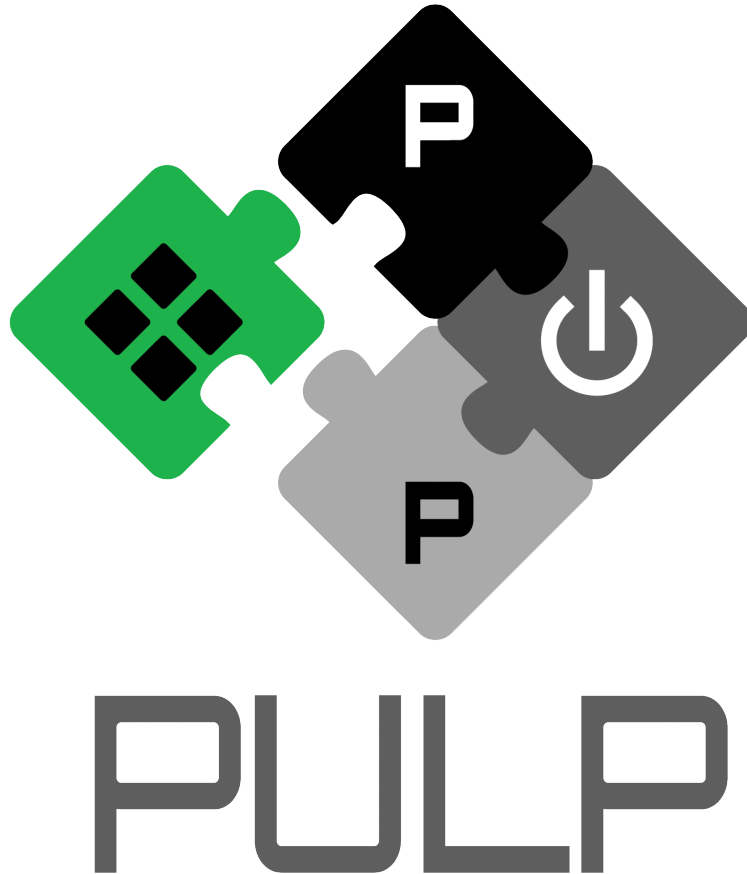


# PULPissimo: Datasheet



The PULP team  
[pulp-info@list.ee.ethz.ch](mailto:pulp-info@list.ee.ethz.ch)

May 11, 2020

# Contents

<b>1</b>	<b>Overview</b>	<b>3</b>
<b>2</b>	<b>Memory Map</b>	<b>5</b>
<b>3</b>	<b>CPU Core</b>	<b>6</b>
<b>4</b>	<b>Peripherals</b>	<b>8</b>
4.1	FLL . . . . .	9
4.1.1	SoC FLL registers . . . . .	9
4.1.2	STATUS . . . . .	9
4.1.3	CFG1 . . . . .	9
4.1.4	CFG2 . . . . .	10
4.1.5	INTEG . . . . .	11
4.2	GPIO . . . . .	12
4.2.1	PADDIR (Pad Direction) . . . . .	12
4.2.2	PADIN (Input Values) . . . . .	12
4.2.3	PADOUT (Output Values) . . . . .	12
4.2.4	INTEN (Interrupt Enable) . . . . .	13
4.2.5	INTTYPE0 (Interrupt Type 0) . . . . .	13
4.2.6	INTTYPE1 (Interrupt Type 1) . . . . .	13
4.2.7	INTSTATUS (Interrupt Status) . . . . .	14
4.2.8	GPIOEN (GPIO Enable) . . . . .	14
4.2.9	PADCFG0-7 (Pad Configuration Registers 0-7) . . . . .	14
4.3	SoC Control . . . . .	16
4.3.1	Info . . . . .	16
4.3.2	Boot Address . . . . .	16
4.3.3	Fetch Enable . . . . .	16
4.3.4	PAD Mux . . . . .	17
4.3.5	PAD Configuration . . . . .	17
4.3.6	JTAG Register . . . . .	17
4.3.7	Core Status . . . . .	18
4.3.8	FLL Clock Select . . . . .	18
4.4	Event/Interrupt Controller . . . . .	19
4.4.1	Mask . . . . .	19
4.4.2	Interrupt . . . . .	19
4.4.3	Int Ack . . . . .	19
4.4.4	FIFO Content . . . . .	20
4.5	SoC Event Generator . . . . .	21
4.5.1	SoC Event Generator Registers . . . . .	21
4.5.2	SW_EVENT . . . . .	22
4.5.3	FC_MASKX, X = 0...7 . . . . .	22

4.5.4	PR_MASKX, X = 0...7 . . . . .	23
4.5.5	ERRX, X = 0...7 . . . . .	23
4.5.6	TIMER_LO . . . . .	23
4.5.7	TIMER_HI . . . . .	24
4.6	APB Timer . . . . .	25
4.6.1	APB Timer Registers . . . . .	25
4.6.2	CFG_LO . . . . .	26
4.6.3	CFG_HI . . . . .	26
4.6.4	CNT_LO . . . . .	27
4.6.5	CNT_HI . . . . .	27
4.6.6	CMP_LO . . . . .	28
4.6.7	CMP_HI . . . . .	28
4.6.8	START_LO . . . . .	28
4.6.9	START_HI . . . . .	29
4.6.10	RESET_LO . . . . .	29
4.6.11	RESET_HI . . . . .	29
4.7	APB Advanced Timer . . . . .	30
4.7.1	APB Advanced Timer Registers . . . . .	30
4.7.2	T0_CMD . . . . .	32
4.7.3	T0_CONFIG . . . . .	32
4.7.4	T0_THRESHOLD . . . . .	33
4.7.5	T0_TH_CHANNEL0 . . . . .	33
4.7.6	T0_TH_CHANNEL1 . . . . .	34
4.7.7	T0_TH_CHANNEL2 . . . . .	34
4.7.8	T0_TH_CHANNEL3 . . . . .	35
4.7.9	T1_CMD . . . . .	35
4.7.10	T1_CONFIG . . . . .	36
4.7.11	T1_THRESHOLD . . . . .	37
4.7.12	T1_TH_CHANNEL0 . . . . .	37
4.7.13	T1_TH_CHANNEL1 . . . . .	37
4.7.14	T1_TH_CHANNEL2 . . . . .	38
4.7.15	T1_TH_CHANNEL3 . . . . .	38
4.7.16	T2_CMD . . . . .	39
4.7.17	T2_CONFIG . . . . .	39
4.7.18	T2_THRESHOLD . . . . .	40
4.7.19	T2_TH_CHANNEL0 . . . . .	40
4.7.20	T2_TH_CHANNEL1 . . . . .	41
4.7.21	T2_TH_CHANNEL2 . . . . .	41
4.7.22	T2_TH_CHANNEL3 . . . . .	42
4.7.23	T3_CMD . . . . .	42
4.7.24	T3_CONFIG . . . . .	43
4.7.25	T3_THRESHOLD . . . . .	44
4.7.26	T3_TH_CHANNEL0 . . . . .	44
4.7.27	T3_TH_CHANNEL1 . . . . .	44
4.7.28	T3_TH_CHANNEL2 . . . . .	45
4.7.29	T3_TH_CHANNEL3 . . . . .	45
4.7.30	CG . . . . .	46

4.8	uDMA Subsystem . . . . .	47
4.8.1	uDMA UART Registers . . . . .	47
4.8.2	RX_SADDR . . . . .	47
4.8.3	RX_SIZE . . . . .	48
4.8.4	RX_CFG . . . . .	48
4.8.5	TX_SADDR . . . . .	49
4.8.6	TX_SIZE . . . . .	49
4.8.7	TX_CFG . . . . .	49
4.8.8	STATUS . . . . .	50
4.8.9	SETUP . . . . .	50
4.8.10	ERROR . . . . .	51
4.8.11	IRQ_EN . . . . .	51
4.8.12	VALID . . . . .	52
4.8.13	DATA . . . . .	52
4.8.14	uDMA SPI Registers . . . . .	53
4.8.15	SPIM_CMD_SADDR . . . . .	53
4.8.16	SPIM_CMD_SIZE . . . . .	54
4.8.17	SPIM_CMD_CFG . . . . .	54
4.8.18	SPIM_RX_SADDR . . . . .	55
4.8.19	SPIM_RX_SIZE . . . . .	55
4.8.20	SPIM_RX_CFG . . . . .	55
4.8.21	SPIM_TX_SADDR . . . . .	56
4.8.22	SPIM_TX_SIZE . . . . .	56
4.8.23	SPIM_TX_CFG . . . . .	57
4.8.24	uDMA I2C0 Registers . . . . .	57
4.8.25	RX_SADDR . . . . .	58
4.8.26	RX_SIZE . . . . .	58
4.8.27	RX_CFG . . . . .	59
4.8.28	TX_SADDR . . . . .	59
4.8.29	TX_SIZE . . . . .	60
4.8.30	TX_CFG . . . . .	60
4.8.31	CMD_SADDR . . . . .	61
4.8.32	CMD_SIZE . . . . .	61
4.8.33	CMD_CFG . . . . .	61
4.8.34	STATUS . . . . .	62
4.8.35	SETUP . . . . .	62
4.8.36	uDMA I2C1 Registers . . . . .	62
4.8.37	RX_SADDR . . . . .	63
4.8.38	RX_SIZE . . . . .	63
4.8.39	RX_CFG . . . . .	64
4.8.40	TX_SADDR . . . . .	64
4.8.41	TX_SIZE . . . . .	65
4.8.42	TX_CFG . . . . .	65
4.8.43	CMD_SADDR . . . . .	66
4.8.44	CMD_SIZE . . . . .	66
4.8.45	CMD_CFG . . . . .	66
4.8.46	STATUS . . . . .	67
4.8.47	SETUP . . . . .	67

4.8.48	uDMA I2S Registers . . . . .	67
4.8.49	I2S_RX_SADDR . . . . .	68
4.8.50	I2S_RX_SIZE . . . . .	69
4.8.51	I2S_RX_CFG . . . . .	69
4.8.52	I2S_TX_SADDR . . . . .	70
4.8.53	I2S_TX_SIZE . . . . .	70
4.8.54	I2S_TX_CFG . . . . .	70
4.8.55	I2S_CLKCFG_SETUP . . . . .	71
4.8.56	I2S_SLV_SETUP . . . . .	72
4.8.57	I2S_MST_SETUP . . . . .	72
4.8.58	I2S_PDM_SETUP . . . . .	73
4.8.59	uDMA Camera Interface Registers . . . . .	73
4.8.60	CAM_RX_SADDR . . . . .	74
4.8.61	CAM_RX_SIZE . . . . .	74
4.8.62	CAM_RX_CFG . . . . .	75
4.8.63	CAM_CFG_GLOB . . . . .	75
4.8.64	CAM_CFG_LL . . . . .	76
4.8.65	CAM_CFG_UR . . . . .	76
4.8.66	CAM_CFG_SIZE . . . . .	77
4.8.67	CAM_CFG_FILTER . . . . .	77
4.8.68	CAM_VSYNC_POLARITY . . . . .	77
4.8.69	uDMA Filter Registers . . . . .	78

## 5    Debug Module for External Debug Support 80

# 1 Overview

PULPissimo is a 32 bit RISC-V single-core System-on-a-Chip. PULPissimo is the second version of the PULPINO system and it can be extended with the multi-core cluster of the PULP project.

Differently from the simpler PULPINO system, PULPissimo uses a more complex memory subsystem, an autonomous I/O subsystem which uses the uDMA, new peripherals (eg the camera interface) and a new SDK.

Figure 1.1 shows a simplified block diagram of the SoC. As for PULPINO, PULPissimo can be configured at design time to use either the RISC-V or ZERO-RISC-V. The peripherals are connected to the uDMA which transfers the data to the memory subsystem efficiently. The JTAG and the AXI plug have also access to the SoC. The AXI plug can be used to extend the microcontroller with a multi-core cluster or an accelerator. As for PULPINO, the advanced debug unit is used to access to system and core registers, memories and memory-mapped IO via JTAG. A logarithmic interconnect allows to link the core and the uDMA to the memory banks simultaneously.

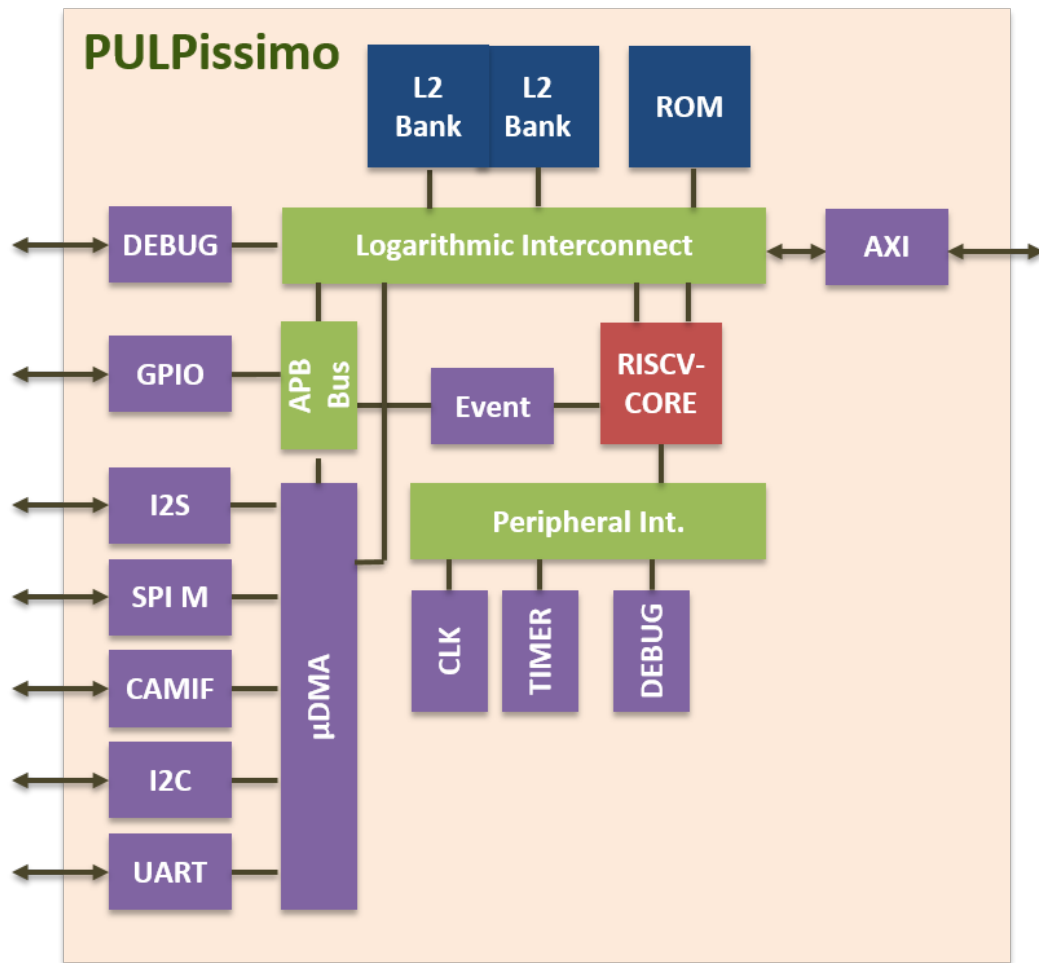


Figure 1.1: PULPissimo Overview.

PULPissimo is mainly targeted at RTL simulation and ASICs. The FPGA versions has not yet been implemented.

## 2 Memory Map

Figure 2.1 shows the default memory-map of PULPiSSIMO, whereas Please, consult the uDMA documentation for the peripherals attached to the uDMA memory-map of configuration.

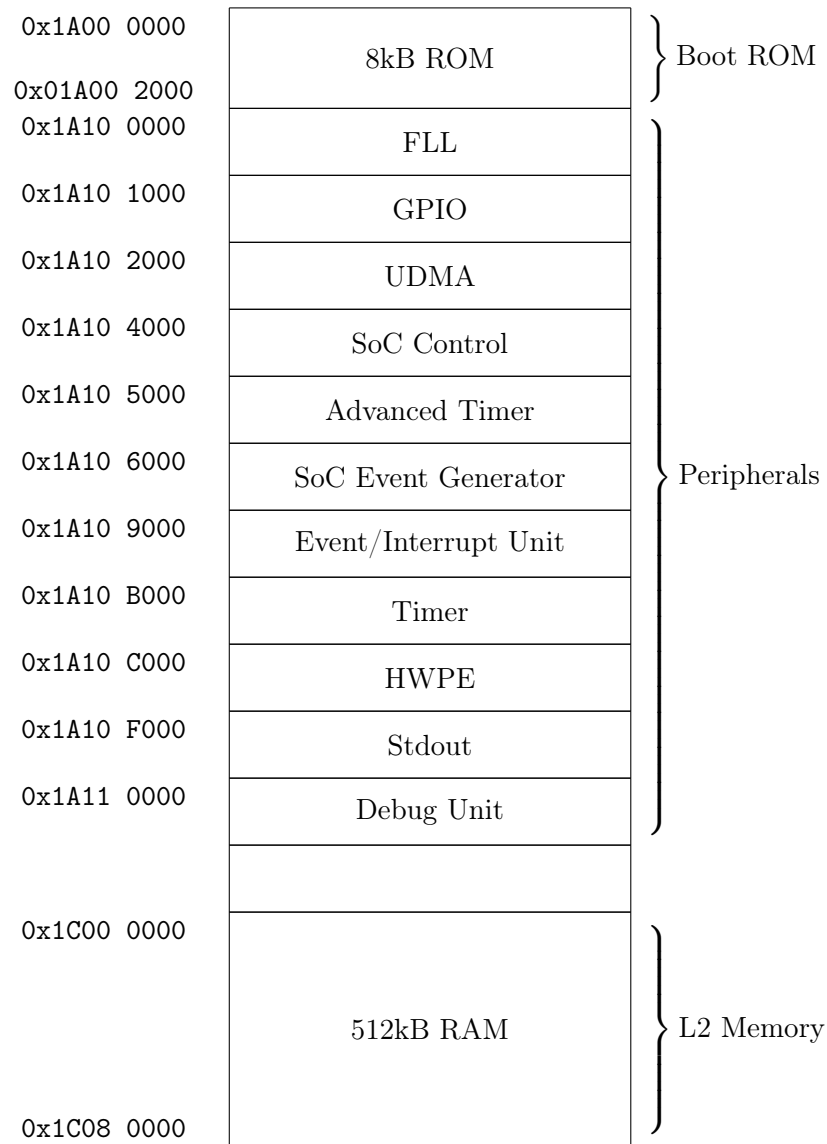


Figure 2.1: PULPiSSIMO memory-map.



### 3 CPU Core

PULPiSSIMO supports both the RISC-V and the ZERO-RISCY RI5CY core. The two cores have the same external interfaces and are thus plug-compatible. Figure 3.1 and 3.2 show the two cores architectures.

For debugging purposes, all core registers have been memory mapped which allows to them to be accessed over the logarithmic-interconnect subsystem. The debug unit inside the core handles the request over this bus and reads/sets the core registers and/or halts the core.

The core supports performance counters. Those are mainly used for counting core internal events like stalls, but it is possible to count core-external events as well. For this purpose there is the `ext_perf_counters_i` port where arbitrary events can be attached. The core then increases its internal performance counter for this event type every time a logic high is seen on this port.

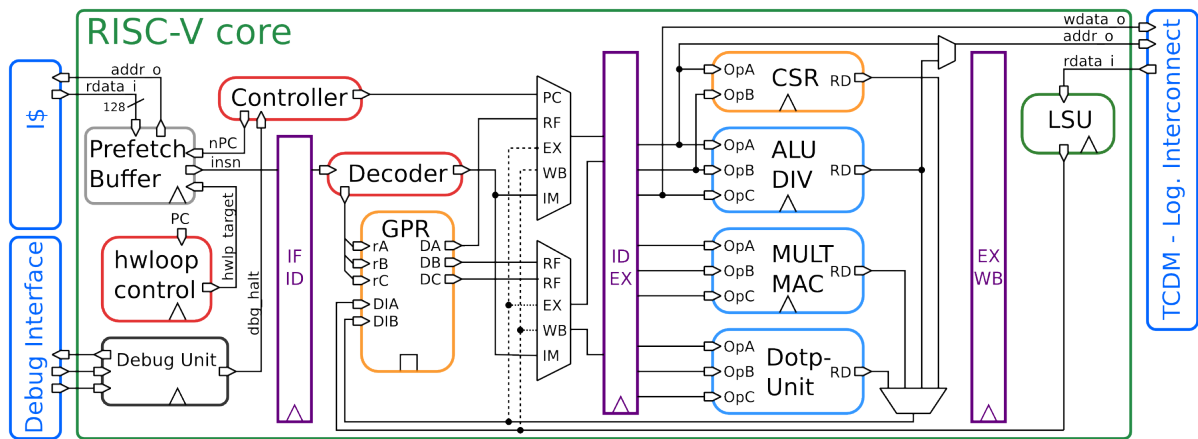


Figure 3.1: RISCY core overview

Take a look at the cores documentation for more details.

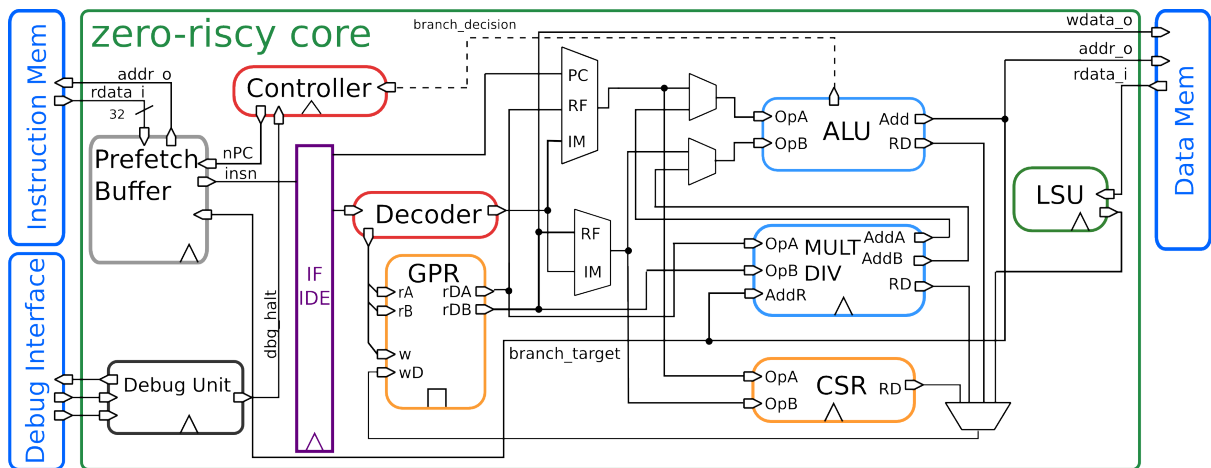


Figure 3.2: zero-riscy core overview

## 4 Peripherals

Most of the peripherals in PULPissimo are connected to the uDMA subsystem which efficiently handles all the data-transfers autonomously. The uDMA must be programmed by the core via memory-mapped read and write operations to receive commands.

