RX_CFG)

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										CLR/PENDING	EN	Reserved			CONTINOUS

Bit 5 - **CLR** *(W)*

RX channel clear and stop transfer:

- *0b0*: disable
- *0b1*: stop and clear - stop and clear the on-going transfer

Bit 5 - **PENDING** *(R)*

RX transfer pending in queue status flag:

- *0b0*: no pending - no pending transfer in the queue
- *0b1*: pending - pending transfer in the queue

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

RX channel enable and start transfer bitfield:

- *0b0*: disable
- *0b1*: start - enable and start the transfer

This signal is used also to queue a transfer if one is already ongoing.

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

RX channel continuous mode bitfield:

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

At the end of the buffer transfer, the uDMA reloads the address / buffer size and starts a new transfer.

6.3.10.6.2.4 uDMA TX MEMCPY buffer base address configuration register. (TX_SADDR)

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**
TX_SADDR															

Bits 15:0 - **TX_SADDR** *(R/W)*

TX buffer base address bitfield:

- Read: returns value of the buffer pointer until transfer is finished. Else returns 0.
- Write: sets buffer base address

6.3.10.6.2.5 uDMA TX MEMCPY buffer size configuration register. (TX_SIZE)

Reset value: 0x0000

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

Bits 16:0 - **TX_SIZE** *(R/W)*

TX buffer size bitfield in bytes. (128kBytes maximum)

- Read: returns remaining buffer size to transfer.
- Write: sets buffer size.

6.3.10.6.2.6 uDMA TX MEMCPY stream configuration register. (TX_CFG)

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										CLR/PENDING	EN	Reserved			CONTINOUS

Bit 5 - **CLR** *(W)*

TX channel clear and stop transfer bitfield:

- *0b0*: disabled
- *0b1*: stop and clear - stop and clear the on-going transfer

Bit 5 - **PENDING** *(R)*

TX transfer pending in queue status flag:

- *0b0*: no pending - no pending transfer in the queue
- *0b1*: pending - pending transfer in the queue

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

TX channel enable and start transfer bitfield:

- *0b0*: disabled
- *0b1*: start - enable and start the transfer

This signal is used also to queue a transfer if one is already ongoing.

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

TX channel continuous mode bitfield:

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

At the end of the buffer transfer, the uDMA reloads the address / buffer size and starts a new transfer.

6.3.10.6.2.7 MEMCPY TX destination address configuration register. (DST_ADDR)

Reset value: 0x0000

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

Bits 16:0 - **DST_ADDR** *(R/W)*

TX destination start address configuration bitfield. MEMCPY TX transfer copy data from L2 TX_CFG.TX_SADDR address to FC_TCDM or L2 memories DST_ADDR address.

6.3.10.6.2.8 MEMCPY RX source address configuration register. (SRC_ADDR)

Reset value: 0x0000

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

Bits 16:0 - **SRC_ADDR** *(R/W)*

RX source start address configuration bitfield. MEMCPY RX transfer copy data from FC_TCDM or L2 memories SRC_ADDR address to L2 RX_CFG.RX_SADDR address.

6.3.10.6.2.9 MEMCPY memory source/destination select configuration register. (MEM_SEL)

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															MEM_SEL

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

Memory selection configuration bitfield:

- *0b0*: L2 from/to FC_TCDM
- *0b1*: L2 from/to L2

6.3.10.7 uDMA I2S interface

I2S component manages the following features:

- Controls all I2S bus specific sequencing, protocol, arbitration and timing.
- Configurable I2S clock frequency related to SoC clock frequency.
- Configurable slave and master modes for clock and WS signals management.
- 2 available clock/WS generators.
- Support PCM and PDM formats.
- PDM filtering feature with decimation and normalization functionalities.
- Support DDR mode.

6.3.10.7.1 uDMA I2S interface registers

Name	Address	Size	Type	Access	Default	Description
RX_SADDR_CH0	0x1A102400	32	Config	R/W	*0x0000*	uDMA RX I2S channel 0 buffer base address configuration register.
RX_SIZE_CH0	0x1A102404	32	Config	R/W	*0x0000*	uDMA RX I2S channel 0 buffer size configuration register.
RX_CFG_CH0	0x1A102408	32	Config	R/W	*0x0004*	uDMA RX I2S channel 0 stream configuration register.
RX_SADDR_CH1	0x1A102410	32	Config	R/W	*0x0000*	uDMA RX I2S channel 1 buffer base address configuration register.
RX_SIZE_CH1	0x1A102414	32	Config	R/W	*0x0000*	uDMA RX I2S channel 1 buffer size configuration register.
RX_CFG_CH1	0x1A102418	32	Config	R/W	*0x0004*	uDMA RX I2S channel 1 stream configuration register.
CFG_EXT	0x1A102420	32	Config	R/W	*0x0000*	I2S external clock configuration register.
CFG_CLKGEN0	0x1A102424	32	Config	R/W	*0x0000*	I2S clock and WS generator 0 configuration register.
CFG_CLKGEN1	0x1A102428	32	Config	R/W	*0x0000*	I2S clock and WS generator 1 configuration register.
CHMODE	0x1A10242C	32	Config	R/W	*0x0000*	I2S channels mode configuration register.
FILT_CH0	0x1A102430	32	Config	R/W	*0x0000*	I2S channel 0 filtering configuration register.
FILT_CH1	0x1A102434	32	Config	R/W	*0x0000*	I2S channel 1 filtering configuration register.

Table 57. uDMA I2S interface registers table

6.3.10.7.2 uDMA I2S interface registers details

6.3.10.7.2.1 uDMA RX I2S channel 0 buffer base address configuration register. (RX_SADDR_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
RX_SADDR															

Bits 15:0 - **RX_SADDR** *(R/W)*

I2S channel 0 RX buffer base address bitfield:

- Read: returns value of the buffer pointer until transfer is finished. Else returns 0.
- Write: sets RX buffer base address

6.3.10.7.2.2 uDMA RX I2S channel 0 buffer size configuration register. (RX_SIZE_CH0)

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															RX_SIZE

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_SIZE															

Bits 16:0 - **RX_SIZE** *(R/W)*

I2S channel 0 RX buffer size bitfield in bytes. (128kBytes maximum)

- Read: returns remaining buffer size to transfer.
- Write: sets buffer size.

6.3.10.7.2.3 uDMA RX I2S channel 0 stream configuration register. (RX_CFG_CH0)

Reset value: 0x0004

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										CLR/PENDING	EN	Reserved	DATASIZE		CONTINOUS

Bit 5 - **CLR** *(W)*
Reset value: 0b0

I2S channel 0 RX Channel clear and stop transfer:

- *0b0*: disable
- *0b1*: stop and clear - stop and clear the on-going transfer

Bit 5 - **PENDING** *(R)*
Reset value: 0b0

I2S channel 0 RX Transfer pending in queue status flag:

- *0b0*: no pending - no pending transfer in the queue
- *0b1*:pending - pending transfer in the queue

Bit 4 - **EN** *(R/W)*
Reset value: 0b0

I2S channel 0 RX Channel enable and start transfer bitfield:

- *0b0*: disable
- *0b1*: start - enable and start the transfer

This signal is used also to queue a transfer if one is already ongoing.

Bits 2:1 - **DATASIZE** *(R/W)*
Reset value: 0b10

RX channel transfer size used to increment uDMA I2S channel 0 RX buffer address pointer:

- *0b00*: plus 1 - +1 (8 bits)
- *0b01*: plus 2 - +2 (16 bits)
- *0b10*: plus 4 - +4 (32 bits)
- *0b11*: plus 0 - +0

Bit 0 - **CONTINOUS** *(R/W)*
Reset value: 0b0

I2S channel 0 RX channel continuous mode bitfield:

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

At the end of the buffer transfer, the uDMA reloads the address / buffer size and starts a new transfer.

6.3.10.7.2.4 uDMA RX I2S channel 1 buffer base address configuration register. (RX_SADDR_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**
TX_SADDR															

Bits 15:0 - **TX_SADDR** *(R/W)*

I2S channel 1 RX buffer base address bitfield:

- Read: returns value of the buffer pointer until transfer is finished. Else returns 0.
- Write: sets RX buffer base address

6.3.10.7.2.5 uDMA RX I2S channel 1 buffer size configuration register. (RX_SIZE_CH1)

Reset value: 0x0000

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

Bits 16:0 - **TX_SIZE** *(R/W)*

I2S channel 1 RX buffer size bitfield in bytes. (128kBytes maximum)

- Read: returns remaining buffer size to transfer.
- Write: sets buffer size.

6.3.10.7.2.6 uDMA RX I2S channel 1 stream configuration register. (RX_CFG_CH1)

Reset value: 0x0004

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										CLR/PENDING	EN	Reserved	DATASIZE		CONTINOUS

Bit 5 - **CLR** *(W)*

Reset value: 0b0

I2S channel 1 RX Channel clear and stop transfer:

- *0b0*: disable
- *0b1*: stop and clear - stop and clear the on-going transfer

Bit 5 - **PENDING** *(R)*

Reset value: 0b0

I2S channel 1 RX Transfer pending in queue status flag:

- *0b0*: no pending - no pending transfer in the queue
- *0b1*: pending - pending transfer in the queue

Bit 4 - **EN** *(R/W)*
Reset value: 0b0

I2S channel 1 RX Channel enable and start transfer bitfield:

- *0b0*: disable
- *0b1*: start - enable and start the transfer

This signal is used also to queue a transfer if one is already ongoing.

Bits 2:1 - **DATASIZE** *(R/W)*
Reset value: 0b10

RX channel transfer size used to increment uDMA I2S channel 1 RX buffer address pointer:

- *0b00*: plus 1 - +1 (8 bits)
- *0b01*: plus 2 - +2 (16 bits)
- *0b10*: plus 4 - +4 (32 bits)
- *0b11*: plus 0 - +0

Bit 0 - **CONTINOUS** *(R/W)*
Reset value: 0b0

I2S channel 1 RX channel continuous mode bitfield:

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

At the end of the buffer transfer, the uDMA reloads the address / buffer size and starts a new transfer.

6.3.10.7.2.7 I2S external clock configuration register. (CFG_EXT)

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											EXT_BITS_WORD				

Bits 4:0 - **EXT_BITS_WORD** *(R/W)*

External clock word length in bits bitfield. The value is (num bits - 1).

6.3.10.7.2.8 I2S clock and WS generator 0 configuration register. (CFG_CLKGEN0)

Reset value: 0x0000

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

Bits 31:16 - **CLK_DIV** *(R/W)*

Clock generator 0 clock divider related to SoC clock frequency.

Bit 8 - **CLK_EN** *(R/W)*

Clock generator 0 enable bitfield:

- *0b0*: disabled
- *0b1*: enabled - enabled. Clock and WS signal are generated.

Bits 4:0 - **BITS_WORD** *(R/W)*

Clock generator 0 word length in bits bitfield. The value is (num bits - 1).

6.3.10.7.2.9 I2S clock and WS generator 1 configuration register. (CFG_CLKGEN1)

Reset value: 0x0000

<table>
<tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th></tr>
<tr><td colspan="16">CLK_DIV</td></tr>
<tr><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="7">Reserved</td><td>CLK_EN</td><td colspan="3">Reserved</td><td colspan="5">BITS_WORD</td></tr>
</table>

Bits 31:16 - **CLK_DIV** *(R/W)*

Clock generator 1 clock divider related to SoC clock frequency.

Bit 8 - **CLK_EN** *(R/W)*

Clock generator 1 enable bitfield:

- *0b0*: disabled
- *0b1*: enabled - enabled. Clock and WS signal are generated.

Bits 4:0 - **BITS_WORD** *(R/W)*

Clock generator 1 word length in bits bitfield. The value is (num bits - 1).

