CORE_V_MCU

Purpose

The purpose of the core-v-mcu is to showcase the cv32e40p fully verified RISC-V core available from the Open Hardware Group. The cv32e40p core is connected to a representative set of peripherals:

- 2xUART
- 2xI2C master
- 1xI2C slave
- 2xQSPI master
- 1xCAMERA
- 1xSDIO
- 4xPWM
- eFPGA with 4 math units

Note: A set of registers in soc_ctrl defines which peripherals and how many were incorporated in the build. The soc_ctrl documenation reports the configuration when the documentation was generated, however that may not be in sync with the configuration when the RTL was built.

The system supports 512KB of SRAM and 3 PLLs.

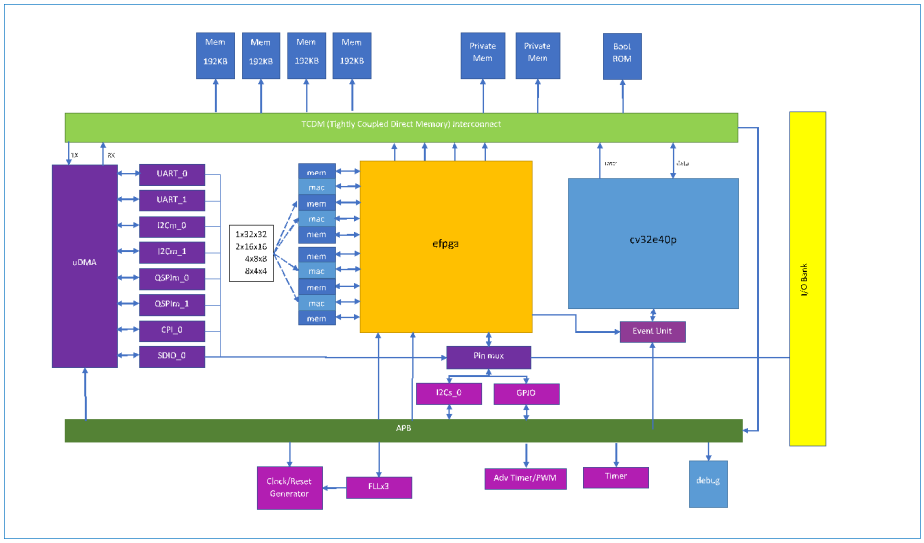

Figure 1: Block Diagram

Evaluation Kits

The core-v-mcu is delivered as either:

- a Xilinx bitstream that runs on the Digilent Nexsy A7 board
- a Xilinx bitstream that runs on the Digilent Genesys2 board
- an SOC implemented with GLOBAFOUNDRIES 22nm fdx SOI technology that runs on an Open Hardware Group EVK board

All boards have multiple PMOD connectors that suport various PMOD modules which are used to connect debug and various peripherals.

Software Support

The Open Hardware Group's Software Task Group supports the software ecosystem.

Software Validation System

The core-v-mcu project has a FreeRTOS/Eclipse based project that runs software on the emulation system to verify the correct operation of the system. The software validation system is in the core-v-mcu-cli-test repo. There are two components:

- a FreeRTOS/Eclipse app called cli-test that uses a Command Line Interface to exercise features
- a Python script that automates using the cli-test to validate correct system operation

Goals of the System Validation Tests

The goals of the whole product tests are to:

- Confirm that the entire address space is accessible by the core
  - All of the TCDM
  - All of the peripheral registers
  - All special registers
- Confirm that gdb can:
  - halt and resume the core
  - can single step the core
  - access entire addressable memory space
    - memories
    - peripherals
    - CSRS
  - NOTE: for performance reasons, may use gdb for sampling test and another mechanism for exhaustive testing
- Confirm that the interrupts work as expected:
  - All sources
  - Masking
- Confirm that the timers work as expected:
  - TBD
- Confirm that the clock trees & divisors work as expected:

  - Can set FLL to desired frequency
  - Clock tree enables work
  - Clock tree divisors work
- uDMA
  - Covered by peripheral tests
- Peripherals (details later in document)
  - UART
  - I2Cm
  - I2Cs
  - SPIm
  - QSPIm
  - CAMI
  - GPIO
  - PWM
- EFPGA
  - TCMD
  - APB
  - Math blocks
  - IO
  - Configuration
  - Test mode
- Confirm I/O control works as expected:
  - Peripherals can access I/O
  - GPIO can access all I/O
  - EFPGA can access all I/O
  - Mux select works properly
  - All IO configurations can be set properly
- Test modes
  - Confirm JTAG can be set in scan mode
  - Confirm JTAG can be set in mem BIST mode
  - Confirm JTAG can be set in EFPGA test mode
- Significant application
  - TfLu person detection application reading from HiMax camera and using eFPGA acceleration

Peripheral tests Peripheral tests use the specified peripherals (typically PMOD) to confirm correct system operation.

UART tests Pmod used for regression testing: PmodUSBUART

Type	Mode	Notes
Blocking		
	Set config	baud, stop, bits, parity
	Read char	
	Write char	
	Read N char	

Type	Mode	Notes
	Write N char	
	Read through '\n'	
	Rx empty	
	Tx empty	
	Wait for idle	
Non-blocking		

I2Cmaster

Pmod used for regression testing: PmodRTCC (has small SRAM)

Type	Mode	Notes
Blocking		
	Set clock frequency	
	Wait for idle	
	Read register	
	Read non-existant register	Test NAK handling
	Write register	
	Write non-existant register	Test NAK handling
	Read N of M registers, M >= N	
	Read N of M registers, M < N	Test NAK handling
	Write N of M registers, M >= N	
	Write N of M registers, M < N	Test NAK handling
Not tested		
	Clock stretching	

I2Cslave

Note: APIs are on host, so used for purposes of describing tested functionality

Type	Mode	Notes
	Read register	
	Read non-existant register	Test NAK handling
	Write register	
	Write non-existant register	Test NAK handling
	Read N of M registers, M >= N	
	Read N of M registers, M < N	Test NAK handling
	Write N of M registers, M >= N	
	Write N of M registers, M < N	Test NAK handling

QSPImaster

Pmod used for regression testing: PmodSF3 (serial NOR flash memory)

Type	Mode	Notes
Blocking		
	Set configuration	clock rate, cpol, cpha, endianness, cs#
	Wait for idle	
Non-blocking		
	Write N bytes in single bit mode	
	Write N bytes in dual bit mode	
	Write N bytes in quad bit mode	
	Read N bytes in single bit mode	
	Read N bytes in dual bit mode	
	Read N bytes in quad bit mode	
	Write/Read N bytes in single mode	

CAMI

GPIO

PWM

Memory Map ^^^^^^^^^^

This table reports the memory address map as described in `core-v-mcu/rtl/includes/soc_mem_map.svh <https://github.com/openhwgroup/core-v-mcu/blob/master/rtl/includes/soc_mem_map.svh>`__. The AXI plug is not reported as not implemented, thus it should be treated as empty memory map space.

Description	**Address Start**	**Address End**
Boot ROM	0x1A000000	0x1A03FFFF
Peripheral Domain	0x1A100000	0x1A2FFFFF
eFPGA Domain	0x1A300000	0x1A3FFFFF
Memory Bank 0	0x1C000000	0x1C007FFF
Memory Bank 1	0x1C008000	0x1C00FFFF
Memory Bank Interleaved	0x1C010000	0x1C08FFFF

The Peripheral Domain memory map is reported below as described in `core-v-mcu/rtl/includes/periph_bus_defines.svh <https://github.com/openhwgroup/core-v-mcu/bl` The Stdout emulator works only with RTL simulation and not with FPGA or ASIC implementations.

Description	Address Start	Address End
Frequency-locked loop	0x1A100000	0x1A100FFF
GPIOs	0x1A101000	0x1A101FFF
uDMA	0x1A102000	0x1A103FFF
SoC Controller	0x1A104000	0x1A104FFF
Advanced Timer	0x1A105000	0x1A105FFF
SoC Event Generator	0x1A106000	0x1A106FFF
I2CS	0x1A107000	0x1A107FFF
Timer	0x1A10B000	0x1A10BFFF
Stdout emulator	0x1A10F000	0x1A10FFFF
Debug	0x1A110000	0x1A11FFFF
eFPGA configuration	0x1A200000	0x1A2F0000
eFPGA HWCE	0x1A300000	0x1A3F0000

The memory map of the **Debug** region of the Peripheral Domain is documented as part of the PULP debug system. With the `Overview <https://github.com/pulp-platform/riscv-dbg/blob/master/doc/debug-system.md>`, *the `Debug Memory Map <https://github.com/pulp-platform/riscv-dbg/blob/master/doc/debug-system`* gives the offsets within the Debug region of the various parts of the debug module.

Pin Assignment

IO	sysio	sel=0	sel=1	sel=2	sel=3
IO_0	jtag_tck				
IO_1	jtag_tdi				
IO_2	jtag_tdo				
IO_3	jtag_tms				
IO_4	jtag_trst				
IO_5	ref_clk				
IO_6	rstn				
IO_7	eth_refclk				
IO_8	eth_rstn				
IO_9	eth_crs_dv				
IO_10	eth_rx_er				
IO_11	eth_rxd0				
IO_12	eth_rxd1				
IO_13	eth_tx_en				
IO_14	eth_txd0				
IO_15	eth_txd1				
IO_16		uart0_rx		apbio_0	fpgaio_0
IO_17		uart0_tx		apbio_1	fpgaio_1

IO	sysio	sel=0	sel=1	sel=2	sel=3
IO_18		uart1_tx		apbio_2	fpgaio_2
IO_19		uart1_rx		apbio_3	fpgaio_3
IO_20		apbio_32	apbio_47	apbio_4	fpgaio_4
IO_21		apbio_0		apbio_5	fpgaio_5
Jxadc[1]		qspim0_csn0	apbio_34	apbio_6	fpgaio_6
Jxadc[2]		qspim0_data0	apbio_35	apbio_7	fpgaio_7
Jxadc[3]		qspim0_data1	apbio_37	apbio_8	fpgaio_8
Jxadc[4]		qspim0_clk	apbio_38	apbio_9	fpgaio_9
Jxadc[7]		apbio_1	apbio_40	apbio_10	fpgaio_10
Jxadc[8]		apbio_2	apbio_41	apbio_11	fpgaio_11
Jxadc[9]		qspim0_data2	apbio_42	apbio_12	fpgaio_12
Jxadc[10]		qspim0_data3	apbio_43	apbio_13	fpgaio_13
IO_30		cam0_vsync	apbio_36	apbio_14	fpgaio_14
IO_31		cam0_hsync	apbio_39	apbio_15	fpgaio_15
IO_32		i2cm0_scl		apbio_16	fpgaio_16
IO_33		i2cm0_sda		apbio_17	fpgaio_17
IO_34		cam0_clk	apbio_33	apbio_18	fpgaio_18
IO_35		apbio_32	qspim0_csn1	apbio_19	fpgaio_19
IO_36		apbio_48	qspim0_csn2	apbio_20	fpgaio_20
IO_37		apbio_49	qspim0_csn3	apbio_21	fpgaio_21
IO_38		cam0_data0	qspim1_csn0	apbio_22	fpgaio_22
IO_39		cam0_data1	qspim1_data0	apbio_23	fpgaio_23
IO_40		cam0_data2	qspim1_data1	apbio_24	fpgaio_24
IO_41		cam0_data3	qspim1_clk	apbio_25	fpgaio_25
IO_42		cam0_data4	qspim1_csn1	apbio_26	fpgaio_26
IO_43		cam0_data5	qspim1_csn2	apbio_27	fpgaio_27
IO_44		cam0_data6	qspim1_data2	apbio_28	fpgaio_28
IO_45		cam0_data7	qspim1_data3	apbio_29	fpgaio_29
IO_46		sdio0_data3		apbio_30	fpgaio_30
IO_47		sdio0_cmd		apbio_31	fpgaio_31
IO_48		sdio0_data0		apbio_32	fpgaio_32
IO_49		sdio0_clk		apbio_43	fpgaio_33
IO_50		sdio0_data1		apbio_44	fpgaio_34
IO_51		sdio0_data2		apbio_45	fpgaio_35
IO_52		apbio_50	apbio_0	apbio_46	fpgaio_36
IO_53	stm				
IO_54	bootsel				fpgaio_37
IO_55		i2cm1_scl			fpgaio_38
IO_56		i2cm1_sda			fpgaio_39

SOC_CTRL

Memory address: SOC_CTRL_START_ADDR (0x1A104000)

This APB peripheral primarily controls I/O configuration and I/O function connection. It also supports a few registers for miscellaneous functions.

I/O control supports two functions:

- I/O configuration
- I/O function selection

I/O configuration is a series of bits that control driver characteristics, such as drive strength and slew rate. I/O function selection controls the select field of a mux that connects the I/O to different signals in the device.