See the uDMA documentation for more details under the uDMA repository.

The GPIO, timers, event unit and event generator, debug and the FLLs are not connected to the uDMA instead but to the APB bus. Following a brief overview about these units is given.

## 4.1 FLL

PULPissimo contains 3 FLLs. One FLL is meant for generating the clock for the peripheral domain, one for the core domain (core, memories, event unit etc) and one is meant for the cluster. The latter is not used.

All the FLLs can be bypassed by writing to the JTAG register before the reset signal is asserted. See Section 4.3 for more details about the bypass register.

### 4.1.1 SoC FLL registers

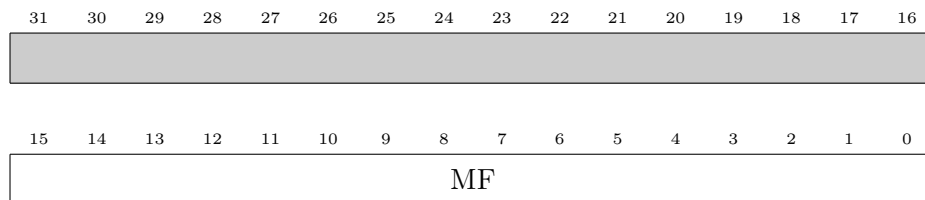
Name	Address	Size	Type	Access	Default	Description
STATUS	0x1A100000	32	Status	R	0x00000000	FLL status register
CFG1	0x1A100004	32	Config	R/W	0x00000000	FLL configuration 1 register
CFG2	0x1A100008	32	Config	R/W	0x00000000	FLL configuration 2 register
INTEG	0x1A10000C	32	Config	R/W	0x00000000	FLL integrator configuration register.

Table 4.2: SoC FLL register table

### 4.1.2 STATUS

Address: 0x1A10\_0000

Reset Value: 0x0000\_0000

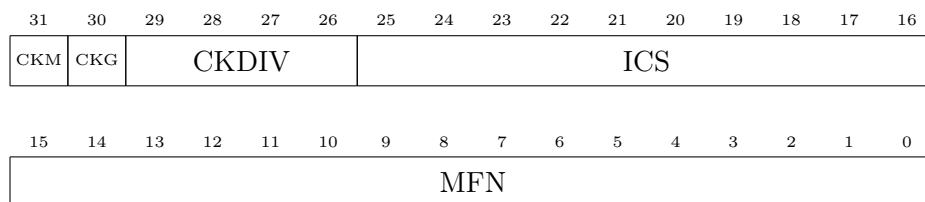


Bit 15-0 **MF** (*R*) Current DCO multiplication factor value bitfield

### 4.1.3 CFG1

Address: 0x1A10\_0004

Reset Value: 0x0000\_0000



Bit 31 **CKM** (*R/W*) FLL operation mode configuration bitfield

- 0b0: standalone
- 0b1: normal

Bit 30 **CKG** (*R/W*) FLL output clock divider configuration

- 0b0: not gated
- 0b1: gated

Bit 29-26 **CKDIV** (*R/W*) FLL output clock divider configuration

Bit 25-16 **ICS** (*R/W*) DCO input code in standalone

Bit 15-0 **MFN** (*R/W*) Target clock multiplication factor in normal mode

#### 4.1.4 CFG2

**Address:** 0x1A10\_0008

**Reset Value:** 0x0000\_0000



Bit 31 **DITH** (*R/W*) Dithering activation

Bit 30 **CKM** (*R/W*) Open loop when locked

- 0b0: disabled
- 0b1: enabled

Bit 29 **CKSEL** (*R/W*) Configuration clock selection in standalone mode

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

Bit 27-16 **LT** (*R/W*) Lock tolerance configuration. It is the margin around the multiplication factor within which the output clock is considered stable.

Bit 15-10 **SCKL** (*R/W*) Number of stable REFCLK cycles until LOCK assert in normal mode. Upper 6 bits of LOCK assert counter target in standalone mode.

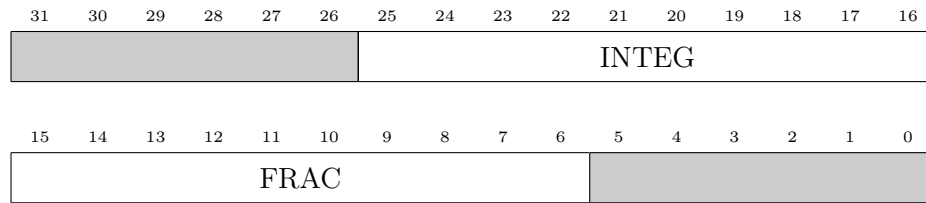
Bit 9-4 **UCKL** (*R/W*) Number of unstable REFCLK cycles until LOCK de-assert in normal mode. Lower 6 bits of LOCK assert counter target in standalone mode.

Bit 3-0 **LG** (*R/W*) FLL loop gain setting

### 4.1.5 INTEG

Address: 0x1A10\_000C

Reset Value: 0x0000\_0000



Bit 25-16 **INTEG** (*R/W*) Integer part of integrator state bitfield. It corresponds to DCO unit bits.

Bit 15-6 **FRAC** (*R/W*) Fractional part of integrator state bitfield. It corresponds to dither unit input.

## 4.2 GPIO

Table 4.3: GPIO Signals

Signal	Direction	Description
gpio_in[31:0]	input	Transmit Data
gpio_out[31:0]	output	Receive Data
gpio_dir[31:0]	output	Request to Send
gpio_padcfg[5:0][31:0]	output	Pad Configuration
interrupt	output	Interrupt (Rise or Fall or Level)

### 4.2.1 PADDIR (Pad Direction)

Address: 0x1A10\_1000

Reset Value: 0x0000\_0000

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

Bit 31:0 **PADDIR**: Pad Direction.

Control the direction of each of the GPIO pads. A value of 1 means it is configured as an output, while 0 configures it as an input.

### 4.2.2 PADIN (Input Values)

Address: 0x1A10\_1004

Reset Value: 0x0000\_0000

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

Bit 31:0 **PADIN**: Input Values.

### 4.2.3 PADOUT (Output Values)

Address: 0x1A10\_1008

Reset Value: 0x0000\_0000

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

Bit 31:0 **PADOUT**: Output Values.

#### 4.2.4 INTEN (Interrupt Enable)

**Address:** 0x1A10\_100C

**Reset Value:** 0x0000\_0000

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

Bit 31:0 **INTEN**: Interrupt Enable.

Interrupt enable per input bit. INTTYPE0 and INTTYPE1 control the interrupt triggering behavior.

There are four triggers available

- INTTYPE0 = 0, INTTYPE1 = 0: Level 1
- INTTYPE0 = 1, INTTYPE1 = 0: Level 0
- INTTYPE0 = 0, INTTYPE1 = 1: Rise
- INTTYPE0 = 1, INTTYPE1 = 1: Fall

#### 4.2.5 INTTYPE0 (Interrupt Type 0)

**Address:** 0x1A10\_1010

**Reset Value:** 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
T0	T0	T0	T0	T0	T0	T0	T0	T0	T0	T0	T0	T0	T0	T0	T0	T0	T0	T0	T0	T0	T0	T0	T0	T0	T0	T0	T0	T0	T0	T0	T0	INTTYPE0

Bit 31:0 **INTTYPE0**: Interrupt Type 0.

Controls the interrupt trigger behavior together with INTTYPE1. Use INTEN to enable interrupts first.

#### 4.2.6 INTTYPE1 (Interrupt Type 1)

**Address:** 0x1A10\_1014

**Reset Value:** 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
T1	T1	T1	T1	T1	T1	T1	T1	T1	T1	T1	T1	T1	T1	T1	T1	T1	T1	T1	T1	T1	T1	T1	T1	T1	T1	T1	T1	T1	T1	T1	T1	INTTYPE1



Bit 31:0 **INTTYPE1**: Interrupt Type 1.

Controls the interrupt trigger behavior together with INTTYPE0. Use INTEN to enable interrupts first.

#### 4.2.7 INTSTATUS (Interrupt Status)

**Address:** 0x1A10\_1018

**Reset Value:** 0x0000\_0000

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

Bit 31:0 **INTSTATUS**: Interrupt Status.

Contains interrupt status per GPIO line. The status register is cleared when read. Similarly the **interrupt** line is high while a bit is set in interrupt status and will be deasserted when the status register is read.

#### 4.2.8 GPIOEN (GPIO Enable)

**Address:** 0x1A10\_101C

**Reset Value:** 0x0000\_0000

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

Bit 31:0 **GPIOEN**: GPIO Enable.

Contains the enable bit per GPIO line.

#### 4.2.9 PADCFG0-7 (Pad Configuration Registers 0-7)

**Address:** 0x1A10\_1020 - 0x1A10\_103C

**Reset Value:** 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	P	PADCFG0-7

Bit 31:0 **PADCFG0-7**: Pad Configuration Registers.

The pad configuration registers control various aspects of the pads that are typically used in ASICs, e.g. drive strength, Schmitt-Triggers, Slew Rate, etc. Since those configuration parameters depend on the exact pads used, each implementation is free to use the PADCFG0-7 registers in every way it wants and also leave them unconnected, if unneeded.

Writing to the PADOUTSET address (0x1A10\_1040), the content of the PADOUT register is updated with its content "ored" with the write data.

Writing to the PADOUTCLR address (0x1A10\_1044), the content of the PADOUT register is updated with its content "anded" with the inverted write data.

## 4.3 SoC Control

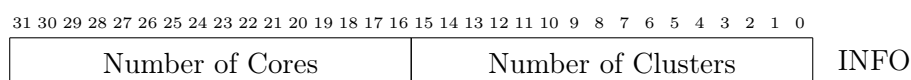
PULPissimo features a small and simple APB peripheral which provides information about the platform and provides the means for pad muxing on the ASIC.

The following registers can be accessed.

### 4.3.1 Info

**Address:** 0x1A10\_4000

**Reset Value:** 0x0000\_0000

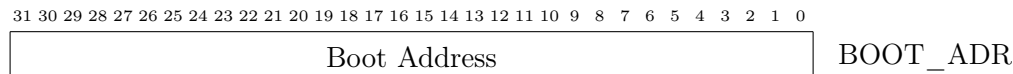


Bit 31:0 **Info:** This register holds the number of clusters and the number of cores in the each cluster. It is a read-only register.

### 4.3.2 Boot Address

**Address:** 0x1A10\_4004

**Reset Value:** 0x1A10\_0000

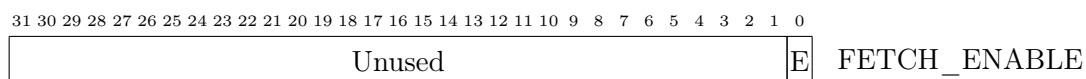


Bit 31:0 **Boot Address:** This register holds the boot address.

### 4.3.3 Fetch Enable

**Address:** 0x1A10\_4008

**Reset Value:** 0x0000\_0001

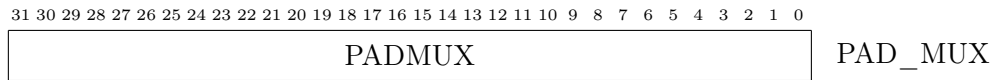


Bit 31:0 **Fetch Enable:** This register contains the value of the fetch enable signal of the core.

#### 4.3.4 PAD Mux

**Address:** 0x1A10\_4010 - 0x1A10\_401C

**Reset Value:** 0x0000\_0000

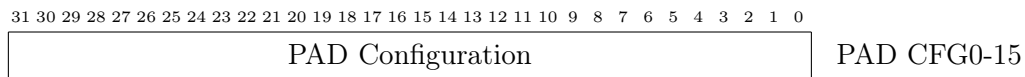


Bit 31:0 **PADMUX**: The content of these registers can be used to multiplex pads when targeting an ASIC. The first register (0x1A10\_4010) can be used to sets the mux (2 bit select) from pin 0 (bits [1:0]) to 15 (bits [31:30]). The second register (0x1A10\_4014) can be used to sets the mux (2 bit select) from pin 16 (bits [1:0]) to 31 (bits [31:30]). The third register (0x1A10\_4018) can be used to sets the mux (2 bit select) from pin 32 (bits [1:0]) to 47 (bits [31:30]). The forth register (0x1A10\_401C) can be used to sets the mux (2 bit select) from pin 48 (bits [1:0]) to 63 (bits [31:30]).

#### 4.3.5 PAD Configuration

**Address:** 0x1A10\_4020 - 0x1A10\_405C

**Reset Value:** 0x0000\_0000

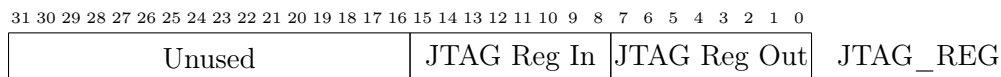


Bit 31:0 **PAD\_CFG0-15**: These 16 registers can be used for ASIC targets to configure pads, e.g. pull up, pull down values.

#### 4.3.6 JTAG Register

**Address:** 0x1A10\_4074

**Reset Value:** 0x0000\_0000

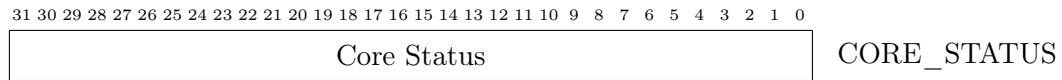


Bit 31:0 **JTAG Register**: This register contains the value of the input from the JTAG and can be used to write 8bit in the JTAG output register for system-to-JTAG communications.

### 4.3.7 Core Status

**Address:** 0x1A10\_40A0 and 0x1A10\_40C0

**Reset Value:** 0x0000\_0001

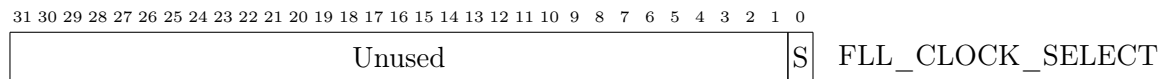


Bit 31:0 **Core Status:** These 2 registers contain the status of the system for testing/verification purposes like End Of Computation. The 0x1A10\_40C0 register is read-only.

### 4.3.8 FLL Clock Select

**Address:** 0x1A10\_40C8

**Reset Value:** 0x0000\_0000



Bit 31:0 **FLL Clock Select:** This register contains whether the system clock is coming from the FLL or the FLL is bypassed. It is a read-only register by the core but it can be written via JTAG.

## 4.4 Event/Interrupt Controller

PULPiSSIMO features a lightweight event and interrupt controller which supports vectorized interrupts and events of up to 32 lines. It contains a FIFO of events from the peripherals or SW events. When an interrupt is ready and it is enabled (not masked), the unit sends the 5-bit ID to the core and the interrupt request line is raised up. If the core takes the interrupt, it replies with the ID of the interrupt taken and the acknowledge signal. The communication between the interrupt controller and the core is completely asynchronous. Note that the interrupt controller can change the interrupt ID anytime but it must rely on the ID sent by the core to know which interrupt has been taken. This is an important feature that covers the situation where a higher priority interrupt request prevent another one that has been already sent to the core. Depending on the core state and core interrupt enable, the interrupt can be accepted within a couple of clock cycles.

### 4.4.1 Mask

**Address:** 0x1A10\_9000

**Reset Value:** 0x0000\_0000

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

Bit 31:0 **MASK**: This register contains the MASK (interrupt enable) for each of the 32 interrupts or events. Writing to 0x1A10\_9004 sets the bits of the MASK register selected. Writing to 0x1A10\_9008 clears the bits of the MASK register selected.

### 4.4.2 Interrupt

**Address:** 0x1A10\_900C

**Reset Value:** 0x0000\_0000

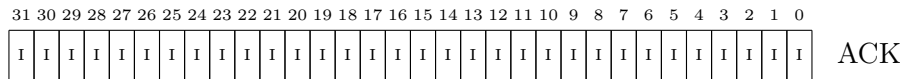
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	INT

Bit 31:0 **INT**: This register contains the pending interrupts or events. Writing to 0x1A10\_9010 sets the bits of the INT register selected. Writing to 0x1A10\_9014 clears the bits of the INT register selected.

### 4.4.3 Int Ack

**Address:** 0x1A10\_9018

**Reset Value:** 0x0000\_0000

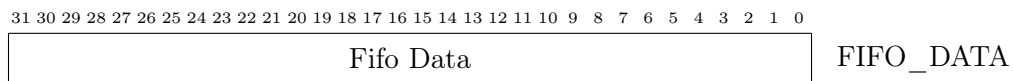


Bit 31:0 **ACK**: This register contains the ACK (interrupt enable) for each of the 32 interrupts or events. Writing to 0x1A10\_901C sets the bits of the ACK register selected. Writing to 0x1A10\_9020 clears the bits of the ACK register selected.

#### 4.4.4 FIFO Content

**Address:** 0x1A10\_9024

**Reset Value:** 0x0000\_0000



Bit 31-0 **FIFO\_DATA**: Fifo Content.

This is a read-only register that contain the first valid value of the FIFO.

## 4.5 SoC Event Generator

Events from peripherals and other sources can be forwarded to the fabric controller, cluster or (back) to certain peripherals, though for PULPissimo we don't have a cluster.

It is the SoC Event Generator's (`soc_event_generator.sv`) job to control which events are to be forwarded and where to. There are three set of masks available to do this:

FC Masks Control which events are to be forwarded to the fabric controller

Cluster Masks Control which events are to be forwarded to the cluster (disabled)

Peripheral Masks Control which events are to be forwarded to peripherals

### 4.5.1 SoC Event Generator Registers

Name	Address	Size	Type	Access	Default	Description
SW_EVENT	0x1A106000	32	Config	W	0x00000000	SoC software events trigger register
FC_MASK0	0x1A106004	32	Config	R/W	0xFFFFFFFF	Events 0-31 dispatch mask to FC
FC_MASK1	0x1A106008	32	Config	R/W	0xFFFFFFFF	Events 32-63 dispatch mask to FC
FC_MASK2	0x1A10600C	32	Config	R/W	0xFFFFFFFF	Events 64-95 dispatch mask to FC
FC_MASK3	0x1A106010	32	Config	R/W	0xFFFFFFFF	Events 96-127 dispatch mask to FC
FC_MASK4	0x1A106014	32	Config	R/W	0xFFFFFFFF	Events 128-159 dispatch mask to FC
FC_MASK5	0x1A106018	32	Config	R/W	0xFFFFFFFF	Events 160-191 dispatch mask to FC
FC_MASK6	0x1A10601C	32	Config	R/W	0xFFFFFFFF	Events 191-223 dispatch mask to FC
FC_MASK7	0x1A106020	32	Config	R/W	0xFFFFFFFF	Events 224-255 dispatch mask to FC
PR_MASK0	0x1A106044	32	Config	R/W	0xFFFFFFFF	Events 0-31 dispatch mask to peripherals
PR_MASK1	0x1A106048	32	Config	R/W	0xFFFFFFFF	Events 32-63 dispatch mask to peripherals
PR_MASK2	0x1A10604C	32	Config	R/W	0xFFFFFFFF	Events 64-95 dispatch mask to peripherals
PR_MASK3	0x1A106050	32	Config	R/W	0xFFFFFFFF	Events 96-127 dispatch mask to peripherals
PR_MASK4	0x1A106054	32	Config	R/W	0xFFFFFFFF	Events 128-159 dispatch mask to peripherals
PR_MASK5	0x1A106058	32	Config	R/W	0xFFFFFFFF	Events 160-191 dispatch mask to peripherals
PR_MASK6	0x1A10605C	32	Config	R/W	0xFFFFFFFF	Events 191-223 dispatch mask to peripherals
PR_MASK7	0x1A106060	32	Config	R/W	0xFFFFFFFF	Events 224-255 dispatch mask to peripherals



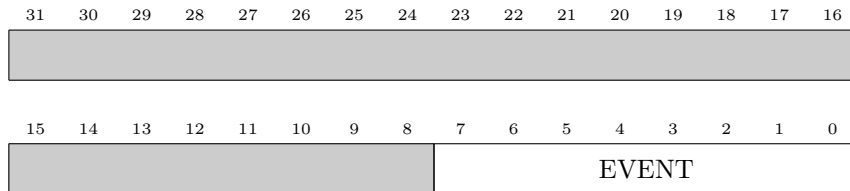
ERR0	0x1A106064	32	Status	R	0x00000000	Events 0-31 event queue overflow
ERR1	0x1A106068	32	Status	R	0x00000000	Events 32-63 event queue overflow
ERR2	0x1A10606C	32	Status	R	0x00000000	Events 64-95 event queue overflow
ERR3	0x1A106070	32	Status	R	0x00000000	Events 96-127 event queue overflow
ERR4	0x1A106074	32	Status	R	0x00000000	Events 128-159 event queue overflow
ERR5	0x1A106078	32	Status	R	0x00000000	Events 160-191 event queue overflow
ERR6	0x1A10607C	32	Status	R	0x00000000	Events 191-223 event queue overflow
ERR7	0x1A106080	32	Status	R	0xFFFFFFFF	Events 224-255 event queue overflow
TIMER_LO	0x1A106084	32	Status	R/W	0xFFFFFFFF	Trigger Timer LO of APB Timer with event
TIMER_HI	0x1A106088	32	Status	R/W	0xFFFFFFFF	Trigger Timer HI of APB Timer with event

Table 4.5: SoC Event Generator register table

## 4.5.2 SW\_EVENT

**Address:** 0x1A10\_6000

**Reset Value:** 0x0000\_0000

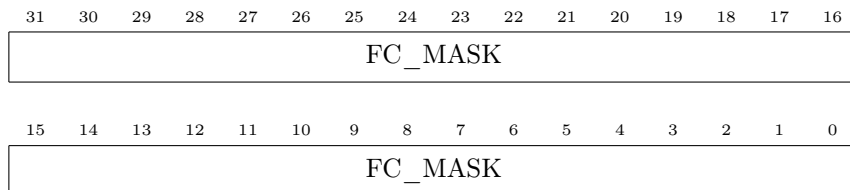


Bit 7-0 **EVENT** (*W*) Writing a one-hot value into EVENT triggers a SoC software event. 8 software events are available.