6.3.10.7.2.10 I2S channels mode configuration register. (CHMODE)

Reset value: 0x0000

<table>
<tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th></tr>
<tr><td colspan="4">Reserved</td><td colspan="2">CH1_MODE</td><td colspan="2">CH0_MODE</td><td colspan="6">Reserved</td><td>CH1_USEDDR</td><td>CH0_USEDDR</td></tr>
<tr><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="2">Reserved</td><td>CH1_PDM_EN</td><td>CH0_PDM_EN</td><td colspan="2">Reserved</td><td>CH1_PDM_USEFILTER</td><td>CH0_PDM_USEFILTER</td><td colspan="2">Reserved</td><td>CH1_LSB_FIRST</td><td>CH0_LSB_FIRST</td><td colspan="4">Reserved</td></tr>
</table>

Bits 27:26 - **CH1_MODE** *(R/W)*

I2S channel 1 clock/WS mode configuration bitfield:

- *0b00*: clock gen 0 - use clock generator 0 (clock and WS generated by clkgen)
- *0b01*: clock gen 1 - use clock generator 1 (clock and WS generated by clkgen)
- *0b10*: ext clock int ws - use external clock but internal generated WS by clock generator 0
- *0b11*: ext clock ext ws - use external clock and external WS

Bits 25:24 - **CH0_MODE** *(R/W)*

I2S channel 0 clock/WS mode configuration bitfield:

- *0b00*: clock gen 0 - use clock generator 0 (clock and WS generated by clkgen)
- *0b01*: clock gen 1 - use clock generator 1 (clock and WS generated by clkgen)
- *0b10*: ext clock int ws - use external clock but internal generated WS by clock generator 0
- *0b11*: ext clock ext ws - use external clock and external WS

Bit 17 - **CH1_USEDDR** *(R/W)*

I2S channel 1 DDR mode activation bitfield:

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

Bit 16 - **CH0_USEDDR** *(R/W)*

I2S channel 0 DDR mode activation bitfield:

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

Bit 13 - **CH1_PDM_EN** *(R/W)*

I2S channel 1 PDM demodulation activation bitfield:

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

Bit 12 - **CH0_PDM_EN** *(R/W)*

I2S channel 0 PDM demodulation activation bitfield:

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

Bit 9 - **CH1_PDM_USEFILTER** *(R/W)*

I2S channel 1 PDM filter activation bitfield:

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

Bit 8 - **CH0_PDM_USEFILTER** *(R/W)*

I2S channel 0 PDM filter activation bitfield:

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

Bit 5 - **CH1_LSB_FIRST** *(R/W)*

I2S channel 1 LSB first configuration for word serialization bitfield:

- *0b0*: MSB first
- *0b1*: LSB first

Bit 4 - **CH0_LSB_FIRST** *(R/W)*

I2S channel 0 LSB first configuration for word serialization bitfield:

- *0b0*: MSB first
- *0b1*: LSB first

6.3.10.7.2.11 I2S channel 0 filtering configuration register. (FILT_CH0)

Reset value: 0x0000

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

Bits 18:16 - **SHIFT** *(R/W)*

I2S channel 0 PDM filter normalisation right shift value bitfield.

Bits 9:0 - **DECIMATION** *(R/W)*

I2S channel 0 PDM filter decimation value bitfield.

6.3.10.7.2.12 I2S channel 1 filtering configuration register. (FILT_CH1)

Reset value: 0x0000

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

Bits 18:16 - **SHIFT** *(R/W)*

I2S channel 1 PDM filter normalisation right shift value bitfield.

Bits 9:0 - **DECIMATION** *(R/W)*

I2S channel 1 PDM filter decimation value bitfield.

6.3.10.8 uDMA CPI interface

CPI component manages the following features:

- Controls all CPI bus specific sequencing, protocol, arbitration and timing.
- Configurable CPI clock frequencey related to SoC clock frequency.
- Frame dropping feature.
- Frame slicing feature.
- RAW, RGB565, RGB555 and RGB444 frame format supported.

6.3.10.8.1 uDMA CPI interface registers

Name	Address	Size	Type	Access	Default	Description
RX_SADDR	0x1A102480	32	Config	R/W	*0x0000*	uDMA RX CPI buffer base address configuration register.

Name	Address	Size	Type	Access	Default	Description
RX_SIZE	0x1A102484	32	Config	R/W	*0x0000*	uDMA RX CPI buffer size configuration register.
RX_CFG	0x1A102488	32	Config	R/W	*0x0000*	uDMA RX CPI stream configuration register.
CFG_GLOB	0x1A1024A0	32	Config	R/W	*0x0000*	uDMA CPI Global configuration register.
CFG_LL	0x1A1024A4	32	Config	R/W	*0x0000*	uDMA CPI Lower Left corner configuration register.
CFG_UR	0x1A1024A8	32	Config	R/W	*0x0000*	uDMA CPI Upper Right corner configuration register.
CFG_SIZE	0x1A1024AC	32	Config	R/W	*0x0000*	uDMA CPI Horizontal Resolution configuration register.
CFG_FILTER	0x1A1024B0	32	Config	R/W	*0x0000*	uDMA CPI RGB coefficients configuration register.

Table 58. uDMA CPI interface registers table

6.3.10.8.2 uDMA CPI interface registers details

6.3.10.8.2.1 uDMA RX CPI buffer base address configuration register. (RX_SADDR)

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**
RX_SADDR															

Bits 15:0 - **RX_SADDR** *(R/W)*

RX buffer base address bitfield:

- Read: returns value of the buffer pointer until transfer is finished. Else returns 0.
- Write: sets RX buffer base address

6.3.10.8.2.2 uDMA RX CPI buffer size configuration register. (RX_SIZE)

Reset value: 0x0000

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

Bits 16:0 - **RX_SIZE** *(R/W)*

RX buffer size bitfield in bytes. (128kBytes maximum)

- Read: returns remaining buffer size to transfer.
- Write: sets buffer size.

6.3.10.8.2.3 uDMA RX CPI stream configuration register. (RX_CFG)

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										CLR/PENDING	EN	Reserved	DATASIZE		CONTINOUS

Bit 5 - **CLR** *(W)*

RX Channel clear and stop transfer:

- *0b0*: disable
- *0b1*: stop and clear - stop and clear the on-going transfer

Bit 5 - **PENDING** *(R)*

RX Transfer pending in queue status flag:

- *0b0*: no pending transfer in the queue
- *0b1*: pending transfer in the queue

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

RX Channel enable and start transfer bitfield:

- *0b0*: disable
- *0b1*: start - enable and start the transfer

This signal is used also to queue a transfer if one is already ongoing.

Bits 2:1 - **DATASIZE** *(R/W)*

RX channel transfer size used to increment uDMA CPI RX buffer address pointer:

- *0b00*: plus 1 - +1 (8 bits)
- *0b01*: plus 2 - +2 (16 bits)
- *0b10*: plus 4 - +4 (32 bits)
- *0b11*: plus 0 - +0

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

RX channel continuous mode bitfield:

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

At the end of the buffer transfer, the uDMA reloads the address / buffer size and starts a new transfer.

6.3.10.8.2.4 uDMA CPI Global configuration register. (CFG_GLOB)

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EN	Reserved														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	SHIFT				FORMAT			FRAMESLICE_EN	FRAMEDROP_VAL						FRAMEDROP_EN

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

CPI interface acquisition enable bitfield:

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

The enable/disable happens only at the beginning of a frame, meaning when VSYNC pulse occurs.

Bits 14:11 - **SHIFT** *(R/W)*

Right shift value for final pixel normalisation bitfield. Not used if CFG_GLOB.FORMAT = BYPASS

Bits 10:8 - **FORMAT** *(R/W)*

Input frame format bitfield:

- *0b000*: RGB565
- *0b001*: RGB555
- *0b010*: RGB444
- *0b100*: BYPASS_LITEND
- 3'b101: BYPASS_BIGEND

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

Input frame slicing bitfield:

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

Bits 6:1 - **FRAMEDROP_VAL** *(R/W)*

Frame dropping value bitfield.

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

Frame dropping bitfield:

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

6.3.10.8.2.5 uDMA CPI Lower Left corner configuration register. (CFG_LL)

Reset value: 0x0000

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

Bits 31:16 - **FRAMESLICE_LLY** *(R/W)*

Y coordinate of lower left corner of slice bitfield. Origin reference of a frame is left up corner.

Bits 15:0 - **FRAMESLICE_LLX** *(R/W)*

X coordinate of lower left corner of slice bitfield. Origin reference of a frame is left up corner.

6.3.10.8.2.6 uDMA CPI Upper Right corner configuration register. (CFG_UR)

Reset value: 0x0000

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

Bits 31:16 - **FRAMESLICE_URY** *(R/W)*

Y coordinate of upper right corner of slice bitfield. Origin reference of a frame is left up corner.

Bits 15:0 - **FRAMESLICE_URX** *(R/W)*

X coordinate of upper right corner of slice bitfield. Origin reference of a frame is left up corner.

6.3.10.8.2.7 uDMA CPI Horizontal Resolution configuration register. (CFG_SIZE)

Reset value: 0x0000

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

Bits 31:16 - **ROWLEN** *(R/W)*

Frame horizontal pixel length bitfield. It is used for slice mode. Value set into the bitfield must be equal to (rowlen - 1).

6.3.10.8.2.8 uDMA CPI RGB coefficients configuration register. (CFG_FILTER)

Reset value: 0x0000

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

Bits 23:16 - **R_COEFF** *(R/W)*

R component coefficient bitfield. Not used if CFG_GLOB.FORMAT = BYPASS.

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

G component coefficient bitfield.Not used if CFG_GLOB.FORMAT = BYPASS.

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

B component coefficient bitfield. Not used if CFG_GLOB.FORMAT = BYPASS.

6.3.10.9 uDMA control unit

uDMA controller component manages the following features:

- uDMA interfaces clock gating
- uDMA interface trigger events configuration

6.3.10.9.1 uDMA control unit registers

Name	Address	Size	Type	Access	Default	Description
CFG_CG	0x1A102780	32	Config	R/W	*0x0000*	uDMA interfaces clock gate configuration register.
CFG_EVENT	0x1A102784	32	Config	R/W	*0x0000*	uDMA interfaces trigger events configuration register.

Table 59. uDMA control unit registers table

6.3.10.9.2 uDMA control unit registers details

6.3.10.9.2.1 uDMA interfaces clock gate configuration register. (CFG_CG)

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						CPI	I2S	TCDM	I2C1	I2C0	UART	HYPER	SPIM1	SPIM0	LVDS

Bit 9 - **CPI** *(R/W)*

uDMA interfaces clock gate configuration for CPI:

- *0b0*: CPI interface clock gate is enabled
- *0b1*: CPI interface clock gate is disabled

Bit 8 - **I2S** *(R/W)*

uDMA interfaces clock gate configuration for I2S:

- *0b0*: I2S interface clock gate is enabled
- *0b1*: I2S interface clock gate is disabled

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

uDMA interfaces clock gate configuration for TCDM:

- *0b0*: TCDM interface clock gate is enabled
- *0b1*: TCDM interface clock gate is disabled

Bit 6 - **I2C1** *(R/W)*

uDMA interfaces clock gate configuration for I2C1:

- *0b0*: I2C1 interface clock gate is enabled
- *0b1*: I2C1 interface clock gate is disabled

Bit 5 - **I2C0** *(R/W)*

uDMA interfaces clock gate configuration for I2C0:

- *0b0*: I2C0 interface clock gate is enabled
- *0b1*: I2C0 interface clock gate is disabled

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

uDMA interfaces clock gate configuration for UART:

- *0b0*: UART interface clock gate is enabled
- *0b1*: UART interface clock gate is disabled

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

uDMA interfaces clock gate configuration for HYPER:

- *0b0*: HYPER interface clock gate is enabled
- *0b1*: HYPER interface clock gate is disabled

Bit 2 - **SPIM1** *(R/W)*

uDMA interfaces clock gate configuration for SPIM1:

- *0b0*: SPIM1 interface clock gate is enabled
- *0b1*: SPIM1 interface clock gate is disabled

Bit 1 - **SPIM0** *(R/W)*

uDMA interfaces clock gate configuration for SPIM0:

- *0b0*: SPIM0 interface clock gate is enabled
- *0b1*: SPIM0 interface clock gate is disabled

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

uDMA interfaces clock gate configuration for LVDS:

- *0b0*: LVDS interface clock gate is enabled
- *0b1*: LVDS interface clock gate is disabled

6.3.10.9.2.2 uDMA interfaces trigger events configuration register. (CFG_EVENT)

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EVT3								EVT2							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVT1								EVT0							

Bits 31:24 - **EVT3** *(R/W)*

uDMA interfaces trigger event 3 configuration bitfield. CFG_EVT3 selects which SoC event is propageted to uDMA interface trigger event 3.

Bits 23:16 - **EVT2** *(R/W)*

uDMA interfaces trigger event 2 configuration bitfield. CFG_EVT2 selects which SoC event is propageted to uDMA interface trigger event 2.

Bits 15:8 - **EVT1** *(R/W)*

uDMA interfaces trigger event 1 configuration bitfield. CFG_EVT1 selects which SoC event is propageted to uDMA interface trigger event 1.

Bits 7:0 - **EVT0** *(R/W)*

uDMA interfaces trigger event 0 configuration bitfield. CFG_EVT0 selects which SoC event is propageted to uDMA interface trigger event 0.