INFO offset = 0x0000

Field	Bits	Type	Default	Description
N_CORES	31:16	RO		Number of cores in design
N_CLUSTERS	15:0	RO		Number of clusters in design

BUILD_DATE offset = 0x000C

Field	Bits	Type	Default	Description
YEAR	31:16	RO		Year in BCD
MONTH	15:8	RO		Month in BCD
DAY	7:0	RO		Day in BCD

BUILD_TIME offset = 0x0010

Field	Bits	Type	Default	Description
HOUR	23:16	RO		Hour in BCD
MINUTES	15:8	RO		Minutes in BCD
SECONDS	7:0	RO		Seconds in BCD

IO_CFG0 offset = 0x0014

Field	Bits	Type	Default	Description
N_GPIO	23:16	RO	32	Number of IO connected to GPIO controller
N_SYSIO	15:8	RO	3	Number of fixed-function IO
N_IO	7:0	RO	57	Number of IO on device (not necessarily on package)

IO_CFG1 offset = 0x0018

Field	Bits	Type	Default	Description
NBIT_PADMUX	15:8	RO	2	Number of bits in pad mux select, which means there are 2^NBIT_PADMUX options
NBIT_PADCFG	7:0	RO	6	Number of bits in padconfiguration

PER_CFG0 offset = 0x0020

Field	Bits	Type	Default	Description
N_I2SC	31:24	RO	0	Number of I2S clients
N_I2CM	23:16	RO	2	Number of I2C masters
N_QSPIM	15:8	RO	2	Number of QSPI masters
N_UART	7:0	RO	2	Number of UARTs

PER_CFG1 offset = 0x0024

Field	Bits	Type	Default	Description
N_CAM	15:8	RO	1	Number of Camera controllers
N_SDIO	7:0	RO	1	Number of SDIO controllers

JTAGREG offset = 0x0074

BOOTSEL offset = 0x00C4

Field	Bits	Type	Default	Description
BootDev	0:0			Selects Boot device 1=SPI, 0=Host mode via I2Cs

CLKSEL offset = 0x00C8

WD_COUNT offset = 0x00D0

Field	Bits	Type	Default	Description
COUNT	30:0	R/W	0x8000	Only writable before Watchdog is enabled

WD_CONTROL offset = 0x00D4

Field	Bits	Type	Default	Description
ENABLE_STATUS	31:31	R0		1=Watchdog Enabled, 0=Watchdog not enabled,Enabled Watch dog, once enable, cannot be disabled
WD_VALUE	15:0	WO	NA	Set to 0x6699 to reset watchdog when enabled, read current WD value

RESET_REASON offset = 0x00D8

Field	Bits	Type	Default	Description
REASON	1:0	R/W		2'b01= reset pin, 2'b11=Watchdog expired

RTO_PERIPHERAL_ERROR offset = 0x00E0

Field	Bits	Type	Default	Description
FCB_RTO	8:8	R/W	0x0	1 indicates that the FCB interface caused a ready timeout
TIMER_RTO	7:7	R/W	0x0	1 indicates that the TIMER interface caused a ready timeout
I2CS_RTO	6:6	R/W	0x0	1 indicates that the I2CS interface caused a ready timeout
EVENT_GEN_RTO	5:5	R/W	0x0	1 indicates that the EVENT GENERATOR interface caused a ready timeout
ADV_TIMER_RTO	4:4	R/W	0x0	1 indicates that the ADVANCED TIMER interface caused a ready timeout
SOC_CONTROL_RTO	3:3	R/W	0x0	1 indicates that the SOC CONTROL interface caused a ready timeout
UDMA_RTO	2:2	R/W	0x0	1 indicates that the UDMA CONTROL interface caused a ready timeout
GPIO_RTO	1:1	R/W	0x0	1 indicates that the GPIO interface caused a ready timeout

Field	Bits	Type	Default	Description
FLL_RTO	0:0	R/W	0x0	1 indicates that the FLL interface caused a ready timeout

READY_TIMEOUT_COUNT offset = 0x00E4

Field	Bits	Type	Default	Description
COUNT	19:0	R/W	0xFF	Number of APB clocks before a ready timeout occurs

RESET_TYPE1_EFPGA offset = 0x00E8

Field	Bits	Type	Default	Description
RESET_LB	3:3	R/W	0x0	Reset eFPGA Left Bottom Quadrant
RESET_RB	2:2	R/W	0x0	Reset eFPGA Right Bottom Quadrant
RESET_RT	1:1	R/W	0x0	Reset eFPGA Right Top Quadrant
RESET_LT	0:0	R/W	0x0	Reset eFPGA Left Top Quadrant

ENABLE_IN_OUT_EFPGA offset = 0x00EC

Field	Bits	Type	Default	Description
ENABLE_EVENTS	5:5	R/W	0x00	Enable events from efpga to SOC
ENABLE_SOC_ACCESS	4:4	R/W	0x0	Enable SOC memory mapped access to EFPGA
ENABLE_TCDM_P3	3:3	R/W	0x0	Enable EFPGA access via TCDM port 3
ENABLE_TCDM_P2	2:2	R/W	1x0	Enable EFPGA access via TCDM port 2
ENABLE_TCDM_P1	1:1	R/W	2x0	Enable EFPGA access via TCDM port 1
ENABLE_TCDM_P0	0:0	R/W	3x0	Enable EFPGA access via TCDM port 0

EFPGA_CONTROL_IN offset = 0x00F0

EFPGA_STATUS_OUT offset = 0x00F4

EFPGA_VERSION offset = 0x00F8

SOFT_RESET offset = 0x00FC

IO_CTRL[57] offset = 0x0400

Field	Bits	Type	Default	Description
CFG	13:8	RW	0x00	Pad configuration (TBD)

Field	Bits	Type	Default	Description
MUX	1:0	RW	0x00	Mux select

Notes:

Access type	Description
RW	Read & Write
RO	Read Only
RC	Read & Clear after read
WO	Write Only
WC	Write Clears (value ignored; always writes a 0)
WS	Write Sets (value ignored; always writes a 1)
RW1S	Read & on Write bits with 1 get set, bits with 0 left unchanged
RW1C	Read & on Write bits with 1 get cleared, bits with 0 left unchanged
RW0C	Read & on Write bits with 0 get cleared, bits with 1 left unchanged

APB_EVENT_CNTRL

Memory address: SOC_EVENT_GEN_START_ADDR(‘SOC_EVENT_START_ADDR)

This APB peripheral device collects all the events which presented to the CPU as IRQ11 (Machine interrupt). Each event is individually maskable by the appropriate bit in the EVENT_MASKx register. When an enabled event (unmasked) is received it is placed in a event FIFO and the IRQ11 signal is presented to the CPU which can then read the EVENT FIFO to determine which event cause the interrupt. each event has a queue of depth four to collect events if the queue for any event overflows an error is logged into the appropriate EVENT_ERR register and IRQ31 is presented to the CPU.

APB_EVENTS offset = 0x000

Field	Bits	Type	Default	Description
apb_event	15:0			16-bits of software generated events

EVENT_MASK0 offset = 0x004

Field	Bits	Type	Default	Description
event_enable	31:30		0xFFFFFFFF	Individual masks for events 0 - 31 1=mask event

EVENT_MASK1 offset = 0x008

Field	Bits	Type	Default	Description
event_enable	31:30			individual masks for events 32 - 63 1=mask event

EVENT_MASK2 offset = 0x00C

Field	Bits	Type	Default	Description
event_enable	31:30		0xFFFFFFFF	individual masks for events 64 - 95 1=mask event

EVENT_MASK3 offset = 0x010

Field	Bits	Type	Default	Description
event_enable	31:30		0xFFFFFFFF	individual masks for events 96 - 127 1=mask event

EVENT_MASK4 offset = 0x014

Field	Bits	Type	Default	Description
event_enable	31:30		0xFFFFFFFF	individual masks for events 128 - 159 1=mask event

EVENT_MASK5 offset = 0x018

Field	Bits	Type	Default	Description
event_enable	31:30		0xFFFFFFFF	individual masks for events 160 - 191 1=mask event

EVENT_MASK6 offset = 0x01C

Field	Bits	Type	Default	Description
event_enable	31:30		0xFFFFFFFF	individual masks for events 192 - 223 1=mask event

EVENT_MASK7 offset = 0x020

Field	Bits	Type	Default	Description
event_enable	31:30		0xFFFFFFFF	individual masks for events 224 - 255 1=mask event

EVENT_ERR0 offset = 0x064

Field	Bits	Type	Default	Description
event_err	31:30		0x0	individual error bits to indicate event queue overflow for events 0 - 31

EVENT_ERR1 offset = 0x068

Field	Bits	Type	Default	Description
event_enable	31:30		0x0	individual error bits to indicate event queue overflow for events 32 - 63

EVENT_ERR2 offset = 0x06C

Field	Bits	Type	Default	Description
event_enable	31:30		0x0	individual error bits to indicate event queue overflow for events 64 - 95

EVENT_ERR3 offset = 0x070

Field	Bits	Type	Default	Description
event_enable	31:30		0x0	individual error bits to indicate event queue overflow for events 96 - 127

EVENT_ERR4 offset = 0x074

Field	Bits	Type	Default	Description
event_enable	31:30	0x0		individual error bits to indicate event queue overflow for events 128 - 159

EVENT_ERR5 offset = 0x078

Field	Bits	Type	Default	Description
event_enable	31:30	0x0		individual error bits to indicate event queue overflow for events 160 - 191

EVENT_ERR6 offset = 0x07C

Field	Bits	Type	Default	Description
event_enable	31:30	0x0		individual error bits to indicate event queue overflow for events 192 - 223

EVENT_ERR7 offset = 0x080

Field	Bits	Type	Default	Description
event_enable	31:30	0x0		individual error bits to indicate event queue overflow for events 224 - 255

TIMER_LO_EVENT offset = 0x0084

Field	Bits	Type	Default	Description
event	7:0		specifies which event should be routed to the lo timer	

TIMER_HI_EVENT offset = 0x0088

Field	Bits	Type	Default	Description
event	7:0		specifies which event should be routed to the hi timer	

EVENT_FIFO offset = 0x090

Field	Bits	Type	Default	Description
EVENT_ID	7:0	RO		ID of triggering event to be read by interrupt handler

Notes:

Access type	Description
RW	Read & Write
RO	Read Only
RC	Read & Clear after read
WO	Write Only
WC	Write Clears (value ignored; always writes a 0)
WS	Write Sets (value ignored; always writes a 1)
RW1S	Read & on Write bits with 1 get set, bits with 0 left unchanged
RW1C	Read & on Write bits with 1 get cleared, bits with 0 left unchanged
RW0C	Read & on Write bits with 0 get cleared, bits with 1 left unchanged

APB_TIMER_UNIT

Memory address: TIMER_START_ADDDR(0x1A10B000)

CFG_REG_LO offset = 0x000

Field	Bits	Type	Default	Description
MODE_64_BIT	31:31	RW		1 = 64-bit mode, 0=32-bit mode
MODE_MTIME_BIT	30:30	RW		1=MTIME mode Changes interrupt to be >= CMP value
PRESCALER_COUNT	15:8	RW		Prescaler divisor
REF_CLK_EN_BIT	7:7	RW		1= use Refclk for counter, 0 = use APB bus clk for counter
PRESCALER_EN_BIT	6:6	RW		1= Use prescaler 0= no prescaler
ONE_SHOT_BIT	5:5	RW		1= disable timer when counter == cmp value
CMP_CLR_BIT	4:4	RW		1=counter is reset once counter == cmp, 0=counter is not reset
IEM_BIT	3:3	RW		1 = event input is enabled
IRQ_BIT	2:2	RW		1 = IRQ is enabled when counter ==cmp value

Field	Bits	Type	Default	Description
RESET_BIT	1:1	RW		1 = reset the counter
ENABLE_BIT	0:0	RW		1 = enable the counter to count

CFG_REG_HI offset = 0x004

Field	Bits	Type	Default	Description
MODE_64_BIT	31:31	RW		
MODE_MTIME_BIT	30:30	RW		
PRESCALER_COUNT	15:8	RW		
REF_CLK_EN_BIT	7:7	RW		
PRESCALER_EN_BIT	6:6	RW		
ONE_SHOT_BIT	5:5	RW		
CMP_CLR_BIT	4:4	RW		
IEM_BIT	3:3	RW		
IRQ_BIT	2:2	RW		
RESET_BIT	1:1	RW		
ENABLE_BIT	0:0	RW		

TIMER_VAL_LO offset = 0x008

Field	Bits	Type	Default	Description
TIMER_VAL_LO	31:0	RW	0x0	

TIMER_VAL_HI offset = 0x00C

Field	Bits	Type	Default	Description
TIMER_VAL_HI	31:0	RW	0x0	

TIMER_CMP_LO offset = 0x010

Field	Bits	Type	Default	Description
TIMER_CMP_LO	31:0	RW	0x0	

TIMER_CMP_HI offset = 0x014

Field	Bits	Type	Default	Description
TIMER_CMP_HI	31:0	RW	0x0	

TIMER_START_LO offset = 0x018

Field	Bits	Type	Default	Description
TIMER_START_LO	0:0	WS	0x0	

TIMER_START_HI offset = 0x01C

Field	Bits	Type	Default	Description
TIMER_START_HI	0:0	WS	0x0	

TIMER_RESET_LO offset = 0x020

Field	Bits	Type	Default	Description
TIMER_RESET_LO	0:0	WS	0x0	

TIMER_RESET_HI offset = 0x024

Field	Bits	Type	Default	Description
TIMER_RESET_HI	0:0	WS	0x0	

Notes:

Access type	Description
RW	Read & Write
RO	Read Only
RC	Read & Clear after read
WO	Write Only
WC	Write Clears (value ignored; always writes a 0)
WS	Write Sets (value ignored; always writes a 1)
RW1S	Read & on Write bits with 1 get set, bits with 0 left unchanged
RW1C	Read & on Write bits with 1 get cleared, bits with 0 left unchanged
RW0C	Read & on Write bits with 0 get cleared, bits with 1 left unchanged

APB_GPIO

Memory address: GPIO_START_ADDR(0x1A101000)

The GPIO module supports S/W access to read and write the values on selected I/O, and configuring selected I/O to generate interrupts.