## 4.5.3 FC\_MASK $X$ , $X = 0 \dots 7$

**Address:** 0x1A10\_6004 + 0x4 \*  $X$

**Reset Value:** 0xFFFF\_FFFF



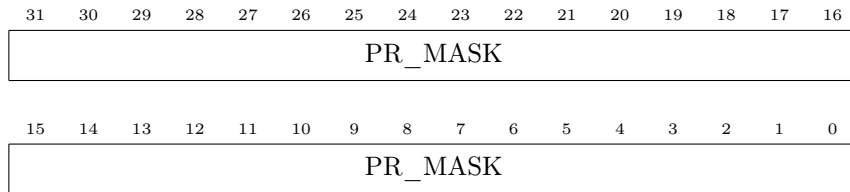
Bit 31-0 **FC\_MASK** (*R/W*) Event Mask to enable/disable event dispatch to FC interrupt controller.

- Setting *bit[i]* to 0b1 disables dispatching *event[32 \* X + i]* to FC interrupt controller.
- Setting *bit[i]* to 0b0 enables dispatching *event[32 \* X + i]* to FC interrupt controller.

#### 4.5.4 PR\_MASKX, X = 0...7

**Address:** 0x1A10\_6044 + 0x4 \* X

**Reset Value:** 0xFFFF\_FFFF



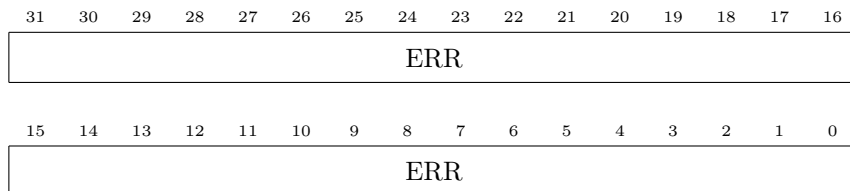
Bit 31-0 **PR\_MASK** (*R/W*) Event Mask to enable/disable event dispatch to peripherals.

- Setting *bit[i]* to 0b1 disables dispatching *event[32 \* X + i]* to peripherals.
- Setting *bit[i]* to 0b0 enables dispatching *event[32 \* X + i]* to peripherals.

#### 4.5.5 ERRX, X = 0...7

**Address:** 0x1A10\_6064 + 0x4 \* X

**Reset Value:** 0x0000\_0000

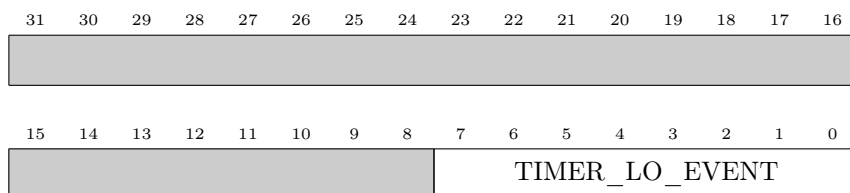


Bit 31-0 **ERR** (*R/W*) Event queue overflow. Clear after read. Reading 0b1 at *ERR[i]* means the event queue of event with id  $32 * X + i$  overflowed.

#### 4.5.6 TIMER\_LO

**Address:** 0x1A10\_6084

**Reset Value:** 0x0000\_0000



Bit 7-0 **TIMER\_LO\_EVENT** (*R/W*) Trigger and start APB Timer LO by the event with id that equals TIMER\_LO\_EVENT

4.5.7 **TIMER\_HI**

Address: 0x1A10\_6088  
Reset Value: 0x0000\_0000



Bit 7-0 **TIMER\_HI\_EVENT** (*R/W*) Trigger and start APB Timer HI by the event with id that equals TIMER\_HI\_EVENT

## 4.6 APB Timer

The APB Timer (`apb_timer_unit.sv`) has the following features:

- 2 general purpose 32-bit upwards counters
- Can be triggered by multiple sources:
  - FLL clock
  - FLL clock + Prescale
  - Reference clock at 32 kHz
  - Any event
- 8-bit programmable prescaler (divides the FLL clock frequency)
- Different counting modes:
  - One shot mode: timer is stopped after the first comparison match
  - Continuous mode: timer continues counting after a match
  - 64-bit cascaded mode: use both 32-bit timers as a 64-bit timer
- Interrupt request generation on comparison match

### 4.6.1 APB Timer Registers

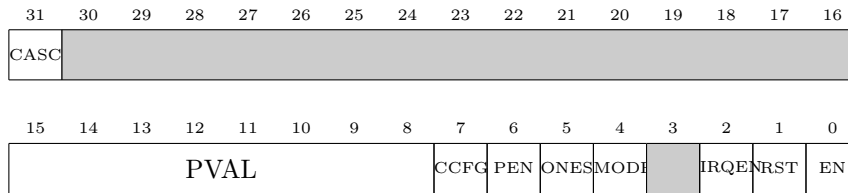
Name	Address	Size	Type	Access	Default	Description
CFG_LO	0x1A10B000	32	Config	R/W	0x00000000	Timer Low Configuration register
CFG_HI	0x1A10B004	32	Config	R/W	0x00000000	Timer High Configuration register
CNT_LO	0x1A10B008	32	Data	R/W	0x00000000	Timer Low counter value register
CNT_HI	0x1A10B00C	32	Data	R/W	0x00000000	Timer High counter value register
CMP_LO	0x1A10B010	32	Config	R/W	0x00000000	Timer Low comparator value register
CMP_HI	0x1A10B014	32	Config	R/W	0x00000000	Timer High comparator value register
START_LO	0x1A10B018	32	Config	R/W	0x00000000	Start Timer Low counting register
START_HI	0x1A10B01C	32	Config	R/W	0x00000000	Start Timer High counting register
RESET_LO	0x1A10B020	32	Config	R/W	0x00000000	Reset Timer Low counter register
RESET_HI	0x1A10B024	32	Config	R/W	0x00000000	Reset Timer High counter register

Table 4.7: APB Timer register table

## 4.6.2 CFG\_LO

Address: 0x1A10\_B000

Reset Value: 0x0000\_0000



Bit 31 **CASC** (*R/W*) Timer low and Timer high 64-bit cascaded mode enable bit

Bit 15-8 **PVAL** (*R/W*) Timer low prescaler value.  $f_{timer} = f_{clk}/(1 + PVAL)$

Bit 7 **CCFG** (*R/W*) Timer low clock source configuration

- 0b0: FLL or FLL plus Prescaler
- 0b1: 32 kHz reference clock

Bit 6 **PEN** (*R/W*) Timer low prescaler enable bit

Bit 5 **ONES** (*R/W*) Timer low one shot configuration

- 0b0: Timer stays enabled after a compare match with CMP\_LO
- 0b1: Timer is disabled after a compare match with CMP\_LO

Bit 4 **MODE** (*R/W*) Timer low continuous mode configuration

- 0b0: Continue incrementing timer low counter after a compare match with CMP\_LO
- 0b1: Reset timer to after a compare match with CMP\_LO

Bit 2 **IRQEN** (*R/W*) Timer low interrupt generation on compare match enable

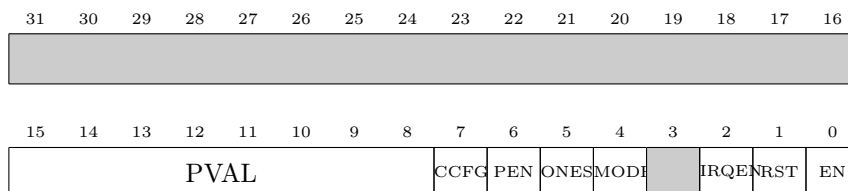
Bit 1 **RST** (*R/W*) Timer low reset, cleared after the reset happened

Bit 0 **EN** (*R/W*) Timer enable (starts counting) bit

## 4.6.3 CFG\_HI

Address: 0x1A10\_B004

Reset Value: 0x0000\_0000



Bit 16-8 **PVAL** (*R/W*) Timer hi prescaler value.  $f_{timer} = f_{clk}/(1 + PVAL)$

Bit 7 **CCFG** (*R/W*) Timer hi clock source configuration

- 0b0: FLL or FLL plus Prescaler

- 0b1: 32 kHz reference clock

Bit 6 **PEN** (*R/W*) Timer hi prescaler enable bit

Bit 5 **ONES** (*R/W*) Timer hi one shot configuration

- 0b0: Timer stays enabled after a compare match with CMP\_HI
- 0b1: Timer is disabled after a compare match with CMP\_HI

Bit 4 **MODE** (*R/W*) Timer hi continuous mode configuration

- 0b0: Continue incrementing timer hi counter after a compare match with CMP\_HI
- 0b1: Reset timer to after a compare match with CMP\_HI

Bit 2 **IRQEN** (*R/W*) Timer hi interrupt generation on compare match enable

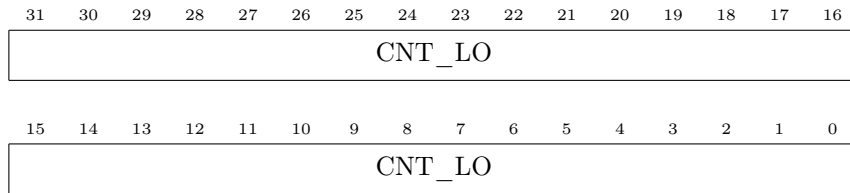
Bit 1 **RST** (*R/W*) Timer hi reset, cleared after the reset happened

Bit 0 **EN** (*R/W*) Timer enable (starts counting) bit

#### 4.6.4 CNT\_LO

**Address:** 0x1A10\_B008

**Reset Value:** 0x0000\_0000

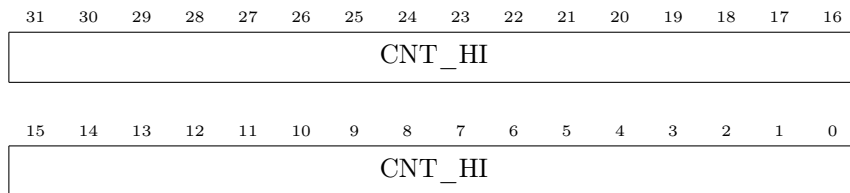


Bit 31-0 **CNT\_LO** (*R/W*) Timer low counter value

#### 4.6.5 CNT\_HI

**Address:** 0x1A10\_B00C

**Reset Value:** 0x0000\_0000

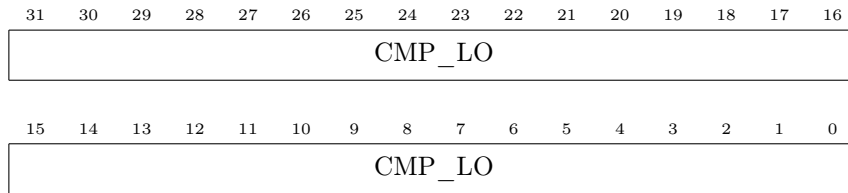


Bit 31-0 **CNT\_HI** (*R/W*) Timer high counter value

## 4.6.6 CMP\_LO

Address: 0x1A10\_B010

Reset Value: 0x0000\_0000

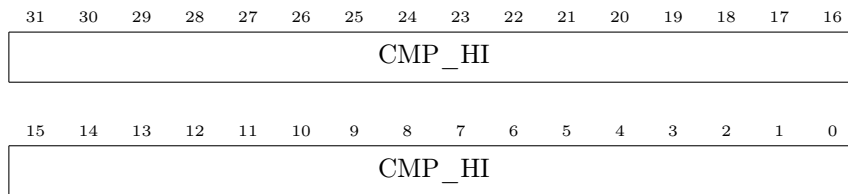


Bit 31-0 **CMP\_LO** (*R/W*) Timer low comparator value

## 4.6.7 CMP\_HI

Address: 0x1A10\_B014

Reset Value: 0x0000\_0000

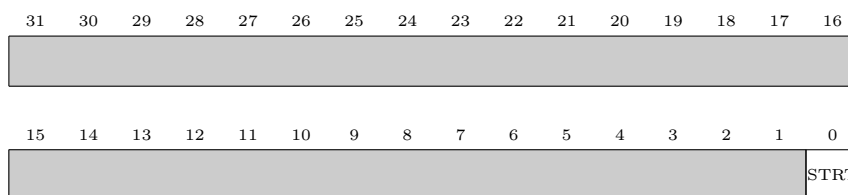


Bit 31-0 **CMP\_HI** (*R/W*) Timer high comparator value

## 4.6.8 START\_LO

Address: 0x1A10\_B018

Reset Value: 0x0000\_0000

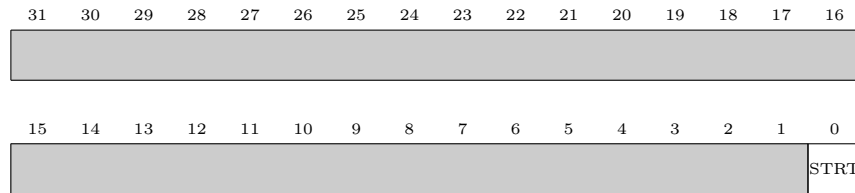


Bit 0 **STRT** (*W*) Timer high start command (sets EN in CFG\_LO)

### 4.6.9 START\_HI

Address: 0x1A10\_B01C

Reset Value: 0x0000\_0000

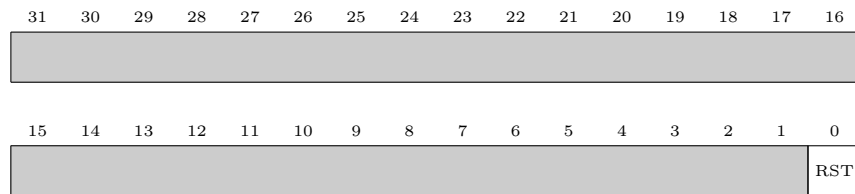


Bit 0 **STRT** (*W*) Timer high start command (sets EN in CFG\_HI)

### 4.6.10 RESET\_LO

Address: 0x1A10\_B020

Reset Value: 0x0000\_0000

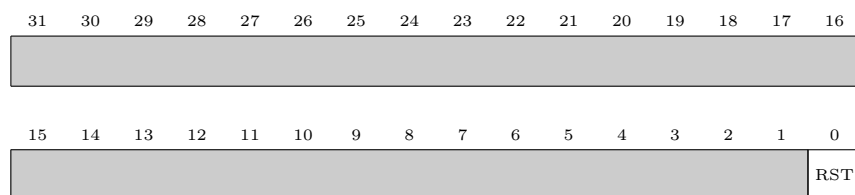


Bit 0 **RST** (*W*) Timer high reset command (writes RST in CFG\_LO)

### 4.6.11 RESET\_HI

Address: 0x1A10\_B024

Reset Value: 0x0000\_0000



Bit 0 **RST** (*W*) Timer high reset command (sets RST in CFG\_HI)



## 4.7 APB Advanced Timer

### 4.7.1 APB Advanced Timer Registers

Name	Address	Size	Type	Access	Default	Description
T0_CMD	0x1A104000	32	Config	R/W	0x00000000	ADV_TIMER0 command register.
T0_CONFIG	0x1A104004	32	Config	R/W	0x00000000	ADV_TIMER0 configuration register.
T0_THRESHOLD	0x1A104008	32	Config	R/W	0x00000000	ADV_TIMER0 threshold configuration register.
T0_TH_CHANNEL0	0x1A10400C	32	Config	R/W	0x00000000	ADV_TIMER0 channel 0 threshold configuration register.
T0_TH_CHANNEL1	0x1A104010	32	Config	R/W	0x00000000	ADV_TIMER0 channel 1 threshold configuration register.
T0_TH_CHANNEL2	0x1A104014	32	Config	R/W	0x00000000	ADV_TIMER0 channel 2 threshold configuration register.
T0_TH_CHANNEL3	0x1A104018	32	Config	R/W	0x00000000	ADV_TIMER0 channel 3 threshold configuration register.
T0_COUNTER	0x1A10402C	32	Status	R	0x00000000	ADV_TIMER0 counter register.
T1_CMD	0x1A104040	32	Config	R/W	0x00000000	ADV_TIMER1 command register.
T1_CONFIG	0x1A104044	32	Config	R/W	0x00000000	ADV_TIMER1 configuration register.
T1_THRESHOLD	0x1A104048	32	Config	R/W	0x00000000	ADV_TIMER1 threshold configuration register.
T1_TH_CHANNEL0	0x1A10404C	32	Config	R/W	0x00000000	ADV_TIMER1 channel 0 threshold configuration register.
T1_TH_CHANNEL1	0x1A104050	32	Config	R/W	0x00000000	ADV_TIMER1 channel 1 threshold configuration register.
T1_TH_CHANNEL2	0x1A104054	32	Config	R/W	0x00000000	ADV_TIMER1 channel 2 threshold configuration register.

T1_TH_CHANNEL3	0x1A104058	32	Config	R/W	0x00000000	ADV_TIMER1 channel 3 thresh- old configuration register.
T1_COUNTER	0x1A10406C	32	Status	R	0x00000000	ADV_TIMER1 counter register.
T2_CMD	0x1A104080	32	Config	R/W	0x00000000	ADV_TIMER2 command register.
T2_CONFIG	0x1A104084	32	Config	R/W	0x00000000	ADV_TIMER2 configuration register.
T2_THRESHOLD	0x1A104088	32	Config	R/W	0x00000000	ADV_TIMER2 threshold configu- ration register.
T2_TH_CHANNEL0	0x1A10408C	32	Config	R/W	0x00000000	ADV_TIMER2 channel 0 thresh- old configuration register.
T2_TH_CHANNEL1	0x1A104090	32	Config	R/W	0x00000000	ADV_TIMER2 channel 1 thresh- old configuration register.
T2_TH_CHANNEL2	0x1A104094	32	Config	R/W	0x00000000	ADV_TIMER2 channel 2 thresh- old configuration register.
T2_TH_CHANNEL3	0x1A104098	32	Config	R/W	0x00000000	ADV_TIMER2 channel 3 thresh- old configuration register.
T2_COUNTER	0x1A1040AC	32	Status	R	0x00000000	ADV_TIMER2 counter register.
T3_CMD	0x1A1040C0	32	Config	R/W	0x00000000	ADV_TIMER3 command register.
T3_CONFIG	0x1A1040C4	32	Config	R/W	0x00000000	ADV_TIMER3 configuration register.
T3_THRESHOLD	0x1A1040C8	32	Config	R/W	0x00000000	ADV_TIMER3 threshold configu- ration register.
T3_TH_CHANNEL0	0x1A1040CC	32	Config	R/W	0x00000000	ADV_TIMER3 channel 0 thresh- old configuration register.
T3_TH_CHANNEL1	0x1A1040D0	32	Config	R/W	0x00000000	ADV_TIMER3 channel 1 thresh- old configuration register.
T3_TH_CHANNEL2	0x1A1040D4	32	Config	R/W	0x00000000	ADV_TIMER3 channel 2 thresh- old configuration register.

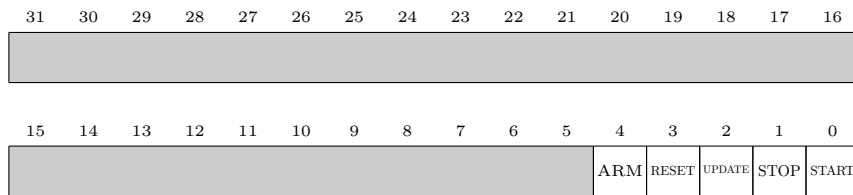
T3_TH_CHANNEL3	0x1A1040D8	32	Config	R/W	0x00000000	ADV_TIMER3 channel 3 threshold configuration register.
T3_COUNTER	0x1A1040EC	32	Status	R	0x00000000	ADV_TIMER3 counter register.
EVENT_CFG	0x1A104100	32	Config	R/W	0x00000000	ADV_TIMERS events configuration register.
CG	0x1A104104	32	Config	R/W	0x00000000	ADV_TIMERS channels clock gating configuration register.

Table 4.8: APB Advanced Timer

## 4.7.2 T0\_CMD

**Address:** 0x1A104000

**Reset Value:** 0x00000000



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.

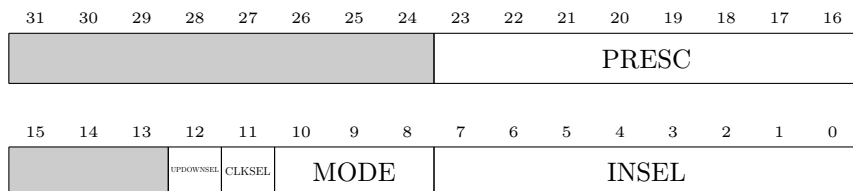
Bit 1 **STOP** (*R/W*) ADV\_TIMER0 stop command bitfield.

Bit 0 **START** (*R/W*) ADV\_TIMER0 start command bitfield.

## 4.7.3 T0\_CONFIG

**Address:** 0x1A104004

**Reset Value:** 0x00000000



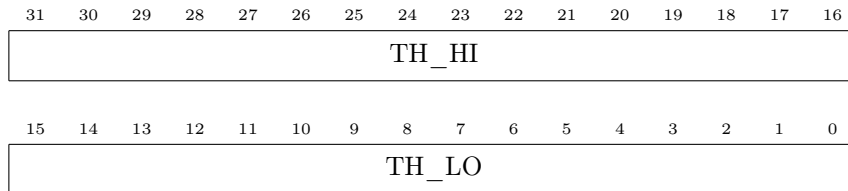
Bit 23 - 16 **PRESC** (*R/W*) ADV\_TIMER0 prescaler value configuration bitfield.