6.4 Fabric Controller Subsystem

6.4.1 Fabric Controller Subsystem Events

Event number	Event name	IP instance name	Direction	Description
0	SW_EVT_0	FC_CORE	Input	Software event 0 from FC_CORE or FC interconnect
1	SW_EVT_1	FC_CORE	Input	Software event 1 from FC_CORE or FC interconnect
2	SW_EVT_2	FC_CORE	Input	Software event 2 from FC_CORE or FC interconnect
3	SW_EVT_3	FC_CORE	Input	Software event 3 from FC_CORE or FC interconnect
4	SW_EVT_4	FC_CORE	Input	Software event 4 from FC_CORE or FC interconnect
5	SW_EVT_5	FC_CORE	Input	Software event 5 from FC_CORE or FC interconnect
6	SW_EVT_6	FC_CORE	Input	Software event 6 from FC_CORE or FC interconnect
7	SW_EVT_7	FC_CORE	Input	Software event 7 from FC_CORE or FC interconnect
8	Reserved			Reserved
9	Reserved			Reserved
10	FC_TIMER_LO_EVT	FC_TIMER_UNIT	Input	FC timer low event
11	FC_TIMER_HI_EVT	FC_TIMER_UNIT	Input	FC timer high event
12	Reserved			Reserved
13	Reserved			Reserved
14	Reserved			Reserved
15	Reserved			Reserved
16	Reserved			Reserved
17	Reserved			Reserved
18	Reserved			Reserved
19	Reserved			Reserved
20	Reserved			Reserved
21	Reserved			Reserved
22	Reserved			Reserved
23	Reserved			Reserved
24	Reserved			Reserved
25	Reserved			Reserved
26	Reserved			Reserved
27	SOC_PERIPH_EVT	SOC_EVENT_GENERATOR	Input	SoC peripherals event
28	MPU_ERR_EVT	FC_MPU	Input	MPU error event
29	SOC_PERIPH_ERR_EVT	SOC_EVENT_GENERATOR	Input	SoC peripherals event fifo overflow error event
30	HP_EVT_0	SOC_EVENT_GENERATOR	Input	High priority event 0
31	HP_EVT_1	SOC_EVENT_GENERATOR	Input	High priority event 1

Table 60. Fabric Controller Subsystem Events table

6.4.2 FC control unit

FC_CTRL component manages the following features:

- End of Computation status flag

- configurable boot address to define where to fetch first instruction in FC_CORE after releasing the reset

6.4.2.1 FC control unit registers

Name	Address	Aliased address	Size	Type	Access	Default	Description
EOC	0x1B200000	0x00200000	32	Status	R	*0x0000*	End of computation status register.
BOOT_ADDR	0x1B200040	0x00200040	32	Config	R/W	*0x1A000000*	Boot address configuration register.

Table 61. FC control unit registers table

6.4.2.2 FC control unit registers details

6.4.2.2.1 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)*

End of computation status bitfield:

- *0b0*: computation still on-going on SoC domain and/or Cluster domain.
- *0b1*: no more computation on SoC domain nor Cluster domain.

6.4.2.2.2 Boot address configuration register. (BOOT_ADDR)

Reset value: 0x1A000000

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)*

Boot address configuration bitfield. The FC_CORE will fetch first instruction at (BA+RESET_EXCEPTION_OFFSET).

6.4.3 FC 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

6.4.3.1 FC timer registers

Name	Address	Aliased address	Size	Type	Access	Default	Description
CFG_LO	0x1B200400	0x00200400	32	Config	R/W	*0x0000*	Timer Low Configuration register.
CFG_HI	0x1B200404	0x00200404	32	Config	R/W	*0x0000*	Timer High Configuration register.
CNT_LO	0x1B200408	0x00200408	32	Data	R/W	*0x0000*	Timer Low counter value register.
CNT_HI	0x1B20040C	0x0020040C	32	Data	R/W	*0x0000*	Timer High counter value register.
CMP_LO	0x1B200410	0x00200410	32	Config	R/W	*0x0000*	Timer Low comparator value register.
CMP_HI	0x1B200414	0x00200414	32	Config	R/W	*0x0000*	Timer High comparator value register.
START_LO	0x1B200418	0x00200418	32	Config	R/W	*0x0000*	Start Timer Low counting register.
START_HI	0x1B20041C	0x0020041C	32	Config	R/W	*0x0000*	Start Timer High counting register.
RESET_LO	0x1B200420	0x00200420	32	Config	R/W	*0x0000*	Reset Timer Low counter register.
RESET_HI	0x1B200424	0x00200424	32	Config	R/W	*0x0000*	Reset Timer High counter register.

Table 62. FC timer registers table

Registers details are identical to Cluster timer defined above

6.4.4 FC event units

FC_CORE event unit component manages the following features:

- FC_CORE software events generation
- FC_CORE clock gate control
- Wait for event functionality
- Input event mask configuration
- FC_CORE IRQ generation
- FC_CORE secured and non-secured mode IRQ mask configuration

Events managed by FC event unit are:

- 2 SoC high prioriy events
- 1 SoC event generator fifo overflow error event
- 1 MPU error event
- 1 SoC peripheral event: when this event occurs, the SoC peripheral events fifo must be read to get the SoC event ID.
- 2 FC timer events
- 8 software events that can come from FC_CORE directly or external triggering.

6.4.4.1 FC event unit registers

Name	Address	Size	Type	Access	Default	Description
EVT_MASK	0x1B200800	32	Config	R/W	*0x0000*	Input event mask configuration register.

Name	Address	Size	Type	Access	Default	Description
EVT_MASK_AND	0x1B200804	32	Config	W	*0x0000*	Input event mask update command register with bitwise AND operation.
EVT_MASK_OR	0x1B200808	32	Config	W	*0x0000*	Input event mask update command register with bitwise OR operation.
IRQ_MASK	0x1B20080C	32	Config	R/W	*0x0000*	Interrupt request mask configuration register.
IRQ_MASK_AND	0x1B200810	32	Config	W	*0x0000*	Interrupt request mask update command register with bitwise AND operation.
IRQ_MASK_OR	0x1B200814	32	Config	W	*0x0000*	Interrupt request mask update command register with bitwise OR operation.
CLOCK_STATUS	0x1B200818	32	Config	R	*0x0000*	FC_CORE clock status register.
EVENT_BUFFER	0x1B20081C	32	Config	R	*0x0000*	Pending input events status register.
EVENT_BUFFER_MASKED	0x1B200820	32	Config	R	*0x0000*	Pending input events status register with EVT_MASK applied.
EVENT_BUFFER_IRQ_MASKED	0x1B200824	32	Config	R	*0x0000*	Pending input events status register with IRQ_MASK applied.
EVENT_BUFFER_CLEAR	0x1B200828	32	Config	W	*0x0000*	Pending input events status clear command register.
SEC_IRQ_MASK	0x1B200840	32	Config	R/W	*0x0000*	FC_CORE secured mode interrupt request mask configuration register.
REMOTE_SW_EVENT_0_TRIG	0x1B200E00	32	Config	W	*0x0000*	Remote software event 0 trigger command register.
REMOTE_SW_EVENT_1_TRIG	0x1B200E04	32	Config	W	*0x0000*	Remote software event 1 trigger command register.
REMOTE_SW_EVENT_2_TRIG	0x1B200E08	32	Config	W	*0x0000*	Remote software event 2 trigger command register.
REMOTE_SW_EVENT_3_TRIG	0x1B200E0C	32	Config	W	*0x0000*	Remote software event 3 trigger command register.
REMOTE_SW_EVENT_4_TRIG	0x1B200E10	32	Config	W	*0x0000*	Remote software event 4 trigger command register.
REMOTE_SW_EVENT_5_TRIG	0x1B200E14	32	Config	W	*0x0000*	Remote software event 5 trigger command register.
REMOTE_SW_EVENT_6_TRIG	0x1B200E18	32	Config	W	*0x0000*	Remote software event 6 trigger command register.
REMOTE_SW_EVENT_7_TRIG	0x1B200E1C	32	Config	W	*0x0000*	Remote software event 7 trigger command register.
SOC_PERIPH_EVENT_ID	0x1B200F00	32	Status	R	*0x0000*	SoC peripheral event ID status register.

Table 63. FC event unit registers table

6.4.4.2 FC event unit (FC private) registers

Name	Aliased address	Size	Type	Access	Default	Description
EVT_MASK	0x00204000	32	Config	R/W	*0x0000*	Input event mask configuration register.
EVT_MASK_AND	0x00204004	32	Config	W	*0x0000*	Input event mask update command register with bitwise AND operation.
EVT_MASK_OR	0x00204008	32	Config	W	*0x0000*	Input event mask update command register with bitwise OR operation.
IRQ_MASK	0x0020400C	32	Config	R/W	*0x0000*	Interrupt request mask configuration register.
IRQ_MASK_AND	0x00204010	32	Config	W	*0x0000*	Interrupt request mask update command register with bitwise AND operation.

Name	Aliased address	Size	Type	Access	Default	Description
IRQ_MASK_OR	0x00204014	32	Config	W	*0x0000*	Interrupt request mask update command register with bitwise OR operation.
CLOCK_STATUS	0x00204018	32	Config	R	*0x0000*	FC_CORE clock status register.
EVENT_BUFFER	0x0020401C	32	Config	R	*0x0000*	Pending input events status register.
EVENT_BUFFER_MASKED	0x00204020	32	Config	R	*0x0000*	Pending input events status register with EVT_MASK applied.
EVENT_BUFFER_IRQ_MASKED	0x00204024	32	Config	R	*0x0000*	Pending input events status register with IRQ_MASK applied.
EVENT_BUFFER_CLEAR	0x00204028	32	Config	W	*0x0000*	Pending input events status clear command register.
EVENT_WAIT	0x00204038	32	Config	R	*0x0000*	Input event wait command register.
EVENT_WAIT_CLEAR	0x0020403C	32	Config	R	*0x0000*	Input event wait and clear command register.
SEC_IRQ_MASK	0x00204040	32	Config	R/W	*0x0000*	FC_CORE secured mode interrupt request mask configuration register.
SEC_IRQ_MASK_AND	0x00204044	32	Config	W	*0x0000*	FC_CORE secured mode interrupt request mask update command register with bitwise AND operation.
SEC_IRQ_MASK_OR	0x00204048	32	Config	W	*0x0000*	FC_CORE secured mode interrupt request mask update command register with bitwise OR operation.
SW_EVENT_0_TRIG	0x00204100	32	Config	W	*0x0000*	FC_CORE Software event 0 trigger command register.
SW_EVENT_1_TRIG	0x00204104	32	Config	W	*0x0000*	FC_CORE Software event 1 trigger command register.
SW_EVENT_2_TRIG	0x00204108	32	Config	W	*0x0000*	FC_CORE Software event 2 trigger command register.
SW_EVENT_3_TRIG	0x0020410C	32	Config	W	*0x0000*	FC_CORE Software event 3 trigger command register.
SW_EVENT_4_TRIG	0x00204110	32	Config	W	*0x0000*	FC_CORE Software event 4 trigger command register.
SW_EVENT_5_TRIG	0x00204114	32	Config	W	*0x0000*	FC_CORE Software event 5 trigger command register.
SW_EVENT_6_TRIG	0x00204118	32	Config	W	*0x0000*	FC_CORE Software event 6 trigger command register.
SW_EVENT_7_TRIG	0x0020411C	32	Config	W	*0x0000*	FC_CORE Software event 7 trigger command register.
SW_EVENT_0_WAIT	0x00204140	32	Config	R	*0x0000*	FC_CORE Software event 0 wait command register.
SW_EVENT_1_WAIT	0x00204144	32	Config	R	*0x0000*	FC_CORE Software event 1 wait command register.
SW_EVENT_2_WAIT	0x00204148	32	Config	R	*0x0000*	FC_CORE Software event 2 wait command register.
SW_EVENT_3_WAIT	0x0020414C	32	Config	R	*0x0000*	FC_CORE Software event 3 wait command register.
SW_EVENT_4_WAIT	0x00204150	32	Config	R	*0x0000*	FC_CORE Software event 4 wait command register.
SW_EVENT_5_WAIT	0x00204154	32	Config	R	*0x0000*	FC_CORE Software event 5 wait command register.
SW_EVENT_6_WAIT	0x00204158	32	Config	R	*0x0000*	FC_CORE Software event 6 wait command register.
SW_EVENT_7_WAIT	0x0020415C	32	Config	R	*0x0000*	FC_CORE Software event 7 wait command register.

Name	Aliased address	Size	Type	Access	Default	Description
SW_EVENT_0_WAIT_CLEAR	0x00204180	32	Config	R	*0x0000*	FC_CORE Software event 0 wait and clear command register.
SW_EVENT_1_WAIT_CLEAR	0x00204184	32	Config	R	*0x0000*	FC_CORE Software event 1 wait and clear command register.
SW_EVENT_2_WAIT_CLEAR	0x00204188	32	Config	R	*0x0000*	FC_CORE Software event 2 wait and clear command register.
SW_EVENT_3_WAIT_CLEAR	0x0020418C	32	Config	R	*0x0000*	FC_CORE Software event 3 wait and clear command register.
SW_EVENT_4_WAIT_CLEAR	0x00204190	32	Config	R	*0x0000*	FC_CORE Software event 4 wait and clear command register.
SW_EVENT_5_WAIT_CLEAR	0x00204194	32	Config	R	*0x0000*	FC_CORE Software event 5 wait and clear command register.
SW_EVENT_6_WAIT_CLEAR	0x00204198	32	Config	R	*0x0000*	FC_CORE Software event 6 wait and clear command register.
SW_EVENT_7_WAIT_CLEAR	0x0020419C	32	Config	R	*0x0000*	FC_CORE Software event 7 wait and clear command register.

Table 64. FC event unit (FC private) registers table

6.4.4.3 FC event unit registers details

6.4.4.3.1 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
EM															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
EM															

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

Input event mask configuration bitfield:

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

6.4.4.3.2 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 EM[i] if EMA[i]=*0b1*.

6.4.4.3.3 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 EM[i] if EMO[i]=*0b1*.

6.4.4.3.4 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
IM															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
IM															

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

Interrupt request mask configuration bitfield:

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

6.4.4.3.5 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 IM[i] if IMA[i]=*0b1*.