Interrupts

Any GPIO can be configured for level type interrupts or edge triggered interrupts. Levels based int

SETGPIO offset = 0x00

Field	Bits	Type	Default	Description
gpio_num	7:0	WO		Set GPIO[gpio_num] = 1

CLRGPIO offset = 0x04

Field	Bits	Type	Default	Description
gpio_num	7:0	WO		Set GPIO[gpio_num] = 0

TOGGPIO offset = 0x08

Field	Bits	Type	Default	Description
gpio_num	7:0	WO		Invert the output of GPIO[gpio_num]

PIN0 offset = 0x10

Field	Bits	Type	Default	Description
gpio_value	31:0	RO		gpio_value[31:0] = GPIO[31:0]

PIN1 offset = 0x14

Field	Bits	Type	Default	Description
gpio_value	31:0	RO		gpio_value[31:0] = GPIO[63:32]

PIN2 offset = 0x18

Field	Bits	Type	Default	Description
gpio_value	31:0	RO		gpio_value[31:0] = GPIO[95:64]

PIN3 offset = 0x1C

Field	Bits	Type	Default	Description
gpio_value	31:0	RO		gpio_value[31:0] = GPIO[127:96]

OUT0 offset = 0x20

Field	Bits	Type	Default	Description
value	31:0	WO		Drive value[31:0] onto GPIO[31:0]

OUT1 offset = 0x24

Field	Bits	Type	Default	Description
value	31:0	WO		Drive value[31:0] onto GPIO[63:32]

OUT2 offset = 0x28

Field	Bits	Type	Default	Description
value	31:0	WO		Drive value[31:0] onto GPIO[95:64]

OUT3 offset = 0x2C

Field	Bits	Type	Default	Description
value	31:0	WO		Drive value[31:0] onto GPIO[127:96]

SETSEL offset = 0x30

Field	Bits	Type	Default	Description
gpio_num	7:0	WO	0x0	Set gpio_num for use by RDSTAT Note: SETGPIO, CLRGPIO, TOGGPIO and SETINT set gpio_num

RDSTAT offset = 0x34

Field	Bits	Type	Default	Description
mode	25:24	RO	0x0	Read the mode control for GPIO[gpio_num] (set gpio_num using SETSEL) 0x0: Input only (output is tri-stated) 0x1: Output active 0x2: Open drain (value=0 drives 0, when value=1 tristated) 0x3: Open drain (value=0 drives 0, when value=1 tristated)
INTTYPE	19:17	RO	0x0	Type of interrupt for GPIO[gpio_num] 0x0: active low, level type interrupt 0x1: rising edge type interupt 0x2: falling edge type interrupt 0x3: no interrupt 0x4: active high, level type interrupt 0x5 to 0x7: no interrupt
INTEN	16:16	RW		Enable interrupt for GPIO[gpio_num]
INPUT	12:12	RO		Input value reported by GPIO[gpio_num]
OUTPUT	8:8	RO		Output value that is set on GPIO[gpio_num]
gpio_sel	7:0	RO		Selected gpio

SETMODE offset = 0x38

Field	Bits	Type	Default	Description
mode	25:24	WO	0x0	mode control for GPIO[gpio_num} 0x0: Input only (output is tri-stated) 0x1: Output active 0x2: Open drain (value=0 drives 0, when value=1 tristated) 0x3: Open drain (value=0 drives 0, when value=1 tristated)
gpio_num	7:0	WO	0x0	Address of GPIO to set mode for

SETINT offset = 0x3C

Field	Bits	Type	Default	Description
INTTYPE	19:17	WO	0x0	Type of interrupt for GPIO[gpio_num] 0x0: active low, level type interrupt 0x1: rising edge type interupt 0x2: falling edge type interrupt 0x3: no interrupt 0x4: active high, level type interrupt 0x5 to 0x7: no interrupt
INTENABLE	16:16	WO	0x0	Enable interrupt on GPIO[GPIO_ADDDR]
gpio_num	7:0	WO	0x0	Address of GPIO to set interrupt type and enable for

INTACK offset = 0x40

Field	Bits	Type	Default	Description
RESERVED	31:8	RW		Enable interrupt for GPIO[gpio_num]
gpio_num	7:0	RW		Input value reported by GPIO[gpio_num]

Notes:

Access type	Description
RW	Read & Write
RO	Read Only
RC	Read & Clear after read
WO	Write Only
WC	Write Clears (value ignored; always writes a 0)
WS	Write Sets (value ignored; always writes a 1)
RW1S	Read & on Write bits with 1 get set, bits with 0 left unchanged
RW1C	Read & on Write bits with 1 get cleared, bits with 0 left unchanged
RW0C	Read & on Write bits with 0 get cleared, bits with 1 left unchanged

APB_I2CS

Memory address: I2CS_START_ADDR(0x1A107000)

I2CS_DEV_ADDRESS offset = 0x000

Field	Bits	Type	Default	Description
RESERVED	7:7	RW		Reserved
SLAVE_ADDR	6:0	RW	0x6F	I2C Device Address

I2CS_ENABLE offset = 0x004

Field	Bits	Type	Default	Description
RESERVED	7:1	RW		
IP_ENABLE	0:0	RW		

I2CS_DEBOUNCE_LENGTH offset = 0x008

Field	Bits	Type	Default	Description
DEB_LEN	7:0	RW	0x14	

I2CS_SCL_DELAY_LENGTH offset = 0x00C

Field	Bits	Type	Default	Description
SCL_DLY_LEN	7:0	RW	0x14	

I2CS_SDA_DELAY_LENGTH offset = 0x010

Field	Bits	Type	Default	Description
SDA_DLY_LEN	7:0	RW	0x14	

I2CS_MSG_I2C_APB offset = 0x040

Field	Bits	Type	Default	Description
I2C_TO_APB	7:0	RW	0x00	

I2CS_MSG_I2C_APB_STATUS offset = 0x044

Field	Bits	Type	Default	Description
RESERVED	7:1	RW		Reserved
I2C_TO_APB_STATUS	0:0	RW	0x00	

I2CS_MSG_APB_I2C offset = 0x048

Field	Bits	Type	Default	Description
APB_TO_I2C	7:0	RW	0x00	

I2CS_MSG_APB_I2C_STATUS offset = 0x04C

Field	Bits	Type	Default	Description
RESERVED	7:1	RW		Reserved
APB_TO_I2C_STATUS	0:0	RW	0x00	

I2CS_FIFO_I2C_APB_WRITE_DATA_PORT offset = 0x080

Field	Bits	Type	Default	Description
I2C_APB_WRITE_DATA_PORT	31:0	RW		

I2CS_FIFO_I2C_APB_READ_DATA_PORT offset = 0x084

Field	Bits	Type	Default	Description
I2C_APB_READ_DATA_PORT	31:0	RW		

I2CS_FIFO_I2C_APB_FLUSH offset = 0x088

Field	Bits	Type	Default	Description
RESERVED	7:1	RW		Reserved
ENABLE	0:0	RW		Writing a 1 to this register bit will flush the I2CtoAPB FIFO, clearing all contents and rendering the FIFO to be empty

I2CS_FIFO_I2C_APB_WRITE_FLAGS offset = 0x08C

Field	Bits	Type	Default	Description
RESERVED	7:3	RW		Reserved
FLAGS	2:0	RW		

I2CS_FIFO_I2C_APB_READ_FLAGS offset = 0x090

Field	Bits	Type	Default	Description
RESERVED	7:3	RW		Reserved
FLAGS	2:0	RW		

I2CS_FIFO_APB_I2C_WRITE_DATA_PORT offset = 0x0C0

Field	Bits	Type	Default	Description
I2C_APB_WRITE_DATA_PORT	31:0	RW		

I2CS_FIFO_APB_I2C_READ_DATA_PORT offset = 0x0C4

Field	Bits	Type	Default	Description
I2C_APB_READ_DATA_PORT	31:0	RW		

I2CS_FIFO_APB_I2C_FLUSH offset = 0x0C8

Field	Bits	Type	Default	Description
RESERVED	7:1	RW		Reserved
ENABLE	0:0	RW		Writing a 1 to this register bit will flush the APBtoI2C FIFO, clearing all contents and rendering the FIFO to be empty

I2CS_FIFO_APB_I2C_WRITE_FLAGS offset = 0x0CC

Field	Bits	Type	Default	Description
RESERVED	7:3	R		Reserved
FLAGS	2:0	R		

I2CS_FIFO_APB_I2C_READ_FLAGS offset = 0x0D0

Field	Bits	Type	Default	Description
RESERVED	7:3	R		Reserved
FLAGS	2:0	R		

I2CS_INTERRUPT_STATUS offset = 0x100

Field	Bits	Type	Default	Description
RESERVED	7:3	R		Reserved
I2C_APB_FIFO_WRITE_STATUS	2:2	R		
APB_I2C_FIFO_READ_STATUS	1:1	R		
APB_I2C_MESSAGE_AVAILABLE	0:0	R		

I2CS_INTERRUPT_ENABLE offset = 0x104

Field	Bits	Type	Default	Description
RESERVED	7:3	RW		Reserved
I2C_APB_FIFO_WRITE_STATUS_INT_ENABLE	2:2	RW		
APB_I2C_FIFO_READ_STATUS_INT_ENABLE	1:1	RW		
APB_I2C_MESSAGE_AVAILABLE_INT_ENABLE	0:0	RW		

I2CS_INTERRUPT_I2C_APB_WRITE_FLAGS_SELECT offset = 0x108

Field	Bits	Type	Default	Description
WRITE_FLAG_FULL	7:7	RW		
WRITE_FLAG_1_SPACE_AVAIL	6:6	RW		
WRITE_FLAG_2_3_SPACE_AVAIL	5:5	RW		
WRITE_FLAG_4_7_SPACE_AVAIL	4:4	RW		
WRITE_FLAG_8_31_SPACE_AVAIL	3:3	RW		
WRITE_FLAG_32_63_SPACE_AVAIL	2:2	RW		
WRITE_FLAG_64_127_SPACE_AVAIL	1:1	RW		
WRITE_FLAG_128_SPACE_AVAIL	0:0	RW		

I2CS_INTERRUPT_APB_I2C_READ_FLAGS_SELECT offset = 0x10C

Field	Bits	Type	Default	Description
READ_FLAG_128_SPACE_AVAIL	7:7	RW		
READ_FLAG_64_127_SPACE_AVAIL	6:6	RW		
READ_FLAG_32_63_SPACE_AVAIL	5:5	RW		
READ_FLAG_8_31_SPACE_AVAIL	4:4	RW		
READ_FLAG_4_7_SPACE_AVAIL	3:3	RW		
READ_FLAG_2_3_SPACE_AVAIL	2:2	RW		
READ_FLAG_1_SPACE_AVAIL	1:1	RW		
READ_FLAG_EMPTY	0:0	RW		

I2CS_INTERRUPT_TO_APB_STATUS offset = 0x140

Field	Bits	Type	Default	Description
RESERVED	7:3	RW		
APB_I2C_FIFO_WRITE_STATUS	2:2	RW		
I2C_APB_FIFO_READ_STATUS	1:1	RW		
NEW_I2C_APB_MSG_AVAIL	0:0	RW		

I2CS_INTERRUPT_TO_APB_ENABLE offset = 0x144

Field	Bits	Type	Default	Description
RESERVED	7:3	RW		
APB_I2C_FIFO_WRITE_STATUS_ENABLE	2:2	RW		
I2C_APB_FIFO_READ_STATUS_ENABLE	1:1	RW		
NEW_I2C_APB_MSG_AVAIL_ENABLE	0:0	RW		

I2CS_INTERRUPT_APB_I2C_WRITE_FLAGS_SELECT offset = 0x148

Field	Bits	Type	Default	Description
WRITE_FLAG_FULL	7:7	RW		
WRITE_FLAG_1_SPACE_AVAIL	6:6	RW		
WRITE_FLAG_2_3_SPACE_AVAIL	5:5	RW		
WRITE_FLAG_4_7_SPACE_AVAIL	4:4	RW		
WRITE_FLAG_8_31_SPACE_AVAIL	3:3	RW		
WRITE_FLAG_32_63_SPACE_AVAIL	2:2	RW		
WRITE_FLAG_64_127_SPACE_AVAIL	1:1	RW		
WRITE_FLAG_128_SPACE_AVAIL	0:0	RW		

I2CS_INTERRUPT_I2C_APB_READ_FLAGS_SELECT offset = 0x14C

Field	Bits	Type	Default	Description
READ_FLAG_128_SPACE_AVAIL	7:7	RW		
READ_FLAG_64_127_SPACE_AVAIL	6:6	RW		
READ_FLAG_32_63_SPACE_AVAIL	5:5	RW		
READ_FLAG_8_31_SPACE_AVAIL	4:4	RW		
READ_FLAG_4_7_SPACE_AVAIL	3:3	RW		
READ_FLAG_2_3_SPACE_AVAIL	2:2	RW		
READ_FLAG_1_SPACE_AVAIL	1:1	RW		
READ_FLAG_EMPTY	0:0	RW		

Notes:

Access type	Description
RW	Read & Write
RO	Read Only
RC	Read & Clear after read
WO	Write Only
WC	Write Clears (value ignored; always writes a 0)
WS	Write Sets (value ignored; always writes a 1)
RW1S	Read & on Write bits with 1 get set, bits with 0 left unchanged
RW1C	Read & on Write bits with 1 get cleared, bits with 0 left unchanged
RW0C	Read & on Write bits with 0 get cleared, bits with 1 left unchanged

eFPGA Template

Memory address: EFPGA_ASYNC_APB_START_ADD(‘ EFPGA_ASYNC_APB_START_ADD)

The eFPGA Template instantiates all the user accessible IO in the eFPGA and makes it available for logic validation .

