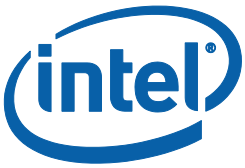

Rocky Lake System FPGA

Architecture and Programming Guide

March, 2011

Revision 0.9.2

Contacts: Michael Konow

Udo Grüning

Matthias Steidl

Contents

Figures

Related Documents

1. Rock Creek System FPGA, External-architecture Specification (EAS), Revision 0.5, June 2008
2. PCI Express Base Specification, Revision 1.1, March 2005
3. Xilinx Virtex-5 Integrated Endpoint Block of PCIE Designs User Guide, Revision 1.3, June 2008
4. Xilinx Virtex-5 FPGA User Guide, Revision 4.2 (UG190), May 2008
5. Xilinx Virtex-5 FPGA Data Sheet: DC and and Switching Characteristics, Revision 4.6 (DS202), June 2008
6. Xilinx XAPP1052: Bus Master DMA Reference Design for the Xilinx Endpoint Block Plus Core for PCI Express, v1.1, August 2008
7. PCI Local Bus Specification, Revision 3.0, February 2004
8. Rock Creek Mesh Interface External-Architecture Specification (EAS), Revision 1.0
9. MCEMU Transactor Implementation, Revision 0.5, December 2008
10. Xilinx Virtex-5 FPGA Embedded Tri-Mode Ethernet MAC
11. Xilinx Virtex-5 FPGA Embedded Tri-Mode Ethernet MAC Wrapper v1.6
12. Xilinx LogiCORE IP Ethernet Statistics v3.3 User Guide

1 Introduction

The Rocky Lake system FPGA is the bridge between the SCC silicon and the management console PC (MCPC). It provides a PCIe connection to the PC allowing the user to setup the environment, control the programs that are running on SCC or develop MCPC applications that can communicate with the SCC silicon.

Beside this bridge functionality the FPGA acts as an I/O hub providing I/O capabilities, which can be used by the silicon directly. The major ones are four stand-alone Ethernet ports and two SATA ports. SW running on SCC can directly access the network ports to communicate with other servers via internet and can use the SATA disk drives to enable fast local storage.

For these functions customized drivers must be developed for the OS that runs on the SCC cores. Drivers are available for the standard SCC Linux.

Figure 1: Rocky Lake FPGA Overview

The following chapter describes the architecture of the different blocks inside the FPGA mainly focusing on the functionality. In chapter 3, the implementation details are described with a focus on specifics that need to be taken into account for implementation inside of a FPGA. A comprehensive list of all FPGA registers and their functions can be found at the end of the document.

2 Architecture

2.1 General Information

2.1.1 Data Types and Interfaces

The Rock Creek 2D mesh uses flits, a 144bits wide data type to transfer information between agents. One SCC packet can consist of 1, 2 or 3 flits. For details see [8]. For simplicity, the SIF defines a new, 384bit wide packet data type which maps most flit fields 1:1 as described in the SIF router section, but also dropped some of the RC mesh specific fields for simplicity. The Mesh Interface Unit (MIU) translates flits into FPGA packets and vice versa.

Another data type found in the FPGA is PCIe transaction frames. The PCIe frame header information is not used outside the host interface and further down the line to RC. Only the data payload of the PCIe transfers is further used inside the SIF and interpreted as a FPGA packet. The PCIe and SIF do not share an address range and are logically separate.

Ethernet and SATA frames are also wrapped into FPGA packets for transportation through the FPGA.

Figure 2: Datatypes and Interfaces

2.1.2 Address translation

SCC and the FPGA use 34bit addresses which are mostly mapped 1:1. The exception being the upper 2 bits in the SCC addresses which are used in the MIU to determine the destination ID inside the FPGA for request transactions received by the FPGA from SCC.

The upper 10 bits (33:24) of the RC System address are configurable via the RC LUTs. It is thereby possible to select a different FPGA destination for every 16 MB block represented by one RC LUT entry for all 48 cores.

Address bits [33:31] indicate the destination:

Destination	Address
Host PC	Addr[33:31] = default
FPGA MIU (debug only)	Addr[33:31] = ‘h1
FPGA Register File	Addr[33:31] = ‘h2

2.1.3 Transaction ID handling

For all requests from the FPGA to the SCC, the response transactions don't have the original address included. Therefore transaction IDs will be used to select the right destination within the FPGA. Each request agent, the MCPC, the four eMAC modules and the two SATA modules have their own transaction ID range.

Transaction ID	Agent
0 … 63	Host PC
64 … 79	eMAC #0
80 … 95	eMAC #1
96 … 111	eMAC #2
112 … 127	eMAC #3
128 … 143	SATA #0
144 … 159	SATA #1
160 … 255	Reserved

Each agent generates the transaction IDs within his own range and increments the IDs with each request it sends to SCC. When an agent receives the response, it checks the ID against its transfer counter. During normal operation, the value is ID(n) = ID(n-1) + 1. In case of a mismatch, an out-of-order response has been found. In this case an error status bit will be set in the in register FPGAStatus(0).

2.2 Mesh Interface Unit (MIU)

The mesh interface unit receives and generates SCC packets. Because the physical link is just half the size of the on-die mesh link, the received data must be assembled into the 144 bit SCC flit format using buffers which reside in the clock domain of the physical interface. The synchronization of the data to the slower system clock domain is done via clock-crossing FIFOs (CCFs). On the SCC side this block receives and generates SCC packets and does the credit handling as described in the SCC EAS document. On the other side the block generates and receives FPGA packets from the different sources of the router. The FPGA packet format is very similar to the SCC packet format and described in the implementation chapter.

2.3 Router

The router does simple blocking one-direction packet forwarding based on the destination field. Only one packet can pass the router at a time per output port. There is a Round-Robin arbiter associated with every output port managing the priorities. A router packet consists of 48 bytes and contains the following fields:

- data: 256bit data (one SCC cache-line)
- byteenable: 8bit byte enable defining valid bytes in non-cacheline packets
- transid: 8bit transaction identification number
- srcid: 8bit source id, indicating the FPGA packet sender
- destid: 8bit destination id, indicating the FPGA packet destination
- addr: 34bit physical target address
- cmd: 12bit command type field
- rc_id: 8bit SCC route id field (x,y tile coordinates)
- rc_subid: 3bit SCC srcid/destid field
- reserved: 39bit reserved for future use

Bytes	Contents	Packet Bits
47-44	reserved[38:7]	packet[383:352]
43-40	{reserved[6:0], rc_subid[2:0], rc_id[7:0], cmd[11:0], addr[33:32]}	packet[351:320]
39-36	addr[31:0]	packet[319:288]
35-32	{destid[7:0], srcid[7:0], transid[7:0], byteenable[7:0]}	packet[287:256]
31-28	data[255:224]	packet[255:224]
	…	…
7-4	data[63:32]	packet[063:032]
3-0	data[31:0]	packet[031:000]

The following happens for a packet transfer:

1. An agent who wants to send a packet must assert his valid out signal.
2. The router expects the destid field. If the destination port is free and has a grant for the input, the packet is accepted into the router and forwarded to the destination port.
3. If multiple agents want to transfer packets to free ports at the same time, the Round-Robin engine will determine the priority.
4. Once the destination agent has consumed the packet, the router outport is idle again, ready to process the next packet.

2.4 PCIe Interface and DMA

The PCIe establishes the connection between the SCC platform and the MCPC. It is based on the PCIe endpoint IP and a DMA application note design from Xilinx. Packets that are sent from or to the MCPC get stored in the SCEMI buffers which are 64kB each. Flow control is implemented in a way that it generates back-pressure into the SCC or to the MCPC to prevent buffer overflow.

The data transfer between the FPGA and the MCPC can be done in two different modes. In PIO mode single dwords are transferred while in DMA mode larger chunks of packets can be transferred with data rates beyond 200MB/s.

Transfers to/from memory or register space are always controlled by the MCPC and needs to be done based on FPGA packets. There is no way to transfer “raw” accesses to the FPGA, as it can handle FPGA packets only. The packets need to be assembled on the MCPC before transferring them via PIO or DMA to the FPGA.

2.4.1 MCPC Linux Driver

The driver manages the data transfer with the FPGA HW via PCIe. It implements the flow control, configures the DMA transfers and handles the data exchange with the application SW..

The following section describes how the driver handles the initialization and read and write transactions. The register references are to Xilinx IP block registers (XLX_) or system FPGA registers (RCK_). The Xilinx DMA registers are seen from the HW perspective. This means that the read registers for the DMA engine read from MCPC memory and the write registers for DMA engine write to MCPC memory; but the system view of reading and writing data from the point of view of the application SW is vice versa.

DMA transfers as described here are done with TLP size of 4 (= 16 Bytes), thus for any number of FPGA packets (48 Bytes each) a whole number of TLP transfers is done. No fragmented packets are sent. However, this approach delivers suboptimal performance.

Better performance can be achieved by using a larger TLP size (only observed power of 2 numbers work reliably), but care must be taken to pass only complete, non-fragmented packets to the application SW. Good performance is achievable with 2 consecutive DMA transfers. The first DMA transfer is with the maximum TLP size possible on the PCIe bus (as defined by HW /

BIOS), and the second DMA transfer is with TLP size that results in the highest whole number of TLPs for the remaining data in the FIFO.

Driver Initialization

1. Use vendor/device ID: 8086/c148
2. Disable ASPM (Active State Power Management, ref PCIe standard) in PCIe root port of Rocky Lake connection (capability ID 0x10):

   Found PCIe Capability structure @ 0x90
   Link Control Reg @ 0xa0: 0x30410001, setting to 0x30410000
3. Map PCI BAR0 and read / check RCK_ID0_REG (bitstream ID), XLX_DCSR1, XLX_DLWSTAT, XLX_DLTRSSTAT.
4. Keep max payloadsize of PCIe endpoint in XLX_DLTRSSTAT[10:8] for later. Access to PCI BAR 0 will be needed later for DMA configuration.
5. SCEMI reset by toggling RCK_CONFIG_SOFTRESET bit in RCK_CONFIG_REG and setting RCK_TRNCT5_INIT bit in RCK_TRNCT5_REG
6. Switch device to MSI interrupt mode, register ISR that resets global waitForDMA flag.

DMA Read (SCC to MCPC)

1. Read FIFO status from RCK_TRNCT4_REG[15:0], check for error in bit [13] → fail, bailout
2. DataAvail = RCK_TRNCT4_REG[14:0] * 2 // in DWords
3. If (DataAvail == 0) return // nothing to do, no read data
4. Assert / DeAssert Reset bit in XLX_DCSR1[0]
5. Write physical address of DMA buffer (to be filled by HW) to XLX_WDMATLPA
6. TLP_size = 4 // (4 DWords), TLP_count = DataAvail / TLP_size, write to XLX_WDMATLPS and XLX_WDMATLPC respectively
7. Set waitForDMA = 1, then start DMA transfer by setting XLX_DDMACR[0]
8. While (waitForDMA) do_nothing();
9. DMA transfer is done when ISR got called and reset the global waitForDMA, data from hardware is now in DMA buffer to be passed to SCC library

DMA Write (MCPC to SCC)

1. Read FIFO status from RCK_TRNCT4_REG[31:16], check for error in bit [29] → fail, bailout
2. RoomAvail = RCK_TRNCT4_REG[30:16] * 2 // in DWords, no more than RoomAvail words can be transferred to HW w/o data loss
3. DataCount = Number of DWords to write
4. dwordsToWrite = min (RoomAvail, DataCount)
5. Assert / DeAssert Reset bit in XLX_DCSR1[0]
6. write physical address of DMA buffer (containing data to write to HW) to XLX_RDMATLPA
7. TLP_size = 4 // (4 DWords), TLP_count = DataAvail / TLP_size, write to XLX_RDMATLPS and XLX_RDMATLPC respectively
8. Set waitForDMA = 1, then start DMA transfer by setting XLX_DDMACR[16]
9. while (waitForDMA) do_nothing();
10. DMA transfer is done when ISR got called and reset the global waitForDMA

2.4.2 PCIe TX Packet Generator

The TX Packet Generator can be used to access large consecutive memory areas with a minimum of MCPC data transfer required. The module is located between the SCEMI TX FIFO (MIP) and the FPGA router. When it receives a packet with a packet generation command prefix, the content of this packet triggers a state machine that sends multiple packets to the router. The contents of these new packets are derived from the original packet and depend on the specific command prefix. Packets with a NOGEN prefix are simply forwarded to the router.

The Packet generator pauses the generation of request packets when the MIU RXFIFO or the MOP FIFO are almost full to prevent the system from being flooded with packets.

The Packet Generator auto-increments the transfer ID of all packets sent out. It uses TIDs within the MCPC range only.

Packet generator supports the following commands:

BLOCK Transaction: Generates up to 2^{32} packets based on the first one. For example, this can be used to read large chunks of DDR3 memory with just sending one request from the MCPC.

BCMC Transaction: Broadcasts the packet received from MCPC to all 4 memory controllers. For example, this can be used to copy an OS image into all 4 memory controllers.

BCTILE Transaction: Broadcasts the packet received to all 24 tiles. For example, this can be used to pull all resets in the tiles by sending just one packet from MCPC.

2.5 Interface to BMC

The FPGA has a dedicated interface to the Board Management Controller (BMC). This interface is used by the BMC firmware to initialize the FPGA as well as the SCC and therefore allows the realization of a self-booting platform which does not need a MCPC to boot. This interface is an asynchronous micro-processor interface with 16 bit data and address lnes. It provides a maximum bandwidth of ~10MB, which is much less than the PCIe interface to the MCPC. This may require an additional MCPC to be connected to the self-booting platform as well (e.g. downloading of large data chunks for debugging).

The interfaces to the MCPC and to the BMC can be used exclusively. There is not HW support to ensure a safe switch from one mode to the other during operation. The SW needs to ensure that no transactions are open, before switching between the modes.

Figure 3: Interface to BMC

2.6 eMAC

The Ethernet interfaces are built upon IP blocks from Xilinx. The Xilinx Ethernet MAC IP contains two eMAC modules, EMAC0 and EMAC1. The IP supports different physical interfaces and for the Rocky Lake system the Tri-Speed operation RGMII v1.3 interface will be used. These physical interfaces are connected to the external PHY modules on the board.

Internally all eMAC modules have a 4 Kbyte FIFO for the TX and RX directions and are connected to the FPGA router via the client interface. Due to the limitation of the internal buffers, jumbo frames won't be supported.

The conversion from internal FPGA packet format to the Ethernet FIFO format is done by a packetizer and de-packetizer. The register interface of the eMAC IP blocks is connected to the internal FPGA register bank to provide a unified way of accessing all internal registers. The registers are mapped into the FPGA register address space.

Sending and receiving of Ethernet frames to/from SCC cores is done via buffers in DDR3 memory space. The sending core writes its frames into the buffer, informs the FPGA HW that new frames are present so that the HW can pull the frames from the buffer and transfers them to the external Ethernet ports. When the HW receives frames on the external Ethernet port, it puts them into the according DDR3 memory buffer, tells the core that new frames are present and the receiving SCC core starts pulling the frames from the buffer. Signaling can be done sending interrupts to the SCC core. Alternatively the core can poll the head of the buffer to determine a change in the write index.

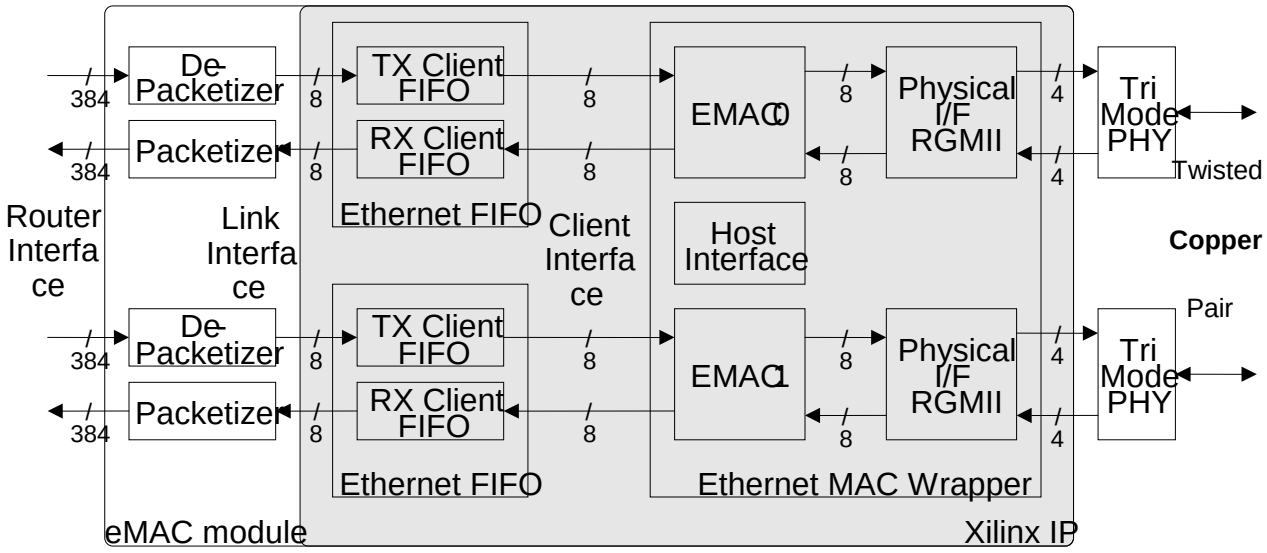

Figure 4: Overview of eMAC IP instance

2.6.1 Memory Buffer Structure

The receive and transmit buffers are build as ring buffers in DDR3 memory. The read and write indexes are stored in FPGA registers. The head address of the buffer is reserved to store copies of the read/write indexes so that the core can look them up by accessing the DDR3 memory instead of the real HW registers in the FPGA.

The buffer is organized in 32 byte chunks, which matches one cache line. The main reason for this is to optimize the data transfer between FPGA and SCC by transferring 32 byte data chunks only. Ethernet frames that are stored in the buffer always start on 32 byte boundaries. When the frame length is not a multiple of 32 bytes, the remaining bytes should be filled with 0.

The start address of the buffer can be somewhere in DDR3 memory or in the on-die MPB. The size of the buffer is limited to ~2 MB. In front of each frame the first two bytes are reserved for the frame length. The HW as well as the SW need to store the frame length there as the L/T field in the Ethernet frame header is usually used for type information. The frame length value must be set to the length of the Ethernet frame and does not include the two bytes of the frame length field itself.

Figure 5: Ethernet Frame Buffer Organization

2.6.2 Ethernet Network Setup

Each core can have multiple independent network interfaces, connected to each of the four eMAC modules. The eMAC modules contain up to 48 independent register sets which can be shared among all cores. This results in a maximum of 192 different network interfaces with individual MAC addresses for one SCC and provides a maximum of flexibility to the SW.

The SW driver is responsible for configuring and enabling the network ports. Each driver does this independently for his own register sets.

1. Disable flow control, enable transmitter, set full duplex mode, set MAC speed and filter mode in Xilinx IP
2. Set up ring buffer space in DDR3 memory

3. Initialize *start address, last index, write/read index, threshold, route* and *MAC address* in FPGA register set
4. Setup signaling mode (interrupt/polling)
5. Activate network port by setting enable bit

Determining the right MAC addresses can be done in different ways. The easiest way is to use the MAC Address Base Register in the FPGA and to calculate the MAC address by adding the core number to it. This address then needs to be written to the MAC address register. Using this approach requires a continuous range of MAC addresses reserved for SCC. The base address can be programmed by the BMC during boot-up of the system.

Another approach could be to store a list of MAC addresses somewhere in the OS image. The cores drivers then look them up and write them to the appropriate registers.

<add description of HW ports on the board and how they are assigned to the internal MAC modules>

Figure 6: MAC Address Assignment

2.6.3 Receiving Ethernet Frames

Frames that are received on the external Ethernet ports are first stored in the internal 4KB buffer of the eMAC IP. When the IP block is ready to provide the data, it signals a start of frame on the client interface. The HW then starts pulling out the data from the IP block, generating FPGA packets and writing them to the DDR3 memory.