6.4.4.3.6 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 IM[i] if IMO[i]=*0b1*.

6.4.4.3.7 FC_CORE 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)*

FC_CORE clock status bitfield:

- CS=*0b0*: FC_CORE clocked is gated
- CS=*0b1*: FC_CORE clocked is running

6.4.4.3.8 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* means that one or more input event i request are pending.

6.4.4.3.9 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 EM applied.

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

6.4.4.3.10 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 IM applied.

IBM[i]=*0b1* means that one or more input events i are pending.

6.4.4.3.11 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*.

6.4.4.3.12 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 FC_CORE clock until at least one unmasked event occurs. The read content of this bitfield is equivalent to EVENT_BUFFER_MASKED.EBM

6.4.4.3.13 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.EM[i]=*0b1*. The read content of this bitfield is equivalent to EVENT_BUFFER_MASKED.EBM

6.4.4.3.14 FC_CORE secured mode interrupt request mask configuration register. (SEC_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
SIM															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SIM															

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

Interrupt request mask configuration bitfield when FC_CORE is in secured mode:

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

6.4.4.3.15 FC_CORE secured mode interrupt request mask update command register with bitwise AND operation. (SEC_IRQ_MASK_AND)

Reset value: 0x0000

Host access bus: DEMUX

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

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

Interrupt request mask configuration bitfield update with bitwise AND operation when FC_CORE is in secured mode. It allows clearing IM[i] if IMA[i]=*0b1*.

6.4.4.3.16 FC_CORE secured mode interrupt request mask update command register with bitwise OR operation. (SEC_IRQ_MASK_OR)

Reset value: 0x0000

Host access bus: DEMUX

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

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

Interrupt request mask configuration bitfield update with bitwise OR operation when FC_CORE is in secured mode. It allows setting IM[i] if IMO[i]=*0b1*.

6.4.4.3.17 FC_CORE Software event 0 trigger command register. (SW_EVENT_0_TRIG)

Reset value: 0x0000

Host access bus: DEMUX

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

Bits 31:0 - **SW0T** *(W)*

Writing to this register triggers FC_CORE software event 0.

6.4.4.3.18 FC_CORE Software event 1 trigger command register. (SW_EVENT_1_TRIG)

Reset value: 0x0000

Host access bus: DEMUX

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

Bits 31:0 - **SW1T** *(W)*

Writing to this register triggers FC_CORE software event 1.

6.4.4.3.19 FC_CORE Software event 2 trigger command register. (SW_EVENT_2_TRIG)

Reset value: 0x0000

Host access bus: DEMUX

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

Bits 31:0 - **SW2T** *(W)*

Writing to this register triggers FC_CORE software event 2.

6.4.4.3.20 FC_CORE Software event 3 trigger command register. (SW_EVENT_3_TRIG)

Reset value: 0x0000

Host access bus: DEMUX

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

Bits 31:0 - **SW3T** *(W)*

Writing to this register triggers FC_CORE software event 3.

6.4.4.3.21 FC_CORE Software event 4 trigger command register. (SW_EVENT_4_TRIG)

Reset value: 0x0000

Host access bus: DEMUX

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

Bits 31:0 - **SW4T** *(W)*

Writing to this register triggers FC_CORE software event 4.

6.4.4.3.22 FC_CORE Software event 5 trigger command register. (SW_EVENT_5_TRIG)

Reset value: 0x0000

Host access bus: DEMUX

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

Bits 31:0 - **SW5T** *(W)*

Writing to this register triggers FC_CORE software event 5.

6.4.4.3.23 FC_CORE Software event 6 trigger command register. (SW_EVENT_6_TRIG)

Reset value: 0x0000

Host access bus: DEMUX

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

Bits 31:0 - **SW6T** *(W)*

Writing to this register triggers FC_CORE software event 6.

6.4.4.3.24 FC_CORE Software event 7 trigger command register. (SW_EVENT_7_TRIG)

Reset value: 0x0000

Host access bus: DEMUX

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

Bits 31:0 - **SW7T** *(W)*

Writing to this register triggers FC_CORE software event 7.

6.4.4.3.25 FC_CORE Software event 0 wait command register. (SW_EVENT_0_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 trigger FC_CORE sofware event 0 and gate the FC_CORE clock until at least one unmasked event occurs. The read content of this bitfield is equivalent to EVENT_BUFFER_MASKED.EBM

6.4.4.3.26 FC_CORE Software event 1 wait command register. (SW_EVENT_1_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 trigger FC_CORE sofware event 1 and gate the FC_CORE clock until at least one unmasked event occurs. The read content of this bitfield is equivalent to EVENT_BUFFER_MASKED.EBM

6.4.4.3.27 FC_CORE Software event 2 wait command register. (SW_EVENT_2_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 trigger FC_CORE sofware event 2 and gate the FC_CORE clock until at least one unmasked event occurs. The read content of this bitfield is equivalent to EVENT_BUFFER_MASKED.EBM

6.4.4.3.28 FC_CORE Software event 3 wait command register. (SW_EVENT_3_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 trigger FC_CORE sofware event 3 and gate the FC_CORE clock until at least one unmasked event occurs. The read content of this bitfield is equivalent to EVENT_BUFFER_MASKED.EBM

6.4.4.3.29 FC_CORE Software event 4 wait command register. (SW_EVENT_4_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 trigger FC_CORE sofware event 4 and gate the FC_CORE clock until at least one unmasked event occurs. The read content of this bitfield is equivalent to EVENT_BUFFER_MASKED.EBM

6.4.4.3.30 FC_CORE Software event 5 wait command register. (SW_EVENT_5_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 trigger FC_CORE sofware event 5 and gate the FC_CORE clock until at least one unmasked event occurs. The read content of this bitfield is equivalent to EVENT_BUFFER_MASKED.EBM

6.4.4.3.31 FC_CORE Software event 6 wait command register. (SW_EVENT_6_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 trigger FC_CORE sofware event 6 and gate the FC_CORE clock until at least one unmasked event occurs. The read content of this bitfield is equivalent to EVENT_BUFFER_MASKED.EBM

6.4.4.3.32 FC_CORE Software event 7 wait command register. (SW_EVENT_7_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 trigger FC_CORE sofware event 7 and gate the FC_CORE clock until at least one unmasked event occurs. The read content of this bitfield is equivalent to EVENT_BUFFER_MASKED.EBM

6.4.4.3.33 FC_CORE Software event 0 wait and clear command register. (SW_EVENT_0_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 will trigger FC_CORE sofware event 0 and gate the FC_CORE clock until at least one unmasked event occurs. In addition, EVENT_BUFFER.EB[i] bits are cleared if EVT_MASK.EM[i]=*0b1* after the read. The read content of this bitfield is equivalent to EVENT_BUFFER_MASKED.EBM

6.4.4.3.34 FC_CORE Software event 1 wait and clear command register. (SW_EVENT_1_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 will trigger FC_CORE sofware event 1 and gate the FC_CORE clock until at least one unmasked event occurs. In addition, EVENT_BUFFER.EB[i] bits are cleared if EVT_MASK.EM[i]=*0b1* after the read. The read content of this bitfield is equivalent to EVENT_BUFFER_MASKED.EBM

6.4.4.3.35 FC_CORE Software event 2 wait and clear command register. (SW_EVENT_2_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 will trigger FC_CORE sofware event 2 and gate the FC_CORE clock until at least one unmasked event occurs. In addition, EVENT_BUFFER.EB[i] bits are cleared if EVT_MASK.EM[i]=*0b1* after the read. The read content of this bitfield is equivalent to EVENT_BUFFER_MASKED.EBM

6.4.4.3.36 FC_CORE Software event 3 wait and clear command register. (SW_EVENT_3_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 will trigger FC_CORE sofware event 3 and gate the FC_CORE clock until at least one unmasked event occurs. In addition, EVENT_BUFFER.EB[i] bits are cleared if EVT_MASK.EM[i]=*0b1* after the read. The read content of this bitfield is equivalent to EVENT_BUFFER_MASKED.EBM

6.4.4.3.37 FC_CORE Software event 4 wait and clear command register. (SW_EVENT_4_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 will trigger FC_CORE sofware event 4 and gate the FC_CORE clock until at least one unmasked event occurs. In addition, EVENT_BUFFER.EB[i] bits are cleared if EVT_MASK.EM[i]=*0b1* after the read. The read content of this bitfield is equivalent to EVENT_BUFFER_MASKED.EBM

6.4.4.3.38 FC_CORE Software event 5 wait and clear command register. (SW_EVENT_5_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 will trigger FC_CORE sofware event 5 and gate the FC_CORE clock until at least one unmasked event occurs. In addition, EVENT_BUFFER.EB[i] bits are cleared if EVT_MASK.EM[i]=*0b1* after the read. The read content of this bitfield is equivalent to EVENT_BUFFER_MASKED.EBM

6.4.4.3.39 FC_CORE Software event 6 wait and clear command register. (SW_EVENT_6_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 will trigger FC_CORE sofware event 6 and gate the FC_CORE clock until at least one unmasked event occurs. In addition, EVENT_BUFFER.EB[i] bits are cleared if EVT_MASK.EM[i]=*0b1* after the read. The read content of this bitfield is equivalent to EVENT_BUFFER_MASKED.EBM

6.4.4.3.40 FC_CORE Software event 7 wait and clear command register. (SW_EVENT_7_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 will trigger FC_CORE sofware event 7 and gate the FC_CORE clock until at least one unmasked event occurs. In addition, EVENT_BUFFER.EB[i] bits are cleared if EVT_MASK.EM[i]=*0b1* after the read. The read content of this bitfield is equivalent to EVENT_BUFFER_MASKED.EBM

6.4.4.3.41 Remote software event 0 trigger command register. (REMOTE_SW_EVENT_0_TRIG)

Reset value: 0x0000

Host access bus: PERIPH

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

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

Writing any value will trigger SoC software event 0

6.4.4.3.42 Remote software event 1 trigger command register. (REMOTE_SW_EVENT_1_TRIG)

Reset value: 0x0000

Host access bus: PERIPH

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

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

Writing any value will trigger SoC software event 1

6.4.4.3.43 Remote software event 2 trigger command register. (REMOTE_SW_EVENT_2_TRIG)

Reset value: 0x0000

Host access bus: PERIPH

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

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

Writing any value will trigger SoC software event 2

6.4.4.3.44 Remote software event 3 trigger command register. (REMOTE_SW_EVENT_3_TRIG)

Reset value: 0x0000

Host access bus: PERIPH

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

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

Writing any value will trigger SoC software event 3

6.4.4.3.45 Remote software event 4 trigger command register. (REMOTE_SW_EVENT_4_TRIG)

Reset value: 0x0000

Host access bus: PERIPH

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

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

Writing any value will trigger SoC software event 4

6.4.4.3.46 Remote software event 5 trigger command register. (REMOTE_SW_EVENT_5_TRIG)

Reset value: 0x0000

Host access bus: PERIPH

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

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

Writing any value will trigger SoC software event 5

6.4.4.3.47 Remote software event 6 trigger command register. (REMOTE_SW_EVENT_6_TRIG)

Reset value: 0x0000

Host access bus: PERIPH

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

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

Writing any value will trigger SoC software event 6

6.4.4.3.48 Remote software event 7 trigger command register. (REMOTE_SW_EVENT_7_TRIG)

Reset value: 0x0000

Host access bus: PERIPH

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

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

Writing any value will trigger SoC software event 7

6.4.4.3.49 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.

6.4.5 FC Memory protection unit

FC_MPU component manages the protection of the FC L1, L2 and peripheral memory mapped areas. Protection rules are positive. i.e. If there is no rule in place access will be denied. The rules protect access of these areas from FC code running in user mode, peripherals and all cluster code. By default the MPU is disabled. It would normally be enabled by a second stage bootloader that would manage loading user code.