The eFPGA provide the following interfaces and resources 1. Asynchronous CPU read/write bus. With 32-bit data, byte enables and 20-bits Address 2. Four TCDM interfaces for eFPGA high speed access to read/write main L2 RAM 3. 32-bits of fpga IO which connects the fpga to the device pins via the pin mux 4. 16-bits of event generation that can interrupt the CPU. 5. 2 Math units each mathunit contains 2 32-bit multipliers that can be fractured into 2-16-bit, 4-8-bit or 8-4-bit multiply with accumulators and 3 4Kbyte Simple dual port RAMS.

TCDM_CTL_P0 offset = 0x00

Field	Bits	Type	Default	Description
tcdm_wen_p0	31:31	RW	0x0	1 = read on TCDM 0, 0 = write on TCDM 0
tcdm_be_p0	23:20	RW	0x0	Set the bye enables for TCDM 0
tcdm_addr_p0	19:0	RW	0x0	Sets the address to be used on TCDM 0

TCDM_CTL_P1 offset = 0x04

Field	Bits	Type	Default	Description
tcdm_wen_p1	31:31	RW	0x0	1 = read on TCDM 1, 0 = write on TCDM 1
tcdm_be_p1	23:20	RW	0x0	Set the bye enables for TCDM 1
tcdm_addr_p1	19:0	RW	0x0	Sets the address to be used on TCDM 1

TCDM_CTL_P2 offset = 0x08

Field	Bits	Type	Default	Description
tcdm_wen_p2	31:31	RW	0x0	1 = read on TCDM 2, 0 = write on TCDM 2
tcdm_be_p2	23:20	RW	0x0	Set the bye enables for TCDM 2
tcdm_addr_p2	19:0	RW	0x0	Sets the address to be used on TCDM 2

TCDM_CTL_P3 offset = 0x0C

Field	Bits	Type	Default	Description
tcdm_wen_p3	31:31	RW	0x0	1 = read on TCDM 3, 0 = write on TCDM 3
tcdm_be_p3	23:20	RW	0x0	Set the bye enables for TCDM 3
tcdm_addr_p3	19:0	RW	0x0	Sets the address to be used on TCDM 3

M0_M0_CONTROL offset = 0x10

Field	Bits	Type	Default	Description
m0_m0_reset	31:31	RW	0x0	Math Unit 0, Multiplier 0 reset accumulator
m0_m0_sat	18:18	RW	0x0	Math Unit 0, Multiplier 0 select saturation
m0_m0_clr	17:17	RW	0x0	Math Unit 0, Multiplier 0 clear accumulator
m0_m0_rnd	16:16	RW	0x0	Math Unit 0, Multiplier 0 select rounding
m0_m0_csel	15:15	RW	0x0	Math Unit 0, Multiplier 0 coefficient selection
m0_m0_osel	14:14	RW	0x0	Math Unit 0, Multiplier 0 operand slection
m0_m0_mode	13:12	RW	0x0	Math Unit 0, Multiplier 0 mode. 00 = 32-bit, 01 = 16-bit, 10= 8-bit, 11 = 4-bit
m0_m0_outsel	5:0	RW	0x0	Math Unit 0, Mutliplier 0 output select

M0_M1_CONTROL offset = 0x14

Field	Bits	Type	Default	Description
m0_m1_reset	31:31	RW	0x0	Math Unit 0, Multiplier 1 reset accumulator

Field	Bits	Type	Default	Description
m0_m1_sat	18:18	RW	0x0	Math Unit 0, Multiplier 1 select saturation
m0_m1_clr	17:17	RW	0x0	Math Unit 0, Multiplier 1 clear accumulator
m0_m1_rnd	16:16	RW	0x0	Math Unit 0, Multiplier 1 select rounding
m0_m1_csel	15:15	RW	0x0	Math Unit 0, Multiplier 1 coefficient selection
m0_m1_osel	14:14	RW	0x0	Math Unit 0, Multiplier 1 operand slection
m0_m1_mode	13:12	RW	0x0	Math Unit 0, Multiplier 1 mode. 00 = 32-bit, 01 = 16-bit, 10= 8-bit, 11 = 4-bit
m0_m1_outsel	5:0	RW	0x0	Math Unit 0, Mutliplier 1 output select

M1_M0_CONTROL offset = 0x18

Field	Bits	Type	Default	Description
m1_m0_reset	31:31	RW	0x0	Math Unit 1, Multiplier 0 reset ac
m1_m1_sat	18:18	RW	0x0	Math Unit 1, Multiplier 0 select sa
m1_m0_clr	17:17	RW	0x0	Math Unit 1, Multiplier 0 clear ac
m1_m0_rnd	16:16	RW	0x0	Math Unit 1, Multiplier 0 select ro
m1_m0_csel	15:15	RW	0x0	Math Unit 1, Multiplier 0 coefficie
m1_m0_osel	14:14	RW	0x0	Math Unit 1, Multiplier 0 operand
m1_m0_mode	13:12	RW	0x0	Math Unit 1, Multiplier 0 mode.
00 = 32-bit, 01 = 16-bit, 10= 8-bit, 11 = 4-bit				
m1_m0_outsel	5:0	RW	0x0	Math Unit 1, Mutliplier 0 output

M1_M1_CONTROL offset = 0x1C

Field	Bits	Type	Default	Description
m1_m1_reset	31:31	RW	0x0	Math Unit 1, Multiplier 1 reset ac
m1_m1_sat	18:18	RW	0x0	Math Unit 1, Multiplier 1 select sa
m1_m1_clr	17:17	RW	0x0	Math Unit 1, Multiplier 1 clear ac
m1_m1_rnd	16:16	RW	0x0	Math Unit 1, Multiplier 1 select ro
m1_m1_csel	15:15	RW	0x0	Math Unit 1, Multiplier 1 coefficie
m1_m1_osel	14:14	RW	0x0	Math Unit 1, Multiplier 1 operand
m1_m1_mode	13:12	RW	0x0	Math Unit 1, Multiplier 1 mode.
00 = 32-bit, 01 = 16-bit, 10= 8-bit, 11 = 4-bit				
m1_m1_outsel	5:0	RW	0x0	Math Unit 1, Mutliplier 1 output

M0_RAM_CONTROL offset = 0x20

Field	Bits	Type	Default	Description
m0_coef_wdsel	14:14	RW	0x0	Math Unit 0 coefficient RAM write data select
m0_oper1_wdsel	13:13	RW	0x0	Math Unit 0 Operand 0 RAM write data select
m0_oper0_wdsel	12:12	RW	0x0	Math Unit 0 Operand 1 RAM write data select
m0_coef_wmode	11:10	RW	0x0	Math Unit 0 coefficient RAM write mode
m0_coef_rmode	9:8	RW	0x0	Math Unit 0 coefficient RAM read mode
m0_oper1_wmode	7:6	RW	0x0	Math Unit 0 operand 0 RAM write mode
m0_oper1_rmode	5:4	RW	0x0	Math Unit 0 operand 0 RAM read mode
m0_oper0_wmode	3:2	RW	0x0	Math Unit 0 operand 1 RAM write mode
m0_oper0_rmode	1:0	RW	0x0	Math Unit 0 operand 1 RAM read mode

M1_RAM_CONTROL offset = 0x24

Field	Bits	Type	Default	Description
m1_coef_wdsel	14:14	RW	0x0	Math Unit 1 coefficient RAM write data select
m1_oper1_wdsel	13:13	RW	0x0	Math Unit 1 Operand 0 RAM write data select
m1_oper0_wdsel	12:12	RW	0x0	Math Unit 1 Operand 1 RAM write data select
m1_coef_wmode	11:10	RW	0x0	Math Unit 1 coefficient RAM write mode
m1_coef_rmode	9:8	RW	0x0	Math Unit 1 coefficient RAM read mode
m1_oper1_wmode	7:6	RW	0x0	Math Unit 1 operand 0 RAM write mode
m1_oper1_rmode	5:4	RW	0x0	Math Unit 1 operand 0 RAM read mode
m1_oper0_wmode	3:2	RW	0x0	Math Unit 1 operand 1 RAM write mode
m1_oper0_rmode	1:0	RW	0x0	Math Unit 1 operand 1 RAM read mode

M0_M0_CLKEN offset = 0x30

Field	Bits	Type	Default	Description
m0_m0_clken	0:0	WO	-	Math Unit 0 Multiplier 0 Clock enable

M0_M1_CLKEN offset = 0x34

Field	Bits	Type	Default	Description
m0_m1_clken	0:0	WO	-	Math Unit 0 Multiplier 1 Clock enable

M1_M0_CLKEN offset = 0x38

Field	Bits	Type	Default	Description
m1_m0_clken	0:0	WO	-	Math Unit 1 Multiplier 0 Clock enable

M1_M1_CLKEN offset = 0x3C

Field	Bits	Type	Default	Description
m1_m1_clken	0:0	WO	-	Math Unit 1 Multiplier 1 Clock enable

FPGAIO_OUT31_00 offset = 0x40

Field	Bits	Type	Default	Description
fpgaio_o_31	31:31	RW	0x0	Sets the fpgio output bit 31
fpgaio_o_30	30:30	RW	0x0	Sets the fpgio output bit 30
fpgaio_o_29	29:29	RW	0x0	Sets the fpgio output bit 29
fpgaio_o_28	28:28	RW	0x0	Sets the fpgio output bit 28
fpgaio_o_27	27:27	RW	0x0	Sets the fpgio output bit 27
fpgaio_o_26	26:26	RW	0x0	Sets the fpgio output bit 26
fpgaio_o_25	25:25	RW	0x0	Sets the fpgio output bit 25
fpgaio_o_24	24:24	RW	0x0	Sets the fpgio output bit 24
fpgaio_o_23	23:23	RW	0x0	Sets the fpgio output bit 23
fpgaio_o_22	22:22	RW	0x0	Sets the fpgio output bit 22
fpgaio_o_21	21:21	RW	0x0	Sets the fpgio output bit 21
fpgaio_o_20	20:20	RW	0x0	Sets the fpgio output bit 20
fpgaio_o_19	19:19	RW	0x0	Sets the fpgio output bit 19
fpgaio_o_18	18:18	RW	0x0	Sets the fpgio output bit 18
fpgaio_o_17	17:17	RW	0x0	Sets the fpgio output bit 17
fpgaio_o_16	16:16	RW	0x0	Sets the fpgio output bit 16
fpgaio_o_15	15:15	RW	0x0	Sets the fpgio output bit 15
fpgaio_o_14	14:14	RW	0x0	Sets the fpgio output bit 14
fpgaio_o_13	13:13	RW	0x0	Sets the fpgio output bit 13
fpgaio_o_12	12:12	RW	0x0	Sets the fpgio output bit 12
fpgaio_o_11	11:11	RW	0x0	Sets the fpgio output bit 11
fpgaio_o_10	10:10	RW	0x0	Sets the fpgio output bit 10
fpgaio_o_9	9:9	RW	0x0	Sets the fpgio output bit 9
fpgaio_o_8	8:8	RW	0x0	Sets the fpgio output bit 8
fpgaio_o_7	7:7	RW	0x0	Sets the fpgio output bit 7
fpgaio_o_6	6:6	RW	0x0	Sets the fpgio output bit 6
fpgaio_o_5	5:5	RW	0x0	Sets the fpgio output bit 5
fpgaio_o_4	4:4	RW	0x0	Sets the fpgio output bit 4
fpgaio_o_3	3:3	RW	0x0	Sets the fpgio output bit 3
fpgaio_o_2	2:2	RW	0x0	Sets the fpgio output bit 2
fpgaio_o_1	1:1	RW	0x0	Sets the fpgio output bit 1