Before the HW starts generating packets, the SW must enable the network port for this core. After reset it is disabled to allow the SW on the SCC core to configure the network interface properly and setup the buffer space before the HW starts writing packets to the buffer.

There are two ways for the SCC core to determine the existence of new frames in the buffer. The first is polling. The SCC core polls the head of the buffer to find out when the write index value has changed. It can then decide when it wants to start reading out first frames from the buffer. The second approach is to use signaling via interrupt. The HW would trigger an interrupt event when it has written a new frame to the buffer.

The core SW stops reading when the buffer is empty while the HW stops writing when the buffer is full. Usually the SW should be able to read out faster than the HW can write into the buffer. Nevertheless when the buffer in DDR3 memory overflows, the HW drops packets because no additional space exists in the FPGA. A bit in the status register section of the eMAC module will be set to indicate an overflow.

When the HW receives an Ethernet broadcast packet, it will send it to all enabled register sets of the physical interface.

1. FPGA HW receives start of new frame
2. Get MAC address from frame header
3. Compare MAC address with all available addresses stored in table
    1. If no match, packet gets dropped
    2. If match, get assigned core number from table
4. Check if network port for this core is enabled
    1. If enabled, continue processing of frame
    2. If not enabled, packet gets dropped
5. Put frame data in FPGA packets and write to according core buffer
    1. Write packet length to buffer
    2. If „end of frame“ update write index and copy value to head of buffer
6. Signal interrupt to core, if in interrupt mode
7. Core starts servicing interrupt or in if in polling mode detects update of write index and starts reading frame data
8. Core updates read index in FPGA to free up buffer space

The following state diagrams describe the control flow on the core side and in the HW

Figure 7: Receiving Ethernet Frames – Control Flow Core Driver

Figure 8: Receiving Ethernet Frames - Control Flow FPGA HW

The figure below illustrates the structure of the packetizer block. Details about the implementation of the block can be found in the implementation chapter.

Figure 9: Ethernet Packetizer Block diagram

2.6.4 Transmitting Ethernet Frames

Ethernet frames that should be sent out through the external network port are written to the transmit buffer by the SCC core. The HW pulls the frames from the transmit buffer by sending 32 byte read requests, unpacks the data and transmits them to the internal buffer of the eMAC IP.

Before the HW starts pulling data from the buffer, it must be enabled by setting the appropriate bit in the eMAC TXCTRL register. After reset all network ports are disabled.

The HW determines new packets in the buffer by looking for updates in the write index register. As soon as the value changes, the HW starts reading out new packets. The de-packetizer block extracts the frame data from the FPGA packets and transmits them to the eMAC IP block. When

the buffer is empty the HW stops reading and waits for the next index register update. When the buffer is full, the SCC core stops writing new packets to the buffer.

The eMAC IP block can signal to the HW when it is not ready to receive new data. In this case, the HW stalls until the block is ready again accepting new packets.

1. Core writes new frames to the memory buffer
2. Core updates write index in TXCTRL register to indicate presence of new frames
3. FPGA checks if network port is enabled
    1. If enabled, HW starts reading of frame data from buffer
    2. If not enabled, HW stalls
4. FPGA reads packet length from buffer
5. Frame data extracted from FPGA packets and send to eMAC module
    1. If last byte reached, „end-of-frame“ indication generated for eMAC
    2. Read index register gets updated
6. HW checks if other cores has data to transfer
    1. “Round-Robin” across all enabled cores – one frame per core

The following state diagrams describe the control flow on the core side and in the HW

Figure 10: Transmitting Ethernet Frames – Control Flow Core Driver

Figure 11: Transmitting Ethernet Frames - Control Flow FPGA HW

The figure below illustrates the structure of the de-packetizer block. Details about the implementation of the block can be found in the implementation chapter.

Figure 12: De-Packetizer Block Diagram

2.6.5 eMAC Driver Description

For the SCC Linux a driver that handles the transfer of Ethernet packets exists. The following section describes the steps that the driver takes to send and receive packets through the eMAC interfaces.

Please note that the write index registers in the RX controllers as well as the read index registers in the TX controllers are read only by the SW and can be written by the HW only.

2.6.5.1 Initialization Xilinx IP

The following description is an example for emac0 (see

POWER: SYSTEM Addr	Reg	Bits	Bit Field	Type	Description
08500	Slave_VMON1	[15:0]	1V0	R 0	Internal use only

POWER: SYSTEM Addr	Reg	Bits	Bit Field	Type	Description
		[31:16]	Unused	R 0	Unused
08504	Slave_VMON2	[15:0]	I_1V0	R 0	Internal use only
		[31:16]	Unused	R 0	Unused
08508	Slave_VMON3	[15:0]	1V1_VCCT	R 0	Internal use only
		[31:16]	Unused	R 0	Unused
0850C	Slave_VMON4	[15:0]	I_1V1_VCCT	R 0	Internal use only
		[31:16]	Unused	R 0	Unused
08510	Slave_VMON5	[15:0]	1V1_VCCA	R 0	Internal use only
		[31:16]	Unused	R 0	Unused
08514	Slave_VMON6	[15:0]	I_1V1_VCCA	R 0	Internal use only
		[31:16]	Unused	R 0	Unused
08518	Slave_VMON9	[15:0]	12V0R1	R 0	Internal use only
		[31:16]	Unused	R 0	Unused
0851C	Slave_VMON10	[15:0]	12V0R2	R 0	Internal use only
		[31:16]	Unused	R 0	Unused
08520	Slave_VMON11	[15:0]	5V0_IN	R 0	Internal use only
		[31:16]	Unused	R 0	Unused
08524	Slave_VMON12	[15:0]	3V3_IN	R 0	Internal use only
		[31:16]	Unused	R 0	Unused
08528	Master_VMON1	[15:0]	3V3	R 0	Internal use only
		[31:16]	Unused	R 0	Unused
0852C	Master_VMON2	[15:0]	I_3V3	R 0	Internal use only
		[31:16]	Unused	R 0	Unused
08530	Master_VMON5	[15:0]	1V65	R 0	Internal use only

POWER: SYSTEM Addr	Reg	Bits	Bit Field	Type	Description
		[31:16]	Unused	R 0	Unused
08534	Master_VMON6	[15:0]	1V65_ADJ	R 0	Internal use only
		[31:16]	Unused	R 0	Unused
08538	Master_VMON7	[15:0]	1V8_SB	R 0	Internal use only
		[31:16]	Unused	R 0	Unused
0853C	Master_VMON8	[15:0]	5V0_PWR	R 0	Internal use only
		[31:16]	Unused	R 0	Unused
08540	Master_VMON9	[15:0]	3V3_PWR	R 0	Internal use only
		[31:16]	Unused	R 0	Unused
08544	Master_VMON10	[15:0]	2V5	R 0	Internal use only
		[31:16]	Unused	R 0	Unused
08548	Master_VMON11	[15:0]	1V8_PHY	R 0	Internal use only
		[31:16]	Unused	R 0	Unused
0854C	Master_UV	[5:0]	Under voltage status	R 0	Internal use only
		[31:6]	Unused	R 0	Unused
08550	Slave_UV	[3:0]	Under voltage status	R 0	Internal use only
		[31:4]	Unused	R 0	Unused
08554	FPGA_Temp	[7:0]	Temperature value	R 0	Internal use only
		[31:8]	Unused	R 0	Unused
08558	SCC_Temp	[7:0]	Temperature value	R 0	Internal use only
		[31:8]	Unused	R 0	Unused

POWER: Control Addr	Reg	Bits	Bit Field	Type	Description

POWER: Control Addr	Reg	Bits	Bit Field	Type	Description
0855C	Timer	[24:0]	Update interval U82	RW	Defines the time that the ADC needs to sample the voltage. Value can be set between 0 and ~268ms.
		[31:25]	Unused	R 0	Unused
08560	Timer	[24:0]	Update interval U83	RW	Defines the time that the ADC needs to sample the voltage. Value can be set between 0 and ~268ms
		[31:25]	Unused	R 0	Unused
08564	Timer	[24:0]	Update interval U5	RW	Defines the time that the ADC needs to sample the voltage. Value can be set between 0 and ~268ms
		[31:25]	Unused	R 0	Unused
08568	Timer	[31:0]	Timeout watchdog low	RW	Defines the time when the HW automatically disables the measurement mode and switches back to normal mode. Programming value 0 disables the watchdog.
0856C	Timer	[7:0]	Timeout watchdog high	RW	
		[31:8]	Unused	R 0	

eMAC IP Configuration Register section):

Map Global Register Bank (GRB) of FPGA (0xF9000000) to local memory, length 0x10000

1. Disable TX and RX flow control (GRB+0x32C0):

   Set top 3 bits of the flow control configuration to zero, therefore disabling tx and rx flow control.

2. Setting the TX configuration bit to enable the transmitter and set to full duplex mode (GRB+0x3280):

   Half duplex (Bit 26) = 0, Transmit enable (Bit 28) = 1, Reset (Bit 31) = 0.

3. Setting the RX configuration bit to enable the transmitter and set to full duplex mode (GRB+0x3240):

   Length/Type Error Check (Bit 25) = 1, Half duplex (Bit 26) = 0, Receiver enable (Bit 28) = 1, Reset (Bit 31) = 0.

4. Setting the speed to 1Gb/s (GRB+0x3300):

   Bit 31 = 1, Bit 30 = 0.

5. Set promiscuous mode (GRB+0x3390):

   Promiscuous mode (Bit 31) = 1.

2.6.5.2 Initialization RX

1. Set RX buffer address (GRB+0x9000):

   Shift 5 times to the right and set address.

2. Read RX write index (GRB+0x9200):

   Read write index (note that the write index registers are read only by SW, thus cannot be set to an initial value).

3. Set RX write index to RX read index (GRB+0x9100):

   Set read index (both point to the same index now, thus the buffer is empty) and store value as driver internal read index.

4. Set RX last index (GRB+0x9300):

   Set last index to define buffer size (size = last index * 32).

5. Set RX route/destination (GRB+0x9500):

   Set route and destination depending on selected core (e.g. for core0 0x600).

6. Set RX hi MAC address (GRB+0x9600):

   Set high MAC address (e.g. 0x0045).

7. Set RX lo MAC address (GRB+0x9700):

   Set low MAC address (e.g. 0x414D4500).

8. Activate RX network port (GRB+0x9800):

   Set to 1.

2.6.5.3 Initialization TX

1. Set TX buffer address (GRB+0x9900):

   Shift 5 times right and set address.

2. Read TX read index (GRB+0x9A00):

   Read TX read index (note that the read index is read only for SW, thus cannot be initialized).

3. Write TX write index (GRB+0x9B00):

   Set write index (both point to the same index now, thus the buffer is empty) and store value as driver internal write index.

4. Set TX last index (GRB+0x9C00):

   Set last index to define buffer size (size = last index * 32).

5. Set TX route/destination (GRB+0x9D00):

   Set route and destination depending on selected core (e.g. for core0 0x600).

6. Activate TX network port (GRB+0x9E00):

   Set to 1.

2.6.5.4 Receiving Frame Data

1. Check for updated write index in head of buffer (or FPGA register) (RX buffer+0x00)
   a. Read 4 bytes from RX buffer at address 0 and compare the write index value to the internal read index value stored in the driver
2. Only if these indexes differ a new frame is ready and needs to be processed
   a. Read first 2 bytes (length of frame) and then the rest of data from the driver internal read index
   b. Set RX buffer read index to new position (GRB+0x9100)
   c. Set driver internal read index to new position
   d. Repeat step 2. as often as the internal read index and the write index register value still differs

2.6.5.5 Sending Frame Data

1. If enough free space is available, write frame length and data to tx buffer (tx buffer+(driver internal write index value+1)*32)
    a. Set frame length to the first 2 bytes, append the frame data
    b. Add padding where necessary
    c. Set TX write index register to new position (GRB+0x9B00)
    d. Set driver internal write index value to new position

2.7 Interrupt Handling

For each SCC core, the FPGA contains an interrupt logic unit in the register bank module which collects the events coming from internal interrupt sources or from IPIs sent by other cores and handles the generation of the interrupt packets. An interrupt packet is a write request from the FPGA to the CRB register within the tile of the receiving core which sets one of the two local interrupt bits. These bits are directly connected to the LINT0/LINT1 pins of the core. The current FPGA architecture supports 6 internal interrupt sources as well as 48 IPI sources: IPI0…IPI47, eMAC0…eMAC3 and SATA0…SATA1. The interrupt events from the eMAC modules are generated when the HW has written new packets into the buffer. For the SATA modules the generation of the event is not yet specified.

Interrupt Control Flow

1. Interrupt event(s) or IPI(s) set status bit(s)
        1. If not masked and no IRQ is pending, send IRQ packet to core.
        2. Otherwise does not send packet.
2. Core receives interrupt and jumps to service routine
        1. Reads status register.
        2. Determines interrupt source(s) and starts processing.
3. When processing has been finished
        1. Clears local interrupt bit in CRB.
        2. Writes to IRQ-Reset register to clear IRQ pending bit and according status bit(s).
4. If one of the status bits is still active, next IRQ packet sent to core

In case one core wants to send an IPI to another core, it writes to the appropriate IPI request register bit in the FPGA. This will set the status bit of the receiving core and triggers an interrupt.

SW can configure which one of the two local interrupt pins of the receiving core will be set by the interrupt packet.

Figure 13: Interrupt Unit Block Diagram

2.8 Atomic Increment Counters

The FPGA contains 96 32-bit wide atomic increment counters split into two sets of 48 counters each. The addresses of the counter sets start at 4k page boundaries at 0xE000 and 0xF000 (Atomic Increment Counters).

Each counter consists of a pair of registers, the atomic increment counter register and the initialization counter register. The atomic counter can be loaded with a 32 bit value though a write access to the initialization counter register. A read to this register provides the actual value of the atomic counter. Reading and writing to the atomic increment counter register increments (read) or decrements (write) the counter value. In case of a read access, the unmodified value will be provided back as read data.

3 Application Software Notes

This chapter describes the layering and functionality between RCK silicon, the system FPGA, the driver and the application SW.

From a physical point of view, the RCK silicon is connected to the system FPGA which also holds the PCIe interface that connects to the MCPC. The FPGA implements registers that provide a FIFO-based packet-oriented interface to RCK, where mesh packets can be read and written. This is basically a SW-controlled port into the RCK on-die mesh.

To write data into RCK memory a packet has to be generated that contains the asswell as an addressing and control header. To read memory, it is necessary to first generate a read request packet containing the packet header and then to wait for a reply packet that can be read out of the FPGA registers by SW, containing the requested data.

This FIF- based interface forms the HW layer of the SCEMI implementation. The SW SCEMI layer handles the low level packet transfer over the FIFO interface, but the application SW still talks to the SCEMI layer in terms of RCK packets, not directly mapped memory accesses.

The driver only has one task: to map the registers that implement the FIFO based interface into application SW memory. This is used for "legacy" code to transfer data word by word under SW control. The DMA implementation is used to pump larger amounts of prepared packets (not raw memory data!) between the FIFOs and host memory.

4 Register Definitions

In this chapter all registers that exist in the FPGA will be described. The registers are located in two independent register banks.

- The PCIe register bank can be accessed through native PCIe read/write accesses from the MCPC and contains all relevant registers to configure and control the data transfer through the PCIe interface.
- The FPGA register bank contains all application specific registers that can be accessed only through FPGA packet read/write requests that can be sent from the MCPC, the BMC or the SCC cores.

4.1 Register Bank Overview

<table>
<tr><th>Start Addr</th><th>End Addr</th><th>Description</th><th>Link</th></tr>
<tr><td>0x0000</td><td>0x0FFF</td><td>Configuration Registers PCIe IP</td><td>PCIe Register Bank</td></tr>
<tr><td>0x1000</td><td>0x1FFF</td><td>Reserved</td><td>n/a</td></tr>
<tr><td>0x2000</td><td>0x2FFF</td><td>Reserved</td><td>n/a</td></tr>
<tr><td>0x3000</td><td>0x31FF</td><td>Reserved</td><td>n/a</td></tr>
<tr><td>0x3200</td><td>0x33FF</td><td>Configuration Registers eMAC IP #0</td><td>
<table>
<tr><th>POWER: SYSTEM Addr</th><th>Reg</th><th>Bits</th><th>Bit Field</th><th>Type</th><th>Description</th></tr>
<tr><td>08500</td><td>Slave_VMON1</td><td>[15:0]</td><td>1V0</td><td>R0</td><td>Internal use only</td></tr>
<tr><td></td><td></td><td>[31:</td><td>Unused</td><td>R</td><td>Unused</td></tr>
</table>
</td></tr>
</table>

Start Addr	End Addr	Description	Link					
					16]		0	
			08504	Slave_VMON2	[15:0]	I_1V0	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08508	Slave_VMON3	[15:0]	1V1_VCCT	R0	Internal use only
					[31:16]	Unused	R0	Unused
			0850C	Slave_VMON4	[15:0]	I_1V1_VCCT	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08510	Slave_VMON5	[15:0]	1V1_VCCA	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08514	Slave_VMON6	[15:0]	I_1V1_VCCA	R0	Internal use only
					[31:16]	Unused	R0	Unused

Start Addr	End Addr	Description	Link					
			08518	Slave_VMON9	[15:0]	12V0R1	R0	Internal use only
					[31:16]	Unused	R0	Unused
			0851C	Slave_VMON10	[15:0]	12V0R2	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08520	Slave_VMON11	[15:0]	5V0_IN	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08524	Slave_VMON12	[15:0]	3V3_IN	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08528	Master_VMON1	[15:0]	3V3	R0	Internal use only
					[31:16]	Unused	R0	Unused
			0852C	Master_VMON2	[15:	I_3V3	R0	Internal use only

Start Addr	End Addr	Description	Link					
					0]			
					[31:16]	Unused	R0	Unused
			08530	Master_VMON5	[15:0]	1V65	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08534	Master_VMON6	[15:0]	1V65_ADJ	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08538	Master_VMON7	[15:0]	1V8_SB	R0	Internal use only
					[31:16]	Unused	R0	Unused
			0853C	Master_VMON8	[15:0]	5V0_PWR	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08540	Master_VMON9	[15:0]	3V3_PWR	R0	Internal use only

Start Addr	End Addr	Description	Link					
					[31:16]	Unused	R0	Unused
			08544	Master_VMON10	[15:0]	2V5	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08548	Master_VMON11	[15:0]	1V8_PHY	R0	Internal use only
					[31:16]	Unused	R0	Unused
			0854C	Master_UV	[5:0]	Under voltage status	R0	Internal use only
					[31:6]	Unused	R0	Unused
			08550	Slave_UV	[3:0]	Under voltage status	R0	Internal use only
					[31:4]	Unused	R0	Unused
			08554	FPGA_Temp	[7:0]	Temperature value	R0	Internal use only
					[31:8]	Unused	R0	Unused
			085	SCC_Te	[7	Temperat	R	Internal use

Start Addr	End Addr	Description	Link
			58 mp :0] ure value 0 only [31:8] Unused R0 Unused

POWER: Control Addr	Reg	Bits	Bit Field	Type	Description
0855C	Timer	[24:0]	Update interval U82	RW	Defines the time that the ADC needs to sample the voltage. Value can be set between 0 and ~268ms.
		[31:25]	Unused	R0	Unused
08560	Timer	[24:0]	Update interval U83	RW	Defines the time that the ADC needs to sample the voltage. Value can be set between 0 and ~268ms
		[31:25]	Unused	R0	Unused
08564	Timer	[24:	Update interval	RW	Defines the time that the