6.4.5.1 FC Memory protection unit registers

Name	Aliased address	Size	Type	Access	Default	Description
MPU_ENABLE	0x00204400	32	Config	R/W	*0x0000*	MPU enable configuration register
FC_TCDM_RULE0	0x00204440	32	Config	R/W	*0x0000*	FC address filter rule 0
FC_TCDM_RULE1	0x00204444	32	Config	R/W	*0x0000*	FC address filter rule 1
FC_TCDM_RULE2	0x00204448	32	Config	R/W	*0x0000*	FC address filter rule 2
FC_TCDM_RULE3	0x0020444C	32	Config	R/W	*0x0000*	FC address filter rule 3
FC_TCDM_RULE4	0x00204450	32	Config	R/W	*0x0000*	FC address filter rule 4
FC_TCDM_RULE5	0x00204454	32	Config	R/W	*0x0000*	FC address filter rule 5
FC_TCDM_RULE6	0x00204458	32	Config	R/W	*0x0000*	FC address filter rule 6
FC_TCDM_RULE7	0x0020445C	32	Config	R/W	*0x0000*	FC address filter rule
L2_RULE0	0x00204480	32	Config	R/W	*0x0000*	L2 address filter rule 0
L2_RULE1	0x00204484	32	Config	R/W	*0x0000*	L2 address filter rule 1
L2_RULE2	0x00204488	32	Config	R/W	*0x0000*	L2 address filter rule 2
L2_RULE3	0x0020448C	32	Config	R/W	*0x0000*	L2 address filter rule 3
L2_RULE4	0x00204490	32	Config	R/W	*0x0000*	L2 address filter rule 4
L2_RULE5	0x00204494	32	Config	R/W	*0x0000*	L2 address filter rule 5
L2_RULE6	0x00204498	32	Config	R/W	*0x0000*	L2 address filter rule 6
L2_RULE7	0x0020449C	32	Config	R/W	*0x0000*	L2 address filter rule
APB_RULE0	0x002044C0	32	Config	R/W	*0x0000*	External peripheral (APB) address filter rule 0
APB_RULE1	0x002044C4	32	Config	R/W	*0x0000*	External peripheral (APB) address filter rule 1
APB_RULE2	0x002044C8	32	Config	R/W	*0x0000*	External peripheral (APB) address filter rule 2
APB_RULE3	0x002044CC	32	Config	R/W	*0x0000*	External peripheral (APB) address filter rule 3
APB_RULE4	0x002044D0	32	Config	R/W	*0x0000*	External peripheral (APB) address filter rule 4
APB_RULE5	0x002044D4	32	Config	R/W	*0x0000*	External peripheral (APB) address filter rule 5
APB_RULE6	0x002044D8	32	Config	R/W	*0x0000*	External peripheral (APB) address filter rule 6
APB_RULE7	0x002044DC	32	Config	R/W	*0x0000*	External peripheral (APB) address filter rule

Table 65. FC Memory protection unit registers table

6.4.5.2 FC Memory protection unit registers details

6.4.5.2.1 MPU enable configuration register (MPU_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															ENABLE

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

MPU enable status

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

6.4.5.2.2 FC address filter rule (FC_TCDM_RULE)

Reset value: 0x0000

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

Bits 31:30 - **A** *(R/W)*

Area (MSB indicates aliased memory area, LSB indicates L1 memory or peripheral memory)

- *0b00* - L1 memory (0x01B00000)
- *0b01* - Peripheral memory (0x1B200000)
- *0b10* - L1 memory aliased (0x10000000)
- *0b11* - Peripheral memory aliased (0x10200000)

The unprotected area will start at the A base + 64K * base and will be 64K * size long

Bits 28:15 - **BASE** *(R/W)*

Area address base in 64kB pages

Bits 13:1 - **SIZE** *(R/W)*

Area size in 64KB pages

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

Rule state

- *0b0*: Rule diabled
- *0b1*: Rule enabled

6.4.5.2.3 L2 address filter rule (L2_RULE)

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
A		Reserved	BASE												

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BASE	Reserved	SIZE													ST

Bits 31:30 - **A** *(R/W)*

Area

Not used. Always 0x1C000000.

Bits 28:15 - **BASE** *(R/W)*

Area address base in 64kB pages

Bits 13:1 - **SIZE** *(R/W)*

Area size in 64KB pages

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

Rule state

- *0b0*: Rule diabled
- *0b1*: Rule enabled

6.4.5.2.4 External peripheral (APB) address filter rule (APB_RULE)

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
A		Reserved	BASE												

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BASE	Reserved	SIZE													ST

Bits 31:30 - **A** *(R/W)*

Area

- *0b00* - L1 memory (0x01B00000)
- *0b01* - Peripheral memory (0x1B200000)
- *0b10* - APB memory (0x1A100000)

The unprotected area will start at the A base + 64K * base and will be 64K * size long

Bits 28:15 - **BASE** *(R/W)*

Area address base in 64kB pages

Bits 13:1 - **SIZE** *(R/W)*

Area size in 64KB pages

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

Rule state

- *0b0*: Rule diabled
- *0b1*: Rule enabled

6.4.6 FC instruction cache control unit

FC_ICACHE_CTRL component manages the following features:

- Bypassable FC instruction cache controller
- Flush and selective flush commands
- FC instruction cache pending action status flag

6.4.6.1 FC instruction cache control unit registers

Name	Address	Aliased address	Size	Type	Access	Default	Description
ENABLE	0x1B201400	0x00201400	32	Config	W	*0x0000*	FC instruction cache unit enable configuration register.
FLUSH	0x1B201404	0x00201404	32	Config	W	*0x0000*	FC instruction cache unit flush command register.
SEL_FLUSH	0x1B201408	0x00201408	32	Config	W	*0x0000*	FC instruction cache unit selective flush command register.
STATUS	0x1B20140C	0x0020140C	32	Status	R	*0x0000*	FC instruction cache unit status register.

Table 66. FC instruction cache control unit registers table

6.4.6.2 FC instruction cache control unit registers details

6.4.6.2.1 FC 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)*

FC instruction cache enable configuration bitfield:

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

6.4.6.2.2 FC 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)*

FC instruction cache fully flush command.

6.4.6.2.3 FC 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)*

FC instruction cache selective flush address configuration bitfield.

6.4.6.2.4 FC instruction cache unit status register. (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															STATUS

Bit 0 - **STATUS** *(R)*

FC instruction cache pending action status flag:

- *0b0*: no pending caching action
- *0b1*: pending caching action

6.4.7 FC secured RISCY core

RI5CY Debug component manages the following features:

- controls break and single step RI5CY core execution modes
- configurable execution behavior on RI5CY core exception occurrence
- access to PC, GPR and CSR RI5CY core registers
- no HW breakpoint are provided

6.4.7.1 FC Core 0 (Debug Unit) registers

Name	Address	Aliased address	Size	Type	Access	Default	Description
CTRL	0x1B300000	0x00300000	32	Config	R/W	*0x0000*	Debug control configuration register.
HIT	0x1B300004	0x00300004	32	Config	R/W	*0x0000*	Debug hit status register.
IE	0x1B300008	0x00300008	32	Config	R/W	*0x0000*	Debug exception trap enable configuration register.
CAUSE	0x1B30000C	0x0030000C	32	Config	R	*0x0000*	Debug trap cause status register.
GPR0	0x1B300400	0x00300400	32	Config	R/W	*0x0000*	Core general purpose register 0 value register.
GPR1	0x1B300404	0x00300404	32	Config	R/W	*0x0000*	Core general purpose register 1 value register.
GPR2	0x1B300408	0x00300408	32	Config	R/W	*0x0000*	Core general purpose register 2 value register.
GPR3	0x1B30040C	0x0030040C	32	Config	R/W	*0x0000*	Core general purpose register 3 value register.
GPR4	0x1B300410	0x00300410	32	Config	R/W	*0x0000*	Core general purpose register 4 value register.

Name	Address	Aliased address	Size	Type	Access	Default	Description
GPR5	0x1B300414	0x00300414	32	Config	R/W	*0x0000*	Core general purpose register 5 value register.
GPR6	0x1B300418	0x00300418	32	Config	R/W	*0x0000*	Core general purpose register 6 value register.
GPR7	0x1B30041C	0x0030041C	32	Config	R/W	*0x0000*	Core general purpose register 7 value register.
GPR8	0x1B300420	0x00300420	32	Config	R/W	*0x0000*	Core general purpose register 8 value register.
GPR9	0x1B300424	0x00300424	32	Config	R/W	*0x0000*	Core general purpose register 9 value register.
GPR10	0x1B300428	0x00300428	32	Config	R/W	*0x0000*	Core general purpose register 10 value register.
GPR11	0x1B30042C	0x0030042C	32	Config	R/W	*0x0000*	Core general purpose register 11 value register.
GPR12	0x1B300430	0x00300430	32	Config	R/W	*0x0000*	Core general purpose register 12 value register.
GPR13	0x1B300434	0x00300434	32	Config	R/W	*0x0000*	Core general purpose register 13 value register.
GPR14	0x1B300438	0x00300438	32	Config	R/W	*0x0000*	Core general purpose register 14 value register.
GPR15	0x1B30043C	0x0030043C	32	Config	R/W	*0x0000*	Core general purpose register 15 value register.
GPR16	0x1B300440	0x00300440	32	Config	R/W	*0x0000*	Core general purpose register 16 value register.
GPR17	0x1B300444	0x00300444	32	Config	R/W	*0x0000*	Core general purpose register 17 value register.
GPR18	0x1B300448	0x00300448	32	Config	R/W	*0x0000*	Core general purpose register 18 value register.
GPR19	0x1B30044C	0x0030044C	32	Config	R/W	*0x0000*	Core general purpose register 19 value register.
GPR20	0x1B300450	0x00300450	32	Config	R/W	*0x0000*	Core general purpose register 20 value register.
GPR21	0x1B300454	0x00300454	32	Config	R/W	*0x0000*	Core general purpose register 21 value register.
GPR22	0x1B300458	0x00300458	32	Config	R/W	*0x0000*	Core general purpose register 22 value register.
GPR23	0x1B30045C	0x0030045C	32	Config	R/W	*0x0000*	Core general purpose register 23 value register.
GPR24	0x1B300460	0x00300460	32	Config	R/W	*0x0000*	Core general purpose register 24 value register.
GPR25	0x1B300464	0x00300464	32	Config	R/W	*0x0000*	Core general purpose register 25 value register.
GPR26	0x1B300468	0x00300468	32	Config	R/W	*0x0000*	Core general purpose register 26 value register.
GPR27	0x1B30046C	0x0030046C	32	Config	R/W	*0x0000*	Core general purpose register 27 value register.
GPR28	0x1B300470	0x00300470	32	Config	R/W	*0x0000*	Core general purpose register 28 value register.
GPR29	0x1B300474	0x00300474	32	Config	R/W	*0x0000*	Core general purpose register 29 value register.
GPR30	0x1B300478	0x00300478	32	Config	R/W	*0x0000*	Core general purpose register 30 value register.
GPR31	0x1B30047C	0x0030047C	32	Config	R/W	*0x0000*	Core general purpose register 31 value register.
NPC	0x1B302000	0x00302000	32	Config	R/W	*0x0000*	Debug next program counter value register.
PPC	0x1B302004	0x00302004	32	Config	R	*0x0000*	Debug previous program counter value register.
CSR_USTATUS	0x1B304000	0x00304000	32	Config	R/W	*0x0000*	Core CSR user status value register.
CSR_UTVEC	0x1B304014	0x00304014	32	Config	R/W	*0x0000*	Core CSR user vector-trap base address value register.
CSR_UHARTID	0x1B304050	0x00304050	32	Config	R	*0x0000*	Core CSR user privilege mode hardware thread ID status register.
CSR_UEPC	0x1B304104	0x00304104	32	Config	R/W	*0x0000*	Core CSR user exception program counter value register.
CSR_UCAUSE	0x1B304108	0x00304108	32	Config	R/W	*0x0000*	Core CSR user trap cause value register.
CSR_MSTATUS	0x1B304C00	0x00304C00	32	Config	R/W	*0x0000*	Core CSR machine status value register.
CSR_MTVEC	0x1B304C14	0x00304C14	32	Config	R/W	*0x0000*	Core CSR machine vector-trap base address value register.
CSR_MEPC	0x1B304D04	0x00304D04	32	Config	R/W	*0x0000*	Core CSR machine exception program counter value register.
CSR_MCAUSE	0x1B304D08	0x00304D08	32	Config	R/W	*0x0000*	Core CSR machine trap cause value register.
CSR_PCCR	0x1B305E00	0x00305E00	32	Config	R/W	*0x0000*	Core CSR performance counter counter register.

Name	Address	Aliased address	Size	Type	Access	Default	Description
CSR_PCER	0x1B305E80	0x00305E80	32	Config	R/W	*0x0000*	Core CSR performance counter enable configuration register.
CSR_PCMR	0x1B305E84	0x00305E84	32	Config	R/W	*0x0000*	Core CSR performance counter mode configuration register.
CSR_HWLP0S	0x1B305EC0	0x00305EC0	32	Config	R/W	*0x0000*	Core CSR hardware loop 0 start configuration register.
CSR_HWLP0E	0x1B305EC4	0x00305EC4	32	Config	R/W	*0x0000*	Core CSR hardware loop 0 end configuration register.
CSR_HWLP0C	0x1B305EC8	0x00305EC8	32	Config	R/W	*0x0000*	Core CSR hardware loop 0 counter configuration register.
CSR_HWLP1S	0x1B305ED0	0x00305ED0	32	Config	R/W	*0x0000*	Core CSR hardware loop 1 start configuration register.
CSR_HWLP1E	0x1B305ED4	0x00305ED4	32	Config	R/W	*0x0000*	Core CSR hardware loop 1 end configuration register.
CSR_HWLP1C	0x1B305ED8	0x00305ED8	32	Config	R/W	*0x0000*	Core CSR hardware loop 1 counter configuration register.
CSR_PRIVLV	0x1B307040	0x00307040	32	Config	R	*0x0000*	Core CSR current privilege level status register.
CSR_MHARTID	0x1B307C50	0x00307C50	32	Config	R	*0x0000*	Core CSR machine privilege mode hardware thread ID status register.

Table 67. FC Core 0 (Debug Unit) registers table

6.4.7.2 FC secured RISCY core registers details

6.4.7.2.1 Debug control configuration register. (CTRL)

Reset value: 0x0000

Always accessible, even when the RI5CY core is running.