Field	Bits	Type	Default	Description
fpgaio_o_0	0:0	RW	0x0	Sets the fpgio output bit 0

FPGAIO_OUT63_32 offset = 0x44

Field	Bits	Type	Default	Description
fpgaio_o_63	31:31	RW	0x0	Sets the fpgio output bit 63
fpgaio_o_62	30:30	RW	0x0	Sets the fpgio output bit 62
fpgaio_o_61	29:29	RW	0x0	Sets the fpgio output bit 61
fpgaio_o_60	28:28	RW	0x0	Sets the fpgio output bit 60
fpgaio_o_59	27:27	RW	0x0	Sets the fpgio output bit 59
fpgaio_o_58	26:26	RW	0x0	Sets the fpgio output bit 58
fpgaio_o_57	25:25	RW	0x0	Sets the fpgio output bit 57
fpgaio_o_56	24:24	RW	0x0	Sets the fpgio output bit 56
fpgaio_o_55	23:23	RW	0x0	Sets the fpgio output bit 55
fpgaio_o_54	22:22	RW	0x0	Sets the fpgio output bit 54
fpgaio_o_53	21:21	RW	0x0	Sets the fpgio output bit 53
fpgaio_o_52	20:20	RW	0x0	Sets the fpgio output bit 52
fpgaio_o_51	19:19	RW	0x0	Sets the fpgio output bit 51
fpgaio_o_50	18:18	RW	0x0	Sets the fpgio output bit 50
fpgaio_o_49	17:17	RW	0x0	Sets the fpgio output bit 49
fpgaio_o_48	16:16	RW	0x0	Sets the fpgio output bit 48
fpgaio_o_47	15:15	RW	0x0	Sets the fpgio output bit 47
fpgaio_o_46	14:14	RW	0x0	Sets the fpgio output bit 46
fpgaio_o_45	13:13	RW	0x0	Sets the fpgio output bit 45
fpgaio_o_44	12:12	RW	0x0	Sets the fpgio output bit 44
fpgaio_o_43	11:11	RW	0x0	Sets the fpgio output bit 43
fpgaio_o_42	10:10	RW	0x0	Sets the fpgio output bit 42
fpgaio_o_41	9:9	RW	0x0	Sets the fpgio output bit 41
fpgaio_o_40	8:8	RW	0x0	Sets the fpgio output bit 40
fpgaio_o_39	7:7	RW	0x0	Sets the fpgio output bit 39
fpgaio_o_38	6:6	RW	0x0	Sets the fpgio output bit 38
fpgaio_o_37	5:5	RW	0x0	Sets the fpgio output bit 37
fpgaio_o_36	4:4	RW	0x0	Sets the fpgio output bit 36
fpgaio_o_35	3:3	RW	0x0	Sets the fpgio output bit 35
fpgaio_o_34	2:2	RW	0x0	Sets the fpgio output bit 34
fpgaio_o_33	1:1	RW	0x0	Sets the fpgio output bit 33
fpgaio_o_32	0:0	RW	0x0	Sets the fpgio output bit 32

FPGAIO_OUT79_64 offset = 0x48

Field	Bits	Type	Default	Description
fpgaio_o_79	15:15	RW	0x0	Sets the fpgio output bit 79
fpgaio_o_78	14:14	RW	0x0	Sets the fpgio output bit 78
fpgaio_o_77	13:13	RW	0x0	Sets the fpgio output bit 77
fpgaio_o_76	12:12	RW	0x0	Sets the fpgio output bit 76
fpgaio_o_75	11:11	RW	0x0	Sets the fpgio output bit 75
fpgaio_o_74	10:10	RW	0x0	Sets the fpgio output bit 74
fpgaio_o_73	9:9	RW	0x0	Sets the fpgio output bit 73
fpgaio_o_72	8:8	RW	0x0	Sets the fpgio output bit 72
fpgaio_o_71	7:7	RW	0x0	Sets the fpgio output bit 71
fpgaio_o_70	6:6	RW	0x0	Sets the fpgio output bit 70
fpgaio_o_69	5:5	RW	0x0	Sets the fpgio output bit 69
fpgaio_o_68	4:4	RW	0x0	Sets the fpgio output bit 68
fpgaio_o_67	3:3	RW	0x0	Sets the fpgio output bit 67
fpgaio_o_66	2:2	RW	0x0	Sets the fpgio output bit 66
fpgaio_o_65	1:1	RW	0x0	Sets the fpgio output bit 65
fpgaio_o_64	0:0	RW	0x0	Sets the fpgio output bit 64

FPGAIO_OE31_00 offset = 0x50

Field	Bits	Type	Default	Description
fpgaio_oe_31	31:31	RW	0x0	Sets the fpgio output enable for bit 31
fpgaio_oe_30	30:30	RW	0x0	Sets the fpgio output enable for bit 30
fpgaio_oe_29	29:29	RW	0x0	Sets the fpgio output enable for bit 29
fpgaio_oe_28	28:28	RW	0x0	Sets the fpgio output enable for bit 28
fpgaio_oe_27	27:27	RW	0x0	Sets the fpgio output enable for bit 27
fpgaio_oe_26	26:26	RW	0x0	Sets the fpgio output enable for bit 26
fpgaio_oe_25	25:25	RW	0x0	Sets the fpgio output enable for bit 25
fpgaio_oe_24	24:24	RW	0x0	Sets the fpgio output enable for bit 24
fpgaio_oe_23	23:23	RW	0x0	Sets the fpgio output enable for bit 23
fpgaio_oe_22	22:22	RW	0x0	Sets the fpgio output enable for bit 22
fpgaio_oe_21	21:21	RW	0x0	Sets the fpgio output enable for bit 21
fpgaio_oe_20	20:20	RW	0x0	Sets the fpgio output enable for bit 20
fpgaio_oe_19	19:19	RW	0x0	Sets the fpgio output enable for bit 19
fpgaio_oe_18	18:18	RW	0x0	Sets the fpgio output enable for bit 18
fpgaio_oe_17	17:17	RW	0x0	Sets the fpgio output enable for bit 17
fpgaio_oe_16	16:16	RW	0x0	Sets the fpgio output enable for bit 16
fpgaio_oe_15	15:15	RW	0x0	Sets the fpgio output enable for bit 15
fpgaio_oe_14	14:14	RW	0x0	Sets the fpgio output enable for bit 14
fpgaio_oe_13	13:13	RW	0x0	Sets the fpgio output enable for bit 13
fpgaio_oe_12	12:12	RW	0x0	Sets the fpgio output enable for bit 12
fpgaio_oe_11	11:11	RW	0x0	Sets the fpgio output enable for bit 11
fpgaio_oe_10	10:10	RW	0x0	Sets the fpgio output enable for bit 10
fpgaio_oe_9	9:9	RW	0x0	Sets the fpgio output enable for bit 9

Field	Bits	Type	Default	Description
fpgaio_oe_8	8:8	RW	0x0	Sets the fpgio output enable for bit 8
fpgaio_oe_7	7:7	RW	0x0	Sets the fpgio output enable for bit 7
fpgaio_oe_6	6:6	RW	0x0	Sets the fpgio output enable for bit 6
fpgaio_oe_5	5:5	RW	0x0	Sets the fpgio output enable for bit 5
fpgaio_oe_4	4:4	RW	0x0	Sets the fpgio output enable for bit 4
fpgaio_oe_3	3:3	RW	0x0	Sets the fpgio output enable for bit 3
fpgaio_oe_2	2:2	RW	0x0	Sets the fpgio output enable for bit 2
fpgaio_oe_1	1:1	RW	0x0	Sets the fpgio output enable for bit 1
fpgaio_oe_0	0:0	RW	0x0	Sets the fpgio output enable for bit 0

FPGAIO_OE63_32 offset = 0x54

Field	Bits	Type	Default	Description
fpgaio_oe_63	31:31	RW	0x0	Sets the fpgio output enable for bit 63
fpgaio_oe_62	30:30	RW	0x0	Sets the fpgio output enable for bit 62
fpgaio_oe_61	29:29	RW	0x0	Sets the fpgio output enable for bit 61
fpgaio_oe_60	28:28	RW	0x0	Sets the fpgio output enable for bit 60
fpgaio_oe_59	27:27	RW	0x0	Sets the fpgio output enable for bit 59
fpgaio_oe_58	26:26	RW	0x0	Sets the fpgio output enable for bit 58
fpgaio_oe_57	25:25	RW	0x0	Sets the fpgio output enable for bit 57
fpgaio_oe_56	24:24	RW	0x0	Sets the fpgio output enable for bit 56
fpgaio_oe_55	23:23	RW	0x0	Sets the fpgio output enable for bit 55
fpgaio_oe_54	22:22	RW	0x0	Sets the fpgio output enable for bit 54
fpgaio_oe_53	21:21	RW	0x0	Sets the fpgio output enable for bit 53
fpgaio_oe_52	20:20	RW	0x0	Sets the fpgio output enable for bit 52
fpgaio_oe_51	19:19	RW	0x0	Sets the fpgio output enable for bit 51
fpgaio_oe_50	18:18	RW	0x0	Sets the fpgio output enable for bit 50
fpgaio_oe_49	17:17	RW	0x0	Sets the fpgio output enable for bit 49
fpgaio_oe_48	16:16	RW	0x0	Sets the fpgio output enable for bit 48
fpgaio_oe_47	15:15	RW	0x0	Sets the fpgio output enable for bit 47
fpgaio_oe_46	14:14	RW	0x0	Sets the fpgio output enable for bit 46
fpgaio_oe_45	13:13	RW	0x0	Sets the fpgio output enable for bit 45
fpgaio_oe_44	12:12	RW	0x0	Sets the fpgio output enable for bit 44
fpgaio_oe_43	11:11	RW	0x0	Sets the fpgio output enable for bit 43
fpgaio_oe_42	10:10	RW	0x0	Sets the fpgio output enable for bit 42
fpgaio_oe_41	9:9	RW	0x0	Sets the fpgio output enable for bit 41
fpgaio_oe_40	8:8	RW	0x0	Sets the fpgio output enable for bit 40
fpgaio_oe_39	7:7	RW	0x0	Sets the fpgio output enable for bit 39
fpgaio_oe_38	6:6	RW	0x0	Sets the fpgio output enable for bit 38
fpgaio_oe_37	5:5	RW	0x0	Sets the fpgio output enable for bit 37
fpgaio_oe_36	4:4	RW	0x0	Sets the fpgio output enable for bit 36
fpgaio_oe_35	3:3	RW	0x0	Sets the fpgio output enable for bit 35
fpgaio_oe_34	2:2	RW	0x0	Sets the fpgio output enable for bit 34

Field	Bits	Type	Default	Description
fpgaio_oe_33	1:1	RW	0x0	Sets the fpgio output enable for bit 33
fpgaio_oe_32	0:0	RW	0x0	Sets the fpgio output enable for bit 32

FPGAIO_OE79_64 offset = 0x58

Field	Bits	Type	Default	Description
fpgaio_oe_79	15:15	RW	0x0	Sets the fpgio output enable for bit 79
fpgaio_oe_78	14:14	RW	0x0	Sets the fpgio output enable for bit 78
fpgaio_oe_77	13:13	RW	0x0	Sets the fpgio output enable for bit 77
fpgaio_oe_76	12:12	RW	0x0	Sets the fpgio output enable for bit 76
fpgaio_oe_75	11:11	RW	0x0	Sets the fpgio output enable for bit 75
fpgaio_oe_74	10:10	RW	0x0	Sets the fpgio output enable for bit 74
fpgaio_oe_73	9:9	RW	0x0	Sets the fpgio output enable for bit 73
fpgaio_oe_72	8:8	RW	0x0	Sets the fpgio output enable for bit 72
fpgaio_oe_71	7:7	RW	0x0	Sets the fpgio output enable for bit 71
fpgaio_oe_70	6:6	RW	0x0	Sets the fpgio output enable for bit 70
fpgaio_oe_69	5:5	RW	0x0	Sets the fpgio output enable for bit 69
fpgaio_oe_68	4:4	RW	0x0	Sets the fpgio output enable for bit 68
fpgaio_oe_67	3:3	RW	0x0	Sets the fpgio output enable for bit 67
fpgaio_oe_66	2:2	RW	0x0	Sets the fpgio output enable for bit 66
fpgaio_oe_65	1:1	RW	0x0	Sets the fpgio output enable for bit 65
fpgaio_oe_64	0:0	RW	0x0	Sets the fpgio output enable for bit 64