<table>
<tr><th>Start Addr</th><th>End Addr</th><th>Description</th><th>Link</th></tr>
<tr><td></td><td></td><td></td><td>
<table>
<tr><td></td><td></td><td>0]</td><td>U5</td><td></td><td>ADC needs to sample the voltage. Value can be set between 0 and ~268ms</td></tr>
<tr><td></td><td></td><td>[31:25]</td><td>Unused</td><td>R0</td><td>Unused</td></tr>
<tr><td>08568</td><td>Timer</td><td>[31:0]</td><td>Timeout watchdog low</td><td>RW</td><td rowspan="3">Defines the time when the HW automatically disables the measurement mode and switches back to normal mode. Programming value 0 disables the watchdog.</td></tr>
<tr><td>0856C</td><td>Timer</td><td>[7:0]</td><td>Timeout watchdog high</td><td>RW</td></tr>
<tr><td></td><td></td><td>[31:8]</td><td>Unused</td><td>R0</td></tr>
</table>
eMAC IP Configuration Register
</td></tr>
<tr><td>0x3400</td><td>0x35FF</td><td>Statistic Registers eMAC IP #0</td><td>
<table>
<tr><th>POWER: SYSTEM Addr</th><th>Reg</th><th>Bits</th><th>Bit Field</th><th>Type</th><th>Description</th></tr>
</table>
</td></tr>
</table>

Start Addr	End Addr	Description	Link					
			08500	Slave_VMON1	[15:0]	1V0	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08504	Slave_VMON2	[15:0]	I_1V0	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08508	Slave_VMON3	[15:0]	1V1_VCCT	R0	Internal use only
					[31:16]	Unused	R0	Unused
			0850C	Slave_VMON4	[15:0]	I_1V1_VCCT	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08510	Slave_VMON5	[15:0]	1V1_VCCA	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08514	Slave_VMON6	[15:	I_1V1_VCCA	R0	Internal use only

Start Addr	End Addr	Description	Link					
					0]			
					[31:16]	Unused	R 0	Unused
			08518	Slave_VMON9	[15:0]	12V0R1	R 0	Internal use only
					[31:16]	Unused	R 0	Unused
			0851C	Slave_VMON10	[15:0]	12V0R2	R 0	Internal use only
					[31:16]	Unused	R 0	Unused
			08520	Slave_VMON11	[15:0]	5V0_IN	R 0	Internal use only
					[31:16]	Unused	R 0	Unused
			08524	Slave_VMON12	[15:0]	3V3_IN	R 0	Internal use only
					[31:16]	Unused	R 0	Unused
			08528	Master_VMON1	[15:0]	3V3	R 0	Internal use only

Start Addr	End Addr	Description	Link					
					[31:16]	Unused	R0	Unused
			0852C	Master_VMON2	[15:0]	I_3V3	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08530	Master_VMON5	[15:0]	1V65	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08534	Master_VMON6	[15:0]	1V65_ADJ	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08538	Master_VMON7	[15:0]	1V8_SB	R0	Internal use only
					[31:16]	Unused	R0	Unused
			0853C	Master_VMON8	[15:0]	5V0_PWR	R0	Internal use only
					[31:1	Unused	R0	Unused

Start Addr	End Addr	Description	Link					
					6]			
			08540	Master_VMON9	[15:0]	3V3_PWR	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08544	Master_VMON10	[15:0]	2V5	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08548	Master_VMON11	[15:0]	1V8_PHY	R0	Internal use only
					[31:16]	Unused	R0	Unused
			0854C	Master_UV	[5:0]	Under voltage status	R0	Internal use only
					[31:6]	Unused	R0	Unused
			08550	Slave_UV	[3:0]	Under voltage status	R0	Internal use only
					[31:4]	Unused	R0	Unused
			08554	FPGA_Temp	[7:0	Temperature value	R0	Internal use only

Start Addr	End Addr	Description	Link

		]			
		[31:8]	Unused	R0	Unused
08558	SCC_Temp	[7:0]	Temperature value	R0	Internal use only
		[31:8]	Unused	R0	Unused

POWER: Control Addr	Reg	Bits	Bit Field	Type	Description
0855C	Timer	[24:0]	Update interval U82	RW	Defines the time that the ADC needs to sample the voltage. Value can be set between 0 and ~268ms.
		[31:25]	Unused	R0	Unused
08560	Timer	[24:0]	Update interval U83	RW	Defines the time that the ADC needs to sample the voltage. Value can be set between 0 and ~268ms

Start Addr	End Addr	Description	Link
			<table><tr><td></td><td></td><td>[31:25]</td><td>Unused</td><td>R0</td><td>Unused</td></tr><tr><td>08564</td><td>Timer</td><td>[24:0]</td><td>Update interval U5</td><td>RW</td><td>Defines the time that the ADC needs to sample the voltage. Value can be set between 0 and ~268ms</td></tr><tr><td></td><td></td><td>[31:25]</td><td>Unused</td><td>R0</td><td>Unused</td></tr><tr><td>08568</td><td>Timer</td><td>[31:0]</td><td>Timeout watchdog low</td><td>RW</td><td rowspan="2">Defines the time when the HW automatically disables the measurement mode and switches back to normal mode. Programming value 0 disables the watchdog.</td></tr><tr><td>0856C</td><td>Timer</td><td>[7:0]</td><td>Timeout watchdog high</td><td>RW</td></tr><tr><td></td><td></td><td>[31:8]</td><td>Unused</td><td>R0</td><td></td></tr></table> eMAC IP Configuration Register
0x3600	0x3FFF	Reserved	n/a
0x4000	0x41FF	Physical interface FPGA – SCC	IODELAY control registers

Start Addr	End Addr	Description	Link
0x4200	0x43FF	Configuration Registers eMAC IP #1	

POWER: SYSTEM Addr	Reg	Bits	Bit Field	Type	Description
08500	Slave_VMON1	[15:0]	1V0	R0	Internal use only
		[31:16]	Unused	R0	Unused
08504	Slave_VMON2	[15:0]	I_1V0	R0	Internal use only
		[31:16]	Unused	R0	Unused
08508	Slave_VMON3	[15:0]	1V1_VCCT	R0	Internal use only
		[31:16]	Unused	R0	Unused
0850C	Slave_VMON4	[15:0]	I_1V1_VCCT	R0	Internal use only
		[31:16]	Unused	R0	Unused
085	Slave_V	[1	1V1_VC	R	Internal use

Start Addr	End Addr	Description	Link					
			10	MON5	5:0]	CA	0	only
					[31:16]	Unused	R0	Unused
			08514	Slave_VMON6	[15:0]	I_1V1_VCCA	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08518	Slave_VMON9	[15:0]	12V0R1	R0	Internal use only
					[31:16]	Unused	R0	Unused
			0851C	Slave_VMON10	[15:0]	12V0R2	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08520	Slave_VMON11	[15:0]	5V0_IN	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08524	Slave_VMON12	[15:0]	3V3_IN	R0	Internal use only

Start Addr	End Addr	Description	Link					
					[31:16]	Unused	R0	Unused
			08528	Master_VMON1	[15:0]	3V3	R0	Internal use only
					[31:16]	Unused	R0	Unused
			0852C	Master_VMON2	[15:0]	I_3V3	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08530	Master_VMON5	[15:0]	1V65	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08534	Master_VMON6	[15:0]	1V65_ADJ	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08538	Master_VMON7	[15:0]	1V8_SB	R0	Internal use only
					[31:1	Unused	R0	Unused

Start Addr	End Addr	Description	Link					
					6]			
			0853C	Master_VMON8	[15:0]	5V0_PWR	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08540	Master_VMON9	[15:0]	3V3_PWR	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08544	Master_VMON10	[15:0]	2V5	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08548	Master_VMON11	[15:0]	1V8_PHY	R0	Internal use only
					[31:16]	Unused	R0	Unused
			0854C	Master_UV	[5:0]	Under voltage status	R0	Internal use only
					[31:6]	Unused	R0	Unused
			085	Slave_U	[3:0	Under voltage	R	Internal use

Start Addr	End Addr	Description	Link

50	V	]	status	0	only
		[31:4]	Unused	R0	Unused
08554	FPGA_Temp	[7:0]	Temperature value	R0	Internal use only
		[31:8]	Unused	R0	Unused
08558	SCC_Temp	[7:0]	Temperature value	R0	Internal use only
		[31:8]	Unused	R0	Unused

POWER: Control Addr	Reg	Bits	Bit Field	Type	Description
0855C	Timer	[24:0]	Update interval U82	RW	Defines the time that the ADC needs to sample the voltage. Value can be set between 0 and ~268ms.
		[31:25]	Unused	R0	Unused
085	Timer	[24:	Update interval	R	Defines the time that the

Start Addr	End Addr	Description	Link				
			60		0]	U83	W	ADC needs to sample the voltage. Value can be set between 0 and ~268ms
					[31:25]	Unused	R 0	Unused
			08564	Timer	[24:0]	Update interval U5	R W	Defines the time that the ADC needs to sample the voltage. Value can be set between 0 and ~268ms
					[31:25]	Unused	R 0	Unused
			08568	Timer	[31:0]	Timeout watchdog low	R W	Defines the time when the HW automatically disables the measurement mode and switches back to normal mode. Programming value 0 disables the watchdog.
			0856C	Timer	[7:0]	Timeout watchdog high	R W	
					[31:8]	Unused	R 0	

<table>
<tr><th>Start Addr</th><th>End Addr</th><th>Description</th><th>Link</th></tr>
<tr><td></td><td></td><td></td><td>eMAC IP Configuration Register</td></tr>
<tr><td>0x4400</td><td>0x45FF</td><td>Statistic Registers eMAC IP #1</td><td>
<table>
<tr><th>POWER: SYSTEM Addr</th><th>Reg</th><th>Bits</th><th>Bit Field</th><th>Type</th><th>Description</th></tr>
<tr><td rowspan="2">08500</td><td rowspan="2">Slave_VMON1</td><td>[15:0]</td><td>1V0</td><td>R0</td><td>Internal use only</td></tr>
<tr><td>[31:16]</td><td>Unused</td><td>R0</td><td>Unused</td></tr>
<tr><td rowspan="2">08504</td><td rowspan="2">Slave_VMON2</td><td>[15:0]</td><td>I_1V0</td><td>R0</td><td>Internal use only</td></tr>
<tr><td>[31:16]</td><td>Unused</td><td>R0</td><td>Unused</td></tr>
<tr><td rowspan="2">08508</td><td rowspan="2">Slave_VMON3</td><td>[15:0]</td><td>1V1_VCCT</td><td>R0</td><td>Internal use only</td></tr>
<tr><td>[31:16]</td><td>Unused</td><td>R0</td><td>Unused</td></tr>
<tr><td rowspan="2">0850C</td><td rowspan="2">Slave_VMON4</td><td>[15:0]</td><td>I_1V1_VCCT</td><td>R0</td><td>Internal use only</td></tr>
<tr><td>[31:1</td><td>Unused</td><td>R0</td><td>Unused</td></tr>
</table>
</td></tr>
</table>

Start Addr	End Addr	Description	Link					
					6]			
			08510	Slave_VMON5	[15:0]	1V1_VCCA	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08514	Slave_VMON6	[15:0]	I_1V1_VCCA	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08518	Slave_VMON9	[15:0]	12V0R1	R0	Internal use only
					[31:16]	Unused	R0	Unused
			0851C	Slave_VMON10	[15:0]	12V0R2	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08520	Slave_VMON11	[15:0]	5V0_IN	R0	Internal use only
					[31:16]	Unused	R0	Unused

Start Addr	End Addr	Description	Link					
			08524	Slave_VMON12	[15:0]	3V3_IN	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08528	Master_VMON1	[15:0]	3V3	R0	Internal use only
					[31:16]	Unused	R0	Unused
			0852C	Master_VMON2	[15:0]	I_3V3	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08530	Master_VMON5	[15:0]	1V65	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08534	Master_VMON6	[15:0]	1V65_ADJ	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08538	Master_VMON7	[15:	1V8_SB	R0	Internal use only

Start Addr	End Addr	Description	Link					
					0]			
					[31:16]	Unused	R0	Unused
			0853C	Master_VMON8	[15:0]	5V0_PWR	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08540	Master_VMON9	[15:0]	3V3_PWR	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08544	Master_VMON10	[15:0]	2V5	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08548	Master_VMON11	[15:0]	1V8_PHY	R0	Internal use only
					[31:16]	Unused	R0	Unused
			0854C	Master_UV	[5:0]	Under voltage status	R0	Internal use only

Start Addr	End Addr	Description	Link
			<table><tr><td></td><td></td><td>[31:6]</td><td>Unused</td><td>R0</td><td>Unused</td></tr><tr><td>08550</td><td>Slave_UV</td><td>[3:0]</td><td>Under voltage status</td><td>R0</td><td>Internal use only</td></tr><tr><td></td><td></td><td>[31:4]</td><td>Unused</td><td>R0</td><td>Unused</td></tr><tr><td>08554</td><td>FPGA_Temp</td><td>[7:0]</td><td>Temperature value</td><td>R0</td><td>Internal use only</td></tr><tr><td></td><td></td><td>[31:8]</td><td>Unused</td><td>R0</td><td>Unused</td></tr><tr><td>08558</td><td>SCC_Temp</td><td>[7:0]</td><td>Temperature value</td><td>R0</td><td>Internal use only</td></tr><tr><td></td><td></td><td>[31:8]</td><td>Unused</td><td>R0</td><td>Unused</td></tr></table> <table><tr><th>POWER: Control Addr</th><th>Reg</th><th>Bits</th><th>Bit Field</th><th>Type</th><th>Description</th></tr><tr><td>0855C</td><td>Timer</td><td>[24:0]</td><td>Update interval U82</td><td>RW</td><td>Defines the time that the ADC needs to sample the voltage. Value can be set between 0 and ~268ms.</td></tr></table>

Start Addr	End Addr	Description	Link					
					[31:25]	Unused	R0	Unused
			08560	Timer	[24:0]	Update interval U83	RW	Defines the time that the ADC needs to sample the voltage. Value can be set between 0 and ~268ms
					[31:25]	Unused	R0	Unused
			08564	Timer	[24:0]	Update interval U5	RW	Defines the time that the ADC needs to sample the voltage. Value can be set between 0 and ~268ms
					[31:25]	Unused	R0	Unused
			08568	Timer	[31:0]	Timeout watchdog low	RW	Defines the time when the HW automatically disables the measurement mode and switches back to normal mode. Programming value 0 disables the
			0856C	Timer	[7:0]	Timeout watchdog high	RW	

<table>
<tr><th>Start Addr</th><th>End Addr</th><th>Description</th><th>Link</th></tr>
<tr><td></td><td></td><td></td><td>
<table>
<tr><td></td><td></td><td></td><td></td><td></td><td>watchdog.</td></tr>
<tr><td></td><td></td><td>[31:8]</td><td>Unused</td><td>R0</td><td></td></tr>
</table>
eMAC IP Configuration Register
</td></tr>
<tr><td>0x4600</td><td>0x4FFF</td><td>Reserved</td><td>n/a</td></tr>
<tr><td>0x5000</td><td>0x51FF</td><td>Reserved</td><td>n/a</td></tr>
<tr><td>0x5200</td><td>0x53FF</td><td>Configuration Registers eMAC IP #2</td><td>
<table>
<tr><th>POWER: SYSTEM Addr</th><th>Reg</th><th>Bits</th><th>Bit Field</th><th>Type</th><th>Description</th></tr>
<tr><td>08500</td><td>Slave_VMON1</td><td>[15:0]</td><td>1V0</td><td>R0</td><td>Internal use only</td></tr>
<tr><td></td><td></td><td>[31:16]</td><td>Unused</td><td>R0</td><td>Unused</td></tr>
<tr><td>08504</td><td>Slave_VMON2</td><td>[15:0]</td><td>I_1V0</td><td>R0</td><td>Internal use only</td></tr>
<tr><td></td><td></td><td>[31:16]</td><td>Unused</td><td>R0</td><td>Unused</td></tr>
<tr><td>08508</td><td>Slave_VMON3</td><td>[15:0]</td><td>1V1_VCCT</td><td>R0</td><td>Internal use only</td></tr>
</table>
</td></tr>
</table>

Start Addr	End Addr	Description	Link					
					[31:16]	Unused	R0	Unused
			0850C	Slave_VMON4	[15:0]	I_1V1_VCCT	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08510	Slave_VMON5	[15:0]	1V1_VCCA	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08514	Slave_VMON6	[15:0]	I_1V1_VCCA	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08518	Slave_VMON9	[15:0]	12V0R1	R0	Internal use only
					[31:16]	Unused	R0	Unused
			0851C	Slave_VMON10	[15:0]	12V0R2	R0	Internal use only
					[31:1	Unused	R0	Unused

Start Addr	End Addr	Description	Link				
					6]		
			08520	Slave_VMON11	[15:0]	5V0_IN	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08524	Slave_VMON12	[15:0]	3V3_IN	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08528	Master_VMON1	[15:0]	3V3	R0	Internal use only
					[31:16]	Unused	R0	Unused
			0852C	Master_VMON2	[15:0]	I_3V3	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08530	Master_VMON5	[15:0]	1V65	R0	Internal use only
					[31:16]	Unused	R0	Unused
			085	Master_	[1	1V65_A	R	Internal use

Start Addr	End Addr	Description	Link					
			34	VMON6	[5:0]	DJ	0	only
					[31:16]	Unused	R0	Unused
			08538	Master_VMON7	[15:0]	1V8_SB	R0	Internal use only
					[31:16]	Unused	R0	Unused
			0853C	Master_VMON8	[15:0]	5V0_PWR	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08540	Master_VMON9	[15:0]	3V3_PWR	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08544	Master_VMON10	[15:0]	2V5	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08548	Master_VMON11	[15:0]	1V8_PHY	R0	Internal use only

Start Addr	End Addr	Description	Link

		[31:16]	Unused	R0	Unused
0854C	Master_UV	[5:0]	Under voltage status	R0	Internal use only
		[31:6]	Unused	R0	Unused
08550	Slave_UV	[3:0]	Under voltage status	R0	Internal use only
		[31:4]	Unused	R0	Unused
08554	FPGA_Temp	[7:0]	Temperature value	R0	Internal use only
		[31:8]	Unused	R0	Unused
08558	SCC_Temp	[7:0]	Temperature value	R0	Internal use only
		[31:8]	Unused	R0	Unused

POWER: Control Addr	Reg	Bits	Bit Field	Type	Description
085	Timer	[2	Update	R	Defines the

Start Addr	End Addr	Description	Link					
			5C		[4:0]	interval U82	W	time that the ADC needs to sample the voltage. Value can be set between 0 and ~268ms.
					[31:25]	Unused	R 0	Unused
			08560	Timer	[24:0]	Update interval U83	R W	Defines the time that the ADC needs to sample the voltage. Value can be set between 0 and ~268ms
					[31:25]	Unused	R 0	Unused
			08564	Timer	[24:0]	Update interval U5	R W	Defines the time that the ADC needs to sample the voltage. Value can be set between 0 and ~268ms
					[31:25]	Unused	R 0	Unused
			08568	Timer	[31:0]	Timeout watchdog low	R W	Defines the time when the HW automatically disables the measurement
			0856C	Timer	[7:0	Timeout watchdo	R W	

Start Addr	End Addr	Description	Link
			<table><tr><td></td><td></td><td>]</td><td>g high</td><td></td><td>mode and switches back to normal mode. Programming value 0 disables the watchdog.</td></tr><tr><td></td><td></td><td>[31:8]</td><td>Unused</td><td>R0</td><td></td></tr></table> eMAC IP Configuration Register
0x5400	0x55FF	Statistic Registers eMAC IP #2	<table><tr><th>POWER: SYSTEM Addr</th><th>Reg</th><th>Bits</th><th>Bit Field</th><th>Type</th><th>Description</th></tr><tr><td>08500</td><td>Slave_VMON1</td><td>[15:0]</td><td>1V0</td><td>R0</td><td>Internal use only</td></tr><tr><td></td><td></td><td>[31:16]</td><td>Unused</td><td>R0</td><td>Unused</td></tr><tr><td>08504</td><td>Slave_VMON2</td><td>[15:0]</td><td>I_1V0</td><td>R0</td><td>Internal use only</td></tr><tr><td></td><td></td><td>[31:16]</td><td>Unused</td><td>R0</td><td>Unused</td></tr><tr><td>085</td><td>Slave_V</td><td>[1</td><td>1V1_VC</td><td>R</td><td>Internal use</td></tr></table>