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

Bit 16 - **HALT** *(W)*

Debug mode configuration bitfield:

- *0b0*: exit debug - exits debug mode
- *0b1*: enter debug – enters debug mode breaking code execution

Bit 16 - **HALT_STATUS** *(R)*

Debug mode status bitfield:

- *0b0*: running mode
- *0b1*: debug mode

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

Single step mode configuration bitfield:

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

6.4.7.2.2 Debug hit status register. (HIT)

Reset value: 0x0000

Always accessible, even when the RI5CY core is running.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															SLEEP

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															SSTH

Bit 16 - **SLEEP** *(R)*

Sleep mode status bitfield:

- *0b0*: running - core is in running state
- *0b1*: sleeping – core is in sleeping state and waits for en event to wake up

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

Single step hit status bitfield:

- *0b0*: disabled - single step mode disabled
- *0b1*: enabled – single step mode enabled

Sticky bit that must be cleared by external debugger.

6.4.7.2.3 Debug exception trap enable configuration register. (IE)

Reset value: 0x0000

Always accessible, even when the RI5CY core is running.

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				ECALL	Reserved			ELSU_DUP	Reserved	ELSU	Reserved	EBRK	EILL	Reserved	

Bit 11 - **ECALL** *(R/W)*

Environment call trap configuration bitfield:

- *0b0*: normal - normal exception behavior mode
- *0b1*: cause trap - exception causes trap and core switch into debug mode

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

Load/store access fault trap configuration bitfield:

- *0b0*: normal - normal exception behavior mode
- *0b1*: cause trap – exception causes trap and core switch into debug mode

This bitfield is duplicates the ELSU bitfield.

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

Load/store access fault trap configuration bitfield:

- *0b0*: normal - normal exception behavior mode
- *0b1*: cause trap – exception causes trap and core switch into debug mode

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

Environment break trap configuration bitfield:

- *0b0*: normal - normal exception behavior mode
- *0b1*: cause trap – exception causes trap and core switch into debug mode

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

Illegal instruction trap configuration bitfield:

- *0b0*: normal - normal exception behavior mode
- *0b1*: cause trap – exception causes trap and core switch into debug mode

6.4.7.2.4 Debug trap cause status register. (CAUSE)

Reset value: 0x0000

Always accessible, even when the RI5CY core is running.

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

Bit 31 - **IRQ** *(R)*

Core in debug mode due to interrupt trap status bitfield:

- *0b0*: false
- *0b1*: true

Bits 4:0 - **CAUSE** *(R)*

Exception ID bitfield. If IRQ is *0b1* contains interrupt number otherwise:

- *0x2*: sigill - Illegal Instruction
- *0x3*: sigtrap - breakpoint
- *0xB*: sigecall - eCall user mode
- *0xB* sigecall - eCall machine mode
- *0x1F* sigstop - core was halted by an external signal

6.4.7.2.5 Core general purpose register 0 value register. (GPR0)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPR0															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPR0															

Bits 31:0 - **GPR0** *(R/W)*

General purpose register 0 value bitfield.

6.4.7.2.6 Core general purpose register 1 value register. (GPR1)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPR1															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPR1															

Bits 31:0 - **GPR1** *(R/W)*

General purpose register 1 value bitfield.

6.4.7.2.7 Core general purpose register 2 value register. (GPR2)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPR2															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPR2															

Bits 31:0 - **GPR2** *(R/W)*

General purpose register 2 value bitfield.

6.4.7.2.8 Core general purpose register 3 value register. (GPR3)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPR3															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPR3															

Bits 31:0 - **GPR3** *(R/W)*

General purpose register 3 value bitfield.

6.4.7.2.9 Core general purpose register 4 value register. (GPR4)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPR4															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPR4															

Bits 31:0 - **GPR4** *(R/W)*

General purpose register 4 value bitfield.

6.4.7.2.10 Core general purpose register 5 value register. (GPR5)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPR5															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPR5															

Bits 31:0 - **GPR5** *(R/W)*

General purpose register 5 value bitfield.

6.4.7.2.11 Core general purpose register 6 value register. (GPR6)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPR6															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPR6															

Bits 31:0 - **GPR6** *(R/W)*

General purpose register 6 value bitfield.

6.4.7.2.12 Core general purpose register 7 value register. (GPR7)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPR7															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPR7															

Bits 31:0 - **GPR7** *(R/W)*

General purpose register 7 value bitfield.

6.4.7.2.13 Core general purpose register 8 value register. (GPR8)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPR8															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPR8															

Bits 31:0 - **GPR8** *(R/W)*

General purpose register 8 value bitfield.

6.4.7.2.14 Core general purpose register 9 value register. (GPR9)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPR9															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPR9															

Bits 31:0 - **GPR9** *(R/W)*

General purpose register 9 value bitfield.

6.4.7.2.15 Core general purpose register 10 value register. (GPR10)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPR10															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPR10															

Bits 31:0 - **GPR10** *(R/W)*

General purpose register 10 value bitfield.

6.4.7.2.16 Core general purpose register 11 value register. (GPR11)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPR11															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPR11															

Bits 31:0 - **GPR11** *(R/W)*

General purpose register 11 value bitfield.

6.4.7.2.17 Core general purpose register 12 value register. (GPR12)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPR12															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPR12															

Bits 31:0 - **GPR12** *(R/W)*

General purpose register 12 value bitfield.

6.4.7.2.18 Core general purpose register 13 value register. (GPR13)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPR13															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPR13															

Bits 31:0 - **GPR13** *(R/W)*

General purpose register 13 value bitfield.

6.4.7.2.19 Core general purpose register 14 value register. (GPR14)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPR14															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPR14															

Bits 31:0 - **GPR14** *(R/W)*

General purpose register 14 value bitfield.

6.4.7.2.20 Core general purpose register 15 value register. (GPR15)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPR15															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPR15															

Bits 31:0 - **GPR15** *(R/W)*

General purpose register 15 value bitfield.

6.4.7.2.21 Core general purpose register 16 value register. (GPR16)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPR16															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPR16															

Bits 31:0 - **GPR16** *(R/W)*

General purpose register 16 value bitfield.

6.4.7.2.22 Core general purpose register 17 value register. (GPR17)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPR17															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPR17															

Bits 31:0 - **GPR17** *(R/W)*

General purpose register 17 value bitfield.

6.4.7.2.23 Core general purpose register 18 value register. (GPR18)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPR18															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPR18															

Bits 31:0 - **GPR18** *(R/W)*

General purpose register 18 value bitfield.

6.4.7.2.24 Core general purpose register 19 value register. (GPR19)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPR19															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPR19															

Bits 31:0 - **GPR19** *(R/W)*

General purpose register 19 value bitfield.

6.4.7.2.25 Core general purpose register 20 value register. (GPR20)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPR20															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPR20															

Bits 31:0 - **GPR20** *(R/W)*

General purpose register 20 value bitfield.

6.4.7.2.26 Core general purpose register 21 value register. (GPR21)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPR21															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPR21															

Bits 31:0 - **GPR21** *(R/W)*

General purpose register 21 value bitfield.

6.4.7.2.27 Core general purpose register 22 value register. (GPR22)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPR22															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPR22															

Bits 31:0 - **GPR22** *(R/W)*

General purpose register 22 value bitfield.

6.4.7.2.28 Core general purpose register 23 value register. (GPR23)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPR23															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPR23															

Bits 31:0 - **GPR23** *(R/W)*

General purpose register 23 value bitfield.

6.4.7.2.29 Core general purpose register 24 value register. (GPR24)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPR24															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPR24															

Bits 31:0 - **GPR24** *(R/W)*

General purpose register 24 value bitfield.

6.4.7.2.30 Core general purpose register 25 value register. (GPR25)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPR25															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPR25															

Bits 31:0 - **GPR25** *(R/W)*

General purpose register 25 value bitfield.

6.4.7.2.31 Core general purpose register 26 value register. (GPR26)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPR26															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPR26															

Bits 31:0 - **GPR26** *(R/W)*

General purpose register 26 value bitfield.

6.4.7.2.32 Core general purpose register 27 value register. (GPR27)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPR27															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPR27															

Bits 31:0 - **GPR27** *(R/W)*

General purpose register 27 value bitfield.

6.4.7.2.33 Core general purpose register 28 value register. (GPR28)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPR28															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPR28															

Bits 31:0 - **GPR28** *(R/W)*

General purpose register 28 value bitfield.

6.4.7.2.34 Core general purpose register 29 value register. (GPR29)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPR29															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPR29															

Bits 31:0 - **GPR29** *(R/W)*

General purpose register 29 value bitfield.

6.4.7.2.35 Core general purpose register 30 value register. (GPR30)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPR30															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPR30															

Bits 31:0 - **GPR30** *(R/W)*

General purpose register 30 value bitfield.

6.4.7.2.36 Core general purpose register 31 value register. (GPR31)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GPR31															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
GPR31															

Bits 31:0 - **GPR31** *(R/W)*

General purpose register 31 value bitfield.

6.4.7.2.37 Debug next program counter value register. (NPC)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

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

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

Next program counter value bitfield.

6.4.7.2.38 Debug previous program counter value register. (PPC)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

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

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

Previous program counter value bitfield.

6.4.7.2.39 Core CSR user status value register. (CSR_USTATUS)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

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											UPIE	Reserved			UIE

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

User privilege mode previous interrupt enable value bitfield. When an interrupis encountered, UPIE will store the value existing in UIE. When uret instruction is executed, the value of UPIE is restored into UIE.

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

User privilege mode interrupt enable configuration bitfield:

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

6.4.7.2.40 Core CSR user vector-trap base address value register. (CSR_UTVEC)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

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

Bits 31:8 - **UTVEC** *(R/W)*

Machine trap-vector base address value bitfield. When an exception is encountered, the core jumps to the corresponding handler using the content of the MTVEC as base address.

6.4.7.2.41 Core CSR user privilege mode hardware thread ID status register. (CSR_UHARTID)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

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					CLUSTER_ID						Reserved	CORE_ID			

Bits 10:5 - **CLUSTER_ID** *(R)*

Cluster ID value bitfield.

Bits 3:0 - **CORE_ID** *(R)*

RI5CY core ID value bitfield.

6.4.7.2.42 Core CSR user exception program counter value register. (CSR_UEPC)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

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

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

Machine exception program counter value bitfield. When an exception is encountered, the current program counter is saved in MEPC, and the core jumps to the exception address. When an mret instruction is executed, the value from MEPC is restored to the current program counter.

6.4.7.2.43 Core CSR user trap cause value register. (CSR_UCAUSE)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

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

Bit 31 - **IRQ** *(R)*

Core triggered an exception due to interrupt status bitfield:

- *0b0*: false
- *0b1*: true

Bits 4:0 - **CAUSE** *(R)*

Exception ID bitfield.

6.4.7.2.44 Core CSR machine status value register. (CSR_MSTATUS)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

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			MPP		Reserved			MPIE	Reserved		UPIE	MIE	Reserved		UIE

Bits 12:11 - **MPP** *(R/W)*

Machine privilege mode previous privilege mode value bitfield:

- *0b00*: User mode
- *0b11*: Machine mode

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

Machine privilege mode previous interrupt enable value bitfield. When an interrupis encountered, MPIE will store the value existing in MIE. When mret instruction is executed, the value of MPIE is restored into MIE.

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

User privilege mode previous interrupt enable value bitfield. When an interrupis encountered, UPIE will store the value existing in UIE. When uret instruction is executed, the value of UPIE is restored into UIE.

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

Machine privilege mode interrupt enable configuration bitfield:

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

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

User privilege mode interrupt enable configuration bitfield:

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

6.4.7.2.45 Core CSR machine vector-trap base address value register. (CSR_MTVEC)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

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

Bits 31:8 - **MTVEC** *(R/W)*

Machine trap-vector base address value bitfield. When an exception is encountered, the core jumps to the corresponding handler using the content of the MTVEC as base address.

6.4.7.2.46 Core CSR machine exception program counter value register. (CSR_MEPC)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

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

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

Machine exception program counter value bitfield. When an exception is encountered, the current program counter is saved in MEPC, and the core jumps to the exception address. When an mret instruction is executed, the value from MEPC is restored to the current program counter.

6.4.7.2.47 Core CSR machine trap cause value register. (CSR_MCAUSE)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

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

Bit 31 - **IRQ** *(R)*

Core triggered an exception due to interrupt status bitfield:

- *0b0*: false
- *0b1*: true

Bits 4:0 - **CAUSE** *(R)*

Exception ID bitfield. If IRQ is *0b1* contains interrupt number otherwise:

- *0x2*: sigill - Illegal Instruction
- *0x3*: sigtrap - breakpoint
- *0xB*: sigecall - eCall user mode
- *0xB* sigecall - eCall machine mode
- *0x1F* sigstop - core was halted by an external signal

6.4.7.2.48 Core CSR performance counter counter register. (CSR_PCCR)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

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

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

Program counter counter value bitfield.

6.4.7.2.49 Core CSR performance counter enable configuration register. (CSR_PCER)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

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

Bits 20:0 - **PCER** *(R/W)*

See documentation on [RI5CY core] for details.