FPGAIO_IN31_00 offset = 0x60

Field	Bits	Type	Default	Description
fpgaio_i_31	31:31	RW	0x0	Reads the fpgaio input value for bit 31
fpgaio_i_30	30:30	RW	0x0	Reads the fpgaio input value for bit 30
fpgaio_i_29	29:29	RW	0x0	Reads the fpgaio input value for bit 29
fpgaio_i_28	28:28	RW	0x0	Reads the fpgaio input value for bit 28
fpgaio_i_27	27:27	RW	0x0	Reads the fpgaio input value for bit 27
fpgaio_i_26	26:26	RW	0x0	Reads the fpgaio input value for bit 26
fpgaio_i_25	25:25	RW	0x0	Reads the fpgaio input value for bit 25
fpgaio_i_24	24:24	RW	0x0	Reads the fpgaio input value for bit 24
fpgaio_i_23	23:23	RW	0x0	Reads the fpgaio input value for bit 23
fpgaio_i_22	22:22	RW	0x0	Reads the fpgaio input value for bit 22
fpgaio_i_21	21:21	RW	0x0	Reads the fpgaio input value for bit 21
fpgaio_i_20	20:20	RW	0x0	Reads the fpgaio input value for bit 20
fpgaio_i_19	19:19	RW	0x0	Reads the fpgaio input value for bit 19
fpgaio_i_18	18:18	RW	0x0	Reads the fpgaio input value for bit 18
fpgaio_i_17	17:17	RW	0x0	Reads the fpgaio input value for bit 17

Field	Bits	Type	Default	Description
fpgaio_i_16	16:16	RW	0x0	Reads the fpgaio input value for bit 16
fpgaio_i_15	15:15	RW	0x0	Reads the fpgaio input value for bit 15
fpgaio_i_14	14:14	RW	0x0	Reads the fpgaio input value for bit 14
fpgaio_i_13	13:13	RW	0x0	Reads the fpgaio input value for bit 13
fpgaio_i_12	12:12	RW	0x0	Reads the fpgaio input value for bit 12
fpgaio_i_11	11:11	RW	0x0	Reads the fpgaio input value for bit 11
fpgaio_i_10	10:10	RW	0x0	Reads the fpgaio input value for bit 10
fpgaio_i_9	9:9	RW	0x0	Reads the fpgaio input value for bit 9
fpgaio_i_8	8:8	RW	0x0	Reads the fpgaio input value for bit 8
fpgaio_i_7	7:7	RW	0x0	Reads the fpgaio input value for bit 7
fpgaio_i_6	6:6	RW	0x0	Reads the fpgaio input value for bit 6
fpgaio_i_5	5:5	RW	0x0	Reads the fpgaio input value for bit 5
fpgaio_i_4	4:4	RW	0x0	Reads the fpgaio input value for bit 4
fpgaio_i_3	3:3	RW	0x0	Reads the fpgaio input value for bit 3
fpgaio_i_2	2:2	RW	0x0	Reads the fpgaio input value for bit 2
fpgaio_i_1	1:1	RW	0x0	Reads the fpgaio input value for bit 1
fpgaio_i_0	0:0	RW	0x0	Reads the fpgaio input value for bit 0

FPGAIO_IN63_32 offset = 0x64

Field	Bits	Type	Default	Description
fpgaio_i_63	31:31	RO		Reads the fpgaio input value for bit 63
fpgaio_i_62	30:30	RO		Reads the fpgaio input value for bit 62
fpgaio_i_61	29:29	RO		Reads the fpgaio input value for bit 61
fpgaio_i_60	28:28	RO		Reads the fpgaio input value for bit 60
fpgaio_i_59	27:27	RO		Reads the fpgaio input value for bit 59
fpgaio_i_58	26:26	RO		Reads the fpgaio input value for bit 58
fpgaio_i_57	25:25	RO		Reads the fpgaio input value for bit 57
fpgaio_i_56	24:24	RO		Reads the fpgaio input value for bit 56
fpgaio_i_55	23:23	RO		Reads the fpgaio input value for bit 55
fpgaio_i_54	22:22	RO		Reads the fpgaio input value for bit 54
fpgaio_i_53	21:21	RO		Reads the fpgaio input value for bit 53
fpgaio_i_52	20:20	RO		Reads the fpgaio input value for bit 52
fpgaio_i_51	19:19	RO		Reads the fpgaio input value for bit 51
fpgaio_i_50	18:18	RO		Reads the fpgaio input value for bit 50
fpgaio_i_49	17:17	RO		Reads the fpgaio input value for bit 49
fpgaio_i_48	16:16	RO		Reads the fpgaio input value for bit 48
fpgaio_i_47	15:15	RO		Reads the fpgaio input value for bit 47
fpgaio_i_46	14:14	RO		Reads the fpgaio input value for bit 46
fpgaio_i_45	13:13	RO		Reads the fpgaio input value for bit 45
fpgaio_i_44	12:12	RO		Reads the fpgaio input value for bit 44
fpgaio_i_43	11:11	RO		Reads the fpgaio input value for bit 43
fpgaio_i_42	10:10	RO		Reads the fpgaio input value for bit 42

Field	Bits	Type	Default	Description
fpgaio_i_41	9:9	RO		Reads the fpgaio input value for bit 41
fpgaio_i_40	8:8	RO		Reads the fpgaio input value for bit 40
fpgaio_i_39	7:7	RO		Reads the fpgaio input value for bit 39
fpgaio_i_38	6:6	RO		Reads the fpgaio input value for bit 38
fpgaio_i_37	5:5	RO		Reads the fpgaio input value for bit 37
fpgaio_i_36	4:4	RO		Reads the fpgaio input value for bit 36
fpgaio_i_35	3:3	RO		Reads the fpgaio input value for bit 35
fpgaio_i_34	2:2	RO		Reads the fpgaio input value for bit 34
fpgaio_i_33	1:1	RO		Reads the fpgaio input value for bit 33
fpgaio_i_32	0:0	RO		Reads the fpgaio input value for bit 32

FPGAIO_IN79_64 offset = 0x68

Field	Bits	Type	Default	Description
fpgaio_i_79	15:15	RO		Reads the fpgaio input value for bit 79
fpgaio_i_78	14:14	RO		Reads the fpgaio input value for bit 78
fpgaio_i_77	13:13	RO		Reads the fpgaio input value for bit 77
fpgaio_i_76	12:12	RO		Reads the fpgaio input value for bit 76
fpgaio_i_75	11:11	RO		Reads the fpgaio input value for bit 75
fpgaio_i_74	10:10	RO		Reads the fpgaio input value for bit 74
fpgaio_i_73	9:9	RO		Reads the fpgaio input value for bit 73
fpgaio_i_72	8:8	RO		Reads the fpgaio input value for bit 72
fpgaio_i_71	7:7	RO		Reads the fpgaio input value for bit 71
fpgaio_i_70	6:6	RO		Reads the fpgaio input value for bit 70
fpgaio_i_69	5:5	RO		Reads the fpgaio input value for bit 69
fpgaio_i_68	4:4	RO		Reads the fpgaio input value for bit 68
fpgaio_i_67	3:3	RO		Reads the fpgaio input value for bit 67
fpgaio_i_66	2:2	RO		Reads the fpgaio input value for bit 66
fpgaio_i_65	1:1	RO		Reads the fpgaio input value for bit 65
fpgaio_i_64	0:0	RO		Reads the fpgaio input value for bit 64

FPGA_EVENT15_00 offset = 0x6C

Field	Bits	Type	Default	Description
Event_15	15:15	RW		sets event 15 to the event unit
Event_14	14:14			sets event 14 to the event unit
Event_13	13:13			sets event 13 to the event unit
Event_12	12:12			sets event 12 to the event unit
Event_11	11:11			sets event 11 to the event unit
Event_10	10:10			sets event 10 to the event unit
Event_9	9:9			sets event 9 to the event unit

Field	Bits	Type	Default	Description
Event_8	8:8			sets event 8 to the event unit
Event_7	7:7			sets event 7 to the event unit
Event_6	6:6			sets event 6 to the event unit
Event_5	5:5			sets event 5 to the event unit
Event_4	4:4			sets event 4 to the event unit
Event_3	3:3			sets event 3 to the event unit
Event_2	2:2			sets event 2 to the event unit
Event_1	1:1			sets event 1 to the event unit
Event_0	0:0	RW		sets event 0 to the event unit

TCDM_RUN_P0 offset = 0x80

Field	Bits	Type	Default	Description
tcdm_wdata_p0	31:0	W		Runs a TCDM operation on P0 with TCDM_CTL_P0 Attributes

TCDM_RUN_P1 offset = 0x84

Field	Bits	Type	Default	Description
tcdm_wdata_p0	31:0	W		Runs a TCDM operation on P1 with TCDM_CTL_P0 Attributes

TCDM_RUN_P2 offset = 0x88

Field	Bits	Type	Default	Description
tcdm_wdata_p0	31:0	W		Runs a TCDM operation on P2 with TCDM_CTL_P0 Attributes

TCDM_RUN_P3 offset = 0x8C

Field	Bits	Type	Default	Description
tcdm_wdata_p0	31:0	W		Runs a TCDM operation on P3 with TCDM_CTL_P0 Attributes

M0_M0_ODATA offset = 0x90

Field	Bits	Type	Default	Description
odata	31:0	RW		Sets the operand data for math unit 0 multiplier 0

M0_M1_ODATA offset = 0x94

Field	Bits	Type	Default	Description
odata	31:0	RW		Sets the operand data for math unit 0 multiplier 1

M0_M0_CDATA offset = 0x98

Field	Bits	Type	Default	Description
cdata	31:0	RW		Sets the coeficient data for math unit 0 multiplier 0

M0_M1_CDATA offset = 0x9C

Field	Bits	Type	Default	Description
cdata	31:0	RW		Sets the coeficient data for math unit 0 multiplier 1

M1_M0_ODATA offset = 0xA0

Field	Bits	Type	Default	Description
odata	31:0	RW		Sets the operand data for math unit 1 multiplier 0

M1_M1_ODATA offset = 0xA4

Field	Bits	Type	Default	Description
odata	31:0	RW		Sets the operand data for math unit 0 multiplier 1

M1_M0_CDATA offset = 0xA8

Field	Bits	Type	Default	Description
cdata	31:0	RW		Sets the coeficient data for math unit 1 multiplier 0

M1_M1_CDATA offset = 0xAC

Field	Bits	Type	Default	Description
cdata	31:0	RW		Sets the coeficient data for math unit 1 multiplier 1

M0_M0_MULTOUT offset = 0x100

Field	Bits	Type	Default	Description
multout	31:0	RO		Reads the output of math unit 0 multiplier 0

M0_M1_MULTOUT offset = 0x104

Field	Bits	Type	Default	Description
multout	31:0	RO		Reads the output of math unit 0 multiplier 1

M1_M0_MULTOUT offset = 0x108

Field	Bits	Type	Default	Description
multout	31:0	RO		Reads the output of math unit 1 multiplier 0

M1_M01MULTOUT offset = 0x10C

Field	Bits	Type	Default	Description
multout	31:0	RO		Reads the output of math unit 1 multiplier 1

M0_OPER0[0x400] offset = 0x1000

M0_OPER1[0x400] offset = 0x2000

M0_COEF[0x400] offset = 0x3000

M1_OPER0[0x400] offset = 0x4000

M1_OPER1[0x400] offset = 0x5000

M1_COEF[0x400] offset = 0x6000

Notes:

Access type	Description
RW	Read & Write
RO	Read Only
RC	Read & Clear after read
WO	Write Only
WC	Write Clears (value ignored; always writes a 0)
WS	Write Sets (value ignored; always writes a 1)
RW1S	Read & on Write bits with 1 get set, bits with 0 left unchanged
RW1C	Read & on Write bits with 1 get cleared, bits with 0 left unchanged
RW0C	Read & on Write bits with 0 get cleared, bits with 1 left unchanged

UDMA_CTRL

Memory address: UDMA_CH_ADDR_CTRL(‘UDMA_CH_ADDR_CTRL)

The UDMA addresses are organized as an array of channels. The first channel, channel 0, is a control channel that is used to:

- enable or disable the peripheral clocks
- reset the periperal controller
- set compare value for the event macthing mechanism

The base address for the UDMA channels is defined as UDMA_START_ADDR in core-v-mcu-config.h The size of each channel is UDMA_CH_SIZE, therefore the address of channels N is UDMA_START_ADDR+N*UDMA_CH_SIZE. core-v-mcu-config.h has explicit defines for each peripheral. For instance, if there are 2 UARTS then there are three defines:

- UDMA_CH_ADDR_UART – address of first UART
- UDMA_CH_ADDR_UART0 – address of UART0
- UDMA_CH_ADDR_UART1 – address of UART1

The reason for having the UDMA_CH_UART define is so that you can programmatically access UART ID by using UDMA_CH_ADDR_UART + ID * UDMA_CH_SIZE

The register definitions for the control channel are specified in this section. The register definitions for each peripheral are specified in sections named UDMA_XXXXX.