- Bit 12 **UPDOWNSEL** (*R/W*) ADV\_TIMER0 center-aligned mode configuration bitfield:
- 1'b0: The counter counts up and down alternatively.
  - 1'b1: The counter counts up and resets to 0 when reach threshold.
- Bit 11 **CLKSEL** (*R/W*) ADV\_TIMER0 clock source configuration bitfield:
- 1'b0: FLL
  - 1'b1: reference clock at 32kHz
- Bit 10 - 8 **MODE** (*R/W*) ADV\_TIMER0 trigger mode configuration bitfield:
- 3'h0: trigger event at each clock cycle.
  - 3'h1: trigger event if input source is 0
  - 3'h2: trigger event if input source is 1
  - 3'h3: trigger event on input source rising edge
  - 3'h4: trigger event on input source falling edge
  - 3'h5: trigger event on input source falling or rising edge
  - 3'h6: trigger event on input source rising edge when armed
  - 3'h7: trigger event on input source falling edge when armed
- Bit 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

#### 4.7.4 T0\_THRESHOLD

**Address:** 0x1A104008

**Reset Value:** 0x00000000



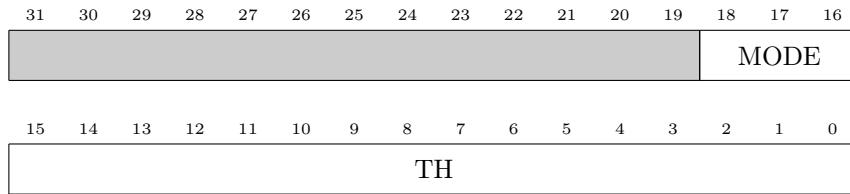
Bit 31 - 16 **TH\_HI** (*R/W*) ADV\_TIMER0 threshold high part configuration bitfield. It defines end counter value.

Bit 15 - 0 **TH\_LO** (*R/W*) ADV\_TIMER0 threshold low part configuration bitfield. It defines start counter value.

#### 4.7.5 T0\_TH\_CHANNEL0

**Address:** 0x1A10400C

**Reset Value:** 0x00000000



Bit 18 - 16 **MODE** (*R/W*) ADV\_TIMER0 channel 0 threshold match action on channel output signal configuration bitfield:

- 3'h0: set.
- 3'h1: toggle then next threshold match action is clear.
- 3'h2: set then next threshold match action is clear.
- 3'h3: toggle.
- 3'h4: clear.
- 3'h5: toggle then next threshold match action is set.
- 3'h6: clear then next threshold match action is set.

Bit 15 - 0 **TH** (*R/W*) ADV\_TIMER0 channel 0 threshold configuration bitfield.

#### 4.7.6 T0\_TH\_CHANNEL1

**Address:** 0x1A104010

**Reset Value:** 0x00000000



Bit 18 - 16 **MODE** (*R/W*) ADV\_TIMER0 channel 1 threshold match action on channel output signal configuration bitfield:

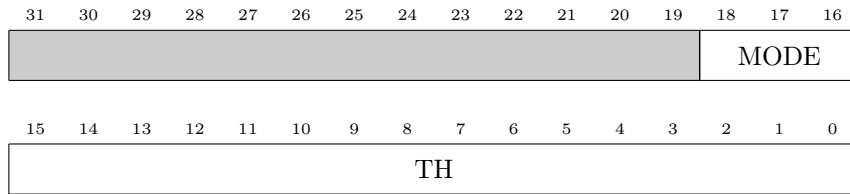
- 3'h0: set.
- 3'h1: toggle then next threshold match action is clear.
- 3'h2: set then next threshold match action is clear.
- 3'h3: toggle.
- 3'h4: clear.
- 3'h5: toggle then next threshold match action is set.
- 3'h6: clear then next threshold match action is set.

Bit 15 - 0 **TH** (*R/W*) ADV\_TIMER0 channel 1 threshold configuration bitfield.

#### 4.7.7 T0\_TH\_CHANNEL2

**Address:** 0x1A104014

**Reset Value:** 0x00000000



Bit 18 - 16 **MODE** (*R/W*) ADV\_TIMER0 channel 2 threshold match action on channel output signal configuration bitfield:

- 3'h0: set.
- 3'h1: toggle then next threshold match action is clear.
- 3'h2: set then next threshold match action is clear.
- 3'h3: toggle.
- 3'h4: clear.
- 3'h5: toggle then next threshold match action is set.
- 3'h6: clear then next threshold match action is set.

Bit 15 - 0 **TH** (*R/W*) ADV\_TIMER0 channel 2 threshold configuration bitfield.

#### 4.7.8 T0\_TH\_CHANNEL3

**Address:** 0x1A104018

**Reset Value:** 0x00000000



Bit 18 - 16 **MODE** (*R/W*) ADV\_TIMER0 channel 3 threshold match action on channel output signal configuration bitfield:

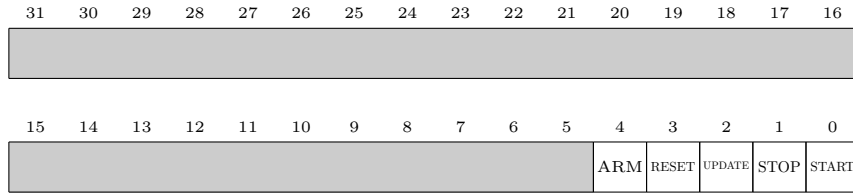
- 3'h0: set.
- 3'h1: toggle then next threshold match action is clear.
- 3'h2: set then next threshold match action is clear.
- 3'h3: toggle.
- 3'h4: clear.
- 3'h5: toggle then next threshold match action is set.
- 3'h6: clear then next threshold match action is set.

Bit 15 - 0 **TH** (*R/W*) ADV\_TIMER0 channel 3 threshold configuration bitfield.

#### 4.7.9 T1\_CMD

**Address:** 0x1A104040

**Reset Value:** 0x00000000



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.

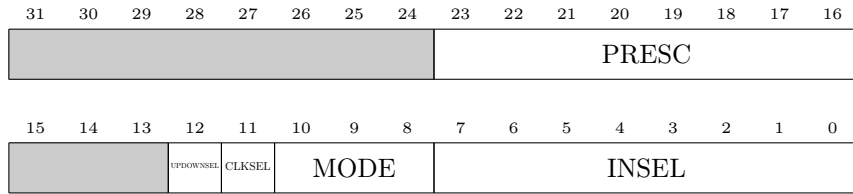
Bit 1 **STOP** (*R/W*) ADV\_TIMER1 stop command bitfield.

Bit 0 **START** (*R/W*) ADV\_TIMER1 start command bitfield.

#### 4.7.10 T1\_CONFIG

**Address:** 0x1A104044

**Reset Value:** 0x00000000



Bit 23 - 16 **PRESC** (*R/W*) ADV\_TIMER1 prescaler value configuration bitfield.

Bit 12 **UPDOWNSEL** (*R/W*) ADV\_TIMER1 center-aligned mode configuration bitfield:

- 1'b0: The counter counts up and down alternatively.
- 1'b1: The counter counts up and resets to 0 when reach threshold.

Bit 11 **CLKSEL** (*R/W*) ADV\_TIMER1 clock source configuration bitfield:

- 1'b0: FLL
- 1'b1: reference clock at 32kHz

Bit 10 - 8 **MODE** (*R/W*) ADV\_TIMER1 trigger mode configuration bitfield:

- 3'h0: trigger event at each clock cycle.
- 3'h1: trigger event if input source is 0
- 3'h2: trigger event if input source is 1
- 3'h3: trigger event on input source rising edge
- 3'h4: trigger event on input source falling edge
- 3'h5: trigger event on input source falling or rising edge
- 3'h6: trigger event on input source rising edge when armed
- 3'h7: trigger event on input source falling edge when armed

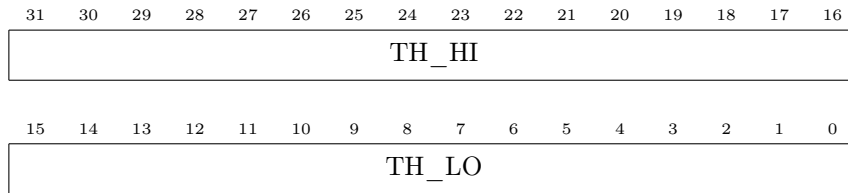
Bit 7 - 0 **INSEL** (*R/W*) ADV\_TIMER1 input source configuration bitfield:

- 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

#### 4.7.11 T1\_THRESHOLD

**Address:** 0x1A104048

**Reset Value:** 0x00000000



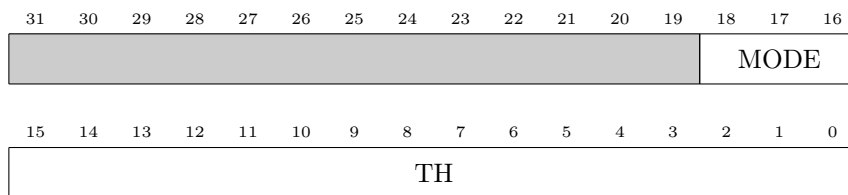
Bit 31 - 16 **TH\_HI** (*R/W*) ADV\_TIMER1 threshold high part configuration bitfield. It defines end counter value.

Bit 15 - 0 **TH\_LO** (*R/W*) ADV\_TIMER1 threshold low part configuration bitfield. It defines start counter value.

#### 4.7.12 T1\_TH\_CHANNEL0

**Address:** 0x1A10404C

**Reset Value:** 0x00000000



Bit 18 - 16 **MODE** (*R/W*) ADV\_TIMER1 channel 0 threshold match action on channel output signal configuration bitfield:

- 3'h0: set.
- 3'h1: toggle then next threshold match action is clear.
- 3'h2: set then next threshold match action is clear.
- 3'h3: toggle.
- 3'h4: clear.
- 3'h5: toggle then next threshold match action is set.
- 3'h6: clear then next threshold match action is set.

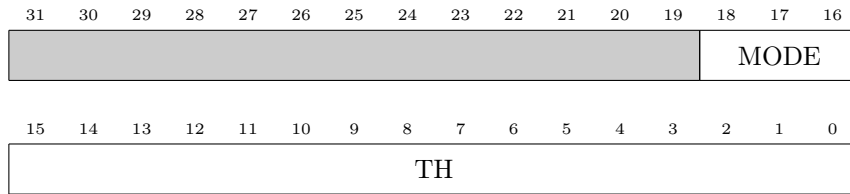
Bit 15 - 0 **TH** (*R/W*) ADV\_TIMER1 channel 0 threshold configuration bitfield.

#### 4.7.13 T1\_TH\_CHANNEL1

**Address:** 0x1A104050

**Reset Value:** 0x00000000





Bit 18 - 16 **MODE** (*R/W*) ADV\_TIMER1 channel 1 threshold match action on channel output signal configuration bitfield:

- 3'h0: set.
- 3'h1: toggle then next threshold match action is clear.
- 3'h2: set then next threshold match action is clear.
- 3'h3: toggle.
- 3'h4: clear.
- 3'h5: toggle then next threshold match action is set.
- 3'h6: clear then next threshold match action is set.

Bit 15 - 0 **TH** (*R/W*) ADV\_TIMER1 channel 1 threshold configuration bitfield.

#### 4.7.14 T1\_TH\_CHANNEL2

**Address:** 0x1A104054

**Reset Value:** 0x00000000



Bit 18 - 16 **MODE** (*R/W*) ADV\_TIMER1 channel 2 threshold match action on channel output signal configuration bitfield:

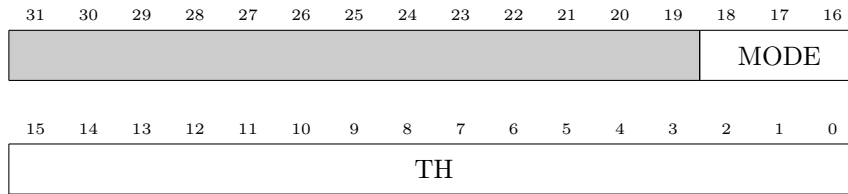
- 3'h0: set.
- 3'h1: toggle then next threshold match action is clear.
- 3'h2: set then next threshold match action is clear.
- 3'h3: toggle.
- 3'h4: clear.
- 3'h5: toggle then next threshold match action is set.
- 3'h6: clear then next threshold match action is set.

Bit 15 - 0 **TH** (*R/W*) ADV\_TIMER1 channel 2 threshold configuration bitfield.

#### 4.7.15 T1\_TH\_CHANNEL3

**Address:** 0x1A104058

**Reset Value:** 0x00000000



Bit 18 - 16 **MODE** (*R/W*) ADV\_TIMER1 channel 3 threshold match action on channel output signal configuration bitfield:

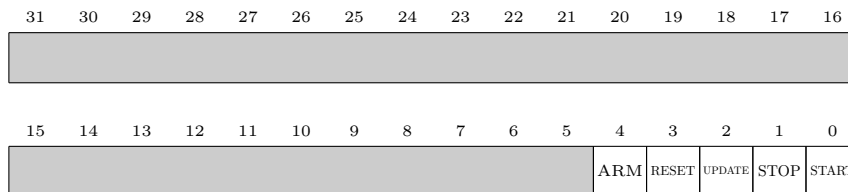
- 3'h0: set.
- 3'h1: toggle then next threshold match action is clear.
- 3'h2: set then next threshold match action is clear.
- 3'h3: toggle.
- 3'h4: clear.
- 3'h5: toggle then next threshold match action is set.
- 3'h6: clear then next threshold match action is set.

Bit 15 - 0 **TH** (*R/W*) ADV\_TIMER1 channel 3 threshold configuration bitfield.

#### 4.7.16 T2\_CMD

**Address:** 0x1A104080

**Reset Value:** 0x00000000



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.

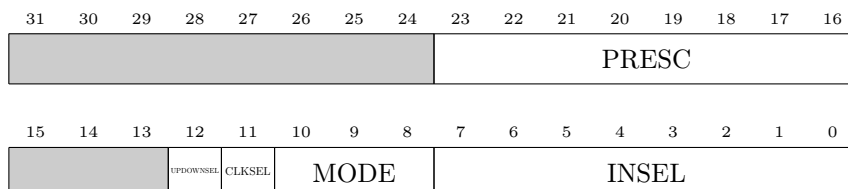
Bit 1 **STOP** (*R/W*) ADV\_TIMER2 stop command bitfield.

Bit 0 **START** (*R/W*) ADV\_TIMER2 start command bitfield.

#### 4.7.17 T2\_CONFIG

**Address:** 0x1A104084

**Reset Value:** 0x00000000



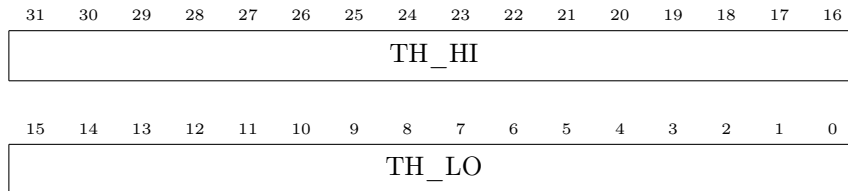
Bit 23 - 16 **PRESC** (*R/W*) ADV\_TIMER2 prescaler value configuration bitfield.

- Bit 12 **UPDOWNSEL** (*R/W*) ADV\_TIMER2 center-aligned mode configuration bitfield:
- 1'b0: The counter counts up and down alternatively.
  - 1'b1: The counter counts up and resets to 0 when reach threshold.
- Bit 11 **CLKSEL** (*R/W*) ADV\_TIMER2 clock source configuration bitfield:
- 1'b0: FLL
  - 1'b1: reference clock at 32kHz
- Bit 10 - 8 **MODE** (*R/W*) ADV\_TIMER2 trigger mode configuration bitfield:
- 3'h0: trigger event at each clock cycle.
  - 3'h1: trigger event if input source is 0
  - 3'h2: trigger event if input source is 1
  - 3'h3: trigger event on input source rising edge
  - 3'h4: trigger event on input source falling edge
  - 3'h5: trigger event on input source falling or rising edge
  - 3'h6: trigger event on input source rising edge when armed
  - 3'h7: trigger event on input source falling edge when armed
- Bit 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

#### 4.7.18 T2\_THRESHOLD

**Address:** 0x1A104088

**Reset Value:** 0x00000000



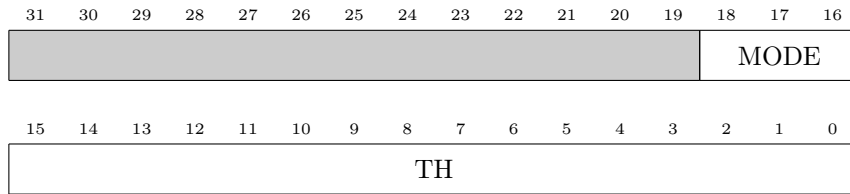
Bit 31 - 16 **TH\_HI** (*R/W*) ADV\_TIMER2 threshold high part configuration bitfield. It defines end counter value.

Bit 15 - 0 **TH\_LO** (*R/W*) ADV\_TIMER2 threshold low part configuration bitfield. It defines start counter value.

#### 4.7.19 T2\_TH\_CHANNEL0

**Address:** 0x1A10408C

**Reset Value:** 0x00000000



Bit 18 - 16 **MODE** (*R/W*) ADV\_TIMER2 channel 0 threshold match action on channel output signal configuration bitfield:

- 3'h0: set.
- 3'h1: toggle then next threshold match action is clear.
- 3'h2: set then next threshold match action is clear.
- 3'h3: toggle.
- 3'h4: clear.
- 3'h5: toggle then next threshold match action is set.
- 3'h6: clear then next threshold match action is set.

Bit 15 - 0 **TH** (*R/W*) ADV\_TIMER2 channel 0 threshold configuration bitfield.

#### 4.7.20 T2\_TH\_CHANNEL1

**Address:** 0x1A104090

**Reset Value:** 0x00000000



Bit 18 - 16 **MODE** (*R/W*) ADV\_TIMER2 channel 1 threshold match action on channel output signal configuration bitfield:

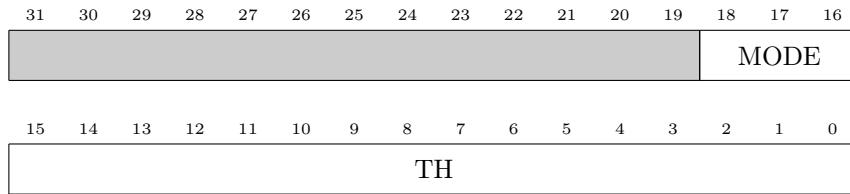
- 3'h0: set.
- 3'h1: toggle then next threshold match action is clear.
- 3'h2: set then next threshold match action is clear.
- 3'h3: toggle.
- 3'h4: clear.
- 3'h5: toggle then next threshold match action is set.
- 3'h6: clear then next threshold match action is set.

Bit 15 - 0 **TH** (*R/W*) ADV\_TIMER2 channel 1 threshold configuration bitfield.

#### 4.7.21 T2\_TH\_CHANNEL2

**Address:** 0x1A104094

**Reset Value:** 0x00000000



Bit 18 - 16 **MODE** (*R/W*) ADV\_TIMER2 channel 2 threshold match action on channel output signal configuration bitfield:

- 3'h0: set.
- 3'h1: toggle then next threshold match action is clear.
- 3'h2: set then next threshold match action is clear.
- 3'h3: toggle.
- 3'h4: clear.
- 3'h5: toggle then next threshold match action is set.
- 3'h6: clear then next threshold match action is set.

Bit 15 - 0 **TH** (*R/W*) ADV\_TIMER2 channel 2 threshold configuration bitfield.

#### 4.7.22 T2\_TH\_CHANNEL3

**Address:** 0x1A104098

**Reset Value:** 0x00000000



Bit 18 - 16 **MODE** (*R/W*) ADV\_TIMER2 channel 3 threshold match action on channel output signal configuration bitfield:

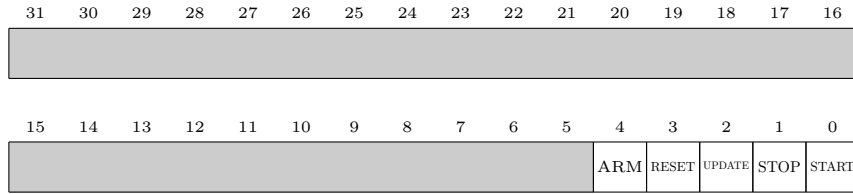
- 3'h0: set.
- 3'h1: toggle then next threshold match action is clear.
- 3'h2: set then next threshold match action is clear.
- 3'h3: toggle.
- 3'h4: clear.
- 3'h5: toggle then next threshold match action is set.
- 3'h6: clear then next threshold match action is set.

Bit 15 - 0 **TH** (*R/W*) ADV\_TIMER2 channel 3 threshold configuration bitfield.

#### 4.7.23 T3\_CMD

**Address:** 0x1A1040C0

**Reset Value:** 0x00000000



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.

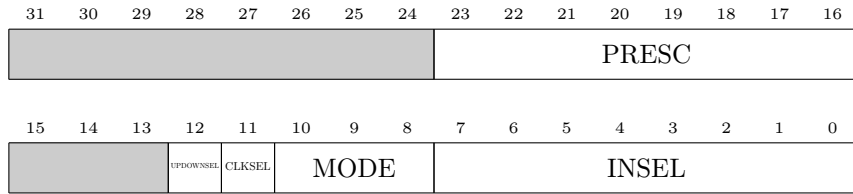
Bit 1 **STOP** (*R/W*) ADV\_TIMER3 stop command bitfield.

Bit 0 **START** (*R/W*) ADV\_TIMER3 start command bitfield.

#### 4.7.24 T3\_CONFIG

**Address:** 0x1A1040C4

**Reset Value:** 0x00000000



Bit 23 - 16 **PRESC** (*R/W*) ADV\_TIMER3 prescaler value configuration bitfield.

Bit 12 **UPDOWNSEL** (*R/W*) ADV\_TIMER3 center-aligned mode configuration bitfield:

- 1'b0: The counter counts up and down alternatively.
- 1'b1: The counter counts up and resets to 0 when reach threshold.