6.4.7.2.50 Core CSR performance counter mode configuration register. (CSR_PCMR)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

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														GE	SAT

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

Performance counter activation configuration bitfield:

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

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

Performance counter saturation mode configuration bitfield:

- *0b0*: wrap around - wrap-around mode
- *0b1*: saturation – saturation mode

6.4.7.2.51 Core CSR hardware loop 0 start configuration register. (CSR_HWLP0S)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

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

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

Hardware loop start address configuration bitfield.

6.4.7.2.52 Core CSR hardware loop 0 end configuration register. (CSR_HWLP0E)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

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

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

Hardware loop end address configuration bitfield.

6.4.7.2.53 Core CSR hardware loop 0 counter configuration register. (CSR_HWLP0C)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

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

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

Hardware loop counter configuration bitfield.

6.4.7.2.54 Core CSR hardware loop 1 start configuration register. (CSR_HWLP1S)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

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

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

Hardware loop start address configuration bitfield.

6.4.7.2.55 Core CSR hardware loop 1 end configuration register. (CSR_HWLP1E)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

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

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

Hardware loop end address configuration bitfield.

6.4.7.2.56 Core CSR hardware loop 1 counter configuration register. (CSR_HWLP1C)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

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

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

Hardware loop counter configuration bitfield.

6.4.7.2.57 Core CSR current privilege level status register. (CSR_PRIVLV)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

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														PLEV	

Bits 1:0 - **PLEV** *(R)*

Current privilege level status bitfield:

- *0b00*: User mode
- *0b11*: Machine mode

6.4.7.2.58 Core CSR machine privilege mode hardware thread ID status register. (CSR_MHARTID)

Reset value: 0x0000

Only accessible if the RI5CY core is halted.

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					CLUSTER_ID						Reserved	CORE_ID			

Bits 10:5 - **CLUSTER_ID** *(R)*

Cluster ID value bitfield.

Bits 3:0 - **CORE_ID** *(R)*

RI5CY core ID value bitfield.

6.5 UDMA specific stream pre-processing protocol for SPIM and I2C interfaces

6.5.1 SPI Master stream pre-processing protocol

SPI Master defines basic types of streams, each of them begins with a chip-select assertion and ends with chip-select de-assertion.

All SPI Master transfers can be assembled into a coherent buffer of commands and data. With different combinations of available commands, any type of SPI transfer can be generated. GAP8's SPI Master interface fetches both commands and data from L2 memory. In this way UDMA stream pre-processing protocol component can create complex SPI transfer fully autonomously and without any intervention of the Fabric Controller RI5CY core. The code below shows an example of 2 SPI transfer buffers that generate a read and a write on a standard SPI flash memory.

```
PULP_L2_DATA unsigned int spi_cmd_readdata[] = {
  SPI_CMD_SOT       (SPI_CMD_SOT_CS0),
  SPI_CMD_SEND_CMD  (0x13, SPI_CMD_CMD_SIZE(0x8), SPI_CMD_QPI_DIS), // send command read memory 4read
  SPI_CMD_SEND_ADDR(32, SPI_CMD_QPI_DIS),
  (SECTOR_4KB_NUM * SECTOR_4KB_SIZE),
  SPI_CMD_RX_DATA   (PAGE_PROGRAM_BUFFER_SIZE*8, SPI_CMD_QPI_DIS, SPI_CMD_BYTE_ALIGN_ENA),
  SPI_CMD_EOT       (SPI_CMD_EOT_EVENT_ENA)
};

PULP_L2_DATA unsigned int spi_cmd_writedata1[] = {
  SPI_CMD_SOT       (SPI_CMD_SOT_CS0),
  SPI_CMD_SEND_CMD  (0x12, SPI_CMD_CMD_SIZE(0x8), SPI_CMD_QPI_DIS) , // 4pp page program
  SPI_CMD_SEND_ADDR(32, SPI_CMD_QPI_DIS),
  (SECTOR_4KB_NUM * SECTOR_4KB_SIZE),
  SPI_CMD_TX_DATA   (PAGE_PROGRAM_BUFFER_SIZE*8, SPI_CMD_QPI_DIS, SPI_CMD_BYTE_ALIGN_ENA),
  0x83828180, 0x87868584, 0x8b8a8988, 0x8f8e8d8c, 0x93929190, 0x97969594, 0x9b9a9998, 0x9f9e9d9c,
  0xa3a2a1a0, 0xa7a6a5a4, 0xabaaa9a8, 0xafaeadac, 0xb3b2b1b0, 0xb7b6b5b4, 0xbbbab9b8, 0xbfbebdbc,
  0xc3c2c1c0, 0xc7c6c5c4, 0xcbcac9c8, 0xcfcecdcc, 0xd3d2d1d0, 0xd7d6d5d4, 0xdbdad9d8, 0xdfdedddc,
  0xe3e2e1e0, 0xe7e6e5e4, 0xebeae9e8, 0xefeeedec, 0xf3f2f1f0, 0xf7f6f5f4, 0xfbfaf9f8, 0xfffefdfc,
  0x03020100, 0x07060504, 0x0b0a0908, 0x0f0e0d0c, 0x13121110, 0x17161514, 0x1b1a1918, 0x1f1e1d1c,
  0x23222120, 0x27262524, 0x2b2a2928, 0x2f2e2d2c, 0x33323130, 0x37363534, 0x3b3a3938, 0x3f3e3d3c,
  0x43424140, 0x47464544, 0x4b4a4948, 0x4f4e4d4c, 0x53525150, 0x57565554, 0x5b5a5958, 0x5f5e5d5c,
```

```
  0x63626160, 0x67666564, 0x6b6a6968, 0x6f6e6d6c, 0x73727170, 0x77767574, 0x7b7a7978, 0x7f7e7d7c,
  SPI_CMD_EOT       (SPI_CMD_EOT_EVENT_ENA),
};
```

6.5.2 I2C stream pre-processing protocol

I^{2}C defines basic types of streams, each of which begins with a START and ends with a STOP:

- Single message where a master writes data to a slave;
- Single message where a master reads data from a slave;
- Combined messages, where a master issues at least two reads and/or writes to one or more slaves.

All I^{2}C transfers can be assembled into a coherent buffer of commands and data. With different combinations of available commands, any type of I^{2}C transfer can be generated. GAP8's I^{2}C interface fetches both commands and data from L2 memory. In this way UDMA stream pre-processing protocol component can create complex I^{2}C transfer fully autonomously and without any intervention of the Fabric Controller RI5CY core. The code below shows an example of a I^{2}C transfer buffer that generates a write on a 24LC1024 EEProm.

```
char i2c_cmd_buffer[] = {
   I2C_CMD_START,
   I2C_CMD_WR,0xA4,
   I2C_CMD_WR,0x00,
   I2C_CMD_WR,0x00,
   I2C_CMD_RPT,0x10,
   I2C_CMD_WR,0x00,0x01,0x02,0x03,
              0x04,0x05,0x06,0x07,
              0x08,0x09,0x0A,0x0B,
              0x0C,0x0D,0x0E,0x0F,
   I2C_CMD_STOP,
   I2C_CMD_WAIT,0x10,
   I2C_CMD_START,
   I2C_CMD_WR,0xA4,
   I2C_CMD_WR,0x00,
   I2C_CMD_WR,0x00,
   I2C_CMD_START,
   I2C_CMD_WR,0xA5,
   I2C_CMD_RPT,0x0F,
   I2C_CMD_RD_ACK,
   I2C_CMD_RD_NACK,
   I2C_CMD_STOP
};
```

The command sequence starts by generating a start bit on the bus followed by a byte write and waiting for the slave acknowledge. The first byte, following the I^{2}C standard sends the 7 bit address with the last bit coding the access type(0 for write 1 for read) so in this case 0x52 is the address and access is a write. The following two writes are the internal address of the EEProm (0x0000). The following instructions tell the I^{2}C IP to repeat the next instructions 16 times. The instruction to be repeated is the write and the data for each write instruction is queued. Here we do write 16 bytes 0x00, 0x01...0x0F. The I2C_CMD_STOP generates the stop bits and ends the transfer. I2C_CMD_WAIT waits some I2C cycles (in this case 16) and the following I2C_CMD_START restart a new I^{2}C transfer. The start is followed by the address of the peripheral and then by the internal address. I2C_CMD_START generates a restart condition needed by the EEPRom and then sends the 7 bit address but this time with a read flag(0xA5). The next command says to read 15 bytes and sends acknowledge at each byte and then read the last byte followed by a not acknowledge to inform the slave that we are done with the transfer. A stop bit then finalizes the transfer. All the commands are read through the TX port while each read pushes data to the RX channel.

7 e-Fuses content

The e-fuses are accessed via the e-fuse registers. The e-fuse registers are indexed by the e-fuse register ID (RegID) and are read and written at an offset from the read and write registers. Each register is 8 bits long however the register offset is calculated as RegID * 4. i.e. the register occupies the LSB of a word where the other bytes are unused.

7.0.3 FC e-fuse details

Name	RegId	No of registers	Description
INFO	0	1	Info register 1
INFO2	1	1	Info register 2
AES_KEY	2	16	Secure boot AES key
AES_IV	18	16	Secure boot AES IV
WAIT_XTAL_DELTA	26	2	XTAL delta threshold
WAIT_XTAL_MIN	28	1	XTAL minimum stabilization time
WAIT_XTAL_MAX	29	1	XTAL maximum stabilization time
HYPER_RDS_DELAY	30	1	HyperFlash read delay
FLL_FREQ	31	1	FLL frequency
FLL_TOLERANCE	32	1	FLL tolerance
FLL_ASSERT_CYCLES	33	1	FLL assert cycles
RESERVED	35	6	Reserved for future use
USER	40	88	Free for application use

Table 68. e-fuses table

7.0.3.1 Info register 1 (INFO)

7	6	5	4	3	2	1	0
WAIT_XTAL	ENCRYPTED	BOOT			PLT		

Bit 7 - **WAIT_XTAL**

Wait for clock to settle:

- *0b0*: Don't wait
- *0b1*: Wait

Bit 6 - **ENCRYPTED**

Encrypted boot code:

- *0b0*: Not encrypted
- *0b1*: Encrypted

Bits 5:3 - **BOOT**

Platform type:

- *0b000*: Other
- *0b001*: SPIS_ALT2
- *0b010*: JTAG
- *0b011*: SPIS_DFLT
- *0b100*: PRELOAD
- *0b101*: HYPER
- *0b110*: SPIM
- *0b111*: SPIM_QPI

Bits 2:0 - **PLT**

Platform type:

- *0b000*: Other
- *0b001*: FPGA
- *0b010*: RTL
- *0b011*: VP
- *0b100*: Chip

7.0.3.2 Info register 2 (INFO2)

7	6	5	4	3	2	1	0
Reserved				SPIM_CLKDIV	FLL_BYPASS_LOCK	FLL_CONF	FLL_FREQ_SET

Bit 3 - **SPIM_CLKDIV**

SPIM clock division factor

- *0b0*: 1
- *0b1*: 2

Bit 2 - **FLL_BYPASS_LOCK**

Gate FLL output:

- *0b0*: Don’t gate
- *0b1*: Gate

Bit 1 - **FLL_CONF**

Configure FLL:

- *0b0*: Don’t configure
- *0b1*: Configure

Bit 0 - **FLL_FREQ_SET**

Set FLL frequency:

- *0b0*: Don't set
- *0b1*: Set

7.0.3.3 XTAL delta threshold (WAIT_XTAL_DELTA)

A fixed point Q15 fraction. The variation between 2 consecutive measurement should not be higher than this percentage.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LSB								MSB							

Bits 15:8 - **LSB**

LSB of delta

Bits 7:0 - **MSB**

MSB of delta

7.0.3.4 XTAL minimum stabilization time (WAIT_XTAL_MIN)

7	6	5	4	3	2	1	0
TIME							

Bits 7:0 - **TIME**

Minimum time that the clock must be under the threshold to be considered stable.

7.0.3.5 XTAL maximum stabilization time (WAIT_XTAL_MAX)

7	6	5	4	3	2	1	0
TIME							

Bits 7:0 - **TIME**

Maximum amount of time to wait for clock stabilization

8 Electrical characteristics

8.1 Parameter condition

Name	Min	Typ	Max	Unit
RAR Switching frequency		1		MHz
RAR DC Output accuracy		3		%
RAR Output current		100		mA
OSC nominal frequency		32.768		kHz
OSC frequency accuracy		50		ppm

Table 69. Parameter condition table

8.2 Absolute maximum ratings

Clock minimum resolution: 32768 Hz, Voltage resolution: 50mV

Voltage (V)	SOC max clock Freq. (MHz)	Cluster max clock Freq. (MHz)
1.20	250	170
1.15	225	149
1.10	200	129
1.05	175	108
1.00	150	87

Table 70. Absolute maximum ratings table

8.3 Operating conditions

Name	Min	Typ	Max	Unit
XTAL_VDD	1.1	1.2	1.3	V
VDD	1.0	1.1	1.2	V
VDD_EXT_CLUSTER	1.0	1.1	1.2	V
VDD_LVDS_1P2V	1.0	1.1	1.2	V
RAR_AVDD	1.6		3.6	V
SAFE_VDDIO	1.6		3.6	V
SPIM_VDDIO	1.6		3.6	V
CAM_VDDIO	1.6		3.6	V
VDDIO_LVDS		2.5		V
XTAL_IN		32768		Hz

Table 71. Operating conditions table

9 Package information

9.1 AQFN88 package mechanical data

The following figure describes the different dimensions of the package elements.