REG_CG offset = 0x000

Field	Bits	Type	Default	Description
PERIPH_CLK_ENABLE	31:0	RW	0x0	Enable for peripheral clocks; see core-v-mcu_config ‘Peripheral clock enable masks’ f

REG_CFG_EVT offset = 0x004

Field	Bits	Type	Default	Description
CMP_EVENT3	31:24		0x00	Compare value for event detection
CMP_EVENT2	23:16		0x01	Compare value for event detection
CMP_EVENT1	15:8		0x02	Compare value for event detection
CMP_EVENT0	7:0		0x03	Compare value for event detection

REG_RST offset = 0x008

Field	Bits	Type	Default	Description
PERIPH_RESET	31:0	RW	0x0	Reset for peripherals; use core-v-mcu_config 'Peripheral clock enable masks' for bit p

Notes:

Access type	Description
RW	Read & Write
RO	Read Only
RC	Read & Clear after read
WO	Write Only
WC	Write Clears (value ignored; always writes a 0)
WS	Write Sets (value ignored; always writes a 1)
RW1S	Read & on Write bits with 1 get set, bits with 0 left unchanged
RW1C	Read & on Write bits with 1 get cleared, bits with 0 left unchanged
RW0C	Read & on Write bits with 0 get cleared, bits with 1 left unchanged

UDMA_UART

Memory address: UDMA_CH_ADDR_UART('UDMA_CH_ADDR_UART)

Basic UART driven by UDMA system

RX_SADDR offset = 0x00

Field	Bits	Type	Default	Description
SADDR	11:0	RW		Address of receive buffer on write; current address on read

RX_SIZE offset = 0x04

Field	Bits	Type	Default	Description
SIZE	15:0	RW		Size of receive buffer on write; bytes left on read

RX_CFG offset = 0x08

Field	Bits	Type	Default	Description
CLR	6:6	WO		Clear the receive channel
PENDING	5:5	RO		Receive transaction is pending
EN	4:4	RW		Enable the receive channel
CONTINUOUS	0:0	RW		0x0: stop after last transfer for channel 0x1: after last transfer for channel, reload buffer size and start address and restart channel

TX_SADDR offset = 0x10

Field	Bits	Type	Default	Description
SADDR	11:0	RW		Address of transmit buffer on write; current address on read

TX_SIZE offset = 0x14

Field	Bits	Type	Default	Description
SIZE	15:0	RW		Size of receive buffer on write; bytes left on read

TX_CFG offset = 0x18

Field	Bits	Type	Default	Description
CLR	6:6	WO		Clear the transmit channel
PENDING	5:5	RO		Transmit transaction is pending
EN	4:4	RW		Enable the transmit channel
CONTINUOUS	0:0	RW		0x0: stop after last transfer for channel 0x1: after last transfer for channel, reload buffer size and start address and restart channel

STATUS offset = 0x20

Field	Bits	Type	Default	Description
RX_BUSY	1:1	RO		0x1: receiver is busy

Field	Bits	Type	Default	Description
TX_BUSY	0:0	RO		0x1: transmitter is busy

UART_SETUP offset = 0x24

Field	Bits	Type	Default	Description
DIV	31:16	RW		
EN_RX	9:9	RW		Enable the reciever
EN_TX	8:8	RW		Enable the transmitter
RX_CLEAN_FIFO	5:5	RW		Empty the receive FIFO
RX_POLLING_EN	4:4	RW		Enable polling mode for receiver
STOP_BITS	3:3	RW		0x0: 1 stop bit 0x1: 2 stop bits
BITS	2:1	RW		0x0: 5 bit transfers 0x1: 6 bit transfers 0x2: 7 bit transfers 0x3: 8 bit transfers
PARITY_EN	0:0	RW		Enable parity

ERROR offset = 0x28

Field	Bits	Type	Default	Description
PARITY_ERR	1:1	RC		0x1 indicates parity error; read clears the bit
OVERFLOW_ERR	0:0	RC		0x1 indicates overflow error; read clears the bit

IRQ_EN offset = 0x2C

Field	Bits	Type	Default	Description
ERR_IRQ_EN	1:1	RW		Enable the error interrupt
RX_IRQ_EN	0:0	RW		Enable the receiver interrupt

VALID offset = 0x30

Field	Bits	Type	Default	Description
RX_DATA_VALID	0:0	RO		Cleared when RX_DATA is read

DATA offset = 0x34

Field	Bits	Type	Default	Description
RX_DATA	7:0	RO		Receive data; reading clears RX_DATA_VALID

Notes:

Access type	Description
RW	Read & Write
RO	Read Only
RC	Read & Clear after read
WO	Write Only
WC	Write Clears (value ignored; always writes a 0)
WS	Write Sets (value ignored; always writes a 1)
RW1S	Read & on Write bits with 1 get set, bits with 0 left unchanged
RW1C	Read & on Write bits with 1 get cleared, bits with 0 left unchanged
RW0C	Read & on Write bits with 0 get cleared, bits with 1 left unchanged

UDMA_I2CM

Memory address: UDMA_CH_ADDR_I2CM(‘UDMA_CH_ADDR_I2CM)

The actions of the I2C controller are controlled using a sequence of commands that are present in the transmit buffer. Therefore, to use the I2C controller the software must assemble the appropriate sequence of commands in a buffer, and use the UDMA to send the buffer to the I2C contoller. And because the UDMA handles data buffers and interrupts, it is important to understand how to operate the UDMA controller.

I2C Command	Value	Description
CMD_START	0x00	Issue an I2C Start sequence
CMD_STOP	0x20	Issue an I2C Stop seqeuence
CMD_RD_ACK	0x40	Read a byte and send an ACK to the device byte read from device is stored in receive buffer ACK implies that the next command will be another read
CMD_RD_NACK	0x60	Read a byte and send a NACK to the device byte read from device is stored in receive buffer NACK implies that the next command will be a Stop (or perhaps Start)
CMD_WR	0x80	Write the next byte in the transmit buffer to the I2C device
CMD_WAIT	0xA0	Next byte specifies number of I2C clocks to wait before procceeding to next command

I2C Command	Value	Description
CMD_RPT	0xC0	Next two bytes are the repeat count and command to be repeated Typical use case would be to read N_BYTES: . . . CMD_RPT, N_BYTES-1, CMD_RD_ACK, CMD_RD_NAK . . .
CMD_CFG	0xE0	Next two bytes in the transmit buffer are the MSB and LSB of the clock divider I2C clock frequency is peripheral clock frequency deviced by divisor
CMD_WAIT_EV	0x10	(Needs more investigation)

RX_SADDR offset = 0x00

Field	Bits	Type	Default	Description
SADDR	11:0	RW		Address of receive buffer on write; current address on read

RX_SIZE offset = 0x04

Field	Bits	Type	Default	Description
SIZE	15:0	RW		Size of receive buffer on write; bytes left on read

RX_CFG offset = 0x08

Field	Bits	Type	Default	Description
CLR	6:6	WO		Clear the receive channel
PENDING	5:5	RO		Receive transaction is pending
EN	4:4	RW		Enable the receive channel
CONTINUOUS	0:0	RW		0x0: stop after last transfer for channel 0x1: after last transfer for channel, reload buffer size and start address and restart channel

TX_SADDR offset = 0x10

Field	Bits	Type	Default	Description
SADDR	11:0	RW		Address of transmit buffer on write; current address on read

TX_SIZE offset = 0x14

Field	Bits	Type	Default	Description
SIZE	15:0	RW		Size of receive buffer on write; bytes left on read

TX_CFG offset = 0x18

Field	Bits	Type	Default	Description
CLR	6:6	WO		Clear the transmit channel
PENDING	5:5	RO		Transmit transaction is pending
EN	4:4	RW		Enable the transmit channel
CONTINUOUS	0:0	RW		0x0: stop after last transfer for channel 0x1: after last transfer for channel, reload buffer size and start address and restart channel

STATUS offset = 0x20

Field	Bits	Type	Default	Description
AL	1:1	RO		Always returns 0
BUSY	0:0	RO		Always returns 0

SETUP offset = 0x24

Field	Bits	Type	Default	Description
RESET	0:0	RW		Reset I2C controller

Notes:

Access type	Description
RW	Read & Write
RO	Read Only
RC	Read & Clear after read
WO	Write Only
WC	Write Clears (value ignored; always writes a 0)
WS	Write Sets (value ignored; always writes a 1)
RW1S	Read & on Write bits with 1 get set, bits with 0 left unchanged
RW1C	Read & on Write bits with 1 get cleared, bits with 0 left unchanged
RW0C	Read & on Write bits with 0 get cleared, bits with 1 left unchanged

UDMA_QSPI

Memory address: UDMA_CH_ADDR_QSPI(‘UDMA_CH_ADDR_QSPI)

The actions of the QSPI controller are controlled using a sequence of commands that are present in the transmit buffer. Therefore, to use theQSPI controller the software must assemble the appropriate sequence of commands in a buffer, and use the UDMA to send the buffer to the QSPI contoller. And because the UDMA handles data buffers and interrupts, it is important to understand how to operate the UDMA controller.

Code	Command/Field	Bits	Description
0x0	SPI_CMD_CFG		Sets the configuration for the SPI Master IP
	CLKDIV	7:0	Sets the clock divider value
	CPHA	8:8	Sets the clock phase: 0x0: 0x1:
	CPOL	9:9	Sets the clock polarity: 0x0: 0x1:
	SPI_CMD	31:28	Command to execute (0x0)
0x1	SPI_CMD_SOT		Sets the Chip Select (CS)
	CS	1:0	Sets the Chip Select (CS): 0x0: select csn0 0x1: select csn1 0x2: select csn2 0x3: select csn3
	SPI_CMD	31:28	Command to execute (0x1)
0x2	SPI_CMD_SEND_CMD		Transmits up to 16bits of data sent in the command
	DATA_VALUE	15:0	Sets the command to send. MSB must always be at bit 15 if cmd size is less than 16
	DATA_SIZE	19:16	N-1, where N is the size in bits of the command to send
	LSB	26:26	Sends the data starting from LSB
	QPI	27:27	Sends the command using QuadSPI
	SPI_CMD	31:28	Command to execute (0x2)
0x4	SPI_CMD_DUMMY		Receives a number of dummy bits which are not sent to the RX interface
	DUMMY_CYCLE	21:16	Number of dummy cycles to perform
	SPI_CMD	31:28	Command to execute (0x4)
0x5	SPI_CMD_WAIT		Waits for an external event to move to the next instruction
	EVENT_ID_CYCLE_COUNT	6:0	External event id or Number of wait cycles
	WAIT_TYPE	9:8	Type of wait: 0x0: wait for and soc event selected by EVENT_ID

Code	Command/Field	Bits	Description
			0x1: wait CYCLE_COUNT cycles 0x2: rfu 0x3: rfu
	SPI_CMD	31:28	Command to execute (0x5)
0x6	SPI_CMD_TX_DATA		Sends data (max 256Kbits)
	WORD_NUM	15:0	N-1, where N is the number of words to send (max 64K)
	WORD_SIZE	20:16	N-1, where N is the number of bits in each word
	WORD_PER_TRANSF	22:21	Number of words transferred from SRAM at each transfer 0x0: 1 word per transfer 0x1: 2 words per transfer 0x2: 4 words per transfer
	LSB	26:26	0x0: MSB first 0x1: LSB first
	QPI	27:27	0x0: single bit data 0x1: quad SPI mode
	SPI_CMD	31:28	Command to execute (0x6)
0x7	SPI_CMD_RX_DATA		Receives data (max 256Kbits)
	WORD_NUM	15:0	N-1, where N is the number of words to send (max 64K)
	WORD_SIZE	20:16	N-1, where N is the number of bits in each word
	WORD_PER_TRANSF	22:21	Number of words transferred from SRAM at each transfer 0x0: 1 word per transfer 0x1: 2 words per transfer 0x2: 4 words per transfer
	LSB	26:26	0x0: MSB first 0x1: LSB first
	QPI	27:27	0x0: single bit data 0x1: quad SPI mode
	SPI_CMD	31:28	Command to execute (0x7)
0x8	SPI_CMD_RPT		Repeat the commands until RTP_END for N times
	RPT_CNT	15:0	Number of repeat iterations (max 64K)
	SPI_CMD	31:28	Command to execute (0x8)
0x9	SPI_CMD_EOT		Clears the Chip Select (CS)
	EVENT_GEN	0:0	Enable EOT event: 0x0: disable 0x1: enable

Code	Command/Field	Bits	Description
	SPI_CMD	31:28	Command to execute (0x9)
0xA	SPI_CMD_RPT_END		End of the repeat loop command
	SPI_CMD	31:28	Command to execute (0xA)
0xB	SPI_CMD_RX_CHECK		Checks up to 16 bits of data against an expected value
	COMP_DATA	15:0	Data to compare
	STATUS_SIZE	19:16	N-1, where N is the size in bits of the word to read
	CHECK_TYPE	25:24	How to compare: 0x0: compare bit by bit 0x1: compare only ones 0x2: compare only zeros
	LSB	26:26	0x0: Receieved data is LSB first 0x1: Received data is MSB first
	QPI	27:27	0x0: single bit data 0x1: quad SPI mode
	SPI_CMD	31:28	Command to execute (0xB)
0xC	SPI_CMD_FULL_DUPL		Activate full duplex mode
	DATA_SIZE	15:0	N-1, where N is the number of bits to send (max 64K)
	LSB	26:26	0x0: Data is LSB first 0x1: Data is MSB first
	SPI_CMD	31:28	Command to execute (0xC)
0xD	SPI_CMD_SETUP_UCA		Sets address for uDMA tx/rx channel
	START_ADDR	20:0	Address of start of buffer
	SPI_CMD	31:28	Command to execute (0xD)
0xE	SPI_CMD_SETUP_UCS		Sets size and starts uDMA tx/rx channel
	SIZE		N-1, where N is the number of bytes to transfer (max size depends on the TRANS_SIZE parameter)
	WORD_PER_TRANSF	26:25	Number of words from SRAM for each transfer: 0x0: 1 word per transfer 0x1: 2 words per transfer 0x2: 4 words per transfer
	TX_RXN	27:27	Selects TX or RX channel: 0x0: RX channel 0x1: TX channel
	SPI_CMD	31:28	Command to execute (0xE)