Bit 11 **CLKSEL** (*R/W*) ADV\_TIMER3 clock source configuration bitfield:

- 1'b0: FLL
- 1'b1: reference clock at 32kHz

Bit 10 - 8 **MODE** (*R/W*) ADV\_TIMER3 trigger mode configuration bitfield:

- 3'h0: trigger event at each clock cycle.
- 3'h1: trigger event if input source is 0
- 3'h2: trigger event if input source is 1
- 3'h3: trigger event on input source rising edge
- 3'h4: trigger event on input source falling edge
- 3'h5: trigger event on input source falling or rising edge
- 3'h6: trigger event on input source rising edge when armed
- 3'h7: trigger event on input source falling edge when armed

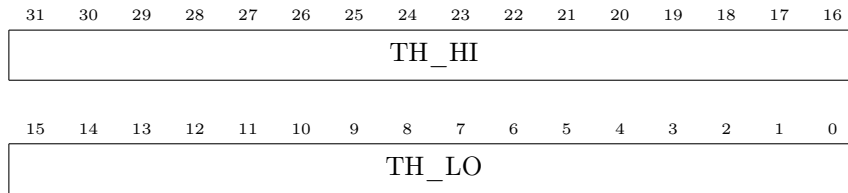
Bit 7 - 0 **INSEL** (*R/W*) ADV\_TIMER3 input source configuration bitfield:

- 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

#### 4.7.25 T3\_THRESHOLD

**Address:** 0x1A1040C8

**Reset Value:** 0x00000000



Bit 31 - 16 **TH\_HI** (*R/W*) ADV\_TIMER3 threshold high part configuration bitfield. It defines end counter value.

Bit 15 - 0 **TH\_LO** (*R/W*) ADV\_TIMER3 threshold low part configuration bitfield. It defines start counter value.

#### 4.7.26 T3\_TH\_CHANNEL0

**Address:** 0x1A1040CC

**Reset Value:** 0x00000000



Bit 18 - 16 **MODE** (*R/W*) ADV\_TIMER3 channel 0 threshold match action on channel output signal configuration bitfield:

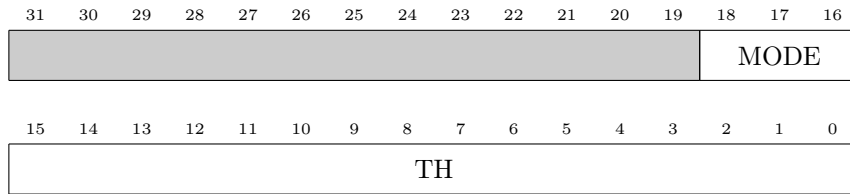
- 3'h0: set.
- 3'h1: toggle then next threshold match action is clear.
- 3'h2: set then next threshold match action is clear.
- 3'h3: toggle.
- 3'h4: clear.
- 3'h5: toggle then next threshold match action is set.
- 3'h6: clear then next threshold match action is set.

Bit 15 - 0 **TH** (*R/W*) ADV\_TIMER3 channel 0 threshold configuration bitfield.

#### 4.7.27 T3\_TH\_CHANNEL1

**Address:** 0x1A1040D0

**Reset Value:** 0x00000000



Bit 18 - 16 **MODE** (*R/W*) ADV\_TIMER3 channel 1 threshold match action on channel output signal configuration bitfield:

- 3'h0: set.
- 3'h1: toggle then next threshold match action is clear.
- 3'h2: set then next threshold match action is clear.
- 3'h3: toggle.
- 3'h4: clear.
- 3'h5: toggle then next threshold match action is set.
- 3'h6: clear then next threshold match action is set.

Bit 15 - 0 **TH** (*R/W*) ADV\_TIMER3 channel 1 threshold configuration bitfield.

#### 4.7.28 T3\_TH\_CHANNEL2

**Address:** 0x1A1040D4

**Reset Value:** 0x00000000



Bit 18 - 16 **MODE** (*R/W*) ADV\_TIMER3 channel 2 threshold match action on channel output signal configuration bitfield:

- 3'h0: set.
- 3'h1: toggle then next threshold match action is clear.
- 3'h2: set then next threshold match action is clear.
- 3'h3: toggle.
- 3'h4: clear.
- 3'h5: toggle then next threshold match action is set.
- 3'h6: clear then next threshold match action is set.

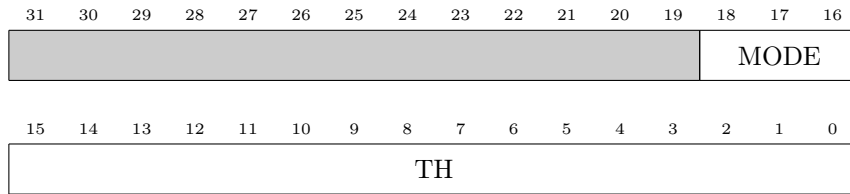
Bit 15 - 0 **TH** (*R/W*) ADV\_TIMER3 channel 2 threshold configuration bitfield.

#### 4.7.29 T3\_TH\_CHANNEL3

**Address:** 0x1A1040D8

**Reset Value:** 0x00000000





Bit 18 - 16 **MODE** (*R/W*) ADV\_TIMER3 channel 3 threshold match action on channel output signal configuration bitfield:

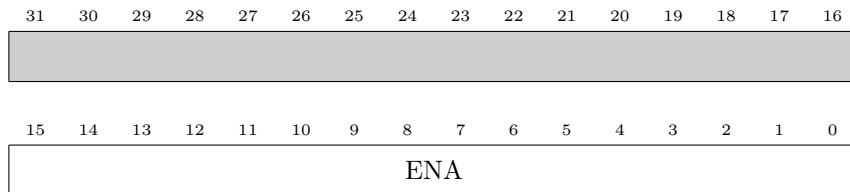
- 3'h0: set.
- 3'h1: toggle then next threshold match action is clear.
- 3'h2: set then next threshold match action is clear.
- 3'h3: toggle.
- 3'h4: clear.
- 3'h5: toggle then next threshold match action is set.
- 3'h6: clear then next threshold match action is set.

Bit 15 - 0 **TH** (*R/W*) ADV\_TIMER3 channel 3 threshold configuration bitfield.

### 4.7.30 CG

**Address:** 0x1A104104

**Reset Value:** 0x00000000



Bit 15 - 0 **ENA** (*R/W*) ADV\_TIMER clock gating configuration bitfield.

- ENA[i]=0: clock gate ADV\_TIMERi.
- ENA[i]=1: enable ADV\_TIMERi.

## 4.8 uDMA Subsystem

### 4.8.1 uDMA UART Registers

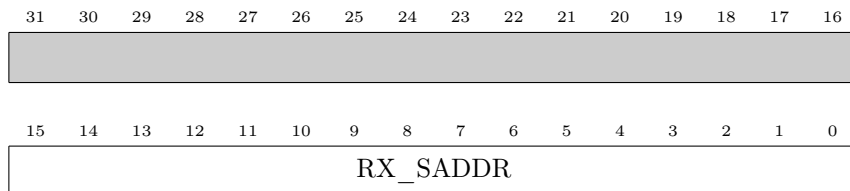
Name	Address	Size	Type	Access	Default	Description
RX_SADDR	0x1A102000	32	Config	R/W	0x00000000	uDMA RX UART buffer base address configuration register.
RX_SIZE	0x1A102004	32	Config	R/W	0x00000000	uDMA RX UART buffer size configuration register.
RX_CFG	0x1A102008	32	Config	R/W	0x00000000	uDMA RX UART stream configuration register.
TX_SADDR	0x1A102010	32	Config	R/W	0x00000000	uDMA TX UART buffer base address configuration register.
TX_SIZE	0x1A102014	32	Config	R/W	0x00000000	uDMA TX UART buffer size configuration register.
TX_CFG	0x1A102018	32	Config	R/W	0x00000000	uDMA TX UART stream configuration register.
STATUS	0x1A102020	32	Status	R	0x00000000	uDMA UART status register.
SETUP	0x1A102024	32	Config	R/W	0x00000000	UDMA UART configuration register.
ERROR	0x1A102028	32	Status	R	0x00000000	uDMA UART Error status
IRQ_EN	0x1A10202C	32	Config	R/W	0x00000000	uDMA UART Read or Error interrupt enable register.
VALID	0x1A102030	32	Status	R	0x00000000	uDMA UART Read polling data valid flag register.
DATA	0x1A102034	32	Data	R	0x00000000	uDMA UART Read polling data register.

Table 4.9: uDMA UART

### 4.8.2 RX\_SADDR

**Address:** 0x1A102000

**Reset Value:** 0x00000000



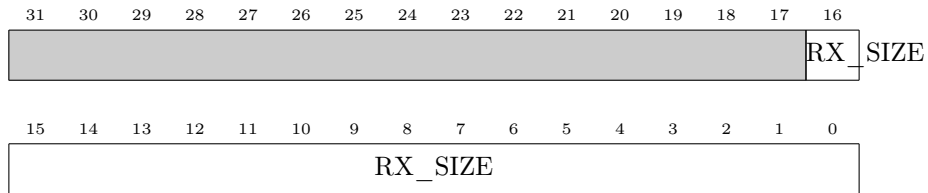
Bit 15 - 0 **RX\_SADDR** (*R/W*) RX buffer base address bitfield:

- Read: returns value of the buffer pointer until transfer is finished. Else returns 0.
- Write: sets RX buffer base address

### 4.8.3 RX\_SIZE

Address: 0x1A102004

Reset Value: 0x00000000

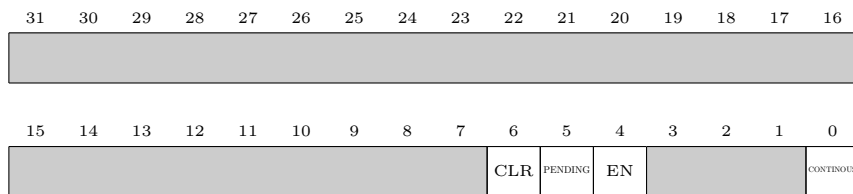


Bit 16 - 0 **RX\_SIZE** (*R/W*) RX buffer size bitfield in bytes. (128kBytes maximum)  
 - Read: returns remaining buffer size to transfer.  
 - Write: sets buffer size.

### 4.8.4 RX\_CFG

Address: 0x1A102008

Reset Value: 0x00000000



Bit 6 **CLR** (*W*) RX channel clear and stop transfer:

- 1'b0: disable
- 1'b1: stop and clear the on-going transfer

Bit 5 **PENDING** (*R*) RX transfer pending in queue status flag:

- 1'b0: no pending transfer in the queue
- 1'b1: pending transfer in the queue

Bit 4 **EN** (*R/W*) RX channel enable and start transfer bitfield:

- 1'b0: disable
  - 1'b1: 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:

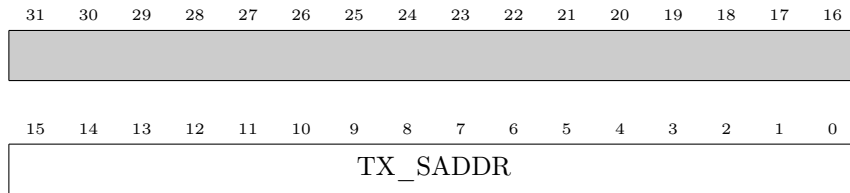
- 1'b0: disabled
- 1'b1: enabled

At the end of the buffer transfer, the uDMA reloads the address / buffer size and starts a new transfer.

## 4.8.5 TX\_SADDR

Address: 0x1A102010

Reset Value: 0x00000000



Bit 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

## 4.8.6 TX\_SIZE

Address: 0x1A102014

Reset Value: 0x00000000



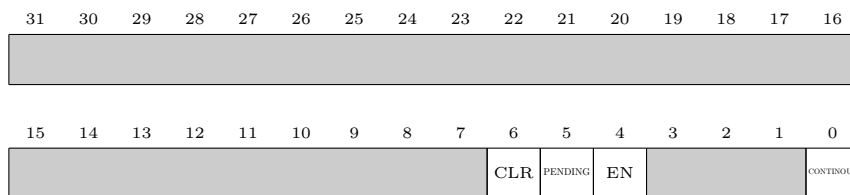
Bit 16 - 0 **TX\_SIZE** (*R/W*) TX buffer size bitfield in bytes. (128kBytes maximum)

- Read: returns remaining buffer size to transfer.
- Write: sets buffer size.

## 4.8.7 TX\_CFG

Address: 0x1A102018

Reset Value: 0x00000000



Bit 6 **CLR** (*W*) TX channel clear and stop transfer bitfield:

- 1'b0: disabled
- 1'b1: stop and clear the on-going transfer

Bit 5 **PENDING** (*R*) TX transfer pending in queue status flag:

- 1'b0: no pending transfer in the queue
- 1'b1: pending transfer in the queue

Bit 4 **EN** (*R/W*) TX channel enable and start transfer bitfield:

- 1'b0: disabled

- 1'b1: 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:

- 1'b0: disabled

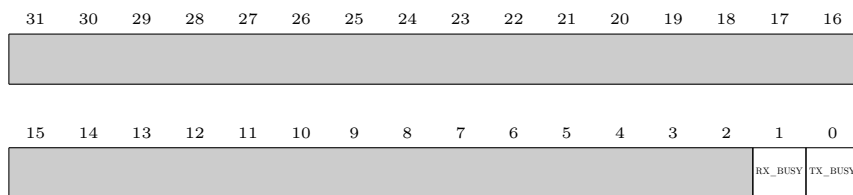
- 1'b1: enabled

At the end of the buffer transfer, the uDMA reloads the address / buffer size and starts a new transfer.

## 4.8.8 STATUS

Address: 0x1A102020

Reset Value: 0x00000000



Bit 1 **RX\_BUSY** (*R*) RX busy status flag:

- 1'b0: no RX transfer on-going

- 1'b1: RX transfer on-going

Bit 0 **TX\_BUSY** (*R*) TX busy status flag:

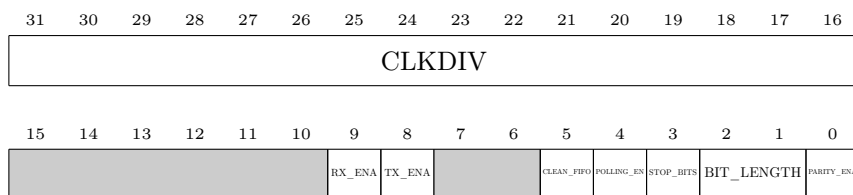
- 1'b0: no TX transfer on-going

- 1'b1: TX transfer on-going

## 4.8.9 SETUP

Address: 0x1A102024

Reset Value: 0x00000000



Bit 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:

- 1'b0: disabled

- 1'b1: enabled

Bit 8 **TX\_ENA** (*R/W*) TX transceiver configuration bitfield:

- 1'b0: disabled

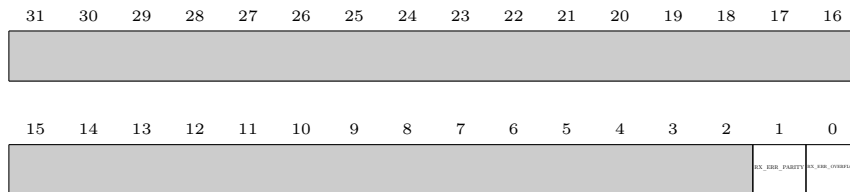
- 1'b1: enabled

- Bit 5 **CLEAN\_FIFO** (*R/W*) In all mode clean the RX fifo, set 1 then set 0 to realize a reset fifo:
- 1'b0: Stop Clean the RX FIFO.
  - 1'b1: Clean the RX FIFO.
- Bit 4 **POLLING\_EN** (*R/W*) When in uart read, use polling method to read the data, read interrupt enable flag will be ignored:
- 1'b0: Do not using polling method to read data.
  - 1'b1: Using polling method to read data. Interrupt enable flag will be ignored.
- Bit 3 **STOP\_BITS** (*R/W*) Stop bits length bitfield:
- 1'b0: 1 stop bit
  - 1'b1: 2 stop bits
- Bit 2 - 1 **BIT\_LENGTH** (*R/W*) Character length bitfield:
- 2'b00: 5 bits
  - 2'b01: 6 bits
  - 2'b10: 7 bits
  - 2'b11: 8 bits
- Bit 0 **PARITY\_ENA** (*R/W*) Parity bit generation and check configuration bitfield:
- 1'b0: disabled
  - 1'b1: enabled

#### 4.8.10 ERROR

**Address:** 0x1A102028

**Reset Value:** 0x00000000

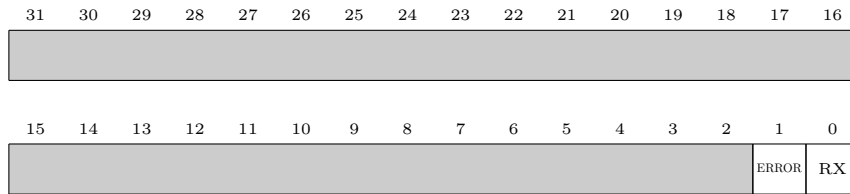


- Bit 1 **RX\_ERR\_PARITY** (*R*) RX parity error status flag:
- 1'b0: no error
  - 1'b1: RX parity error occurred
- Bit 0 **RX\_ERR\_OVERFLOW** (*R*) RX overflow error status flag:
- 1'b0: no error
  - 1'b1: RX overflow error occurred

#### 4.8.11 IRQ\_EN

**Address:** 0x1A10202C

**Reset Value:** 0x00000000



Bit 1 **ERROR** (*R/W*) Error interrupt in enable flag:

- 1'b0: Error IRQ disable
- 1'b1: Error IRQ enable

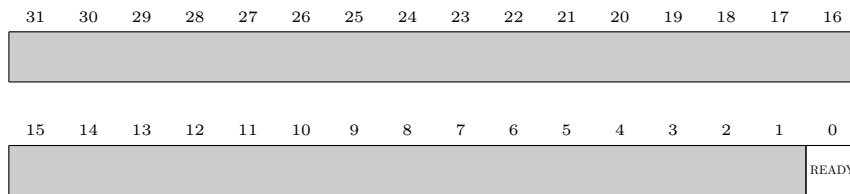
Bit 0 **RX** (*R/W*) Rx interrupt in enable flag:

- 1'b0: RX IRQ disable
- 1'b1: RX IRQ enable

#### 4.8.12 VALID

Address: 0x1A102030

Reset Value: 0x00000000



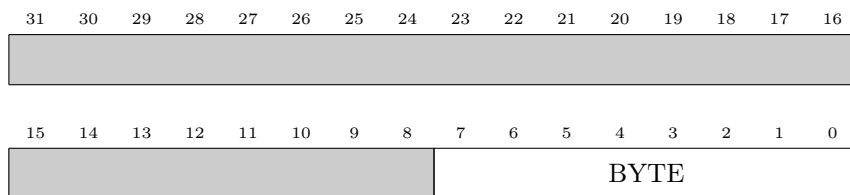
Bit 0 **READY** (*R*) Used only in RX polling method to indicate data is ready for read:

- 1'b0: Data is not ready to read
- 1'b1: Data is ready to read

#### 4.8.13 DATA

Address: 0x1A102034

Reset Value: 0x00000000



Bit 7 - 0 **BYTE** (*R*) RX read data for polling or interrupt

#### 4.8.14 uDMA SPI Registers

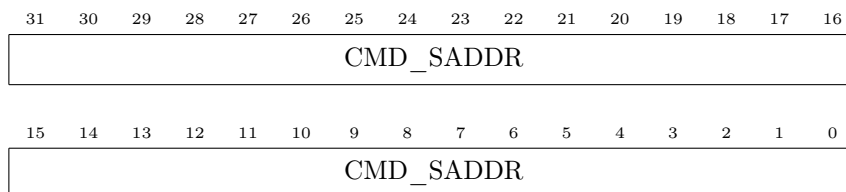
Name	Address	Size	Type	Access	Default	Description
SPIM_RX_SADDR	0x1A102080	32	Config	R/W	0x00000000	RX SPI uDMA transfer address of associated buffer
SPIM_RX_SIZE	0x1A102084	32	Config	R/W	0x00000000	RX SPI uDMA transfer size of buffer
SPIM_RX_CFG	0x1A102088	32	Config	R/W	0x00000004	RX SPI uDMA transfer configuration
SPIM_TX_SADDR	0x1A102090	32	Config	R/W	0x00000000	TX SPI uDMA transfer address of associated buffer
SPIM_TX_SIZE	0x1A102094	32	Config	R/W	0x00000000	TX SPI uDMA transfer size of buffer
SPIM_TX_CFG	0x1A102098	32	Config	R/W	0x00000000	TX SPI uDMA transfer configuration
SPIM_CMD_SADDR	0x1A1020A0	32	Config	R/W	0x00000000	CMD SPI uDMA transfer address of associated buffer
SPIM_CMD_SIZE	0x1A1020A4	32	Config	R/W	0x00000000	CMD SPI uDMA transfer size of buffer
SPIM_CMD_CFG	0x1A1020A8	32	Config	R/W	0x00000004	CMD SPI uDMA transfer configuration

Table 4.10: uDMA SPI

#### 4.8.15 SPIM\_CMD\_SADDR

**Address:** 0x1A1020A0

**Reset Value:** 0x00000000



Bit 31 - 0 **CMD SADDR** (*R/W*) Configure pointer to memory buffer:

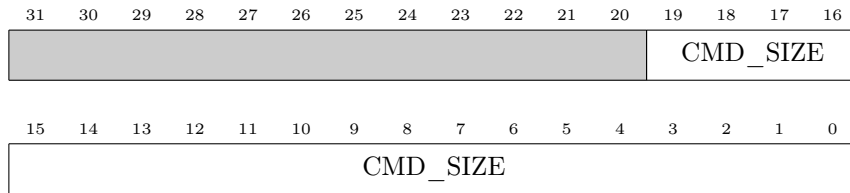
- Read: value of the pointer until transfer is over. Else returns 0
- Write: set Address Pointer to memory buffer start address



#### 4.8.16 SPIM\_CMD\_SIZE

Address: 0x1A1020A4

Reset Value: 0x00000000



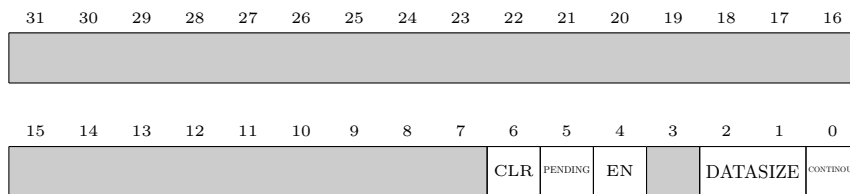
Bit 19 - 0 **CMD\_SIZE** (*R/W*) Buffer size in bytes. (1MBytes maximum)

- Read: buffer size left
- Write: set buffer size

#### 4.8.17 SPIM\_CMD\_CFG

Address: 0x1A1020A8

Reset Value: 0x00000004



Bit 6 **CLR** (*W*) Channel clear and stop transfer:

- 1'b0: disable
- 1'b1: enable

Bit 5 **PENDING** (*R*) Transfer pending in queue status flag:

- 1'b0: free
- 1'b1: pending

Bit 4 **EN** (*R/W*) Channel enable and start transfer:

- 1'b0: disable
- 1'b1: enable

This signal is used also to queue a transfer if one is already ongoing.