Start Addr	End Addr	Description	Link					
			08	MON3	[5:0]	CT	0	only
					[31:16]	Unused	R0	Unused
			0850C	Slave_VMON4	[15:0]	I_1V1_VCCT	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08510	Slave_VMON5	[15:0]	1V1_VCCA	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08514	Slave_VMON6	[15:0]	I_1V1_VCCA	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08518	Slave_VMON9	[15:0]	12V0R1	R0	Internal use only
					[31:16]	Unused	R0	Unused
			0851C	Slave_VMON10	[15:0]	12V0R2	R0	Internal use only

Start Addr	End Addr	Description	Link					
					[31:16]	Unused	R0	Unused
			08520	Slave_VMON11	[15:0]	5V0_IN	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08524	Slave_VMON12	[15:0]	3V3_IN	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08528	Master_VMON1	[15:0]	3V3	R0	Internal use only
					[31:16]	Unused	R0	Unused
			0852C	Master_VMON2	[15:0]	I_3V3	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08530	Master_VMON5	[15:0]	1V65	R0	Internal use only
					[31:1	Unused	R0	Unused

Start Addr	End Addr	Description	Link					
					6]			
			0853 4	Master_VMON6	[15:0]	1V65_ADJ	R0	Internal use only
					[31:16]	Unused	R0	Unused
			0853 8	Master_VMON7	[15:0]	1V8_SB	R0	Internal use only
					[31:16]	Unused	R0	Unused
			0853C	Master_VMON8	[15:0]	5V0_PWR	R0	Internal use only
					[31:16]	Unused	R0	Unused
			0854 0	Master_VMON9	[15:0]	3V3_PWR	R0	Internal use only
					[31:16]	Unused	R0	Unused
			0854 4	Master_VMON10	[15:0]	2V5	R0	Internal use only
					[31:16]	Unused	R0	Unused
			085	Master_	[1	1V8_PH	R	Internal use

Start Addr	End Addr	Description	Link					
			48	VMON11	5:0]	Y	0	only
					[31:16]	Unused	R 0	Unused
			0854C	Master_UV	[5:0]	Under voltage status	R 0	Internal use only
					[31:6]	Unused	R 0	Unused
			08550	Slave_UV	[3:0]	Under voltage status	R 0	Internal use only
					[31:4]	Unused	R 0	Unused
			08554	FPGA_Temp	[7:0]	Temperature value	R 0	Internal use only
					[31:8]	Unused	R 0	Unused
			08558	SCC_Temp	[7:0]	Temperature value	R 0	Internal use only
					[31:8]	Unused	R 0	Unused

Start Addr	End Addr	Description	Link

POWER: Control Addr	Reg	Bits	Bit Field	Type	Description
0855C	Timer	[24:0]	Update interval U82	RW	Defines the time that the ADC needs to sample the voltage. Value can be set between 0 and ~268ms.
		[31:25]	Unused	R0	Unused
08560	Timer	[24:0]	Update interval U83	RW	Defines the time that the ADC needs to sample the voltage. Value can be set between 0 and ~268ms
		[31:25]	Unused	R0	Unused
08564	Timer	[24:0]	Update interval U5	RW	Defines the time that the ADC needs to sample the voltage. Value can be set between 0 and ~268ms
		[31:	Unused	R0	Unused

<table>
<tr><th>Start Addr</th><th>End Addr</th><th>Description</th><th>Link</th></tr>
<tr><td></td><td></td><td></td><td>
<table>
<tr><td></td><td></td><td>25]</td><td></td><td></td><td></td></tr>
<tr><td>08568</td><td>Timer</td><td>[31:0]</td><td>Timeout watchdog low</td><td>RW</td><td rowspan="2">Defines the time when the HW automatically disables the measurement mode and switches back to normal mode. Programming value 0 disables the watchdog.</td></tr>
<tr><td>0856C</td><td>Timer</td><td>[7:0]</td><td>Timeout watchdog high</td><td>RW</td></tr>
<tr><td></td><td></td><td>[31:8]</td><td>Unused</td><td>R0</td><td></td></tr>
</table>
eMAC IP Configuration Register
</td></tr>
<tr><td>0x5600</td><td>0x5FFF</td><td>Reserved</td><td>n/a</td></tr>
<tr><td>0x6000</td><td>0x61FF</td><td>Reserved</td><td>n/a</td></tr>
<tr><td>0x6200</td><td>0x63FF</td><td>Configuration Registers eMAC IP #3</td><td>
<table>
<tr><th>POWER: SYSTEM Addr</th><th>Reg</th><th>Bits</th><th>Bit Field</th><th>Type</th><th>Description</th></tr>
<tr><td>08500</td><td>Slave_VMON1</td><td>[15:0]</td><td>1V0</td><td>R0</td><td>Internal use only</td></tr>
<tr><td></td><td></td><td>[3</td><td>Unused</td><td>R</td><td>Unused</td></tr>
</table>
</td></tr>
</table>

Start Addr	End Addr	Description	Link					
					1:16]		0	
			08504	Slave_VMON2	[15:0]	I_1V0	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08508	Slave_VMON3	[15:0]	1V1_VCCT	R0	Internal use only
					[31:16]	Unused	R0	Unused
			0850C	Slave_VMON4	[15:0]	I_1V1_VCCT	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08510	Slave_VMON5	[15:0]	1V1_VCCA	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08514	Slave_VMON6	[15:0]	I_1V1_VCCA	R0	Internal use only
					[31:1	Unused	R0	Unused

Start Addr	End Addr	Description	Link					
					6]			
			08518	Slave_VMON9	[15:0]	12V0R1	R0	Internal use only
					[31:16]	Unused	R0	Unused
			0851C	Slave_VMON10	[15:0]	12V0R2	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08520	Slave_VMON11	[15:0]	5V0_IN	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08524	Slave_VMON12	[15:0]	3V3_IN	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08528	Master_VMON1	[15:0]	3V3	R0	Internal use only
					[31:16]	Unused	R0	Unused

Start Addr	End Addr	Description	Link					
			0852C	Master_VMON2	[15:0]	I_3V3	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08530	Master_VMON5	[15:0]	1V65	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08534	Master_VMON6	[15:0]	1V65_ADJ	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08538	Master_VMON7	[15:0]	1V8_SB	R0	Internal use only
					[31:16]	Unused	R0	Unused
			0853C	Master_VMON8	[15:0]	5V0_PWR	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08540	Master_VMON9	[15:	3V3_PWR	R0	Internal use only

Start Addr	End Addr	Description	Link					
					0]			
					[31:16]	Unused	R0	Unused
			08544	Master_VMON10	[15:0]	2V5	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08548	Master_VMON11	[15:0]	1V8_PHY	R0	Internal use only
					[31:16]	Unused	R0	Unused
			0854C	Master_UV	[5:0]	Under voltage status	R0	Internal use only
					[31:6]	Unused	R0	Unused
			08550	Slave_UV	[3:0]	Under voltage status	R0	Internal use only
					[31:4]	Unused	R0	Unused
			08554	FPGA_Temp	[7:0]	Temperature value	R0	Internal use only
					[31:	Unused	R0	Unused

Start Addr	End Addr	Description	Link

		8]			
08558	SCC_Temp	[7:0]	Temperature value	R0	Internal use only
		[31:8]	Unused	R0	Unused

POWER: Control Addr	Reg	Bits	Bit Field	Type	Description
0855C	Timer	[24:0]	Update interval U82	RW	Defines the time that the ADC needs to sample the voltage. Value can be set between 0 and ~268ms.
		[31:25]	Unused	R0	Unused
08560	Timer	[24:0]	Update interval U83	RW	Defines the time that the ADC needs to sample the voltage. Value can be set between 0 and ~268ms
		[31:25]	Unused	R0	Unused

Start Addr	End Addr	Description	Link
			08564 Timer [24:0] Update interval U5 RW Defines the time that the ADC needs to sample the voltage. Value can be set between 0 and ~268ms 08564 Timer [31:25] Unused R 0 Unused 08568 Timer [31:0] Timeout watchdog low RW Defines the time when the HW automatically disables the measurement mode and switches back to normal mode. Programming value 0 disables the watchdog. 086C Timer [7:0] Timeout watchdog high RW 086C Timer [31:8] Unused R 0 eMAC IP Configuration Register
0x6400	0x65FF	Statistic Registers eMAC IP #3	

Start Addr	End Addr	Description	Link

POWER: SYSTEM Addr	Reg	Bits	Bit Field	Type	Description
08500	Slave_VMON1	[15:0]	1V0	R0	Internal use only
		[31:16]	Unused	R0	Unused
08504	Slave_VMON2	[15:0]	I_1V0	R0	Internal use only
		[31:16]	Unused	R0	Unused
08508	Slave_VMON3	[15:0]	1V1_VCCT	R0	Internal use only
		[31:16]	Unused	R0	Unused
0850C	Slave_VMON4	[15:0]	I_1V1_VCCT	R0	Internal use only
		[31:16]	Unused	R0	Unused
08510	Slave_VMON5	[15:	1V1_VCCA	R0	Internal use only

Start Addr	End Addr	Description	Link					
					0]			
					[31:16]	Unused	R0	Unused
			08514	Slave_VMON6	[15:0]	I_1V1_VCCA	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08518	Slave_VMON9	[15:0]	12V0R1	R0	Internal use only
					[31:16]	Unused	R0	Unused
			0851C	Slave_VMON10	[15:0]	12V0R2	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08520	Slave_VMON11	[15:0]	5V0_IN	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08524	Slave_VMON12	[15:0]	3V3_IN	R0	Internal use only

Start Addr	End Addr	Description	Link					
					[31:16]	Unused	R0	Unused
			08528	Master_VMON1	[15:0]	3V3	R0	Internal use only
					[31:16]	Unused	R0	Unused
			0852C	Master_VMON2	[15:0]	I_3V3	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08530	Master_VMON5	[15:0]	1V65	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08534	Master_VMON6	[15:0]	1V65_ADJ	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08538	Master_VMON7	[15:0]	1V8_SB	R0	Internal use only
					[31:1	Unused	R0	Unused

Start Addr	End Addr	Description	Link					
					6]			
			0853C	Master_VMON8	[15:0]	5V0_PWR	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08540	Master_VMON9	[15:0]	3V3_PWR	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08544	Master_VMON10	[15:0]	2V5	R0	Internal use only
					[31:16]	Unused	R0	Unused
			08548	Master_VMON11	[15:0]	1V8_PHY	R0	Internal use only
					[31:16]	Unused	R0	Unused
			0854C	Master_UV	[5:0]	Under voltage status	R0	Internal use only
					[31:6]	Unused	R0	Unused
			085	Slave_U	[3:0	Under voltage	R	Internal use

Start Addr	End Addr	Description	Link

50	V	]	status	0	only
		[31:4]	Unused	R0	Unused
08554	FPGA_Temp	[7:0]	Temperature value	R0	Internal use only
		[31:8]	Unused	R0	Unused
08558	SCC_Temp	[7:0]	Temperature value	R0	Internal use only
		[31:8]	Unused	R0	Unused

POWER: Control Addr	Reg	Bits	Bit Field	Type	Description
0855C	Timer	[24:0]	Update interval U82	RW	Defines the time that the ADC needs to sample the voltage. Value can be set between 0 and ~268ms.
		[31:25]	Unused	R0	Unused
085	Timer	[24:	Update interval	R	Defines the time that the

Start Addr	End Addr	Description	Link					
			60		0]	U83	W	ADC needs to sample the voltage. Value can be set between 0 and ~268ms
					[31:25]	Unused	R 0	Unused
			08564	Timer	[24:0]	Update interval U5	R W	Defines the time that the ADC needs to sample the voltage. Value can be set between 0 and ~268ms
					[31:25]	Unused	R 0	Unused
			08568	Timer	[31:0]	Timeout watchdog low	R W	Defines the time when the HW automatically disables the measurement mode and switches back to normal mode. Programming value 0 disables the watchdog.
			0856C	Timer	[7:0]	Timeout watchdog high	R W	
					[31:8]	Unused	R 0	

Start Addr	End Addr	Description	Link
			eMAC IP Configuration Register
0x6600	0x6FFF	Reserved	n/a
0x7000	0x70FF	Status Registers eMAC IP #0	eMAC Application Registers
0x7100	0x71FF	Status Registers eMAC IP #1	eMAC Application Registers
0x7200	0x72FF	Status Registers eMAC IP #2	eMAC Application Registers
0x7300	0x73FF	Status Registers eMAC IP #3	eMAC Application Registers
0x7400	0x7FFF	General Config Registers eMAC	eMAC Application Registers
0x8000	0x8FFF	FPGA Status / Config Registers	SIF status info registers
0x9000	0x9FFF	RX-/TX-Control Registers eMAC #0	eMAC Application Registers
0xA000	0xAFFF	RX-/TX-Control Registers eMAC #1	eMAC Application Registers
0xB000	0xBFFF	RX-/TX-Control Registers eMAC #2	eMAC Application Registers
0xC000	0xCFFF	RX-/TX-Control Registers eMAC #3	eMAC Application Registers
0xD000	0xDFFF	Interrupt Controller	Interrupt Registers
0xE000	0xEFFF	Atomic Increment Counters	Atomic Increment Counters
0xF000	0xFFFF	Atomic Increment Counters	Atomic Increment Counters

4.2 PCIe Register Bank

The following registers are accessible via PCIe reads and write from the MCPC and are used for the configuration of the PCIe DMA engine.

DCSR Addr	Reg	Bits	Bit Field	Type	Description
000	DCSR	[31:24]	FPGA Family	RO	8’b00000000 = Invalid Entry 8’b00010001 = Virtex-2 PRO

					8’b00010010 = Virtex-4 FX 8’b00010011 = Virtex-5 LXT/SXT 8’b00011000 = Spartan™-3 8’b00101000 = Spartan-3E 8’b00111000 = Spartan-3A
		[23:20]	R1	RO	Reserved
		[19:16]	Data Path Width	RO	4’b0000 = Invalid Entry 4’b0001 = 32 bit 4’b0010 = 64 bit All other = Reserved
		[15:8]	Version Number	RO	Build Version Number; Corresponds to Document Revision Number
		[7:1]	R0	RO	Reserved
		[0]	Initiator Reset	RW	1’b1 = Resets and holds Read/Write Initiator engines in reset. Clears RO Status Registers. 1’b0 = Enable Read/Write Initiator engine operation.

DDMACR Addr	Reg	Bits	Bit Field	Type	Description
004	DDMACR	[31]	Read DMA Operation Data Error	RO	1’b1 = When expected Read Completion Data Pattern not equal to expected Data Pattern; Cleared by Initiator Reset
		[30:25]	R3	RO	Reserved
		[24]	Read DMA Done	RO	1’b1 = Read DMA Operation Done; Cleared by Initiator Reset
		[23]	Read DMA Done Interrupt Disable	RW	1’b1 = Disable transmission of interrupts
		[22]	Read DMA No	RW	1’b1 = Sets No Snoop attribute bit on

DDMACR Addr	Reg	Bits	Bit Field	Type	Description
			Snoop		TLP
		[21]	Read DMA Relaxed Ordering	RW	1’b1 = Sets Relaxed Ordering attribute bit on TLP
		[20:17]	R2	RO	Reserved
		[16]	Read DMA Start	RW	1’b1 = Start the Read DMA Engine; Cleared by Initiator Reset
		[15:9]	R1	R0	Reserved
		[8]	Write DMA Done	R0	1’b1 = Write DMA Operation Done; Cleared by Initiator Reset
		[7]	Write DMA Interrupt Disable	RW	1’b1 = Disable transmission of interrupts
		[6]	Write DMA No Snoop	RW	1’b1 = Sets No Snoop attribute bit on TLP
		[5]	Write DMA Relaxed Ordering	RW	1’b1 = Sets Relaxed Ordering attribute bit on TLP
		[4:1]	R0	RO	Reserved
		[0]	Write DMA Start	RW	1’b1 = Start the Write DMA Engine; Cleared by Initiator Reset;

WDMATLPA Addr	Reg	Bits	Bit Field	Type	Description
008	WDMATLPA	[31:2]	Write DMA Lower TLP Address	RW	Write DMA Lower TLP Address; This address will be placed on the first Write TLP. Subsequent TLP address field values are derived from this address and TLP size.
		[1:0]	R1	RO	Reserved

WDMATLPS Addr	Reg	Bits	Bit Field	Type	Description

00C	WDMATLPS	[31:24]	Write DMA Upper TLP	RW	Write DMA Upper TLP Address; Specifies 64b TLP Transaction Address bits[39:32]. TLP Transaction address bits[63:40] will always be 0
		[23:20]	R1	RO	Reserved
		[19]	64 Bit Write TLP Enable	RW	64bit Write TLP Enable; Setting this bit enables 64b Memory Write TLP generation.
		[18:16]	Write DMA TLP TC	RW	Memory Write TLP Traffic Class; Controls Traffic Class field of the generated TLP
		[15:13]	R0	RO	Reserved
		[12:0]	Write DMA TLP Size	RW	Memory Write TLP Payload Length in DWORDs (1 DWORD = 4 Bytes); 01H<=Length<=1FFFH

WDMATLPC Addr	Reg	Bits	Bit Field	Type	Description
010	WDMATLPC	[31:16]	R0	RO	Reserved
		[15:0]	Write DMA TLP Count	RW	Memory Write 32 TLP Count; 01D<=Count<=65535D

WDMATLPP Addr	Reg	Bits	Bit Field	Type	Description
014	WDMATLPP	[31:0]	Write DMA TLP Data Pattern	RW	Memory Write 32 TLP Data Pattern; All Write TLP Payload DWORDs

RDMATLPP Addr	Reg	Bits	Bit Field	Type	Description

018	RDMATLPP	[31:0]	Read DMA Expected Data Pattern	RW	Data Pattern expected in Completion with Data TLPs; All Completion TLP Payload DWORDs

RDMATLPA Addr	Reg	Bits	Bit Field	Type	Description
01C	RDMATLPA	[31:2]	Read DMA Low TLP Address	RW	Read DMA Lower TLP Address; This address will be placed on the first Read TLP. Subsequent TLP address field values are derived from this address and TLP size
		[1:0]	R1	RO	Reserved

RDMATLPS Addr	Reg	Bits	Bit Field	Type	Description
020	RDMATLPS	[31:24]	Read DMA Upper TLP Address	RW	Read DMA Upper TLP Address; 64b Transaction Address bits[39:32]. Bits [63:40] will always be 0.
		[23:20]	R1	RO	Reserved
		[19]	64 Bit Read TLP Enable	RW	64bit Read TLP Enable; Setting this bit enables 64b Memory Read TLP generation.
		[18:16]	Read DMA TLP TC	RW	Memory Read TLP Traffic Class; Controls Traffic Class field of the generated TLP
		[15:13]	R0	RO	Reserved
		[12:0]	Read DMA TLP Size	RW	Memory Read TLP Read Length in DWORDs (1 DWORD = 4 Bytes);

					01H<=Length<=1FFFH

RDMATLPC Addr	Reg	Bits	Bit Field	Type	Description
024	RDMATLPC	[31:16]	R0	RO	Reserved
		[15:0]	Read DMA TLP Count	RW	Memory Read 32 TLP Count; 01D<=Count<=65535D

WDMAPERF Addr	Reg	Bits	Bit Field	Type	Description
028	WDMAPERF	[31:0]	Write DMA Performance Counter	RO	Number of Interface Clock Cycles for Write DMA transfer to complete; Cycle time depends on Core Interface Data Path (DCSR) value. • x8 = 4 ns cycle time • x4 = 8 ns cycle time for 64 bit, 4 ns cycle time for 32 bit • x1 = 32 ns cycle time for 64 bit, 16 ns cycle time or 32 bit Cleared by Initiator Reset

RDMAPERF Addr	Reg	Bits	Bit Field	Type	Description
02C	RDMAPERF	[31:0]	Read DMA Performance	RO	Number of Interface Clock Cycles for Read DMA transfer to complete; Cycle

			Counter		time depends on Core Interface Data Path (DCSR) value. • x8 = 4 ns cycle time • x4 = 8 ns cycle time for 64 bit, 4 ns cycle time for 32 bit • x1 = 32 ns cycle time for 64 bit, 16 ns cycle time or 32 bit Cleared by Initiator Reset