Code	Command/Field	Bits	Description

RX_SADDR offset = 0x00

Field	Bits	Type	Default	Description
SADDR	11:0	RW		Address of receive buffer on write; current address on read

RX_SIZE offset = 0x04

Field	Bits	Type	Default	Description
SIZE	15:0	RW		Size of receive buffer on write; bytes left on read

RX_CFG offset = 0x08

Field	Bits	Type	Default	Description
CLR	6:6	WO		Clear the receive channel
PENDING	5:5	RO		Receive transaction is pending
EN	4:4	RW		Enable the receive channel
DATASIZE	2:1	RW	0x02	Controls uDMA address increment 0x00: increment address by 1 (data is 8 bits) 0x01: increment address by 2 (data is 16 bits) 0x02: increment address by 4 (data is 32 bits) 0x03: increment address by 0
CONTINUOUS	0:0	RW		0x0: stop after last transfer for channel 0x1: after last transfer for channel, reload buffer size and start address and restart channel

TX_SADDR offset = 0x10

Field	Bits	Type	Default	Description
SADDR	11:0	RW		Address of transmit buffer on write; current address on read

TX_SIZE offset = 0x14

Field	Bits	Type	Default	Description
SIZE	15:0	RW		Size of receive buffer on write; bytes left on read

TX_CFG offset = 0x18

Field	Bits	Type	Default	Description
CLR	6:6	WO		Clear the transmit channel
PENDING	5:5	RO		Transmit transaction is pending
EN	4:4	RW		Enable the transmit channel
DATASIZE	2:1	WO	0x02	Controls uDMA address increment (Reads as 0x00) 0x00: increment address by 1 (data is 8 bits) 0x01: increment address by 2 (data is 16 bits) 0x02: increment address by 4 (data is 32 bits) 0x03: increment address by 0
CONTINUOUS	0:0	RW		0x0: stop after last transfer for channel 0x1: after last transfer for channel, reload buffer size and start address and restart channel

CMD_SADDR offset = 0x20

Field	Bits	Type	Default	Description
SADDR	31:0	RW	0x00	Address of command memory buffer: Read: current address until transfer is complete, then 0x00 Write: start addrress of command buffer

CMD_SIZE offset = 0x24

Field	Bits	Type	Default	Description
SIZE	20:0			Buffer size in bytes (1MByte maximum) Read: bytes remaining to be transferred Write: number of bytes to transmit

CMD_CFG offset = 0x28

Field	Bits	Type	Default	Description
CLR	6:6	WO		Clear the transmit channel
PENDING	5:5	RO		Transmit transaction is pending
EN	4:4	RW		Enable the transmit channel
DATASIZE	2:1	WO	0x02	Controls uDMA address increment (Reads as 0x02)

Field	Bits	Type	Default	Description
CONTINUOUS	0:0	RW		0x00: increment address by 1 (data is 8 bits) 0x01: increment address by 2 (data is 16 bits) 0x02: increment address by 4 (data is 32 bits) 0x03: increment address by 0 0x0: stop after last transfer for channel 0x1: after last transfer for channel, reload buffer size and start address and restart channel

STATUS offset = 0x30

Field	Bits	Type	Default	Description
BUSY	1:0	RO		Status: 0x00: STAT_NONE 0x01: STAT_CHECK (matched) 0x02: STAT_EOL (end of loop)

Notes:

Access type	Description
RW	Read & Write
RO	Read Only
RC	Read & Clear after read
WO	Write Only
WC	Write Clears (value ignored; always writes a 0)
WS	Write Sets (value ignored; always writes a 1)
RW1S	Read & on Write bits with 1 get set, bits with 0 left unchanged
RW1C	Read & on Write bits with 1 get cleared, bits with 0 left unchanged
RW0C	Read & on Write bits with 0 get cleared, bits with 1 left unchanged

UDMA_SDIO

Memory address: UDMA_CH_ADDR_SDIO(‘UDMA_CH_ADDR_SDIO)

RX_SADDR offset = 0x00

Field	Bits	Type	Default	Description
SADDR	31:0	RW		Address of receive memory buffer: - Read: value of pointer until transfer is over, then 0 - Write: set memory buffer start address

RX_SIZE offset = 0x04

Field	Bits	Type	Default	Description
SIZE	15:0	RW		Buffer size in bytes (1MB max) - Read: bytes remaining until transfer complete - Write: set number of bytes to transfer

RX_CFG offset = 0x08

Field	Bits	Type	Default	Description
CLR	5:5	WO		Clear the receive channel
PENDING	5:5	RO		Receive transaction is pending
EN	4:4	RW		Enable the receive channel
DATASIZE	2:1	RW	0x02	Controls uDMA address increment 0x00: increment address by 1 (data is 8 bits) 0x01: increment address by 2 (data is 16 bits) 0x02: increment address by 4 (data is 32 bits) 0x03: increment address by 0
CONTINUOUS	0:0	RW		0x0: stop after last transfer for channel 0x1: after last transfer for channel, reload buffer size and start address and restart channel

TX_SADDR offset = 0x10

Field	Bits	Type	Default	Description
SADDR	11:0	RW		Address of transmit memory buffer: - Read: value of pointer until transfer is over, then 0 #NAME?

TX_SIZE offset = 0x14

Field	Bits	Type	Default	Description
SIZE	15:0	RW		Buffer size in bytes (1MB max) #NAME? #NAME?

TX_CFG offset = 0x18

Field	Bits	Type	Default	Description
CLR	5:5	WO		Clear the transmit channel
PENDING	5:5	RO		Transmit transaction is pending
EN	4:4	RW		Enable the transmit channel
DATASIZE	2:1	WO	0x02	Controls uDMA address increment (Reads as 0x00) 0x00: increment address by 1 (data is 8 bits) 0x01: increment address by 2 (data is 16 bits) 0x02: increment address by 4 (data is 32 bits) 0x03: increment address by 0
CONTINUOUS	0:0	RW		0x0: stop after last transfer for channel 0x1: after last transfer for channel, reload buffer size and start address and restart channel

CMD_OP offset = 0x20

Field	Bits	Type	Default	Description
CMD_OP	13:8	WO		SDIO command opcode
CMD_RSP_TYPE	2:0	WO		Response type: 0x0: No response 0x1: 48 bits with CRC 0x2: 48 bits with no CRC 0x3: 136 bits 0x4: 48 bits with BUSY check

CMD_ARG offset = 0x24

Field	Bits	Type	Default	Description
CMD_ARG	31:0	WO		Argument to be sent with the command

DATA_SETUP offset = 0x28

Field	Bits	Type	Default	Description
BLOCK_SIZE	25:16	W		Block size
BLOCK_NUM	15:8	W		Number of blocks
DATA_QUAD	2:2	W		Use QUAD mode
DATA_RWN	1:1	W		Set transfer direction: 0x0: write 0x1: read
DATA_EN	0:0	W		Enable data transfer for current command

START offset = 0x2C

Field	Bits	Type	Default	Description
START	0:0	W		Start SDIO transfer

RSP0 offset = 0x30

Field	Bits	Type	Default	Description
RSP0	31:0	R		Bytes[3:0] of the response

RSP1 offset = 0x34

Field	Bits	Type	Default	Description
RSP1	31:0	R		Bytes[7:4] of the response

RSP2 offset = 0x38

Field	Bits	Type	Default	Description
RSP2	31:0	R		Bytes[11:8] of the response

RSP3 offset = 0x3C

Field	Bits	Type	Default	Description
RSP3	31:0	R		Bytes[15:12] of the response

CLK_DIV offset = 0x40

Field	Bits	Type	Default	Description
VALID	8:8	RW		??
CLK_DIV	7:0	RW		Clock divisor

STATUS offset = 0x44

Field	Bits	Type	Default	Description
CMD_ERR_STATUS	21:16	R		Error status of commnd transfer: 0x0: no error 0x1: response timeout

Field	Bits	Type	Default	Description
				0x2: response wrong direction 0x4: response BUSy timeout
ERROR	1:1	RWC		Indicate error in command or data 0x0: no error 0x1: error
EOT	0:0	RWC		Indicate the end of transfer (command or data) 0x0: not ended 0x1: ended

Notes:

Access type	Description
RW	Read & Write
RO	Read Only
RC	Read & Clear after read
WO	Write Only
WC	Write Clears (value ignored; always writes a 0)
WS	Write Sets (value ignored; always writes a 1)
RW1S	Read & on Write bits with 1 get set, bits with 0 left unchanged
RW1C	Read & on Write bits with 1 get cleared, bits with 0 left unchanged
RW0C	Read & on Write bits with 0 get cleared, bits with 1 left unchanged

UDMA_CAMERA

Memory address: UDMA_CH_ADDR_CAMERA(‘UDMA_CH_ADDR_CAMERA)

RX_SADDR offset = 0x00

Field	Bits	Type	Default	Description
SADDR	31:0	RW		Address of receive memory buffer: - Read: value of pointer until transfer is over, then 0 - Write: set memory buffer start address

RX_SIZE offset = 0x04

Field	Bits	Type	Default	Description
SIZE	15:0	RW		Buffer size in bytes (1MB max) - Read: bytes remaining until transfer complete - Write: set number of bytes to transfer

RX_CFG offset = 0x08

Field	Bits	Type	Default	Description
CLR	5:5	WO		Clear the receive channel
PENDING	5:5	RO		Receive transaction is pending
EN	4:4	RW		Enable the receive channel
DATASIZE	2:1	RW	0x02	Controls uDMA address increment 0x00: increment address by 1 (data is 8 bits) 0x01: increment address by 2 (data is 16 bits) 0x02: increment address by 4 (data is 32 bits) 0x03: increment address by 0
CONTINUOUS	0:0	RW		0x0: stop after last transfer for channel 0x1: after last transfer for channel, reload buffer size and start address and restart channel

CFG_GLOB offset = 0x20

Field	Bits	Type	Default	Description
EN	31:31	RW		Enable data RX from camera interface Enable/disable only happens at start of frame 0x0: disable 0x1: enable
SHIFT	14:11			Number of bits to right shift final pixel value Note: not used if FORMAT == BYPASS
FORMAT	10:8			Input frame format: 0x0: RGB565 0x1: RGB555 0x2: RGB444 0x4: BYPASS_LITTLEEND 0x5: BYPASS_BIGEND
FRAMEWINDOW_EN	7:7			Windowing enable: 0x0: disable 0x1: enable
FRAMEDROP_VAL	6:1			How many frames dropped between receievd frame
FRAMEDROP_EN	0:0			Frame dropping enable: 0x0: disable frame dropping 0x1: enable frame dropping

CFG_LL offset = 0x24

Field	Bits	Type	Default	Description
FRAMEWINDOW_LLY	31:16			Y coordinate of lower left corner of window
FRAMEWINDOW_LLX	15:0			X coordinate of lower left corner of window

CFG_UR offset = 0x28

Field	Bits	Type	Default	Description
FRAMEWINDOW_URY	31:16			Y coordinate of upper right corner of window
FRAMEWINDOW_URX	15:0			X coordinate of upper right corner of window

CFG_SIZE offset = 0x2C

Field	Bits	Type	Default	Description
ROWLEN	31:16			N-1 where N is the number of horizontal pixels (used in window mode)

CFG_FILTER offset = 0x30

Field	Bits	Type	Default	Description
R_COEFF	23:16			Coefficent that multiplies R component Note: not used if FORMAT == BYPASS
G_COEFF	15:8			Coefficent that multiplies G component Note: not used if FORMAT == BYPASS
B_COEFF	15:8			Coefficent that multiplies B component Note: not used if FORMAT == BYPASS

VSYNC_POLARITY offset = 0x34

Field	Bits	Type	Default	Description
VSYNC_POLARITY	0:0	R/W		Set vsync polarity: 0x0: Active low 0x1: Active high

Notes:

Access type	Description
RW	Read & Write
RO	Read Only

Access type	Description
RC	Read & Clear after read
WO	Write Only
WC	Write Clears (value ignored; always writes a 0)
WS	Write Sets (value ignored; always writes a 1)
RW1S	Read & on Write bits with 1 get set, bits with 0 left unchanged
RW1C	Read & on Write bits with 1 get cleared, bits with 0 left unchanged
RW0C	Read & on Write bits with 0 get cleared, bits with 1 left unchanged