Bit 2 - 1 **DATASIZE** (*R/W*) Channel transfer size used to increment uDMA buffer address pointer:

- 2'b00: +1 (8 bits)
- 2'b01: +2 (16 bits)
- 2'b10: +4 (32 bits)(default)
- 2'b11: +0

Bit 0 **CONTINUOUS** (*R/W*) Channel continuous mode:

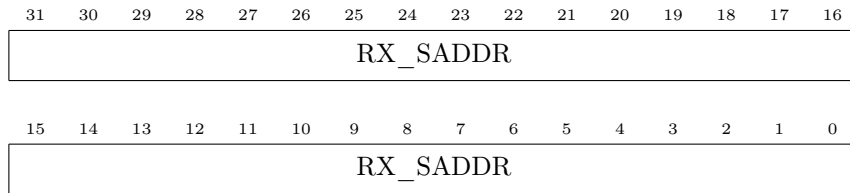
- 1'b0: disable
- 1'b1: enable

At the end of the buffer the uDMA reloads the address and size and starts a new transfer.

#### 4.8.18 SPIM\_RX\_SADDR

Address: 0x1A102080

Reset Value: 0x00000000



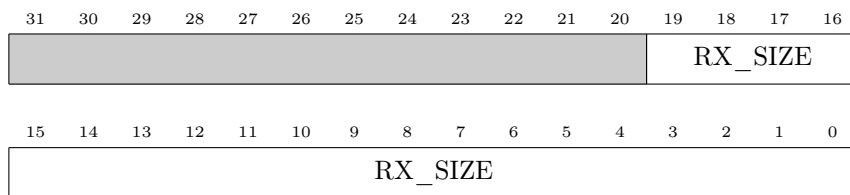
Bit 31 - 0 **RX\_SADDR** (*R/W*) Configure pointer to memory buffer:

- Read: value of the pointer until transfer is over. Else returns 0
- Write: set Address Pointer to memory buffer start address

#### 4.8.19 SPIM\_RX\_SIZE

Address: 0x1A102084

Reset Value: 0x00000000



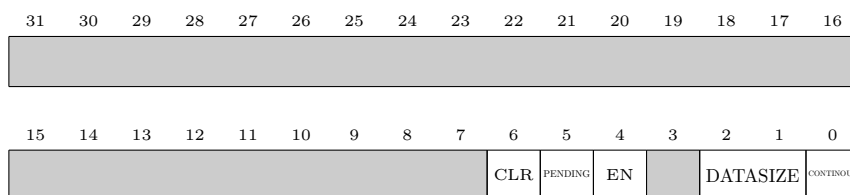
Bit 19 - 0 **RX\_SIZE** (*R/W*) Buffer size in bytes. (1MBytes maximum)

- Read: buffer size left
- Write: set buffer size

#### 4.8.20 SPIM\_RX\_CFG

Address: 0x1A102088

Reset Value: 0x00000004



Bit 6 **CLR** (*W*) Channel clear and stop transfer:

- 1'b0: disable
- 1'b1: enable

Bit 5 **PENDING** (*R*) Transfer pending in queue status flag:

- 1'b0: free
- 1'b1: pending

Bit 4 **EN** (*R/W*) Channel enable and start transfer:

-1'b0: disable

-1'b1: enable

This signal is used also to queue a transfer if one is already ongoing.

Bit 2 - 1 **DATASIZE** (*R/W*) Channel transfer size used to increment uDMA buffer address pointer:

- 2'b00: +1 (8 bits)

- 2'b01: +2 (16 bits)

- 2'b10: +4 (32 bits)(default)

- 2'b11: +0

Bit 0 **CONTINUOUS** (*R/W*) Channel continuous mode:

-1'b0: disable

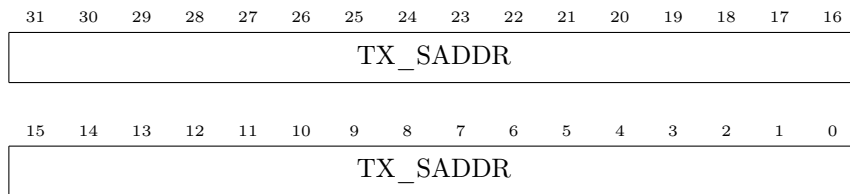
-1'b1: enable

At the end of the buffer the uDMA reloads the address and size and starts a new transfer.

#### 4.8.21 SPIM\_TX\_SADDR

**Address:** 0x1A102090

**Reset Value:** 0x00000000



Bit 31 - 0 **TX\_SADDR** (*R/W*) Configure pointer to memory buffer:

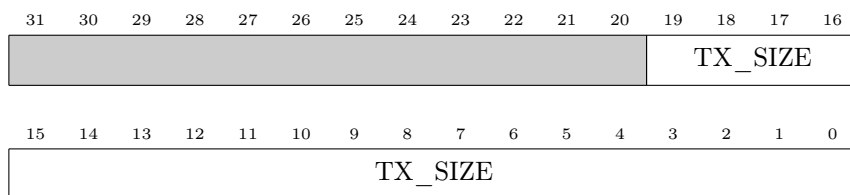
- Read: value of the pointer until transfer is over. Else returns 0

- Write: set Address Pointer to memory buffer start address

#### 4.8.22 SPIM\_TX\_SIZE

**Address:** 0x1A102094

**Reset Value:** 0x00000000



Bit 19 - 0 **TX\_SIZE** (*R/W*) Buffer size in bytes. (1MBytes maximum)

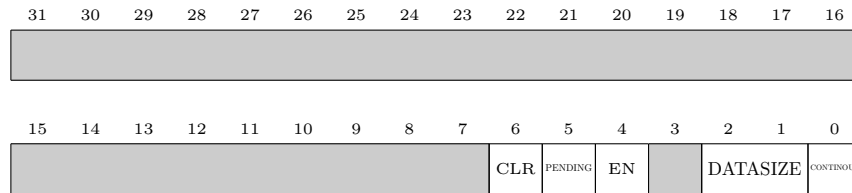
- Read: buffer size left

- Write: set buffer size

### 4.8.23 SPIM\_TX\_CFG

Address: 0x1A102098

Reset Value: 0x00000000



Bit 6 **CLR** (*W*) Channel clear and stop transfer:

-1'b0: disable

-1'b1: enable

Bit 5 **PENDING** (*R*) Transfer pending in queue status flag:

-1'b0: free

-1'b1: pending

Bit 4 **EN** (*R/W*) Channel enable and start transfer:

-1'b0: disable

-1'b1: enable

This signal is used also to queue a transfer if one is already ongoing.

Bit 2 - 1 **DATASIZE** (*R/W*) Channel transfer size used to increment uDMA buffer address pointer:

- 2'b00: +1 (8 bits)

- 2'b01: +2 (16 bits)

- 2'b10: +4 (32 bits)(default)

- 2'b11: +0

Bit 0 **CONTINUOUS** (*R/W*) Channel continuous mode:

-1'b0: disable

-1'b1: enable

At the end of the buffer the uDMA reloads the address and size and starts a new transfer.

### 4.8.24 uDMA I2C0 Registers

Name	Address	Size	Type	Access	Default	Description
RX_SADDR	0x1A102100	32	Config	R/W	0x00000000	uDMA RX I2C buffer base address configuration register.
RX_SIZE	0x1A102104	32	Config	R/W	0x00000000	uDMA RX I2C buffer size configuration register.
RX_CFG	0x1A102108	32	Config	R/W	0x00000000	uDMA RX I2C stream configuration register.
TX_SADDR	0x1A102110	32	Config	R/W	0x00000000	uDMA TX I2C buffer base address configuration register.
TX_SIZE	0x1A102114	32	Config	R/W	0x00000000	uDMA TX I2C buffer size configuration register.
TX_CFG	0x1A102118	32	Config	R/W	0x00000000	uDMA TX I2C stream configuration register.

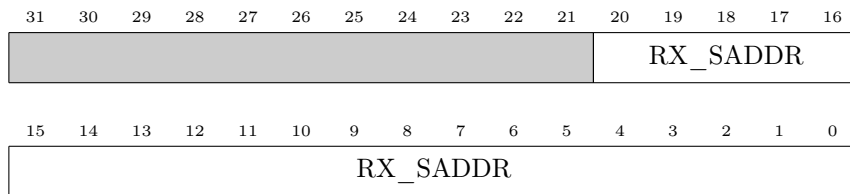
CMD_SADDR	0x1A102120	32	Config	R/W	0x00000000	uDMA CMD I2C buffer base address configuration register.
CMD_SIZE	0x1A102124	32	Config	R/W	0x00000000	uDMA CMD I2C buffer size configuration register.
CMD_CFG	0x1A102128	32	Config	R/W	0x00000000	uDMA CMD I2C stream configuration register.
STATUS	0x1A102130	32	Status	R/W	0x00000000	uDMA I2C Status register.
SETUP	0x1A102134	32	Config	R/W	0x00000000	uDMA I2C Configuration register.

Table 4.11: uDMA I2C0

#### 4.8.25 RX\_SADDR

**Address:** 0x1A102100

**Reset Value:** 0x00000000



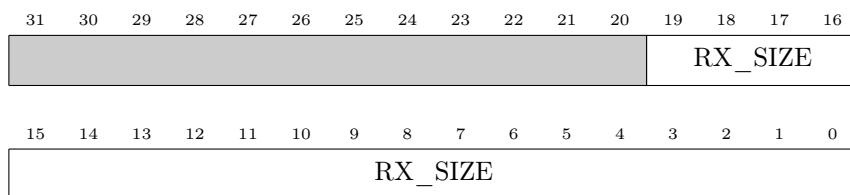
Bit 20 - 0 **RX\_SADDR** (*R/W*) RX buffer base address bitfield:

- Read: returns value of the buffer pointer until transfer is finished. Else returns 0.
- Write: sets RX buffer base address

#### 4.8.26 RX\_SIZE

**Address:** 0x1A102104

**Reset Value:** 0x00000000



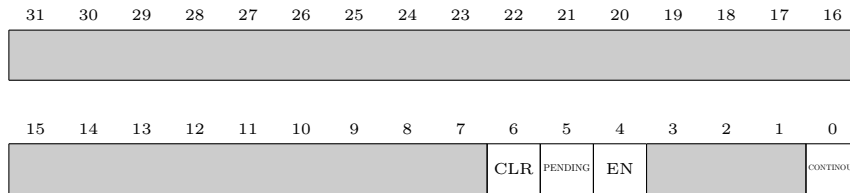
Bit 19 - 0 **RX\_SIZE** (*R/W*) RX buffer size bitfield in bytes. (128kBytes maximum)

- Read: returns remaining buffer size to transfer.
- Write: sets buffer size.

## 4.8.27 RX\_CFG

**Address:** 0x1A102108

**Reset Value:** 0x00000000



**Bit 6 CLR** (*W*) RX channel clear and stop transfer:

-1'b0: disable

-1'b1: stop and clear the on-going transfer

**Bit 5 PENDING** (*R*) RX transfer pending in queue status flag:

-1'b0: no pending transfer in the queue

-1'b1: pending transfer in the queue

**Bit 4 EN** (*R/W*) RX channel enable and start transfer bitfield:

-1'b0: disable

-1'b1: enable and start the transfer

This signal is used also to queue a transfer if one is already ongoing.

**Bit 0 CONTINUOUS** (*R/W*) RX channel continuous mode bitfield:

-1'b0: disabled

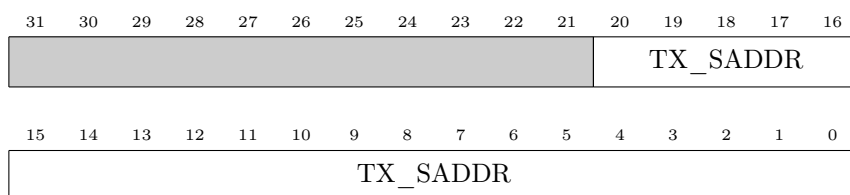
-1'b1: enabled

At the end of the buffer transfer, the uDMA reloads the address / buffer size and starts a new transfer.

## 4.8.28 TX\_SADDR

**Address:** 0x1A102110

**Reset Value:** 0x00000000



**Bit 20 - 0 TX\_SADDR** (*R/W*) TX buffer base address bitfield:

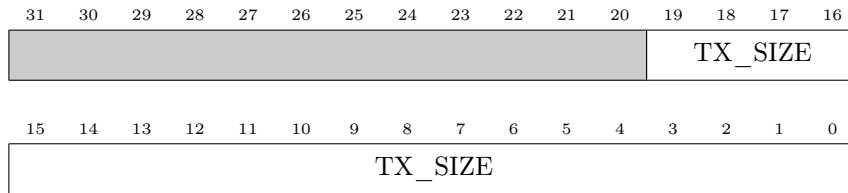
- Read: returns value of the buffer pointer until transfer is finished. Else returns 0.

- Write: sets buffer base address

### 4.8.29 TX\_SIZE

Address: 0x1A102114

Reset Value: 0x00000000

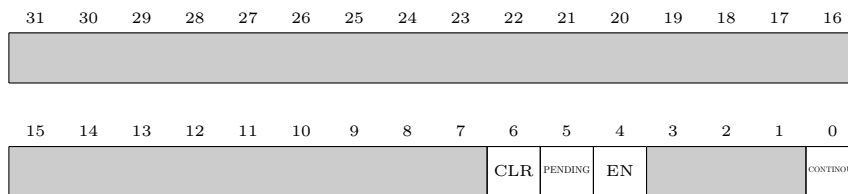


Bit 19 - 0 **TX\_SIZE** (*R/W*) TX buffer size bitfield in bytes. (128kBytes maximum)  
- Read: returns remaining buffer size to transfer.  
- Write: sets buffer size.

### 4.8.30 TX\_CFG

Address: 0x1A102118

Reset Value: 0x00000000



Bit 6 **CLR** (*W*) TX channel clear and stop transfer bitfield:

- 1'b0: disabled
- 1'b1: stop and clear the on-going transfer

Bit 5 **PENDING** (*R*) TX transfer pending in queue status flag:

- 1'b0: no pending transfer in the queue
- 1'b1: pending transfer in the queue

Bit 4 **EN** (*R/W*) TX channel enable and start transfer bitfield:

- 1'b0: disabled
  - 1'b1: enable and start the transfer
- This signal is used also to queue a transfer if one is already ongoing.

Bit 0 **CONTINUOUS** (*R/W*) TX channel continuous mode bitfield:

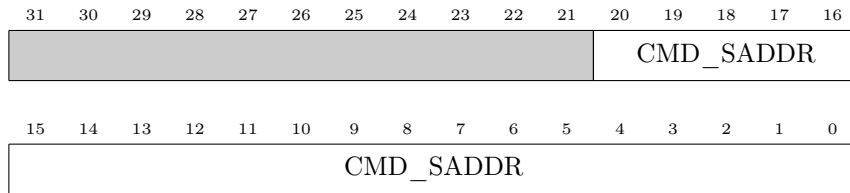
- 1'b0: disabled
- 1'b1: enabled

At the end of the buffer transfer, the uDMA reloads the address / buffer size and starts a new transfer.

### 4.8.31 CMD\_SADDR

Address: 0x1A102120

Reset Value: 0x00000000



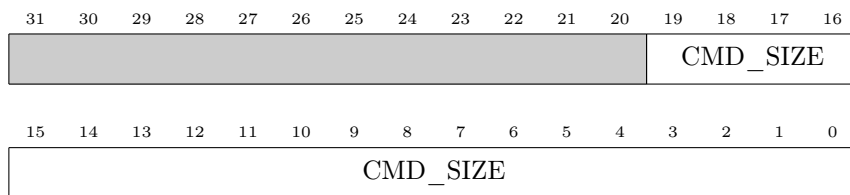
Bit 20 - 0 **CMD\_SADDR** (*R/W*) CMD buffer base address bitfield:

- Read: returns value of the buffer pointer until transfer is finished. Else returns 0.
- Write: sets buffer base address

### 4.8.32 CMD\_SIZE

Address: 0x1A102124

Reset Value: 0x00000000



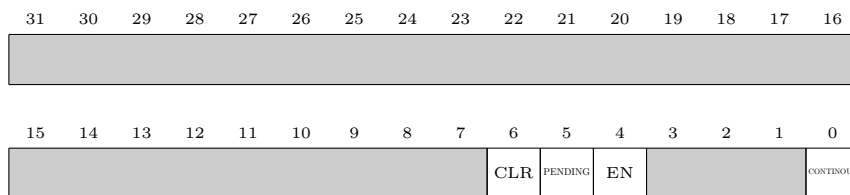
Bit 19 - 0 **CMD\_SIZE** (*R/W*) CMD buffer size bitfield in bytes. (128kBytes maximum)

- Read: returns remaining buffer size to transfer.
- Write: sets buffer size.

### 4.8.33 CMD\_CFG

Address: 0x1A102128

Reset Value: 0x00000000



Bit 6 **CLR** (*W*) CMD channel clear and stop transfer bitfield:

- 1'b0: disabled
- 1'b1: stop and clear the on-going transfer

Bit 5 **PENDING** (*R*) CMD transfer pending in queue status flag:

- 1'b0: no pending transfer in the queue
- 1'b1: pending transfer in the queue



Bit 4 **EN** (*R/W*) CMD channel enable and start transfer bitfield:

-1'b0: disabled

-1'b1: enable and start the transfer

This signal is used also to queue a transfer if one is already ongoing.

Bit 0 **CONTINOUS** (*R/W*) CMD channel continuous mode bitfield:

-1'b0: disabled

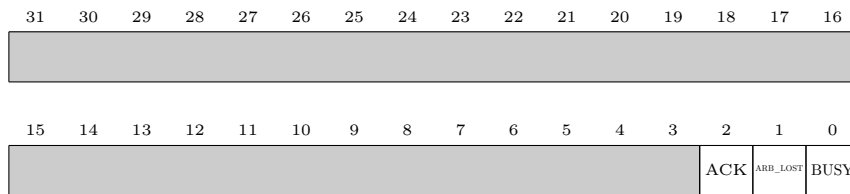
-1'b1: enabled

At the end of the buffer transfer, the uDMA reloads the address / buffer size and starts a new transfer.

#### 4.8.34 STATUS

Address: 0x1A102130

Reset Value: 0x00000000



Bit 2 **ACK** (*R*) I2C ack flag, can be polling for busy:

- 1'b0: ACK

- 1'b1: NAK

Bit 1 **ARB\_LOST** (*R/W*) I2C arbitration lost status flag:

- 1'b0: no error

- 1'b1: arbitration lost error

Bit 0 **BUSY** (*R/W*) I2C bus busy status flag:

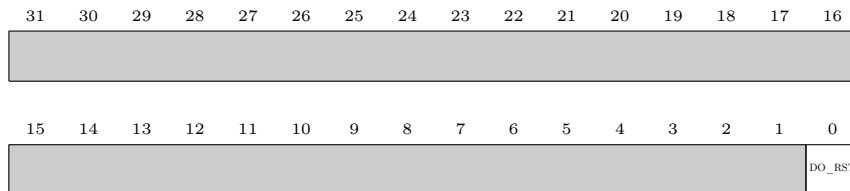
- 1'b0: no transfer on-going

- 1'b1: transfer on-going

#### 4.8.35 SETUP

Address: 0x1A102134

Reset Value: 0x00000000



Bit 0 **DO\_RST** (*R/W*) Reset command used to abort the on-going transfer and clear busy and arbitration lost status flags.

#### 4.8.36 uDMA I2C1 Registers

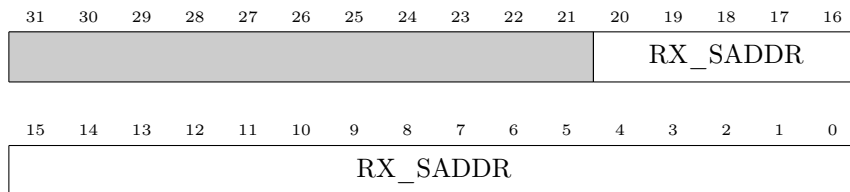
Name	Address	Size	Type	Access	Default	Description
RX_SADDR	0x1A102180	32	Config	R/W	0x00000000	uDMA RX I2C buffer base address configuration register.
RX_SIZE	0x1A102184	32	Config	R/W	0x00000000	uDMA RX I2C buffer size configuration register.
RX_CFG	0x1A102188	32	Config	R/W	0x00000000	uDMA RX I2C stream configuration register.
TX_SADDR	0x1A102190	32	Config	R/W	0x00000000	uDMA TX I2C buffer base address configuration register.
TX_SIZE	0x1A102194	32	Config	R/W	0x00000000	uDMA TX I2C buffer size configuration register.
TX_CFG	0x1A102198	32	Config	R/W	0x00000000	uDMA TX I2C stream configuration register.
CMD_SADDR	0x1A1021A0	32	Config	R/W	0x00000000	uDMA CMD I2C buffer base address configuration register.
CMD_SIZE	0x1A1021A4	32	Config	R/W	0x00000000	uDMA CMD I2C buffer size configuration register.
CMD_CFG	0x1A1021A8	32	Config	R/W	0x00000000	uDMA CMD I2C stream configuration register.
STATUS	0x1A1021B0	32	Status	R/W	0x00000000	uDMA I2C Status register.
SETUP	0x1A1021B4	32	Config	R/W	0x00000000	uDMA I2C Configuration register.