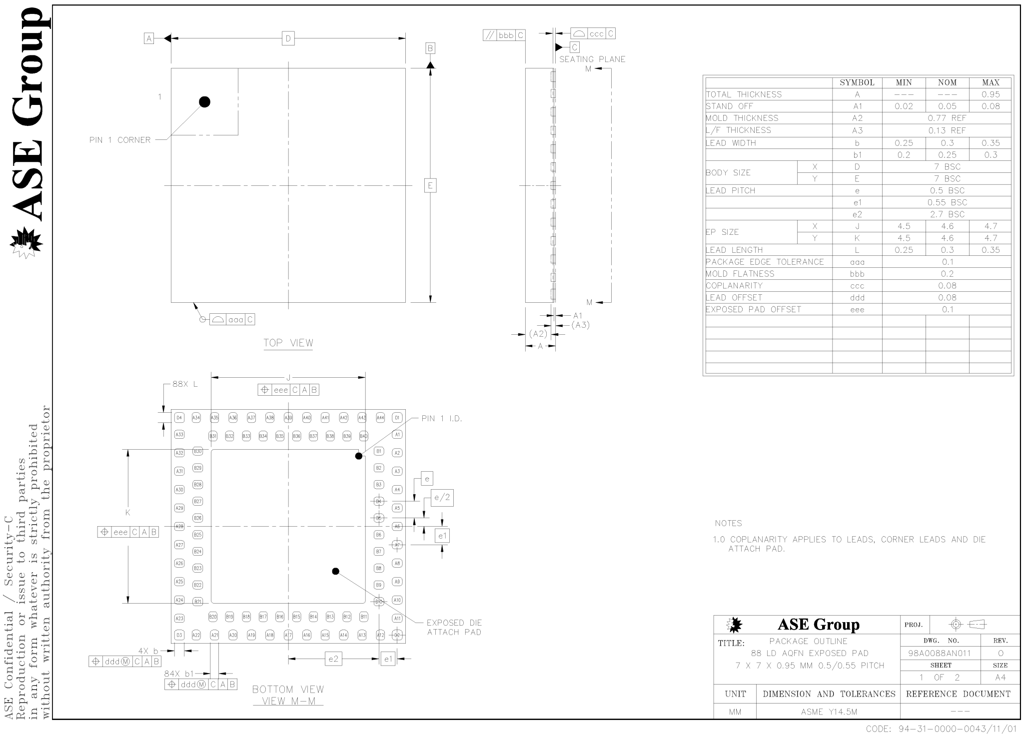

Figure 9. AFN88 package

9.2 AQFN88 recommended footprint

The following figure describes the recommended land patterns on the PCB. Corner pattern sizes are 0.35mm. Other patterns are 0.30mm.

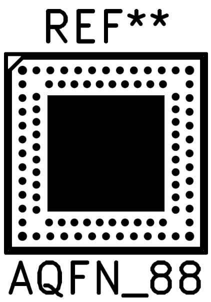

Figure 10. AFN88 Land Patterns

9.3 AQFN88 re-flow curves

The following figure describes the recommended temperatures during the re-flow process

	A.Ramp Up Rate (25~150C)	B.Pre-HeatTime (155~165C)	C.Ramp Up (200C~ peak)	D.Reflow Time (Above 220C)	E.Peak Temp	G.Cooling Rate (peak ~25C)
SPEC.	<1.2C/s	60~90 sec	<1C/s	60~90 sec	230~255C	<6C/s

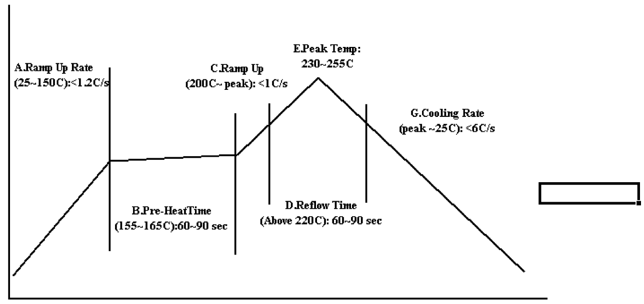

Figure 11. Re-flow Curve

9.4 AQFN88 thermal characteristics

Rating	Value	Unit
Storage temperature range	-65 to 150	°C
Operating temperature range	-40 to 125	°C
Maximum junction temperature	150	°C

Table 72. GAP8 AQFN88 thermal characteristics

10 Chip part numbering

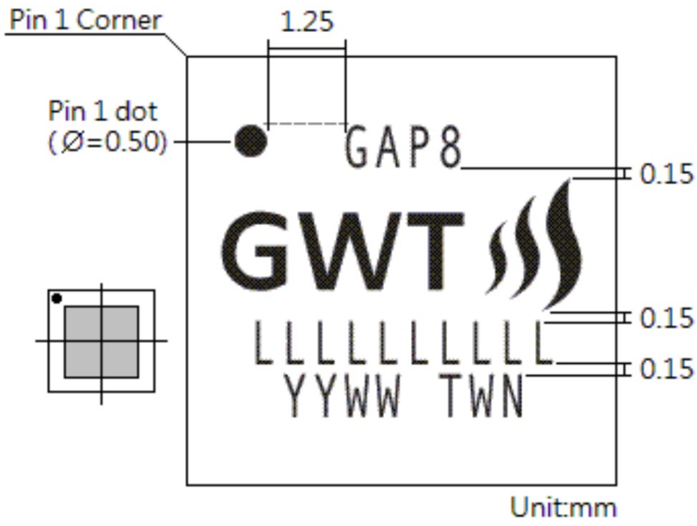

Figure 12. GAP8 chip part numbering

Line Number	Parameter	Description
2	LLLLLLLLLL	Assembly Lot Number
3	YY	Last digit of the year of the assembly date
3	WW	Week number of the assembly date
3	TWN	Location of the assembly process : TAIWAN

Table 73. GAP8 chip part numbering

11 PCB Design

11.1 Internal Oscillator

The GAP8 internal oscillator is an ultra-low power oscillator and is thus sensitive to parasitic coupling and noise. The following guidelines for PCB design and signal routing must be followed with care.

11.1.1 Schematic Guidelines

When using on-board capacitors, two configurations are available:

1. In-parallel configuration: a capacitor is connected between XTAL_IN (noted XA on pictures) and XTAL_OUT(noted XB on pictures)
2. In-series configuration: a capacitor is connected between XTAL_IN and ground, another between XTAL_OUT and ground.

The in-series configuration is recommended to achieve high precision especially when the recommended load capacitance value is low. The in-parallel configuration requires only one capacitor with half of the value of the in-series configuration. In-parallel configuration is recommended when area and cost need to be optimized.

The following picture illustrates the in-series configuration :

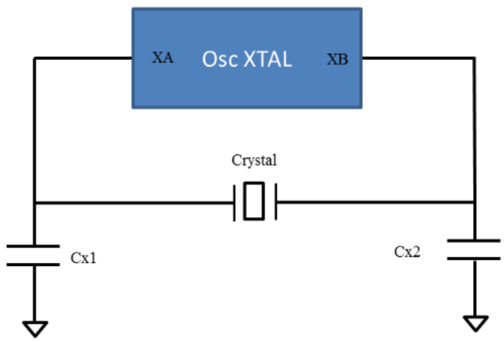

Figure 13. In-series configuration

The following picture illustrates the in-parallel configuration :

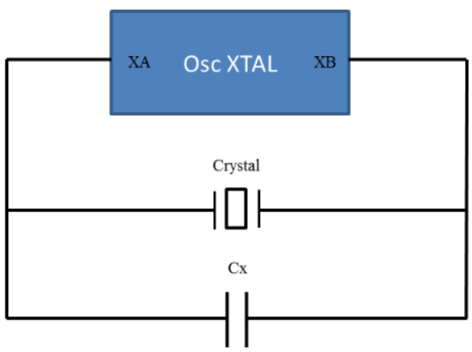

Figure 14. In-parallel configuration

11.1.2 Placement Rules

In both configurations, the crystal has to be placed as close as possible (less than ** 0.4inch or 10mm**) to the chip to reduce parasitic capacitance. External capacitors must be placed as close as possible to the pin they are connected to.

11.1.3 Routing Rules

1. When using a multi-layer PCB, it is recommended that a ground plane layer below the top layer is used to provide better noise

immunity. Guard rings around external signals XTAL_IN/XTAL_OUT and external components are recommended to increase isolation. Guards have to be implemented on the top layer and underneath ground layer(s). Note that when using a ground layer under top layer, parasitic coupling to ground will increase and thus has to be taken into account in C L calculations.
2. When using a single-layer PCB, it is recommended that you implement a guard ring around external signals XTAL_IN/XTAL_OUT and external components in order to increase isolation.
3. It is recommended that you have an unclosed guard ring instead of a closed guard ring.
4. Return currents across the GAP8 oscillator region must be avoided.
5. To prevent frequency accuracy degradation, avoid coupling with other clock signals or signals with fast transitions. Do not route high-frequency signals underneath the GAP8 oscillator region.
6. If on-board capacitances with in-series configuration are used, the connection of the two capacitors to the ground should be as short as possible and made at a single point.
7. Do not use wide PCB traces for crystal and capacitor connections in order to minimize coupling with other parasitic signals.
8. PCB layout has to be symmetrical (refer to the figures below for an example).

The following picture illustrates layout placement and constraints in a single layer environment:

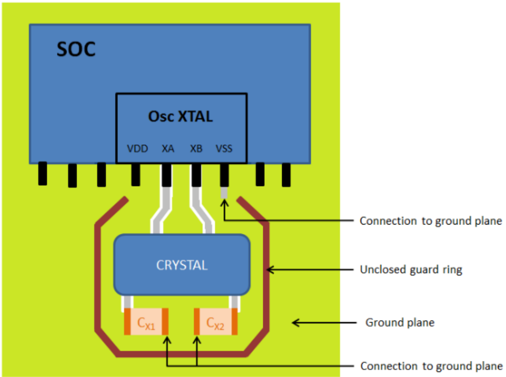

Figure 15. Single layer layout

The following picture illustrates layout placement and constraints in a multiple layer environment:

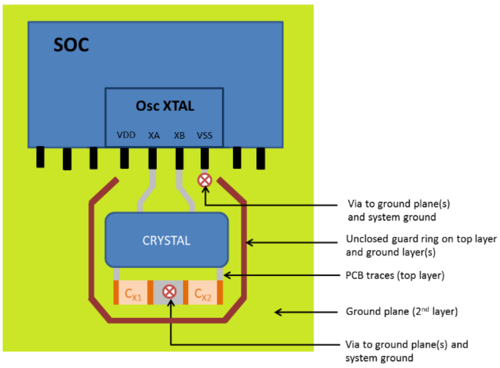

Figure 16. Multiple layer layout

11.2 Internal Regulator (RAR)

11.2.1 Schematic Guidelines

The following figure illustrates the recommended power scheme :

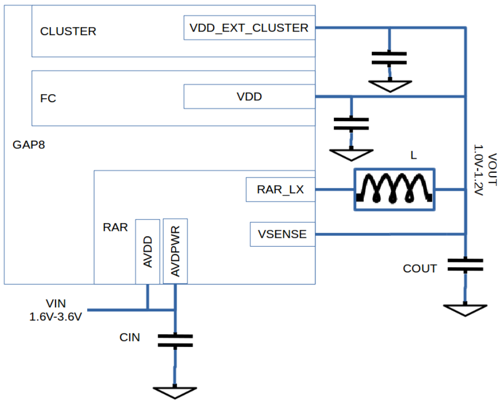

Figure 17. Power connections

11.2.2 Placement Rules

External capacitor CIN and COUT must be placed as close as possible to the pins they are connected to.

11.2.3 Routing Rules

1. Routing between AVD pin, AVDPWR pin, and the battery must be wide and short in order to minimize the parasitic resistance of this connection.
2. It is recommended to connect the ground pins to PCB plane with wide traces.

11.2.4 Device Selection Rules

1. CIN and COUT capacitance should be 4.7uF
2. CIN and COUT should be X5R or X7R
3. L inductance should be 22uH

12 Revision history

Date	Revision	Changes
15 Dec 2017	0.2	First Draft
31 Jan 2018	1.0	Initial Release
28 Feb 2018	1.1	B14 name update and PCB layout guidelines
8 March 2018	1.2	Further corrections
19 March 2018	1.3	VDD operating Condition
17 April 2018	1.4	Include all IP information
15 May 2018	1.5	Complete Device overview and add content
09 Sept 2018	1.5.1	Detail default HyperBus chip selects
11 Oct 2018	1.5.2	Correct efuse list
25 Oct 2018	1.5.3	Correct PADCFG bit location
07 Nov 2018	1.5.4	Correct register description
30 Jan 2019	1.5.5	Correct DC/DC inductance value

13 IMPORTANT NOTICE – PLEASE READ CAREFULLY

14 Copyright Notice