RDMASTAT Addr	Reg	Bits	Bit Field	Type	Description
030	RDMASTAT	[31:16]	R0	RO	Reserved
		[15:8]	Completion w/ UR Tag	RO	Tag received on the last completion w/ UR; Cleared by Initiator Reset
		[7:0]	Completions w/ UR Received	RO	Tag received on the last completion w/ UR; Cleared by Initiator Reset

NRDCOMP Addr	Reg	Bits	Bit Field	Type	Description
034	NRDCOMP	[31:0]	Number of Completion w/ Data TLPs Received	RO	Number of completions w/ Data Received; Cleared by Initiator Reset

RCOMPDSIZW Addr	Reg	Bits	Bit Field	Type	Description
038	RCOMPDSIZW	[31:0]	Total size of data received	RO	Total size of data received; Cleared by Initiator Reset

DLWSTAT Addr	Reg	Bits	Bit Field	Type	Description
03C	DLWSTAT	[31:14]	R1	RO	Reserved
		[13:8]	Negotiated Max. Link Width	RO	Negotiated Maximum Link Width for the Device; Encoding is as follows: 000000b Reserved 000001b x 1 000010b x 2 000100b x 4 001000b x 8 001100b x 12 010000b x 16 100000b x 32
		[7:6]	R0	RO	Reserved
		[5:0]	Capability Max. Link Width	RO	Capability Maximum Link Width for the Device; Encoding is as follows: 000000b Reserved 000001b x 1 000010b x 2 000100b x 4 001000b x 8 001100b x 12 010000b x 16 100000b x 32

DLTRSSTAT Addr	Reg	Bits	Bit Field	Type	Description
040	DLTRSSTAT	[31:19]	R2	RO	Reserved
		[18:16]	Max. Read Request Size	RO	Maximum Read Request Size; Device must not generate read requests with size exceeding the set value. Encoding is as follows: 000000b Reserved 000b 128B 001b 256B 010b 512B 011b 1024B 100b 2048B 101b 4096B 110b Reserved 111b Reserved
		[15:11]	R1	RO	Reserved
		[10:8]	Programmed Max. Payload Size	RO	Programmed Maximum Payload Size for the Device; Encoding is as follows: 000b 128B 001b 256B 010b 512B 011b 1024B 100b 2048B 101b 4096B 110b Reserved 111b Reserved
		[7:3]	R0	RO	Reserved
		[2:0]	Capability Max. Payload Size	RO	Capability Maximum Payload Size for the Device;

DLTRSSTAT Addr	Reg	Bits	Bit Field	Type	Description
					Encoding is as follows: 000b 128B 001b 256B 010b 512B 011b 1024B 100b 2048B 101b 4096B 110b Reserved 111b Reserved

DMISCCON Addr	Reg	Bits	Bit Field	Type	Description
044	DMISCCONT	[31:9]	R1	RO	Reserved
		[8]	Receive Non-Posted OK	RW	1’b0 = trn_rnp_ok_n driven to logic 0 1’b1= trn_rnp_ok_n driven to logic 1
		[7:2]	R0	RO	Reserved
		[1]	Read Metering Enable	RW	1’b1= Read Metering Enabled
		[0]	Completion Streaming Enable	RW	1’b1= Completion Streaming Enabled

The following registers are also accessible via direct PCIe read/write accesses. They contain general configuration/status information as well as some debug hooks.

ID Addr	Reg	Bits	Bit Field	Type	Description
300	ID(0)	[31:0]	BitSID	R	release ID of the SIF design (year/month/day/count)
304	ID(1)	[31:0]	grbTest[31:0]	R/W A5A5A5A 5_h	basic memory for test purposes
308	ID(2)	[31:0]	grbTest[63:32]	R/W DEADBEEF$_h$	

TRNCT Addr	Reg	Bits	Bit Field	Type	Description
30C	TRNCT(0)	[31:0]	barWin[0][31:0]	R	Message Out Port (MOP) fifo data
310	TRNCT(1)	[31:0]	barWin[0][63:32]	R	Message In Port (MIP) fifo data
314	TRNCT(2)	[31:0]	fooWin[0][31:0]	R/W 0	
318	TRNCT(3)	[31:0]	fooWin[0][63:32]	R/W 0	
31C	TRNCT(4)	[15:0]	FIFOFI[0]	R	MOP Fifo Fill level
		[31:16]	FIFOFI[1]	R	MIP Fifo Fill level
320	TRNCT(5)	[0]	INIT	TrgWr	SCEMI reset bit
		[1]	TRNCTE	R/W all 1	
		[31:2]	N.U.	R 0	unused

CONFIG Addr	Reg	Bits	Bit Field	Type	Description
324	CONFIG	[0]	SoftReset	R/W 0_d	Holds the application part of the SIF FPGA in reset when asserted (resets all SIF logic but the PCIE IF)
		[31:1]	N.U.	R 0	unused

SCEMI CONFIG Addr	Reg	Bits	Bit Field	Type	Description
328	SCEMICONFIG	[0]	MipDisableRd	R/W 0_d	Disables automatic packet forwarding from MIP to transactor (and router). Stalls all transmission into the design
		[1]	ScemiLoopback	R/W 0_d	Loopback of data from MIP to MOP directly for debugging purposes
		[2]	MipRdtrig	TrgWr	Triggers the read pointer in the MIP fifo to increment to show the next 64bit data in MipFifoDat
		[31:3]	N.U.	R 0	unused

SCEMIDATA Addr	Reg	Bits	Bit Field	Type	Description
32C	SCEMIDATA(0)	[31:0]	MipFifoDat[31:0]	R	64bit data that is at the current location of the read pointer in the MIP FIFO
330	SCEMIDATA(1)	[31:0]	MipFifoDat[63:32]	R	

RTRCOUNT Addr	Reg	Bits	Bit Field	Type	Description
334	RTRCOUNT(0)	[31:0]	RTRCntP0I	R	Software can read the number of RTR packets received by the port 0 input so far.
338	RTRCOUNT(1)	[31:0]	RTRCntP0O	R	Software can read the number of RTR packets sent by the port 0 output so far.
33C	RTRCOUNT(2)	[31:0]	RTRCntP1I	R	Software can read the number of RTR packets received by the port 1 input so far.
340	RTRCOUNT(3)	[31:0]	RTRCntP1O	R	Software can read the number of RTR packets sent by the port 1 output so far.
344	RTRCOUNT(4)	[31:0]	RTRCntP2I	R	Software can read the number of RTR packets received by the port 2 input so far.
348	RTRCOUNT(5)	[31:0]	RTRCntP2O	R	Software can read the number of RTR packets sent by the port 2 output so far.
34C	RTRCOUNT(6)	[31:0]	RTRCntP3I	R	Software can read the number of RTR packets received by the port 3 input so far.
350	RTRCOUNT(7)	[31:0]	RTRCntP3O	R	Software can read the number of RTR packets sent by the port 3 output so far.

RESERVED Addr	Reg	Bits	Bit Field	Type	Description
354 – 3B0	RESERVED	[31:0]	Reserved	R	Internal Use Only

FPGA CONF Addr	Reg	Bits	Bit Field	Type	Description
3B4	FPGA Configuration	[8:0]	SCEMI Buffer Size	RO	Size of SCEMI buffers in KB
		[10:9]	eMAC Available	RO	eMAC IP blocks exist in design when bit is set [9] eMAC#0 [10] eMAC#1
		[12:11]	SATA Available	RO	SATA IP blocks exist in design when bit is set [11] SATA#0 [12] SATA#1

4.3 FPGA Register Bank

The following registers can be accessed through FPGA packet accesses only, either from the MCPC, from the BMC or directly from SW that is running on the SCC cores.

4.3.1 IODELAY control registers

The FPGA elements used for the input deskew are IODELAY cells in variable delay mode. Their delay settings can be configured at run-time via register file accesses.

SIFPHY: PHYiod Control Addr	Reg	Bits	Bit Field	Type	Description
04000	PHYiodControl(0)	[31:0]	PHYiodEnDataIn[31:0]	R/W 0_d	DATA input IODELAY enable (CE), 0: no delay change 1: modify delay according to direction setting
04004	PHYiodControl(1)	[31:0]	PHYiodEnDataIn[63:32]	R/W 0_d	
04008	PHYiodControl(2)	[7:0]	PHYiodEnDataIn[71:64]	R/W 0_d	
		[8]	PHYiodEnTXIn	R/W 0_d	TXCLK input IODELAY enable (CE), 0: no delay change 1: modify delay according to direction setting
		[9]	PHYiodEnRXIn	R/W 0_d	RXCLK input IODELAY enable (CE), 0: no delay change 1: modify delay according to direction setting
		[31:10]	N.U.	R 0	unused
0400C	PHYiodControl(3)	[31:0]	PHYiodDirDataIn[31:0]	R/W 0_d	DATA input IODELAY direction (INC), 0: decrement delay 1: increment delay
04010	PHYiodControl(4)	[31:0]	PHYiodDirDataIn[63:32]	R/W 0_d	
04014	PHYiodControl(5)	[7:0]	PHYiodDirDataIn[71:64]	R/W 0_d	
		[8]	PHYiodDirTXIn	R/W 0_d	TXCLK input IODELAY direction (INC), 0: decrement delay 1: increment delay
		[9]	PHYiodDirRXIn	R/W 0_d	RXCLK input IODELAY direction (INC), 0: decrement delay 1: increment delay
		[31:10]	N.U.	R 0	unused
04018	PHYiodControl(6)	[0]	PHYiodClockPulse	TrgWr	Modify input IODELAY values, Trigger on Write Signal: 0: no

SIFPHY: PHYiod Control Addr	Reg	Bits	Bit Field	Type	Description
					delay change 1: enable one clock pulse to the IODELAY control interface to modify the delay by one according to programmed settings for each bit
		[31:1]	N.U.	R 0	unused

4.3.2 Pattern generation and training control registers

SIFPHY: PHYpg Control Addr	Reg	Bits	Bit Field	Type	Description
0401C	PHYpgControl(0)	[31:0]	PHYFlitData[31:0]	R	Captured data of the rx_ccf fifo (32 flits, incremented via PHYNextFlit)
04020	PHYpgControl(1)	[31:0]	PHYFlitData[63:32]	R	
04024	PHYpgControl(2)	[31:0]	PHYFlitData[95:64]	R	
04028	PHYpgControl(3)	[31:0]	PHYFlitData[127:96]	R	
0402C	PHYpgControl(4)	[15:0]	PHYFlitData[143:128]	R	
		[31:16]	N.U.	R 0	unused
04030	PHYpgControl(5)	[0]	PHYPatternGenOn	R/W 0_d	Enables the pattern generation in the TX path to continuously sent a pre-determined flit pattern for delay adaptation
		[1]	PHYMonitorEn	R/W 0_d	Enables pattern monitor in the RX path (disables MIU RX de-packetizer)
		[2]	PHYIoLoopBackOn	R/W 0_d	Enables loopback mode to test RX/TX packetizers without attached Rock Creek
		[3]	PHYCaptureFlits	TrgWr	Trigger on Write: enables capturing 32 received flits in RX clock crossing fifo (rx_ccf)
		[4]	PHYNextFlit	TrgWr	Trigger on Write: selects next captured flit to be available in register PHYFlitData
		[31:5]	N.U.	R 0	unused

4.3.3 SIF status info registers

SIFCFG: PCI Config Addr	Reg	Bits	Bit Field	Type	Description
08000	PCIConfig(0)	[15:0]	PCICfgStatus	R	Status Register: Status register from the Configuration Space Header.
		[31:16]	N.U.	R 0	unused
08004	PCIConfig(1)	[15:0]	PCICfgDStatus	R	Device Status Register: Device status register from the PCI Express Extended Capability Structure.
		[31:16]	N.U.	R 0	unused
08008	PCIConfig(2)	[15:0]	PCICfgLStatus	R	Link Status Register: Link status register from the PCI Express Extended Capability Structure.
		[31:16]	N.U.	R 0	unused
0800C	Debug Reg	[0]	MOPfull	R/W	Indicates MOP FIFO almost full
		[1]	PacketError	R/W	Packet Error received
		[31:2]	Not used	R 0	Unused

SIFCFG: ID Addr	Reg	Bits	Bit Field	Type	Description
08010	ID(0)	[31:0]	BitSID	R	release ID of the SIF design (year/month/day/count)
08014	ID(1)	[31:0]	grbTest[31:0]	R/W $A5A5A5A5_h$	basic memory for test purposes
08018	ID(2)	[31:0]	grbTest[63:32]	R/W $DEADBEEF_h$	

SIFCFG: CONFIG Addr	Reg	Bits	Bit Field	Type	Description
0801C	CONFIG	[0]	N.U.	R 0	unused
		[1]	cfgGenRCMemComp	R/W 1_d	Configures the SIF MIU to generate Memory Completion (without data) packets upon RC writes. Relieves host software from having to do this. Performance feature, use with care.
		[2]	cfgDropRCMemComp	R/W 0_d	Configures the SIF MIU to drop Memory Completion (without data) packets received from RC. Relieves host software from having to do this. Performance feature, use with care.
		[3]	RCInitDone	R	This flag indicates that the Rock Creek system initialization (power, reset, clock config, etc.) is done and the chip is ready for interaction with the SIF FPGA. The flag is driven by FFU_GPIO[0] from the board.
		[4]	RtrCntRst	TrgWr	This bit triggers a reset to 0 of the packet counters in the rtr module. The counters are not reset via a normal system reset.
		[5]	scemiTxRetyEn	R/W 0_d	Enables the retry of host read/write requests at the SCEMI port after a watchdog has timed out and no response was received from RC.
		[31:6]	N.U.	R 0	unused

SIFCFG: FPGA Status Addr	Reg	Bits	Bit Field	Type	Description
08020	FPGAStatus(0)	[15:0]	RtrStat	R	Router enable and valid bits. 4 * enable IN/OUT, valid IN/OUT, port 3 down to 0.
		[22:16]	OOOResponse	R	TID of response packets out of order SATA#1, SATA#0, eMAC#3…eMAC#0, Host
08024	FPGAStatus(1)	[15:0]	RtrDebug	R/W 0_d	Enable packet forwarding to host of router traffic
		[16]	EthernetDebug	R/W	0 – All frames get routed to SIF 1 – All frames get routed to MCPC/BMC
		[31:17]	Reserved	R 0	Reserved
08028	FPGAStatus(2)	[31:0]	ScemiFifoLvl	R	Scemi Fifo lvl of RX and TX in a RC packet domain as status info.

SIFCFG: RTR COUNT Addr	Reg	Bits	Bit Field	Type	Description
0802C	RTRCOUNT(0)	[31:0]	RTRCntP0I	R	Software can read the number of RTR packets received by the port 0 input so far.
08030	RTRCOUNT(1)	[31:0]	RTRCntP0O	R	Software can read the number of RTR packets sent by the port 0 output so far.
08034	RTRCOUNT(2)	[31:0]	RTRCntP1I	R	Software can read the number of RTR packets received by the port 1 input so far.
08038	RTRCOUNT(3)	[31:0]	RTRCntP1O	R	Software can read the number of RTR packets sent by the port 1 output so far.
0803C	RTRCOUNT(4)	[31:0]	RTRCntP2I	R	Software can read the number of RTR packets received by the port 2 input so far.
08040	RTRCOUNT(5)	[31:0]	RTRCntP2O	R	Software can read the number of RTR packets sent by the port 2 output so far.
08044	RTRCOUNT(6)	[31:0]	RTRCntP3I	R	Software can read the number of RTR packets received by the port 3 input so far.
08048	RTRCOUNT(7)	[31:0]	RTRCntP3O	R	Software can read the number of RTR packets sent by the port 3 output so far.
0804C	RTRCOUNT(8)	[31:0]	RCKReqMonitor[31:0]	R	Request Monitoring: {InvalidID,Timeout,retry,retry_count[4:0]}[63:0] - one 8bit vector per core
08050	RTRCOUNT(9)	[31:0]	RCKReqMonitor[63:32]	R	
... 08088	... RTRCOUNT(23)	[31:0]	RCKReqMonitor[511:480]	R	
0808C	RTRCOUNT(24)	[7:0]	N.U.	R 0	unused
		[31:8]	N.U.	R 0	unused
08090	RTRCOUNT(25)	[10:0]	RXFifoLvlMax	R	Mesh RX FiFo maximum level (ever) information (reset via

SIFCFG: RTR COUNT Addr	Reg	Bits	Bit Field	Type	Description
					RtrCntRst)
		[15:11]	N.U.	R 0	unused
		[26:16]	RXFifoLvl	R	Mesh RX FiFo current level information (must be below d512, otherwise there was an overflow)
		[31:27]	N.U.	R 0	unused
08094	RTRCOUNT(26)	[31:0]	RtrPktMon[31:0]	R	Software can read the header of NCIOWR and NCIORD packets received at a sif_rtr port (4*96bit per port, packet bits [351:256]). packet assignment is in increasing port order, i.e. [95:0] is first received packet from port 0 (host_port)
08098	RTRCOUNT(27)	[31:0]	RtrPktMon[63:32]	R	
... 08150	... RTRCOUNT(73)	[31:0]	RtrPktMon[1535:1504]	R	

SIFCFG: BMC PORT Addr	Reg	Bits	Bit Field	Type	Description
08154	BMCPORT	[31:0]	BMCGPIO	R	System state as reported from BMC and to BMC: 15: PLLLOCKLOST, 14: PLLLOCK, 9: RCKPRESENT, 8: RCKPOWERISON

SIFCFG: RTR PKT DEBUG Addr	Reg	Bits	Bit Field	Type	Description
08158	RTRPKTDEBUG(0)	[0]	PortLoggingEn	R/W 0_d	Enable the port logging for the SIF_RC_PORT.
		[9:1]	N.U.	R 0	Not used
		[10]	Port1InPktRd	Trg Wr	Trigger on Write: selects next packet to be read from fifo storing port1 input packets
		[11]	Port1OutPktRd	Trg Wr	
		[31:12]	N.U.	R 0	Not used
0815C	RTRPKTDEBUG(1)	[3:0]	MAC IP Reset	Trg Wr	Resets MAC IP block when writing a 1 to this register bits [0] – eMAC#0 [1] – eMAC#1 [2] – eMAC#2 [3] – eMAC#3
		[31:4]	N.U.	R 0	Not used
08160	RTRPKTDEBUG(2)	[31:0]	N.U.[63:32]	R 0	Not used
... 0818C	... RTRPKTDEBUG(12)	[31:0]	N.U.[383:352]	R 0	Not used
08190	RTRPKTDEBUG(13)	[31:0]	N.U.[31:0]	R 0	Not used
08194	RTRPKTDEBUG(14)	[31:0]	N.U.[63:32]	R 0	Not used
... 081BC	... RTRPKTDEBUG(23)	[31:0]	N.U.[351:320]	R 0	Not used
081C0	RTRPKTDEBUG(24)	[0]	MOPfull Enable	R/W	Enable for MOPfull status bit Enabled by default
	RTRPKTDEBUG(24)	[1]	PacketError Enable	R/W	Enable for PacketError Status bit Enabled by default
	RTRPKTDEBUG(24)	[31:2]	N.U.	R 0	Not used

SIFCFG: RTR PKT DEBUG Addr	Reg	Bits	Bit Field	Type	Description
081C4	RTRPKTDEBUG(25)	[31:0]	Port1InPkt[31:0]	R	Captured packet from the router port 1 (RC) (up to 512 packets, select via Port1InPktRd)
081C8	RTRPKTDEBUG(26)	[31:0]	Port1InPkt[63:32]	R	
... 081F0	... RTRPKTDEBUG(36)	[31:0]	Port1InPkt[383:352]	R	
081F4	RTRPKTDEBUG(37)	[31:0]	Port1OutPkt[31:0]	R	Captured packet to the router port 1 (RC) (up to 512 packets, select via Port1OutPktRd)
081F8	RTRPKTDEBUG(38)	[31:0]	Port1OutPkt[63:32]	R	
... 08220	... RTRPKTDEBUG(48)	[31:0]	Port1OutPkt[383:352]	R	