Table 4.12: uDMA I2C1

#### 4.8.37 RX\_SADDR

**Address:** 0x1A102180

**Reset Value:** 0x00000000



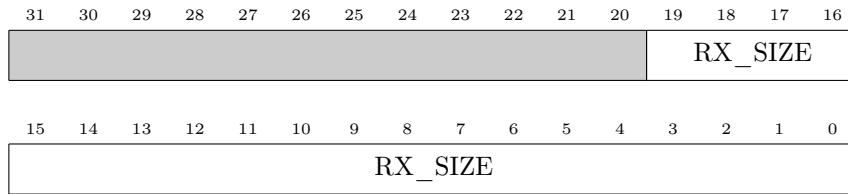
Bit 20 - 0 **RX\_SADDR** (*R/W*) RX buffer base address bitfield:

- Read: returns value of the buffer pointer until transfer is finished. Else returns 0.
- Write: sets RX buffer base address

#### 4.8.38 RX\_SIZE

**Address:** 0x1A102184

**Reset Value:** 0x00000000

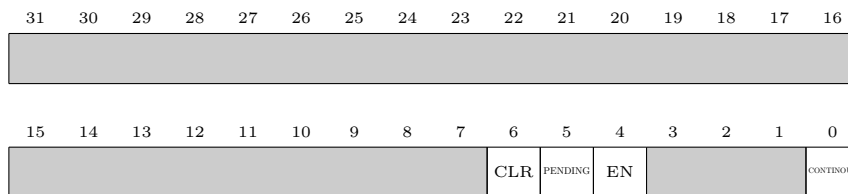


Bit 19 - 0 **RX\_SIZE** (*R/W*) RX buffer size bitfield in bytes. (128kBytes maximum)  
 - Read: returns remaining buffer size to transfer.  
 - Write: sets buffer size.

#### 4.8.39 RX\_CFG

**Address:** 0x1A102188

**Reset Value:** 0x00000000



Bit 6 **CLR** (*W*) RX channel clear and stop transfer:

- 1'b0: disable
- 1'b1: stop and clear the on-going transfer

Bit 5 **PENDING** (*R*) RX transfer pending in queue status flag:

- 1'b0: no pending transfer in the queue
- 1'b1: pending transfer in the queue

Bit 4 **EN** (*R/W*) RX channel enable and start transfer bitfield:

- 1'b0: disable
  - 1'b1: enable and start the transfer
- This signal is used also to queue a transfer if one is already ongoing.

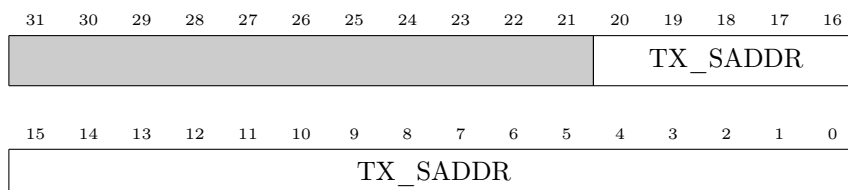
Bit 0 **CONTINUOUS** (*R/W*) RX channel continuous mode bitfield:

- 1'b0: disabled
  - 1'b1: enabled
- At the end of the buffer transfer, the uDMA reloads the address / buffer size and starts a new transfer.

#### 4.8.40 TX\_SADDR

**Address:** 0x1A102190

**Reset Value:** 0x00000000

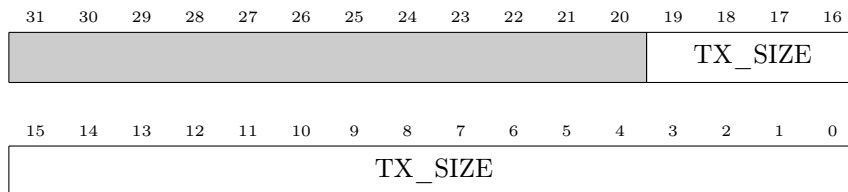


Bit 20 - 0 **TX\_SADDR** (*R/W*) TX buffer base address bitfield:  
 - Read: returns value of the buffer pointer until transfer is finished. Else returns 0.  
 - Write: sets buffer base address

#### 4.8.41 TX\_SIZE

Address: 0x1A102194

Reset Value: 0x00000000

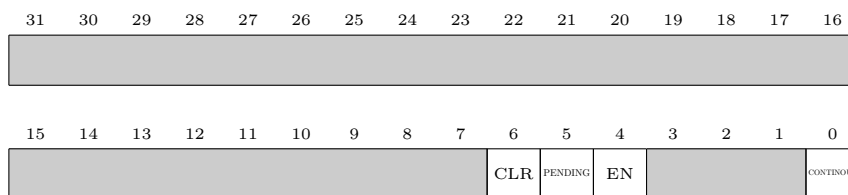


Bit 19 - 0 **TX\_SIZE** (*R/W*) TX buffer size bitfield in bytes. (128kBytes maximum)  
 - Read: returns remaining buffer size to transfer.  
 - Write: sets buffer size.

#### 4.8.42 TX\_CFG

Address: 0x1A102198

Reset Value: 0x00000000



Bit 6 **CLR** (*W*) TX channel clear and stop transfer bitfield:

- 1'b0: disabled
- 1'b1: stop and clear the on-going transfer

Bit 5 **PENDING** (*R*) TX transfer pending in queue status flag:

- 1'b0: no pending transfer in the queue
- 1'b1: pending transfer in the queue

Bit 4 **EN** (*R/W*) TX channel enable and start transfer bitfield:

- 1'b0: disabled
  - 1'b1: enable and start the transfer
- This signal is used also to queue a transfer if one is already ongoing.

Bit 0 **CONTINUOUS** (*R/W*) TX channel continuous mode bitfield:

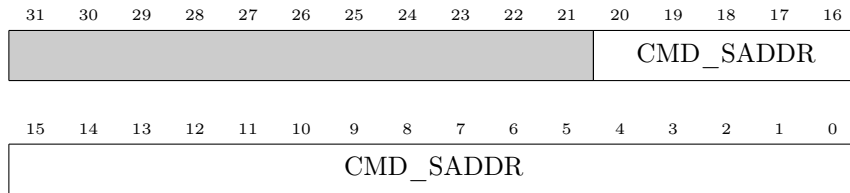
- 1'b0: disabled
- 1'b1: enabled

At the end of the buffer transfer, the uDMA reloads the address / buffer size and starts a new transfer.

#### 4.8.43 CMD\_SADDR

Address: 0x1A1021A0

Reset Value: 0x00000000



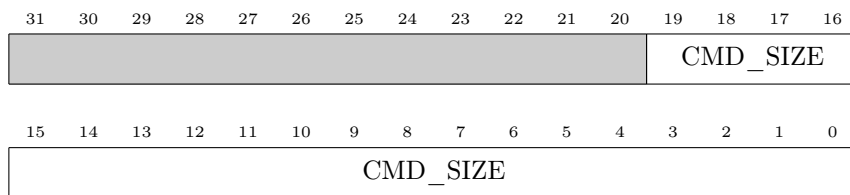
Bit 20 - 0 **CMD\_SADDR** (*R/W*) CMD buffer base address bitfield:

- Read: returns value of the buffer pointer until transfer is finished. Else returns 0.
- Write: sets buffer base address

#### 4.8.44 CMD\_SIZE

Address: 0x1A1021A4

Reset Value: 0x00000000



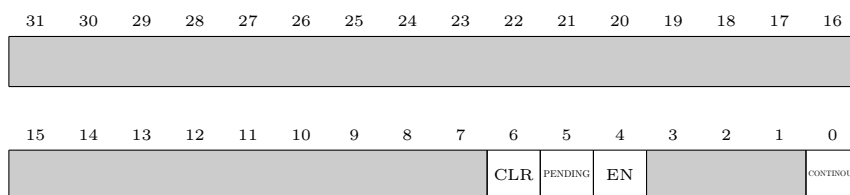
Bit 19 - 0 **CMD\_SIZE** (*R/W*) CMD buffer size bitfield in bytes. (128kBytes maximum)

- Read: returns remaining buffer size to transfer.
- Write: sets buffer size.

#### 4.8.45 CMD\_CFG

Address: 0x1A1021A8

Reset Value: 0x00000000



Bit 6 **CLR** (*W*) CMD channel clear and stop transfer bitfield:

- 1'b0: disabled
- 1'b1: stop and clear the on-going transfer

Bit 5 **PENDING** (*R*) CMD transfer pending in queue status flag:

- 1'b0: no pending transfer in the queue
- 1'b1: pending transfer in the queue

Bit 4 **EN** (*R/W*) CMD channel enable and start transfer bitfield:

-1'b0: disabled

-1'b1: enable and start the transfer

This signal is used also to queue a transfer if one is already ongoing.

Bit 0 **CONTINOUS** (*R/W*) CMD channel continuous mode bitfield:

-1'b0: disabled

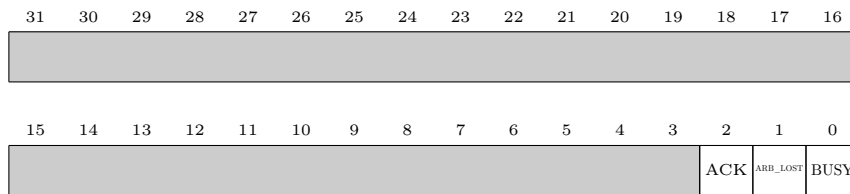
-1'b1: enabled

At the end of the buffer transfer, the uDMA reloads the address / buffer size and starts a new transfer.

#### 4.8.46 STATUS

Address: 0x1A1021B0

Reset Value: 0x00000000



Bit 2 **ACK** (*R*) I2C ack flag, can be polling for busy:

- 1'b0: ACK

- 1'b1: NAK

Bit 1 **ARB\_LOST** (*R/W*) I2C arbitration lost status flag:

- 1'b0: no error

- 1'b1: arbitration lost error

Bit 0 **BUSY** (*R/W*) I2C bus busy status flag:

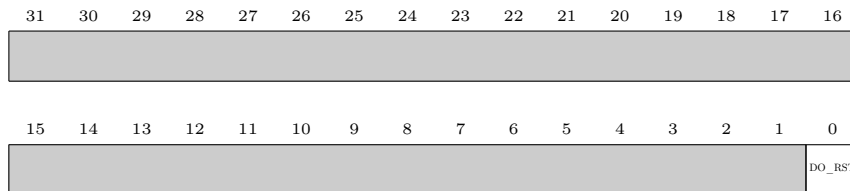
- 1'b0: no transfer on-going

- 1'b1: transfer on-going

#### 4.8.47 SETUP

Address: 0x1A1021B4

Reset Value: 0x00000000



Bit 0 **DO\_RST** (*R/W*) Reset command used to abort the on-going transfer and clear busy and arbitration lost status flags.

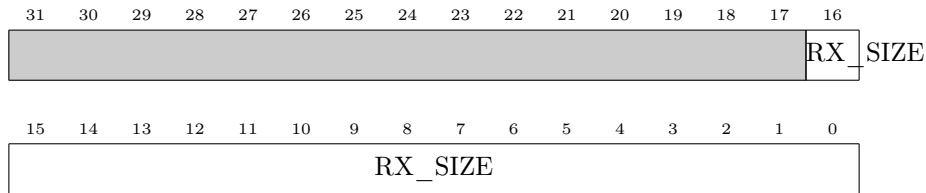
#### 4.8.48 uDMA I2S Registers



#### 4.8.50 I2S\_RX\_SIZE

Address: 0x1A102284

Reset Value: 0x00000000



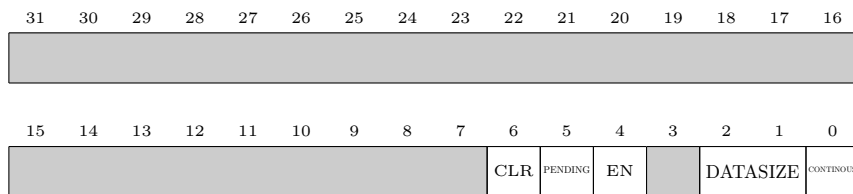
Bit 16 - 0 **RX\_SIZE** (*R/W*) Buffer size in byte. (128kBytes maximum)

- Read: buffer size left
- Write: set buffer size

#### 4.8.51 I2S\_RX\_CFG

Address: 0x1A102288

Reset Value: 0x00000004



Bit 6 **CLR** (*W*) Channel clear and stop transfer:

- 1'b0: disable
- 1'b1: enable

Bit 5 **PENDING** (*R*) Transfer pending in queue status flag:

- 1'b0: free
- 1'b1: pending

Bit 4 **EN** (*R/W*) Channel enable and start transfer:

- 1'b0: disable
- 1'b1: enable

This signal is used also to queue a transfer if one is already ongoing.

Bit 2 - 1 **DATASIZE** (*R/W*) Channel transfer size used to increment uDMA buffer address pointer:

- 2'b00: +1 (8 bits)
- 2'b01: +2 (16 bits)
- 2'b10: +4 (32 bits)
- 2'b11: +0

Bit 0 **CONTINUOUS** (*R/W*) Channel continuous mode:

- 1'b0: disable
- 1'b1: enable

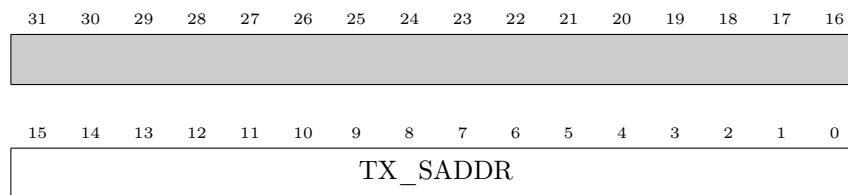
At the end of the buffer the uDMA reloads the address and size and starts a new transfer.



#### 4.8.52 I2S\_TX\_SADDR

**Address:** 0x1A102290

Reset Value: 0x00000000



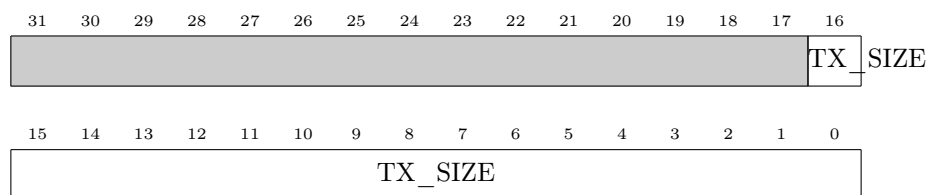
Bit 15 - 0 **TX\_SADDR** (*R/W*) Configure pointer to memory buffer:

- Read: value of the pointer until transfer is over. Else returns 0
- Write: set Address Pointer to memory buffer start address

#### 4.8.53 I2S\_TX\_SIZE

**Address:** 0x1A102294

Reset Value: 0x00000000



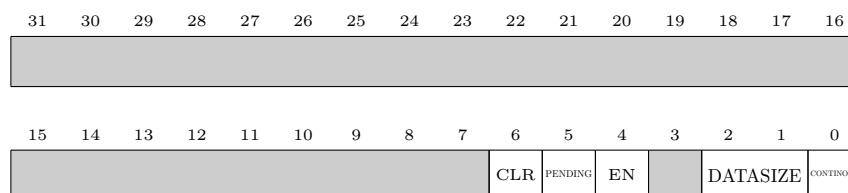
Bit 16 - 0 **TX SIZE** (*R/W*) Buffer size in byte. (128kBytes maximum)

- Read: buffer size left
- Write: set buffer size

#### 4.8.54 I2S TX CFG

**Address:** 0x1A102298

**Reset Value:** 0x00000004



Bit 6 **CLR** (*R/W*) Channel clear and stop transfer:

- 1'b0: disable  
-1'b1: enable

Bit 5 **PENDING** (*R*) Transfer pending in queue status flag:

- ```
-1'b0: free
-1'b1: pending
```

Bit 4 **EN** (*R/W*) Channel enable and start transfer:

-1'b0: disable

-1'b1: enable

This signal is used also to queue a transfer if one is already ongoing.

Bit 2 - 1 **DATASIZE** (*R/W*) Channel transfer size used to increment uDMA buffer address pointer:

- 2'b00: +1 (8 bits)

- 2'b01: +2 (16 bits)

- 2'b10: +4 (32 bits)

- 2'b11: +0

Bit 0 **CONTINUOUS** (*R/W*) Channel continuous mode:

-1'b0: disable

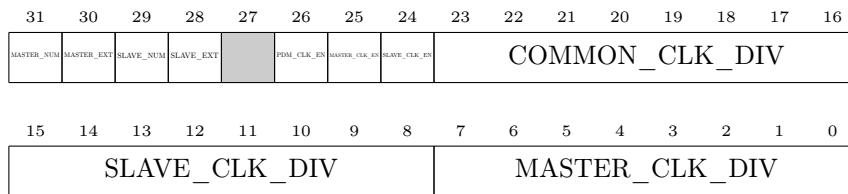
-1'b1: enable

At the end of the buffer the uDMA reloads the address and size and starts a new transfer.

#### 4.8.55 I2S\_CLKCFG\_SETUP

Address: 0x1A1022A0

Reset Value: 0x00000000



Bit 31 **MASTER\_NUM** (*R/W*) Selects master clock source(either ext or generated):

-1'b0:selects master

-1'b1:selects slave

Bit 30 **MASTER\_EXT** (*R/W*) When set uses external clock for master

Bit 29 **SLAVE\_NUM** (*R/W*) Selects slave clock source(either ext or generated):

-1'b0:selects master

-1'b1:selects slave

Bit 28 **SLAVE\_EXT** (*R/W*) When set uses external clock for slave

Bit 26 **PDM\_CLK\_EN** (*R/W*) When enabled slave output clock is taken from PDM module

Bit 25 **MASTER\_CLK\_EN** (*R/W*) Enables Master clock

Bit 24 **SLAVE\_CLK\_EN** (*R/W*) Enables Slave clock

Bit 23 - 16 **COMMON\_CLK\_DIV** (*R/W*) MSBs of both master and slave clock divider

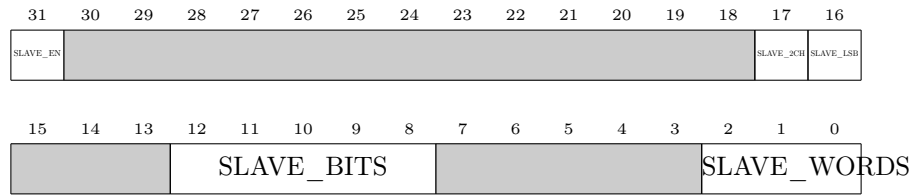
Bit 15 - 8 **SLAVE\_CLK\_DIV** (*R/W*) LSB of slave clock divider

Bit 7 - 0 **MASTER\_CLK\_DIV** (*R/W*) LSB of master clock divider

#### 4.8.56 I2S\_SLV\_SETUP

Address: 0x1A1022A4

Reset Value: 0x00000000



Bit 31 **SLAVE\_EN** (*R/W*) Enables the Slave

Bit 17 **SLAVE\_2CH** (*R/W*) Enables both channels

Bit 16 **SLAVE\_LSB** (*R/W*) Enables LSB shifting

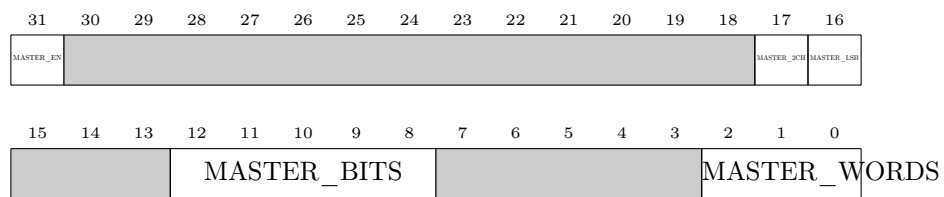
Bit 12 - 8 **SLAVE\_BITS** (*R/W*) Sets how many bits per word

Bit 2 - 0 **SLAVE\_WORDS** (*R/W*) Sets how many words for each I2S phase

#### 4.8.57 I2S\_MST\_SETUP

Address: 0x1A1022A8

Reset Value: 0x00000000



Bit 31 **MASTER\_EN** (*R/W*) Enables the Master

Bit 17 **MASTER\_2CH** (*R/W*) Enables both channels

Bit 16 **MASTER\_LSB** (*R/W*) Enables LSB shifting

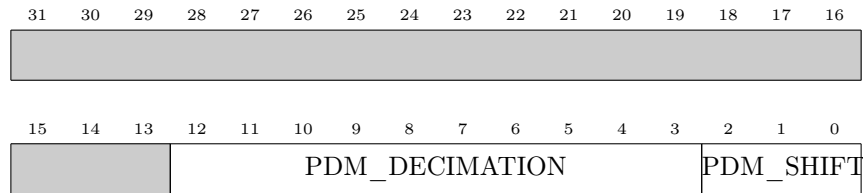
Bit 12 - 8 **MASTER\_BITS** (*R/W*) Sets how many bits per word

Bit 2 - 0 **MASTER\_WORDS** (*R/W*) Sets how many words for each I2S phase

#### 4.8.58 I2S\_PDM\_SETUP

Address: 0x1A1022AC

Reset Value: 0x00000000



Bit 12 - 3 **PDM\_DECIMATION** (*R/W*) Sets the decimation ratio of the filter

Bit 2 - 0 **PDM\_SHIFT** (*R/W*) Shifts the output of the filter

#### 4.8.59 uDMA Camera Interface Registers

| Name         | Address    | Size | Type   | Access | Default    | Description                                                   |
|--------------|------------|------|--------|--------|------------|---------------------------------------------------------------|
| CAM_RX_SADDR | 0x1A102300 | 32   | Config | R/W    | 0x00000000 | RX Camera uDMA transfer address of associated buffer register |
| CAM_RX_SIZE  | 0x1A102304 | 32   | Config | R/W    | 0x00000000 | RX Camera uDMA transfer size of buffer register               |
| CAM_RX_CFG   | 0x1A102308 | 32   | Config | R/W    | 0x00000000 | RX Camera uDMA transfer configuration register                |
| CAM_CFG_GLOB | 0x1A102320 | 32   | Config | R/W    | 0x00000000 | Global configuration register                                 |
| CAM_CFG_LL   | 0x1A102324 | 32   | Config | R/W    | 0x00000000 | Lower Left corner configuration register                      |
| CAM_CFG_UR   | 0x1A102328 | 32   | Config | R/W    | 0x00000000 | Upper Right corner configuration register                     |

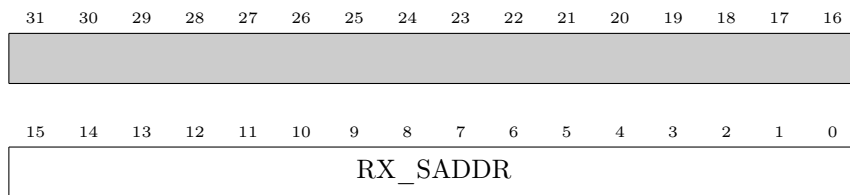
|                    |            |    |        |     |            |                                              |
|--------------------|------------|----|--------|-----|------------|----------------------------------------------|
| CAM_CFG_SIZE       | 0x1A10232C | 32 | Config | R/W | 0x00000000 | Horizontal Resolution configuration register |
| CAM_CFG_FILTER     | 0x1A102330 | 32 | Config | R/W | 0x00000000 | RGB coefficients configuration register      |
| CAM_VSYNC_POLARITY | 0x1A102334 | 32 | Config | R/W | 0x00000000 | VSYNC Polarity register                      |

Table 4.14: uDMA Camera Interface

#### 4.8.60 CAM\_RX\_SADDR

Address: 0x1A102300

Reset Value: 0x00000000



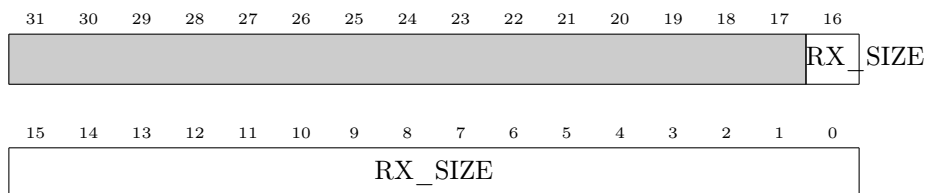
Bit 15 - 0 **RX\_SADDR** (*R/W*) Configure pointer to memory buffer:

- Read: value of the pointer until transfer is over. Else returns 0
- Write: set Address Pointer to memory buffer start address

#### 4.8.61 CAM\_RX\_SIZE

Address: 0x1A102304

Reset Value: 0x00000000



Bit 16 - 0 **RX\_SIZE** (*R/W*) Buffer size in bytes. (128kBytes maximum)

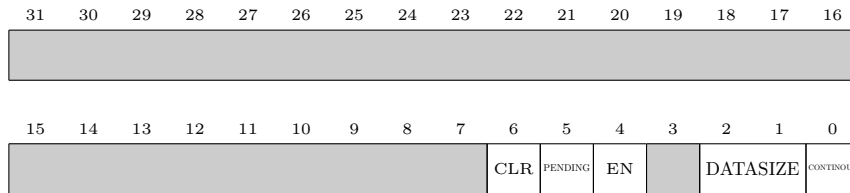
- Read: buffer size left
- Write: set buffer size

NOTE: Careful with size in byte. If you use uncompressed pixel data mapped on 16 bits, you have to declare buffer size in bytes even if buffer type is short.

## 4.8.62 CAM\_RX\_CFG

Address: 0x1A102308

Reset Value: 0x00000000



Bit 6 **CLR** (*W*) Channel clear and stop transfer:

-1'b0: disable

-1'b1: enable

Bit 5 **PENDING** (*R*) Transfer pending in queue status flag:

-1'b0: free

-1'b1: pending

Bit 4 **EN** (*R/W*) Channel enable and start transfer:

-1'b0: disable

-1'b1: enable

This signal is used also to queue a transfer if one is already ongoing.

Bit 2 - 1 **DATASIZE** (*R/W*) Channel transfer size used to increment uDMA buffer address pointer:

- 2'b00: +1 (8 bits)

- 2'b01: +2 (16 bits)

- 2'b10: +4 (32 bits)

- 2'b11: +0

Bit 0 **CONTINUOUS** (*R/W*) Channel continuous mode:

-1'b0: disable

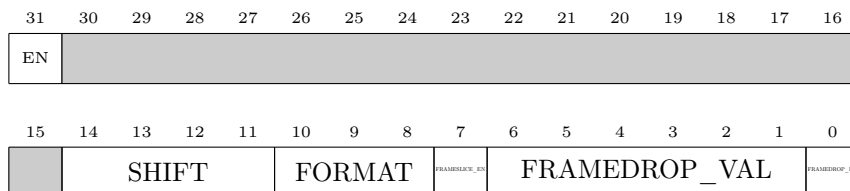
-1'b1: enable

At the end of the buffer the uDMA reloads the address and size and starts a new transfer.

## 4.8.63 CAM\_CFG\_GLOB

Address: 0x1A102320

Reset Value: 0x00000000



Bit 31 **EN** (*R/W*) Enable data rx from camera interface.

The enable/disable happens only at the start of a frame.

- 1'b0: disable

- 1'b1: enable

Bit 14 - 11 **SHIFT** (*R/W*) Right shift of final pixel value (DivFactor)

NOTE: not used if FORMAT == BYPASS

Bit 10 - 8 **FORMAT** (*R/W*) Input frame format:

- 3'b000: RGB565
- 3'b001: RGB555
- 3'b010: RGB444
- 3'b100: BYPASS\_LITEND
- 3'b101: BYPASS\_BIGEND

Bit 7 **FRAMESLICE\_EN** (*R/W*) Input frame slicing:

- 1'b0: disable
- 1'b1: enable

Bit 6 - 1 **FRAMEDROP\_VAL** (*R/W*) Sets how many frames should be dropped after each received.

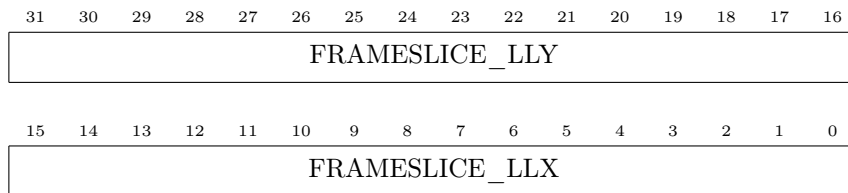
Bit 0 **FRAMEDROP\_EN** (*R/W*) Frame dropping:

- 1'b0: disable
- 1'b1: enable

#### 4.8.64 CAM\_CFG\_LL

Address: 0x1A102324

Reset Value: 0x00000000



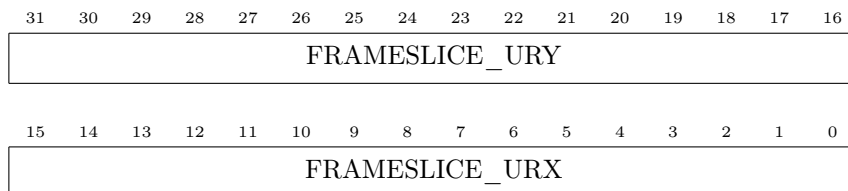
Bit 31 - 16 **FRAMESLICE\_LLY** (*R/W*) Y coordinate of lower left corner of slice

Bit 15 - 0 **FRAMESLICE\_LLX** (*R/W*) X coordinate of lower left corner of slice

#### 4.8.65 CAM\_CFG\_UR

Address: 0x1A102328

Reset Value: 0x00000000



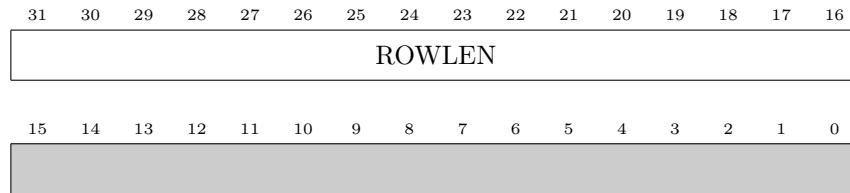
Bit 31 - 16 **FRAMESLICE\_URY** (*R/W*) Y coordinate of upper right corner of slice

Bit 15 - 0 **FRAMESLICE\_URX** (*R/W*) X coordinate of upper right corner of slice

#### 4.8.66 CAM\_CFG\_SIZE

Address: 0x1A10232C

Reset Value: 0x00000000

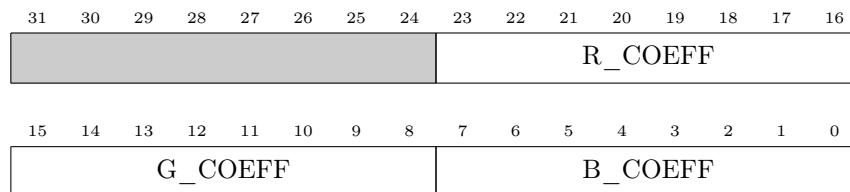


Bit 31 - 16 **ROWLEN** (*R/W*) Horizontal Resolution. It is used for slice mode. Value set into the bitfield must be equal to (rowlen-1).

#### 4.8.67 CAM\_CFG\_FILTER

Address: 0x1A102330

Reset Value: 0x00000000



Bit 23 - 16 **R\_COEFF** (*R/W*) Coefficient that multiplies the R component  
NOTE: not used if FORMAT == BYPASS

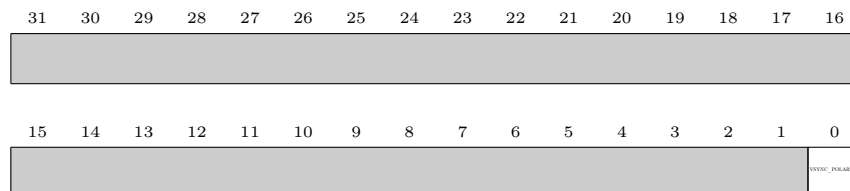
Bit 15 - 8 **G\_COEFF** (*R/W*) Coefficient that multiplies the G component  
NOTE: not used if FORMAT == BYPASS

Bit 7 - 0 **B\_COEFF** (*R/W*) Coefficient that multiplies the B component  
NOTE: not used if FORMAT == BYPASS

#### 4.8.68 CAM\_VSYNC\_POLARITY

Address: 0x1A102334

Reset Value: 0x00000000



Bit 0 **VSYNC\_POLARITY** (*R/W*) Set vsync polarity of CPI.  
- 1'b0: Active 0  
- 1'b1: Active 1



#### 4.8.69 uDMA Filter Registers

| Name            | Address    | Size | Type   | Access | Default    | Description                                   |
|-----------------|------------|------|--------|--------|------------|-----------------------------------------------|
| REG_TX_CH0_ADD  | 0x1A102380 | 32   | Config | R/W    | 0x00000000 | FILTER tx channel 0 address register          |
| REG_TX_CH0_CFG  | 0x1A102384 | 32   | Config | R/W    | 0x00000000 | FILTER tx channel 0 configuration register    |
| REG_TX_CH0_LEN0 | 0x1A102388 | 32   | Config | R/W    | 0x00000000 | FILTER tx channel 0 length0 register          |
| REG_TX_CH0_LEN1 | 0x1A10238C | 32   | Config | R/W    | 0x00000000 | FILTER tx channel 0 length1 register          |
| REG_TX_CH0_LEN2 | 0x1A102390 | 32   | Config | R/W    | 0x00000000 | FILTER tx channel 0 length2 register          |
| REG_TX_CH1_ADD  | 0x1A102394 | 32   | Config | R/W    | 0x00000000 | FILTER tx channel 1 address register          |
| REG_TX_CH1_CFG  | 0x1A102398 | 32   | Config | R/W    | 0x00000000 | FILTER tx channel 1 configuration register    |
| REG_TX_CH1_LEN0 | 0x1A10239C | 32   | Config | R/W    | 0x00000000 | FILTER tx channel 1 length0 register          |
| REG_TX_CH1_LEN1 | 0x1A1023A0 | 32   | Config | R/W    | 0x00000000 | FILTER tx channel 1 length1 register          |
| REG_TX_CH1_LEN2 | 0x1A1023A4 | 32   | Config | R/W    | 0x00000000 | FILTER tx channel 1 length2 register          |
| REG_RX_CH_ADD   | 0x1A1023A8 | 32   | Config | R/W    | 0x00000000 | FILTER RX channel address register            |
| REG_RX_CH_CFG   | 0x1A1023AC | 32   | Config | R/W    | 0x00000000 | FILTER RX channel configuration register      |
| REG_RX_CH_LEN0  | 0x1A1023B0 | 32   | Config | R/W    | 0x00000000 | FILTER RX channel length0 register            |
| REG_RX_CH_LEN1  | 0x1A1023B4 | 32   | Config | R/W    | 0x00000000 | FILTER RX channel length1 register            |
| REG_RX_CH_LEN2  | 0x1A1023B8 | 32   | Config | R/W    | 0x00000000 | FILTER RX channel length2 register            |
| REG_AU_CFG      | 0x1A1023BC | 32   | Config | R/W    | 0x00000000 | FILTER arithmetic unit configuration register |

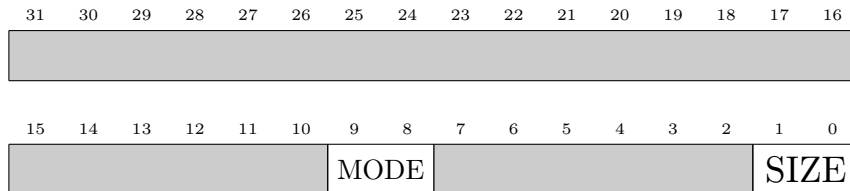
|                  |            |    |        |     |            |                                              |
|------------------|------------|----|--------|-----|------------|----------------------------------------------|
| REG_AU_REG0      | 0x1A1023C0 | 32 | Config | R/W | 0x00000000 | FILTER arithmetic unit 0 register            |
| REG_AU_REG1      | 0x1A1023C4 | 32 | Config | R/W | 0x00000000 | FILTER arithmetic unit 1 register            |
| REG_BINCUI_TH    | 0x1A1023C8 | 32 | Config | R/W | 0x00000000 | FILTER binarization threshold register       |
| REG_BINCUI_CNT   | 0x1A1023CC | 32 | Config | R/W | 0x00000000 | FILTER binarization count register           |
| REG_BINCUI_SETUP | 0x1A1023D0 | 32 | Config | R/W | 0x00000000 | FILTER binarization datasize format register |
| REG_BINCUI_VAL   | 0x1A1023D4 | 32 | Status | R   | 0x00000000 | FILTER binarization result count register    |
| REG_FILTER       | 0x1A1023D8 | 32 | Config | R/W | 0x00000000 | FILTER control mode register                 |
| REG_FILTER_CMD   | 0x1A1023DC | 32 | Config | R/W | 0x00000000 | FILTER start register                        |
| REG_STATUS       | 0x1A1023E0 | 32 | Status | R/W | 0x00000000 | FILTER status register                       |

Table 4.15: uDMA Filter

#### 4.8.70 REG\_TX\_CH0\_CFG

**Address:** 0x1A102384

**Reset Value:** 0x00000000



Bit 9 - 8 **MODE** (*R*) Data transfer mode:

- 2'b00: Linear
- 2'b01: Sliding
- 2;b10: Circular
- 2;b11: 2D

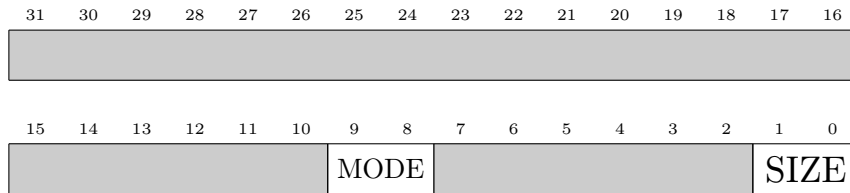
Bit 1 - 0 **SIZE** (*R*) Data transfer format:

- 2'b00: 8-bit
- 2'b01: 16-bit
- 2;b10: 32-bit

#### 4.8.71 REG\_TX\_CH1\_CFG

Address: 0x1A102398

Reset Value: 0x00000000



Bit 9 - 8 **MODE** (*R/W*) Data transfer mode:

- 2'b00: Linear
- 2'b01: Sliding
- 2;b10: Circular
- 2;b11: 2D

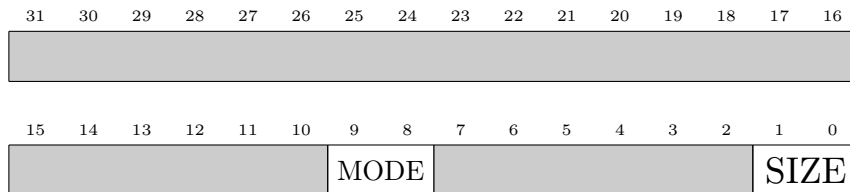
Bit 1 - 0 **SIZE** (*R*) Data transfer format:

- 2'b00: 8-bit
- 2'b01: 16-bit
- 2'b10: 32-bit

#### 4.8.72 REG\_RX\_CH\_CFG

Address: 0x1A1023AC

Reset Value: 0x00000000



Bit 9 - 8 **MODE** (*R/W*) Data transfer mode:

- 2'b00: Linear
- 2'b01: Sliding
- 2;b10: Circular
- 2;b11: 2D

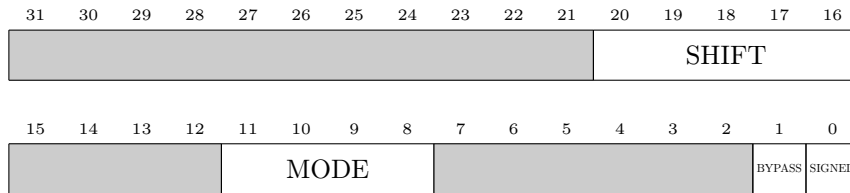
Bit 1 - 0 **SIZE** (*R/W*) Data transfer format:

- 2'b00: 8-bit
- 2'b01: 16-bit
- 2;b10: 32-bit

### 4.8.73 REG\_AU\_CFG

**Address:** 0x1A1023BC

**Reset Value:** 0x00000000



Bit 20 - 16 **SHIFT** (*R/W*) Arithmetic Unit shift window size, (0 - 31).

Bit 11 - 8 **MODE** (*R/W*) Arithmetic Unit mode:

- 4'b0000: AU\_MODE\_AxB
- 4'b0001: AU\_MODE\_AxB+REG0
- 4'b0010: AU\_MODE\_AxB accumulation
- 4'b0011: AU\_MODE\_AxA
- 4'b0100: AU\_MODE\_AxA+B
- 4'b0101: AU\_MODE\_AxA-B
- 4'b0110: AU\_MODE\_AxA accumulation
- 4'b0111: AU\_MODE\_AxA+REG0
- 4'b1000: AU\_MODE\_AxREG1
- 4'b1001: AU\_MODE\_AxREG1+B
- 4'b1010: AU\_MODE\_AxREG1-B
- 4'b1011: AU\_MODE\_AxREG1+REG0
- 4'b1100: AU\_MODE\_AxREG1 accumulation
- 4'b1101: AU\_MODE\_A+B
- 4'b1110: AU\_MODE\_A-B
- 4'b1111: AU\_MODE\_A+REG0

Bit 1 **BYPASS** (*R/W*) Arithmetic Unit bypass or not.

- 1'b0: not bypass AU
- 1'b1: bypass AU

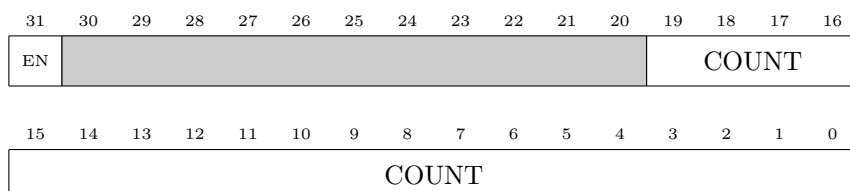
Bit 0 **SIGNED** (*R/W*) Arithmetic Unit result signed or not.

- 1'b0: not signed
- 1'b1: signed

### 4.8.74 REG\_BINCUN\_CNT

**Address:** 0x1A1023CC

**Reset Value:** 0x00000000



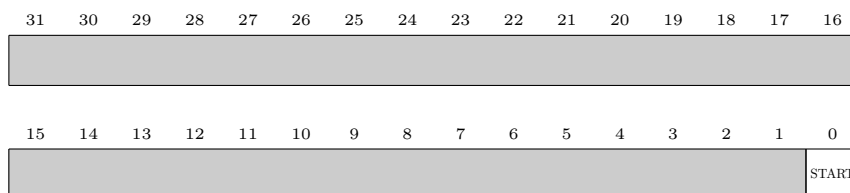
Bit 31 **EN** (*R/W*) Binarization and counting unit enable:  
 -1'b0: not enable  
 -1'b1: enable

Bit 19 - 0 **COUNT** (*R/W*) Binarization and counting unit count value set.

#### 4.8.75 REG\_FILT\_CMD

**Address:** 0x1A1023DC

**Reset Value:** 0x00000000

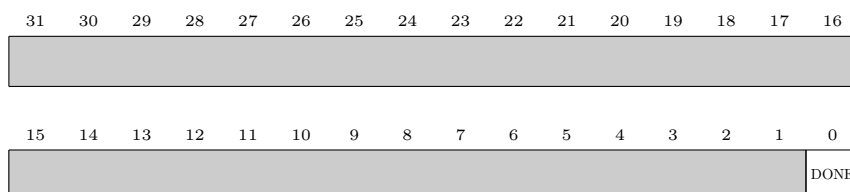


Bit 0 **START** (*R/W*) Filter start flag, write only, write 1 to start the filter :

#### 4.8.76 REG\_STATUS

**Address:** 0x1A1023E0

**Reset Value:** 0x00000000



Bit 0 **DONE** (*R/W*) Filter done flag, write 1 to clear the flag :  
 -1'b0: Filter process is not finished  
 -1'b1: Filter process is finished

## 5 Debug Module for External Debug Support

The debug module in PULPissimo is compliant with the RISC-V External Debug Support specification *v1.13.1*. For more details please take a look at the documentation in the debug module folder and consult the RISC-V External Debug Support specification.