SIFCFG: TESET Addr	Reg	Bits	Bit Field	Type	Description
08224 …08228	Timestamp	[31:0]		RO	Provides lower 32 bits of global timestamp counter based on 125MHz system clock of the FPGA
		[31:0]		RO	Provides upper 32 bits of global timestamp counter based on 125MHz system clock of the FPGA

SIFCFG: SYSTEM Addr	Reg	Bits	Bit Field	Type	Description
0822C	FPGA Configuration	[8:0]	SCEMI Buffer Size	RO	Size of SCEMI buffers in KB
		[12:9]	eMAC Available	RO	eMAC IP blocks exist in design when bit is set [9] eMAC#0 [10] eMAC#1 [11] eMAC#2 [12] eMAC#3
		[14:13]	SATA Available	RO	SATA IP blocks exist in design when bit is set [13] SATA#0 [14] SATA#1
		[31:13]		R 0	Unused
08230	Clock Frequency	[15:0]	Fast Clock	R/W	Actual frequency of the fast clock
		[31:16]	Unused	R 0	Unused
08234	Memory Configuration	[1:0]	MC#0, Slot#0	RW	Defines the size of the DIMM modules applied to slot#0 and slot#1 on the board 00 – 0GB 01 – 2GB 10 – 4GB 11 – 8GB
		[3:2]	MC#0, Slot#1	RW	
		[5:4]	MC#1, Slot#0	RW	
		[7:6]	MC#1, Slot#1	RW	
		[9:8]	MC#2, Slot#0	RW	
		[11:10]	MC#2, Slot#1	RW	
		[13:12]	MC#3, Slot#0	RW	
		[15:14]	MC#3, Slot#1	RW	
		[31:16]	Unused	R 0	Unused
08238	Linux Frame Buffer Resolution	[15:0]	X Resolution	RW	Resolution of frame buffer in X dimension
		[31:16]	Unused	R 0	Unused

SIFCFG: SYSTEM Addr	Reg	Bits	Bit Field	Type	Description
0823C	Linux Frame Buffer Resolution	[15:0]	Y Resolution	RW	Resolution of frame buffer in Y dimension
		[31:16]	Unused	R 0	Unused
08240	Linux Frame Buffer Resolution	[7:0]	Color Depth	RW	Color depth of frame buffer
		[31:8]	Unused	R 0	Unused
08244	Linux Private Memory	[7:0]	Private Memory Slots	RW	Number of private memory slots used by one Linux instance Default is set to 0x14 (320 MB)
		[31:8]	Unused	R 0	Unused
08248	SATA IP Config	[7:0]	Tile ID	RW	Route to the MC where the SATA buffer is located
		[31:8]	Unused	R 0	Unused
0824C	SATA IP Config	[2:0]	Destination ID	RW	Router port where the MC is connected
		[31:3]	Unused	R 0	Unused
08250	SATA IP Config	[5:0]	IRQ Core Number	RW	Core that receives the IRQs from SATA IP
		[31:6]	Unused	R 0	Unused
08254	SATA IP Config	[1:0]	MC Select	RW	Address bits 33:32 to select MC
		[31:2]	Unused	R 0	Unused
08258	SATA IP Config	[0]	SATA IP reset	TrgWr	Resets the SATA IP block
		[31:1]	Unused	R 0	Unused

4.3.4 Power Measurement Registers

These registers are located in the power measurement control block and divided into two register sets. The user register set can be read by the SCC application through the SIF. The system register block is for internal use and can be read by the BMC only.

POWER: USER Addr	Reg	Bits	Bit Field	Type	Description
08400	LTC2495_CH0	[15:0]	SCC_VCC0	RO	Supply voltage for tiles 0-2, 0-3, 1-2, 1-3
		[31:16]	Unused	R 0	Unused
08404	LTC2495_CH1	[15:0]	SCC_VCC1	RO	Supply voltage for tiles 2-2, 2-3, 3-2, 3-3
		[31:16]	Unused	R 0	Unused
08408	LTC2495_CH2	[15:0]	SCC_VCC2	RO	Supply voltage for mesh network / system interface
		[31:16]	Unused	R 0	Unused
0840C	LTC2495_CH3	[15:0]	SCC_VCC3	RO	Supply voltage for tiles 4-2, 4-3, 5-2, 5-3
		[31:16]	Unused	R 0	Unused
08410	LTC2495_CH4	[15:0]	SCC_VCC4	RO	Supply voltage for tiles 0-0, 0-1, 1-0, 1-1
		[31:16]	Unused	R 0	Unused
08414	LTC2495_CH5	[15:0]	SCC_VCC5	RO	Supply voltage for tiles 2-0, 2-1, 3-0, 3-1
		[31:16]	Unused	R 0	Unused
08418	LTC2495_CH7	[15:0]	SCC_VCC7	RO	Supply voltage for tiles 4-0, 4-1, 5-0, 5-1
		[31:16]	Unused	R 0	Unused
0841C	Master_VMON3	[15:0]	SCC_3V3	RO	Supply voltage of SCC tile silicon region
		[31:16]	Unused	R 0	Unused
08420	Slave_VMON7	[15:0]	MC_1V5	RO	Supply voltage of Memory Controller silicon region
		[31:16]	Unused	R 0	Unused
08424	Master_VMON4	[15:0]	SCC_3V3C	RO	Supply current of SCC tile silicon region

POWER: USER Addr	Reg	Bits	Bit Field	Type	Description
		[31:16]	Unused	R 0	Unused
08428	Slave_VMON8	[15:0]	SCC_1V5C	RO	Supply current of Memory Controller silicon region
		[31:16]	Unused	R 0	Unused
0842C	DomainConfig	[0]	Enable SCC_VCC0	RW	1 – Domain will be measured 0 – Measurement disabled
		[1]	Enable SCC_VCC1	RW	
		[2]	Enable SCC_VCC2	RW	
		[3]	Enable SCC_VCC3	RW	
		[4]	Enable SCC_VCC4	RW	
		[5]	Enable SCC_VCC5	RW	
		[6]	Enable SCC_VCC7	RW	
		[7]	Enable SCC_3V3	RW	
		[8]	Enable MC_1V5	RW	
		[9]	Enable SCC_3V3C	RW	
		[10]	Enable MC_1V5C	RW	
		[11]	Enable Measurement Mode	RW	1 – Measurement Mode enabled 0 – Measurement Mode disabled
		[31:12]	Unused	R 0	Unused

POWER: SYSTEM Addr	Reg	Bits	Bit Field	Type	Description
08500	Slave_VMON1	[15:0]	1V0	R 0	Internal use only
		[31:16]	Unused	R 0	Unused
08504	Slave_VMON2	[15:0]	I_1V0	R 0	Internal use only

POWER: SYSTEM Addr	Reg	Bits	Bit Field	Type	Description
		[31:16]	Unused	R 0	Unused
08508	Slave_VMON3	[15:0]	1V1_VCCT	R 0	Internal use only
		[31:16]	Unused	R 0	Unused
0850C	Slave_VMON4	[15:0]	I_1V1_VCCT	R 0	Internal use only
		[31:16]	Unused	R 0	Unused
08510	Slave_VMON5	[15:0]	1V1_VCCA	R 0	Internal use only
		[31:16]	Unused	R 0	Unused
08514	Slave_VMON6	[15:0]	I_1V1_VCCA	R 0	Internal use only
		[31:16]	Unused	R 0	Unused
08518	Slave_VMON9	[15:0]	12V0R1	R 0	Internal use only
		[31:16]	Unused	R 0	Unused
0851C	Slave_VMON10	[15:0]	12V0R2	R 0	Internal use only
		[31:16]	Unused	R 0	Unused
08520	Slave_VMON11	[15:0]	5V0_IN	R 0	Internal use only
		[31:16]	Unused	R 0	Unused
08524	Slave_VMON12	[15:0]	3V3_IN	R 0	Internal use only
		[31:16]	Unused	R 0	Unused
08528	Master_VMON1	[15:0]	3V3	R 0	Internal use only
		[31:16]	Unused	R 0	Unused
0852C	Master_VMON2	[15:0]	I_3V3	R 0	Internal use only
		[31:16]	Unused	R 0	Unused
08530	Master_VMON5	[15:0]	1V65	R 0	Internal use only
		[31:16]	Unused	R 0	Unused
08534	Master_VMON6	[15:0]	1V65_ADJ	R 0	Internal use only

POWER: SYSTEM Addr	Reg	Bits	Bit Field	Type	Description
		[31:16]	Unused	R 0	Unused
08538	Master_VMON7	[15:0]	1V8_SB	R 0	Internal use only
		[31:16]	Unused	R 0	Unused
0853C	Master_VMON8	[15:0]	5V0_PWR	R 0	Internal use only
		[31:16]	Unused	R 0	Unused
08540	Master_VMON9	[15:0]	3V3_PWR	R 0	Internal use only
		[31:16]	Unused	R 0	Unused
08544	Master_VMON10	[15:0]	2V5	R 0	Internal use only
		[31:16]	Unused	R 0	Unused
08548	Master_VMON11	[15:0]	1V8_PHY	R 0	Internal use only
		[31:16]	Unused	R 0	Unused
0854C	Master_UV	[5:0]	Under voltage status	R 0	Internal use only
		[31:6]	Unused	R 0	Unused
08550	Slave_UV	[3:0]	Under voltage status	R 0	Internal use only
		[31:4]	Unused	R 0	Unused
08554	FPGA_Temp	[7:0]	Temperature value	R 0	Internal use only
		[31:8]	Unused	R 0	Unused
08558	SCC_Temp	[7:0]	Temperature value	R 0	Internal use only
		[31:8]	Unused	R 0	Unused

POWER: Control Addr	Reg	Bits	Bit Field	Type	Description
0855C	Timer	[24:0]	Update interval U82	RW	Defines the time that the ADC needs to sample the voltage. Value can be set between 0 and ~268ms.

POWER: Control Addr	Reg	Bits	Bit Field	Type	Description
		[31:25]	Unused	R 0	Unused
08560	Timer	[24:0]	Update interval U83	RW	Defines the time that the ADC needs to sample the voltage. Value can be set between 0 and ~268ms
		[31:25]	Unused	R 0	Unused
08564	Timer	[24:0]	Update interval U5	RW	Defines the time that the ADC needs to sample the voltage. Value can be set between 0 and ~268ms
		[31:25]	Unused	R 0	Unused
08568	Timer	[31:0]	Timeout watchdog low	RW	Defines the time when the HW automatically disables the measurement mode and switches back to normal mode. Programming value 0 disables the watchdog.
0856C	Timer	[7:0]	Timeout watchdog high	RW	
		[31:8]	Unused	R 0	

4.3.5 eMAC IP Configuration Register

These registers are located in the eMAC IP block from Xilinx. Detailed description can be found in the user guide document.

SIFCFG: eMAC #0 CFG Addr	Reg	Bits	Bit Field	Type	Description
03200	Receiver Configuration(0)	[31:0]		R/W	see Xilinx eMAC user guide
03240	Receiver Configuration(1)	[31:0]		R/W	
03280	Transmitter Configuration	[31:0]		R/W	
032C0	Flow Control Configuration	[31:0]		R/W	
03300	Ethernet MAC Mode	[31:0]		R/W	
03320	RGMII Configuration	[31:0]		R/W	
03340	Management Configuration	[31:0]		R/W	see Xilinx eMAC user guide
03380	Unicast Address (Word 0)	[31:0]		R/W	
03384	Unicast Address (Word 1)	[31:0]		R/W	
03388	Additional Address Table Access (Word 0)	[31:0]		R/W	see Xilinx eMAC user guide
0338C	Additional Address Table Access (Word 1)	[31:0]		R/W	
03390	Address Filter Mode	[31:0]		R/W	

SIFCFG: eMAC #1 CFG Addr	Reg	Bits	Bit Field	Type	Description
04200	Receiver Configuration(0)	[31:0]		R/W	see Xilinx eMAC user guide
04240	Receiver Configuration(1)	[31:0]		R/W	
04280	Transmitter Configuration	[31:0]		R/W	
042C0	Flow Control Configuration	[31:0]		R/W	
04300	Ethernet MAC Mode	[31:0]		R/W	
04320	RGMII Configuration	[31:0]		R/W	
04340	Management Configuration	[31:0]		R/W	see Xilinx eMAC user guide
04380	Unicast Address (Word 0)	[31:0]		R/W	
04384	Unicast Address (Word 1)	[31:0]		R/W	
04388	Additional Address Table Access (Word 0)	[31:0]		R/W	see Xilinx eMAC user guide
0438C	Additional Address Table Access (Word 1)	[31:0]		R/W	
04390	Address Filter Mode	[31:0]		R/W	

SIFCFG: eMAC #2 CFG Addr	Reg	Bits	Bit Field	Type	Description
05200	Receiver Configuration(0)	[31:0]		R/W	see Xilinx eMAC user guide
05240	Receiver Configuration(1)	[31:0]		R/W	
05280	Transmitter Configuration	[31:0]		R/W	
052C0	Flow Control Configuration	[31:0]		R/W	
05300	Ethernet MAC Mode	[31:0]		R/W	
05320	RGMII Configuration	[31:0]		R/W	
05340	Management Configuration	[31:0]		R/W	see Xilinx eMAC user guide
05380	Unicast Address (Word 0)	[31:0]		R/W	
05384	Unicast Address (Word 1)	[31:0]		R/W	
05388	Additional Address Table Access (Word 0)	[31:0]		R/W	see Xilinx eMAC user guide
0538C	Additional Address Table Access (Word 1)	[31:0]		R/W	
05390	Address Filter Mode	[31:0]		R/W	

SIFCFG: eMAC #3 CFG Addr	Reg	Bits	Bit Field	Type	Description
06200	Receiver Configuration(0)	[31:0]		R/W	see Xilinx eMAC user guide
06240	Receiver Configuration(1)	[31:0]		R/W	
06280	Transmitter Configuration	[31:0]		R/W	
062C0	Flow Control Configuration	[31:0]		R/W	
06300	Ethernet MAC Mode	[31:0]		R/W	
06320	RGMII Configuration	[31:0]		R/W	
06340	Management Configuration	[31:0]		R/W	see Xilinx eMAC user guide
06380	Unicast Address (Word 0)	[31:0]		R/W	
06384	Unicast Address (Word 1)	[31:0]		R/W	
06388	Additional Address Table Access (Word 0)	[31:0]		R/W	see Xilinx eMAC user guide
0638C	Additional Address Table Access (Word 1)	[31:0]		R/W	
06390	Address Filter Mode	[31:0]		R/W	

The statistics registers are located in the statistics block for the Xilinx eMAC IP. Details about this block can be found in the user guide document. All registers have 64bit size.

SIFCFG: eMAC#0 STATISTIC Addr	Reg	Bits	Bit Field	Type	Description
03400	Transmitted bytes low	[31:0]		RO	see Xilinx eMAC statistics user guide
03404	Transmitted bytes high	[31:0]		RO	
03408	Received bytes low	[31:0]		RO	
0340C	Received bytes high	[31:0]		RO	
03410	Undersize frames received low	[31:0]		RO	
03414	Undersize frames received high	[31:0]		RO	
03418	Fragment frames received low	[31:0]		RO	
0341C	Fragment frames received high	[31:0]		RO	
03420	64 byte Frames Received OK low	[31:0]		RO	
03424	64 byte Frames Received OK high	[31:0]		RO	
03428	65-127 byte Frames Received OK low	[31:0]		RO	
0342C	65-127 byte Frames Received OK high	[31:0]		RO	
03430	128-255 byte Frames Received OK low	[31:0]		RO	
03434	128-255 byte Frames Received OK high	[31:0]		RO	
03438	256-511 byte Frames Received OK low	[31:0]		RO	
0343C	256-511 byte Frames Received OK high	[31:0]		RO	
03440	512-1023 byte Frames Received OK low	[31:0]		RO	
03444	512-1023 byte Frames Received OK high	[31:0]		RO	
03448	1024-MaxFrameSize byte Frames Received OK low	[31:0]		RO	
0344C	1024-MaxFrameSize byte Frames Received OK high	[31:0]		RO	

SIFCFG: eMAC#0 STATISTIC Addr	Reg	Bits	Bit Field	Type	Description
03450	Oversize Frames Received OK low	[31:0]		RO	
03454	Oversize Frames Received OK high	[31:0]		RO	
03458	64 byte Frames Transmitted OK low	[31:0]		RO	
0345C	64 byte Frames Transmitted OK high	[31:0]		RO	
03460	65-127 byte Frames Transmitted OK low	[31:0]		RO	
03464	65-127 byte Frames Transmitted OK high	[31:0]		RO	
03468	128-255 byte Frames Transmitted OK low	[31:0]		RO	
0346C	128-255 byte Frames Transmitted OK high	[31:0]		RO	
03470	256-511 byte Frames Transmitted OK low	[31:0]		RO	
03474	256-511 byte Frames Transmitted OK high	[31:0]		RO	
03478	512-1023 byte Frames Transmitted OK low	[31:0]		RO	
0347C	512-1023 byte Frames Transmitted OK high	[31:0]		RO	
03480	1024-MaxFrameSize byte Frames Transmitted OK low	[31:0]		RO	
03484	1024-MaxFrameSize byte Frames Transmitted OK high	[31:0]		RO	
03488	Oversize Frames Transmitted OK low	[31:0]		RO	
0348C	Oversize Frames Transmitted OK high	[31:0]		RO	
03490	Frames Received OK low	[31:0]		RO	
03494	Frames Received OK high	[31:0]		RO	
03498	Frame Check Sequence Errors low	[31:0]		RO	
0349C	Frame Check Sequence Errors high	[31:0]		RO	
034A0	Broadcast Frames Received OK low	[31:0]		RO	
034A4	Broadcast Frames Received OK high	[31:0]		RO	

SIFCFG: eMAC#0 STATISTIC Addr	Reg	Bits	Bit Field	Type	Description
034A8	Multicast Frames Received OK low	[31:0]		RO	
034AC	Multicast Frames Received OK high	[31:0]		RO	
034B0	Control Frames Received OK low	[31:0]		RO	
034B4	Control Frames Received OK high	[31:0]		RO	
034B8	Length/Type Out of Range low	[31:0]		RO	
034BC	Length/Type Out of Range high	[31:0]		RO	
034C0	VLAN Tagged Frames Received OK low	[31:0]		RO	
034C4	VLAN Tagged Frames Received OK high	[31:0]		RO	
034C8	Pause Frames Received OK low	[31:0]		RO	
034CC	Pause Frames Received OK high	[31:0]		RO	
034D0	Control Frames Received with Unsupported Opcode low	[31:0]		RO	
034D4	Control Frames Received with Unsupported Opcode high	[31:0]		RO	
034D8	Frames Transmitted OK low	[31:0]		RO	
034DC	Frames Transmitted OK high	[31:0]		RO	
034E0	Broadcast Frames Transmitted OK low	[31:0]		RO	
034E4	Broadcast Frames Transmitted OK high	[31:0]		RO	
034E8	Multicast Frames Transmitted OK low	[31:0]		RO	
034EC	Multicast Frames Transmitted OK high	[31:0]		RO	
034F0	Underrun Errors low	[31:0]		RO	
034F4	Underrun Errors high	[31:0]		RO	
034F8	Control Frames Transmitted OK low	[31:0]		RO	
034FC	Control Frames Transmitted OK high	[31:0]		RO	

SIFCFG: eMAC#0 STATISTIC Addr	Reg	Bits	Bit Field	Type	Description
03500	VLAN Tagged Frames Transmitted OK low	[31:0]		RO	
03504	VLAN Tagged Frames Transmitted OK high	[31:0]		RO	
03508	Pause Frames Transmitted OK low	[31:0]		RO	
0350C	Pause Frames Transmitted OK high	[31:0]		RO	

SIFCFG: eMAC#1 STATISTIC Addr	Reg	Bits	Bit Field	Type	Description
04400	Transmitted bytes low	[31:0]		RO	see Xilinx eMAC statistics user guide
04404	Transmitted bytes high	[31:0]		RO	
04408	Received bytes low	[31:0]		RO	
0440C	Received bytes high	[31:0]		RO	
04410	Undersize frames received low	[31:0]		RO	
04414	Undersize frames received high	[31:0]		RO	
04418	Fragment frames received low	[31:0]		RO	
0441C	Fragment frames received high	[31:0]		RO	
04420	64 byte Frames Received OK low	[31:0]		RO	
04424	64 byte Frames Received OK high	[31:0]		RO	
04428	65-127 byte Frames Received OK low	[31:0]		RO	
0442C	65-127 byte Frames Received OK high	[31:0]		RO	
04430	128-255 byte Frames Received OK low	[31:0]		RO	
04434	128-255 byte Frames Received OK high	[31:0]		RO	
04438	256-511 byte Frames Received OK low	[31:0]		RO	
0443C	256-511 byte Frames Received OK high	[31:0]		RO	
04440	512-1023 byte Frames Received OK low	[31:0]		RO	
04444	512-1023 byte Frames Received OK high	[31:0]		RO	
04448	1024-MaxFrameSize byte Frames Received OK low	[31:0]		RO	
0444C	1024-MaxFrameSize byte Frames Received OK high	[31:0]		RO	
04450	Oversize Frames Received OK low	[31:0]		RO	
04454	Oversize Frames Received OK high	[31:0]		RO	

SIFCFG: eMAC#1 STATISTIC Addr	Reg	Bits	Bit Field	Type	Description
04458	64 byte Frames Transmitted OK low	[31:0]		RO	
0445C	64 byte Frames Transmitted OK high	[31:0]		RO	
04460	65-127 byte Frames Transmitted OK low	[31:0]		RO	
04464	65-127 byte Frames Transmitted OK high	[31:0]		RO	
04468	128-255 byte Frames Transmitted OK low	[31:0]		RO	
0446C	128-255 byte Frames Transmitted OK high	[31:0]		RO	
04470	256-511 byte Frames Transmitted OK low	[31:0]		RO	
04474	256-511 byte Frames Transmitted OK high	[31:0]		RO	
04478	512-1023 byte Frames Transmitted OK low	[31:0]		RO	
0447C	512-1023 byte Frames Transmitted OK high	[31:0]		RO	
04480	1024-MaxFrameSize byte Frames Transmitted OK low	[31:0]		RO	
04484	1024-MaxFrameSize byte Frames Transmitted OK high	[31:0]		RO	
04488	Oversize Frames Transmitted OK low	[31:0]		RO	
0448C	Oversize Frames Transmitted OK high	[31:0]		RO	
04490	Frames Received OK low	[31:0]		RO	
04494	Frames Received OK high	[31:0]		RO	
04498	Frame Check Sequence Errors low	[31:0]		RO	
0449C	Frame Check Sequence Errors high	[31:0]		RO	
044A0	Broadcast Frames Received OK low	[31:0]		RO	
044A4	Broadcast Frames Received OK high	[31:0]		RO	
044A8	Multicast Frames Received OK low	[31:0]		RO	
044AC	Multicast Frames Received OK high	[31:0]		RO	

SIFCFG: eMAC#1 STATISTIC Addr	Reg	Bits	Bit Field	Type	Description
044B0	Control Frames Received OK low	[31:0]		RO	
044B4	Control Frames Received OK high	[31:0]		RO	
044B8	Length/Type Out of Range low	[31:0]		RO	
044BC	Length/Type Out of Range high	[31:0]		RO	
044C0	VLAN Tagged Frames Received OK low	[31:0]		RO	
044C4	VLAN Tagged Frames Received OK high	[31:0]		RO	
044C8	Pause Frames Received OK low	[31:0]		RO	
044CC	Pause Frames Received OK high	[31:0]		RO	
044D0	Control Frames Received with Unsupported Opcode low	[31:0]		RO	
044D4	Control Frames Received with Unsupported Opcode high	[31:0]		RO	
044D8	Frames Transmitted OK low	[31:0]		RO	
044DC	Frames Transmitted OK high	[31:0]		RO	
044E0	Broadcast Frames Transmitted OK low	[31:0]		RO	
044E4	Broadcast Frames Transmitted OK high	[31:0]		RO	
044E8	Multicast Frames Transmitted OK low	[31:0]		RO	
044EC	Multicast Frames Transmitted OK high	[31:0]		RO	
044F0	Underrun Errors low	[31:0]		RO	
044F4	Underrun Errors high	[31:0]		RO	
044F8	Control Frames Transmitted OK low	[31:0]		RO	
044FC	Control Frames Transmitted OK high	[31:0]		RO	
04500	VLAN Tagged Frames Transmitted OK low	[31:0]		RO	
04504	VLAN Tagged Frames Transmitted OK high	[31:0]		RO	

SIFCFG: eMAC#1 STATISTIC Addr	Reg	Bits	Bit Field	Type	Description
04508	Pause Frames Transmitted OK low	[31:0]		RO	
0450C	Pause Frames Transmitted OK high	[31:0]		RO	

SIFCFG: eMAC#2 STATISTIC Addr	Reg	Bits	Bit Field	Type	Description
05400	Transmitted bytes low	[31:0]		RO	see Xilinx eMAC statistics user guide
05404	Transmitted bytes high	[31:0]		RO	
05408	Received bytes low	[31:0]		RO	
0540C	Received bytes high	[31:0]		RO	
05410	Undersize frames received low	[31:0]		RO	
05414	Undersize frames received high	[31:0]		RO	
05418	Fragment frames received low	[31:0]		RO	
0541C	Fragment frames received high	[31:0]		RO	
05420	64 byte Frames Received OK low	[31:0]		RO	
05424	64 byte Frames Received OK high	[31:0]		RO	
05428	65-127 byte Frames Received OK low	[31:0]		RO	
0542C	65-127 byte Frames Received OK high	[31:0]		RO	
05430	128-255 byte Frames Received OK low	[31:0]		RO	
05434	128-255 byte Frames Received OK high	[31:0]		RO	
05438	256-511 byte Frames Received OK low	[31:0]		RO	
0543C	256-511 byte Frames Received OK high	[31:0]		RO	
05440	512-1023 byte Frames Received OK low	[31:0]		RO	
05444	512-1023 byte Frames Received OK high	[31:0]		RO	
05448	1024-MaxFrameSize byte Frames Received OK low	[31:0]		RO	
0544C	1024-MaxFrameSize byte Frames Received OK high	[31:0]		RO	
05450	Oversize Frames Received OK low	[31:0]		RO	
05454	Oversize Frames Received OK high	[31:0]		RO	

SIFCFG: eMAC#2 STATISTIC Addr	Reg	Bits	Bit Field	Type	Description
05458	64 byte Frames Transmitted OK low	[31:0]		RO	
0545C	64 byte Frames Transmitted OK high	[31:0]		RO	
05460	65-127 byte Frames Transmitted OK low	[31:0]		RO	
05464	65-127 byte Frames Transmitted OK high	[31:0]		RO	
05468	128-255 byte Frames Transmitted OK low	[31:0]		RO	
0546C	128-255 byte Frames Transmitted OK high	[31:0]		RO	
05470	256-511 byte Frames Transmitted OK low	[31:0]		RO	
05474	256-511 byte Frames Transmitted OK high	[31:0]		RO	
05478	512-1023 byte Frames Transmitted OK low	[31:0]		RO	
0547C	512-1023 byte Frames Transmitted OK high	[31:0]		RO	
05480	1024-MaxFrameSize byte Frames Transmitted OK low	[31:0]		RO	
05484	1024-MaxFrameSize byte Frames Transmitted OK high	[31:0]		RO	
05488	Oversize Frames Transmitted OK low	[31:0]		RO	
0548C	Oversize Frames Transmitted OK high	[31:0]		RO	
05490	Frames Received OK low	[31:0]		RO	
05494	Frames Received OK high	[31:0]		RO	
05498	Frame Check Sequence Errors low	[31:0]		RO	
0549C	Frame Check Sequence Errors high	[31:0]		RO	
054A0	Broadcast Frames Received OK low	[31:0]		RO	
054A4	Broadcast Frames Received OK high	[31:0]		RO	
054A8	Multicast Frames Received OK low	[31:0]		RO	
054AC	Multicast Frames Received OK high	[31:0]		RO	

SIFCFG: eMAC#2 STATISTIC Addr	Reg	Bits	Bit Field	Type	Description
054B0	Control Frames Received OK low	[31:0]		RO	
054B4	Control Frames Received OK high	[31:0]		RO	
054B8	Length/Type Out of Range low	[31:0]		RO	
054BC	Length/Type Out of Range high	[31:0]		RO	
054C0	VLAN Tagged Frames Received OK low	[31:0]		RO	
054C4	VLAN Tagged Frames Received OK high	[31:0]		RO	
054C8	Pause Frames Received OK low	[31:0]		RO	
054CC	Pause Frames Received OK high	[31:0]		RO	
054D0	Control Frames Received with Unsupported Opcode low	[31:0]		RO	
054D4	Control Frames Received with Unsupported Opcode high	[31:0]		RO	
054D8	Frames Transmitted OK low	[31:0]		RO	
054DC	Frames Transmitted OK high	[31:0]		RO	
054E0	Broadcast Frames Transmitted OK low	[31:0]		RO	
054E4	Broadcast Frames Transmitted OK high	[31:0]		RO	
054E8	Multicast Frames Transmitted OK low	[31:0]		RO	
054EC	Multicast Frames Transmitted OK high	[31:0]		RO	
054F0	Underrun Errors low	[31:0]		RO	
054F4	Underrun Errors high	[31:0]		RO	
054F8	Control Frames Transmitted OK low	[31:0]		RO	
054FC	Control Frames Transmitted OK high	[31:0]		RO	
05500	VLAN Tagged Frames Transmitted OK low	[31:0]		RO	
05504	VLAN Tagged Frames Transmitted OK high	[31:0]		RO	

SIFCFG: eMAC#2 STATISTIC Addr	Reg	Bits	Bit Field	Type	Description
05508	Pause Frames Transmitted OK low	[31:0]		RO	
0550C	Pause Frames Transmitted OK high	[31:0]		RO	

SIFCFG: eMAC#3 STATISTIC Addr	Reg	Bits	Bit Field	Type	Description
06400	Transmitted bytes low	[31:0]		RO	see Xilinx eMAC statistics user guide
06404	Transmitted bytes high	[31:0]		RO	
06408	Received bytes low	[31:0]		RO	
0640C	Received bytes high	[31:0]		RO	
06410	Undersize frames received low	[31:0]		RO	
06414	Undersize frames received high	[31:0]		RO	
06418	Fragment frames received low	[31:0]		RO	
0641C	Fragment frames received high	[31:0]		RO	
06420	64 byte Frames Received OK low	[31:0]		RO	
06424	64 byte Frames Received OK high	[31:0]		RO	
06428	65-127 byte Frames Received OK low	[31:0]		RO	
0642C	65-127 byte Frames Received OK high	[31:0]		RO	
06430	128-255 byte Frames Received OK low	[31:0]		RO	
06434	128-255 byte Frames Received OK high	[31:0]		RO	
06438	256-511 byte Frames Received OK low	[31:0]		RO	
0643C	256-511 byte Frames Received OK high	[31:0]		RO	
06440	512-1023 byte Frames Received OK low	[31:0]		RO	
06444	512-1023 byte Frames Received OK high	[31:0]		RO	
06448	1024-MaxFrameSize byte Frames Received OK low	[31:0]		RO	
0644C	1024-MaxFrameSize byte Frames Received OK high	[31:0]		RO	
06450	Oversize Frames Received OK low	[31:0]		RO	
06454	Oversize Frames Received OK high	[31:0]		RO	

SIFCFG: eMAC#3 STATISTIC Addr	Reg	Bits	Bit Field	Type	Description
06458	64 byte Frames Transmitted OK low	[31:0]		RO	
0645C	64 byte Frames Transmitted OK high	[31:0]		RO	
06460	65-127 byte Frames Transmitted OK low	[31:0]		RO	
06464	65-127 byte Frames Transmitted OK high	[31:0]		RO	
06468	128-255 byte Frames Transmitted OK low	[31:0]		RO	
0646C	128-255 byte Frames Transmitted OK high	[31:0]		RO	
06470	256-511 byte Frames Transmitted OK low	[31:0]		RO	
06474	256-511 byte Frames Transmitted OK high	[31:0]		RO	
06478	512-1023 byte Frames Transmitted OK low	[31:0]		RO	
0647C	512-1023 byte Frames Transmitted OK high	[31:0]		RO	
06480	1024-MaxFrameSize byte Frames Transmitted OK low	[31:0]		RO	
06484	1024-MaxFrameSize byte Frames Transmitted OK high	[31:0]		RO	
06488	Oversize Frames Transmitted OK low	[31:0]		RO	
0648C	Oversize Frames Transmitted OK high	[31:0]		RO	
06490	Frames Received OK low	[31:0]		RO	
06494	Frames Received OK high	[31:0]		RO	
06498	Frame Check Sequence Errors low	[31:0]		RO	
0649C	Frame Check Sequence Errors high	[31:0]		RO	
064A0	Broadcast Frames Received OK low	[31:0]		RO	
064A4	Broadcast Frames Received OK high	[31:0]		RO	
064A8	Multicast Frames Received OK low	[31:0]		RO	
064AC	Multicast Frames Received OK high	[31:0]		RO	

SIFCFG: eMAC#3 STATISTIC Addr	Reg	Bits	Bit Field	Type	Description
064B0	Control Frames Received OK low	[31:0]		RO	
064B4	Control Frames Received OK high	[31:0]		RO	
064B8	Length/Type Out of Range low	[31:0]		RO	
064BC	Length/Type Out of Range high	[31:0]		RO	
064C0	VLAN Tagged Frames Received OK low	[31:0]		RO	
064C4	VLAN Tagged Frames Received OK high	[31:0]		RO	
064C8	Pause Frames Received OK low	[31:0]		RO	
064CC	Pause Frames Received OK high	[31:0]		RO	
064D0	Control Frames Received with Unsupported Opcode low	[31:0]		RO	
064D4	Control Frames Received with Unsupported Opcode high	[31:0]		RO	
064D8	Frames Transmitted OK low	[31:0]		RO	
064DC	Frames Transmitted OK high	[31:0]		RO	
064E0	Broadcast Frames Transmitted OK low	[31:0]		RO	
064E4	Broadcast Frames Transmitted OK high	[31:0]		RO	
064E8	Multicast Frames Transmitted OK low	[31:0]		RO	
064EC	Multicast Frames Transmitted OK high	[31:0]		RO	
064F0	Underrun Errors low	[31:0]		RO	
064F4	Underrun Errors high	[31:0]		RO	
064F8	Control Frames Transmitted OK low	[31:0]		RO	
064FC	Control Frames Transmitted OK high	[31:0]		RO	
06500	VLAN Tagged Frames Transmitted OK low	[31:0]		RO	
06504	VLAN Tagged Frames Transmitted OK high	[31:0]		RO	

SIFCFG: eMAC#3 STATISTIC Addr	Reg	Bits	Bit Field	Type	Description
06508	Pause Frames Transmitted OK low	[31:0]		RO	
0650C	Pause Frames Transmitted OK high	[31:0]		RO	

4.3.6 eMAC Application Registers

These registers are part of the eMAC implementation and reside either in the packetizer / de-packetizer of the Ethernet blocks or in the global FPGA register bank. Through these registers the sending and receiving of Ethernet frames by/from SCC cores can be configured and controlled.

48 register sets exists per Ethernet block. These sets can be used by individual SCC cores or the MCPC. The route and destination ID specifies the location of the buffer and the source/sink of the traffic. Individual cores or the MCPC can use multiple sets in parallel to use more than one network interface at the same time.

SIFCFG: eMAC #0 RXCTRL Addr	Reg	Bits	Bit Field	Type	Description
09000 …090BC	RX Buffer Start Address 0…47	[28:0]		R/W	Upper 29 bit of physical start address [33:5] of memory buffer. Lower 5 bits are always 0 because access granularity to buffer will be always 32 bytes – one cache-line.
		[31:29]		R 0	Unused
09100 …091BC	RX Buffer Read Index 0…47	[15:0]		R/W	Read index of RX buffer Points to 32 byte entry in the buffer.
		[31:16]		R 0	Unused
09200 …092BC	RX Buffer Write Index 0…47	[15:0]		RO	Write index of RX buffer Points to 32 byte entry in the buffer
		[31:16]		R 0	Unused
09300 …093BC	RX Buffer Last Index 0…47	[15:0]		R/W	Last valid index in buffer. Buffer size = last index * 32 bytes
		[31:17]		R 0	Unused
09400 …094BC	Reserved	[31:0]		R 0	Unused

SIFCFG: eMAC #0 RXCTRL Addr	Reg	Bits	Bit Field	Type	Description
09500 …095BC	RX Buffer Route 0…47	[7:0]		R/W	Route to the tile where the MC for this buffer is located
	RX Buffer Destination ID 0…47	[10:8]		R/W	Defines the port at which the MC is connected
	Reserved	[15:11]		R 0	Unused
	RX Interrupt Route 0…47	[23:16]		R/W	Route to the tile where the core for this buffer is located
	RX Interrupt Destination ID 0…47	[26:24]		R/W	Selects core 0 or core 1 in the tile
	Reserved	[31:27]		R 0	Unused
09600 …096BC	Unused	[31:16]		R 0	Unused
	RX Network Port MAC Address 0…47 (high)	[15:0]		R/W	Higher 16 bits of MAC address
09700 …097BC	RX Network Port MAC Address 0…47 (low)	[31:0]		R/W	Lower 32 bits of MAC addresses
09800 …098BC	RX Network Port Enable 0…47	[0]		R/W	1 – Enabled 0 – Disabled
		[31:1]		R 0	Unused

SIFCFG: eMAC #1 RXCTRL Addr	Reg	Bits	Bit Field	Type	Description
0A000 …0A0BC	RX Buffer Start Address 0…47	[28:0]		R/W	Upper 29 bit of physical start address [33:5] of memory buffer. Lower 5 bits are always 0 because access granularity to buffer will be always 32 bytes – one cache-line.
		[31:29]		R 0	Unused
0A100 …0A1BC	RX Buffer Read Index 0…47	[15:0]		R/W	Read index of RX buffer Points to 32 byte entry in the buffer.
		[31:16]		R 0	Unused
0A200 …0A2BC	RX Buffer Write Index 0…47	[15:0]		RO	Write index for RX buffer Points to 32 byte entry in the buffer.
		[31:16]		R 0	Unused
0A300 …0A3BC	RX Buffer Last Index 0…47	[15:0]		R/W	Last valid index in buffer. Buffer size = last index * 32 bytes
		[31:17]		R 0	Unused
0A400 …0A4BC	Reserved	[31:0]		R 0	Unused
0A500 …0A5BC	RX Buffer Route 0…47	[7:0]		R/W	Route to the tile where the MC for this buffer is located
	RX Buffer Destination ID 0…47	[10:8]		R/W	Defines the port at which the MC is connected
	Reserved	[15:11]		R 0	Unused
	RX Interrupt Route 0…47	[23:16]		R/W	Route to the tile where the core for this buffer is located
	RX Interrupt Destination ID 0…47	[26:24]		R/W	Selects core 0 or core 1 in the tile

SIFCFG: eMAC #1 RXCTRL Addr	Reg	Bits	Bit Field	Type	Description
	Reserved	[31:27]		R 0	Unused
0A600 …0A6BC	Unused	[31:16]		R 0	Unused
	RX Network Port MAC Address 0…47 (high)	[15:0]		R/W	Higher 16 bits of MAC address
0A700 …0A7BC	RX Network Port MAC Address 0…47 (low)	[31:0]		R/W	Lower 32 bits of MAC addresses
0A800 …0A8BC	RX Network Port Enable 0…47	[0]		R/W	1 – Enabled 0 – Disabled
		[31:1]		R 0	Unused

SIFCFG: eMAC #2 RXCTRL Addr	Reg	Bits	Bit Field	Type	Description
0B000 …0B0BC	RX Buffer Start Address 0…47	[28:0]		R/W	Upper 29 bit of physical start address [33:5] of memory buffer. Lower 5 bits are always 0 because access granularity to buffer will be always 32 bytes – one cache-line.
		[31:29]		R 0	Unused
0B100 …0B1BC	RX Buffer Read Index 0…47	[15:0]		R/W	Read index of RX buffer Points to 32 byte entry in the buffer.
		[31:16]		R 0	Unused
0B200 …0B2BC	RX Buffer Write Index 0…47	[15:0]		RO	Write index of RX buffer Points to 32 byte entry in the buffer.
		[31:16]		R 0	Unused
0B300 …0B3BC	RX Buffer Last Index 0…47	[15:0]		R/W	Last valid index in buffer. Buffer size = last index * 32 bytes
		[31:17]		R 0	Unused
0B400 …0B4BC	Reserved	[31:0]		R 0	Unused
0B500 …0B5BC	RX Buffer Route 0…47	[7:0]		R/W	Route to the tile where the MC for this buffer is located
	RX Buffer Destination ID 0…47	[10:8]		R/W	Defines the port at which the MC is connected
	Reserved	[15:11]		R 0	Unused
	RX Interrupt Route 0…47	[23:16]		R/W	Route to the tile where the core for this buffer is located
	RX Interrupt Destination ID 0…47	[26:24]		R/W	Selects core 0 or core 1 in the tile

SIFCFG: eMAC #2 RXCTRL Addr	Reg	Bits	Bit Field	Type	Description
	Reserved	[31:27]		R 0	Unused
0B600 …0B6BC	Unused	[31:16]		R 0	Unused
	RX Network Port MAC Address 0…47 (high)	[15:0]		R/W	Higher 16 bits of MAC address
0B700 …0B7BC	RX Network Port MAC Address 0…47 (low)	[31:0]		R/W	Lower 32 bits of MAC addresses
0B800 …0B8BC	RX Network Port Enable 0…47	[0]		R/W	1 – Enabled 0 – Disabled
		[31:1]		R 0	Unused

SIFCFG: eMAC #3 RXCTRL Addr	Reg	Bits	Bit Field	Type	Description
0C000 …0C0BC	RX Buffer Start Address 0…47	[28:0]		R/W	Upper 29 bit of physical start address [33:5] of memory buffer. Lower 5 bits are always 0 because access granularity to buffer will be always 32 bytes – one cache-line.
		[31:29]		R 0	Unused
0C100 …0C1BC	RX Buffer Read Index 0…47	[15:0]		R/W	Read index of RX buffer Points to 32 byte entry in the buffer.
		[31:16]		R 0	Unused
0C200 …0C2BC	RX Buffer Write Index 0…47	[15:0]		RO	Write index of RX buffer Points to 32 byte entry in the buffer.
		[31:16]		R 0	Unused
0C300 …0C3BC	RX Buffer Last Index 0…47	[15:0]		R/W	Last valid index in buffer. Buffer size = last index * 32 bytes
		[31:17]		R 0	Unused
0C400 …0C4BC	Reserved	[31:0]		R/W	Unused
0C500 …0C5BC	RX Buffer Route 0…47	[7:0]		R/W	Route to the tile where the MC for this buffer is located
	RX Buffer Destination ID 0…47	[10:8]		R/W	Defines the port at which the MC is connected
	Reserved	[15:11]		R 0	Unused
	RX Interrupt Route 0…47	[23:16]		R/W	Route to the tile where the core for this buffer is located
	RX Interrupt Destination ID 0…47	[26:24]		R/W	Selects core 0 or core 1 in the tile

SIFCFG: eMAC #3 RXCTRL Addr	Reg	Bits	Bit Field	Type	Description
	Reserved	[31:27]		R 0	Unused
0C600 …0C6BC	Unused	[31:16]		R 0	Unused
	RX Network Port MAC Address 0…47 (high)	[15:0]		R/W	Higher 16 bits of MAC address
0C700 …0C7BC	RX Network Port MAC Address 0…47 (low)	[31:0]		R/W	Lower 32 bits of MAC addresses
0C800 …0C8BC	RX Network Port Enable 0…47	[0]		R/W	1 – Enabled 0 – Disabled
		[31:1]		R 0	Unused

SIFCFG: eMAC #0 TXCTRL Addr	Reg	Bits	Bit Field	Type	Description
09900 …099BC	TX Buffer Start Address 0…47	[28:0]		R/W	Upper 29 bit of physical start address [33:5] of memory buffer. Lower 5 bits are always 0 because access granularity to buffer will be always 32 bytes – one cache-line.
		[31:29]		R 0	Unused
09A00 …09ABC	TX Buffer Read Index 0…47	[15:0]		RO	Read index of TX buffer Points to 32 byte entry in the buffer
		[31:16]		R 0	Unused
09B00 …09BBC	TX Buffer Write Index 0…47	[15:0]		R/W	Write index of TX buffer Points to 32 byte entry in the buffer.
		[31:16]		R 0	Unused
09C00 …09CBC	TX Buffer Last Index 0…47	[16:0]		R/W	Last valid index in buffer. Buffer size = last index * 32 bytes
		[31:17]		R 0	Unused
09D00 …09DBC	TX Buffer Route 0…47	[7:0]		R/W	Route to the tile where the MC for this buffer is located
	TX Buffer Destination ID 0…47	[10:8]		R/W	Defines the port at which the MC is connected
	Unused	[31:11]		R 0	Unused
09E00 …09EBC	TX Network Port Enable 0…47	[0]		R/W	1 – Enabled 0 – Disabled
		[31:1]		R 0	Unused

SIFCFG: eMAC #1 TXCTRL Addr	Reg	Bits	Bit Field	Type	Description
0A900 …0A9BC	TX Buffer Start Address 0…47	[28:0]		R/W	Upper 29 bit of physical start address [33:5] of memory buffer. Lower 5 bits are always 0 because access granularity to buffer will be always 32 bytes – one cache-line.
		[31:29]		R 0	Unused
0AA00 …0AABC	TX Buffer Read Index 0…47	[15:0]		RO	Read index of TX buffer Points to 32 byte entry in the buffer.
		[31:16]		R 0	Unused
0AB00 …0ABBC	TX Buffer Write Index 0…47	[15:0]		R/W	Write index of TX buffer Points to 32 byte entry in the buffer.
		[31:16]		R 0	Unused
0AC00 …0ACBC	TX Buffer Last Index 0…47	[16:0]		R/W	Last valid index in buffer. Buffer size = last index * 32 bytes
		[31:17]		R 0	Unused
0AD00 …0ADBC	TX Buffer Route 0…47	[7:0]		R/W	Route to the tile where the MC for this buffer is located
	TX Buffer Destination ID 0…47	[10:8]		R/W	Defines the port at which the MC is connected
	Unused	[31:11]		R 0	Unused
0AE00 …0AEBC	TX Network Port Enable 0…47	[0]		R/W	1 – Enabled 0 – Disabled
		[31:1]		R 0	Unused

SIFCFG: eMAC #2 TXCTRL Addr	Reg	Bits	Bit Field	Type	Description
0B900 …0B9BC	TX Buffer Start Address 0…47	[28:0]		R/W	Upper 29 bit of physical start address [33:5] of memory buffer. Lower 5 bits are always 0 because access granularity to buffer will be always 32 bytes – one cache-line.
		[31:29]		R 0	Unused
0BA00 …0BABC	TX Buffer Read Index 0…47	[15:0]		RO	Read index of TX buffer Points to 32 byte entry in the buffer.
		[31:16]		R 0	Unused
0BB00 …0BBBC	TX Buffer Write Index 0…47	[15:0]		R/W	Write index of TX buffer Points to 32 byte entry in the buffer.
		[31:16]		R 0	Unused
0BC00 …0BCBC	TX Buffer Last Index 0…47	[16:0]		R/W	Last valid index in buffer. Buffer size = last index * 32 bytes
		[31:17]		R 0	Unused
0BD00 …0BDBC	TX Buffer Route 0…47	[7:0]		R/W	Route to the tile where the MC for this buffer is located
	TX Buffer Destination ID 0…47	[10:8]		R/W	Defines the port at which the MC is connected
	Unused	[31:11]		R 0	Unused
0BE00 …0BEBC	TX Network Port Enable 0…47	[0]		R/W	1 – Enabled 0 – Disabled
		[31:1]		R 0	Unused

SIFCFG: eMAC #3 TXCTRL Addr	Reg	Bits	Bit Field	Type	Description
0C900 …0C9BC	TX Buffer Start Address 0…47	[28:0]		R/W	Upper 29 bit of physical start address [33:5] of memory buffer. Lower 5 bits are always 0 because access granularity to buffer will be always 32 bytes – one cache-line.
		[31:29]		R 0	Unused
0CA00 …0CABC	TX Buffer Read Index 0…47	[15:0]		RO	Read index of TX buffer Points to 32 byte entry in the buffer.
		[31:16]		R 0	Unused
0CB00 …0CBBC	TX Buffer Write Index 0…47	[15:0]		R/W	Write index of TX buffer Points to 32 byte entry in the buffer.
		[31:16]		R 0	Unused
0CC00 …0CCBC	TX Buffer Last Index 0…47	[16:0]		R/W	Last valid index in buffer. Buffer size = last index * 32 bytes
		[31:17]		R 0	Unused
0CD00 …0CDBC	TX Buffer Route 0…47	[7:0]		R/W	Route to the tile where the MC for this buffer is located
	TX Buffer Destination ID 0…47	[10:8]		R/W	Defines the port at which the MC is connected
	Unused	[31:11]		R 0	Unused
0CE00 …0CEBC	TX Network Port Enable 0…47	[0]		R/W	1 – Enabled 0 – Disabled
		[31:1]		R 0	Unused

SIFCFG: eMAC #0 STATUS Addr	Reg	Bits	Bit Field	Type	Description
07000 …0702F	RX Frame Dropped Channel 0..47	[0]		RW	Set when frame dropped due to RX buffer overflow in DDR3
		[31:1]		R 0	Unused
07030	TX Fifo Buffer Full	[0]		RO	Set in case of an overflow of TX fifo buffer in eMAC IP block
		[31:1]		R 0	Unused
07034	RX Fifo Buffer Full	[0]		RO	Set in case of an overflow of RX fifo buffer in eMAC IP block
		[31:1]		R 0	Unused
07038	TX eMAC not ready	[0]		RO	eMAC does not accept frames
		[31:1]		R 0	Unused
0703C	RX MAC address error	[0]		RO	MAC address not found in table
		[31:1]		R 0	Unused
07040	RX MAC address lower part	[31:0]		RO	Lower 32 bits of MAC address that has not been found
07044	RX MAC address upper part	[15:0]		RO	Upper 16 bits of MAC address that has not been found

SIFCFG: eMAC #1 STATUS Addr	Reg	Bits	Bit Field	Type	Description
07100 …0712F	RX Frame Dropped Channel 0..47	[0]		RW	Set when frame dropped due to RX buffer overflow in DDR3
		[31:1]		R 0	Unused
07130	TX Fifo Buffer Full	[0]		RO	Set in case of an overflow of TX fifo buffer in eMAC IP block
		[31:1]		R 0	Unused
07134	RX Fifo Buffer Full	[0]		RO	Set in case of an overflow of RX fifo buffer in eMAC IP block
		[31:1]		R 0	Unused
07138	TX eMAC not ready	[0]		RO	eMAC does not accept frames
		[31:1]		R 0	Unused
0713C	RX MAC address error	[0]		RO	MAC address not found in table
		[31:1]		R 0	Unused
07140	RX MAC address lower part	[31:0]		RO	Lower 32 bits of MAC address that has not been found
07144	RX MAC address upper part	[15:0]		RO	Upper 16 bits of MAC address that has not been found

SIFCFG: eMAC #2 STATUS Addr	Reg	Bits	Bit Field	Type	Description
07200 …0722F	RX Frame Dropped Channel 0..47	[0]		RW	Set when frame dropped due to RX buffer overflow in DDR3
		[31:1]		R 0	Unused
07230	TX Fifo Buffer Full	[0]		RO	Set in case of an overflow of TX fifo buffer in eMAC IP block
		[31:1]		R 0	Unused
07234	RX Fifo Buffer Full	[0]		RO	Set in case of an overflow of RX fifo buffer in eMAC IP block
		[31:1]		R 0	Unused
07238	TX eMAC not ready	[0]		RO	eMAC does not accept frames
		[31:1]		R 0	Unused
0723C	RX MAC address error	[0]		RO	MAC address not found in table
		[31:1]		R 0	Unused
07240	RX MAC address lower part	[31:0]		RO	Lower 32 bits of MAC address that has not been found
07244	RX MAC address upper part	[15:0]		RO	Upper 16 bits of MAC address that has not been found

SIFCFG: eMAC #3 STATUS Addr	Reg	Bits	Bit Field	Type	Description
07300 …0732F	RX Frame Dropped Channel 0..47	[0]		RW	Set when frame dropped due to RX buffer overflow in DDR3
		[31:1]		R 0	Unused
07330	TX Fifo Buffer Full	[0]		RO	Set in case of an overflow of TX fifo buffer in eMAC IP block
		[31:1]		R 0	Unused
07334	RX Fifo Buffer Full	[0]		RO	Set in case of an overflow of RX fifo buffer in eMAC IP block
		[31:1]		R 0	Unused
07338	TX eMAC not ready	[0]		RO	eMAC does not accept frames
		[31:1]		R 0	Unused
0733C	RX MAC address error	[0]		RO	MAC address not found in table
		[31:1]		R 0	Unused
07340	RX MAC address lower part	[31:0]		RO	Lower 32 bits of MAC address that has not been found
07344	RX MAC address upper part	[15:0]		RO	Upper 16 bits of MAC address that has not been found

SIFCFG: eMAC CTRL Addr	Reg	Bits	Bit Field	Type	Description
07e00	MAC Base Address Register (high)	[31:16]		R 0	Unused
		[15:0]		R/W	Upper 16 bit of base MAC address
07e04	MAC Base Address Register (low)	[31:0]		R/W	Lower 32 bits of MAC addresses
07e08	Start IP Address of SCC network	[31:0]		R/W	IP Address for SCC Core 0. IPs for other cores are assigned in order.
07e0C	Host IP Address	[31:0]		R/W	IP Address of Host computer where /shared is mounted
07e10	Host Gateway Address	[31:0]		R/W	Gateway of Host computer

4.3.7 Interrupt Registers

SIFCFG: IRQ Addr	Reg	Bits	Bit Field	Type	Description
0D000	Interrupt Status Core 0	[31:0]		RO	High-active status set by HW can be read. [5:0] = SATA1..SATA0, eMAC3..eMAC0 [31:6] = IPI25…IPI0
0D004	Interrupt Status Core 0	[22:0]		RO	High-active status set by HW can be read. [22:0] = MCPC, IPI47…IPI26
		[31:23]		R0	Unused
…	…				
0D178	Interrupt Status Core 47	[31:0]		RO	High-active status set by HW can be read. [5:0] = SATA1..SATA0, eMAC3..eMAC0 [31:6] = IPI25…IPI0
0D17C	Interrupt Status Core 47	[22:0]		RO	High-active status set by HW can be read. [22:0] = MCPC, IPI47…IPI26
		[31:23]		R0	Unused
0D200	Interrupt Mask Core 0	[31:0]		R/W	High-active mask bit to disable interrupt delivery individually [5:0] = SATA1..SATA0, eMAC3..eMAC0 [31:6] = IPI25…IPI0
0D204	Interrupt Mask Core 0	[22:0]		R/W	High-active mask bit to disable interrupt delivery individually [22:0] = MCPC, IPI47…IPI26

SIFCFG: IRQ Addr	Reg	Bits	Bit Field	Type	Description
		[31:23]		R0	Unused
…	…				
0D378	Interrupt Mask Core 47	[31:0]		R/W	High-active mask bit to disable interrupt delivery individually [5:0] = SATA1..SATA0, eMAC3..eMAC0 [31:6] = IPI25…IPI0
0D37C	Interrupt Mask Core 47	[22:0]		R/W	High-active mask bit to disable interrupt delivery individually [22:0] = MCPC, IPI47…IPI26
		[31:23]		R0	Unused
0D400	Interrupt Reset Core 0	[31:0]		WO	Writing 1 to a bit position clears the according bit in the status register. [5:0] = SATA1..SATA0, eMAC3..eMAC0 [31:6] = IPI25…IPI0
0D404	Interrupt Reset Core 0	[22:0]		WO	Writing 1 to a bit position clears the according bit in the status register. [22:0] = MCPC, IPI47…IPI26
		[31:23]		R0	Unused
…	…				
0D578	Interrupt Reset Core 47	[31:0]		WO	Writing 1 to a bit position clears the according bit in the status register. [5:0] = SATA1..SATA0, eMAC3..eMAC0 [31:6] = IPI25…IPI0
0D57C	Interrupt Reset Core 47	[22:0]		WO	Writing 1 to a bit position clears the according bit in the status

SIFCFG: IRQ Addr	Reg	Bits	Bit Field	Type	Description
					register. [22:0] = MCPC, IPI47…IPI26
		[31:23]		R0	Unused
0D600	IPI Request Core 0	[31:0]		WO	Writing 1 to a bit position sets the according bit in the status register. [31:0] = IPI31…IPI0
0D604	IPI Request Core 0	[15:0]		WO	Writing 1 to a bit position sets the according bit in the status register. [15:0] = IPI47…IPI32
		[31:16]		R0	Unused
…	…				
0D778	IPI Request Core 47	[31:0]		WO	Writing 1 to a bit position sets the according bit in the status register. [31:0] = IPI31…IPI0
0D77C	IPI Request Core 47	[15:0]		WO	Writing 1 to a bit position sets the according bit in the status register. [15:0] = IPI47…IPI32
		[31:16]		R0	Unused
0D800 …0D8BC	Interrupt Config Core 0…47	[0]		RW	Configures the local interrupt pins of the core that will be used 0 – LINT 0 1 – LINT 1
		[31:1]		RO	Reserved
0D900	IRQ Request MCPC	[31:0]		WO	Writing 1 to a bit position sets the according bit in the status register. [31:0] = MCPC31…MCPC0
0D904	IRQ Request MCPC	[15:0]		WO	Writing 1 to a bit position sets the according bit in the status register. [15:0] = MCPC47…MCPC32
		[31:16]		R0	Unused

4.3.8 Atomic Increment Counters

SIFCFG: TSET Addr	Reg	Bits	Bit Field	Type	Description
0E000	Atomic Increment Counter #0	[31:0]		R/W	Read access causes an atomic increment of the register value. The old value gets returned. Write access causes an atomic decrement of the register value.
0E004	Initialization Counter #0	[31:0]		R/W	Read access returns the current register value. Write access initializes the register with new value.
0E008	Atomic Increment Counter #1	[31:0]		R/W	Read access causes an atomic increment of the register value. The old value gets returned. Write access causes an atomic decrement of the register value.
0E00C	Initialization Counter #1	[31:0]		R/W	Read access returns the current register value. Write access initializes the register with new value.
...0E178	Atomic Increment Counter #47	[31:0]		R/W	Read access causes an atomic increment of the register value. The old value gets returned. Write access causes an atomic decrement of the register value.
…0E17C	Initialization Counter #47	[31:0]		R/W	Read access returns the current register value. Write access initializes the register with new value.
0F000	Atomic Increment Counter #48	[31:0]		R/W	Read access causes an atomic increment of the register value. The old value gets returned. Write access causes an atomic decrement of the register value.
0F004	Initialization Counter #48	[31:0]		R/W	Read access returns the current register value. Write access initializes the register with new

SIFCFG: TSET Addr	Reg	Bits	Bit Field	Type	Description
					value.
0F008	Atomic Increment Counter #49	[31:0]		R/W	Read access causes an atomic increment of the register value. The old value gets returned. Write access causes an atomic decrement of the register value.
0F00C	Initialization Counter #49	[31:0]		R/W	Read access returns the current register value. Write access initializes the register with new value.
...0F178	Atomic Increment Counter #95	[31:0]		R/W	Read access causes an atomic increment of the register value. The old value gets returned. Write access causes an atomic decrement of the register value.
…0F17C	Initialization Counter #95	[31:0]		R/W	Read access returns the current register value. Write access initializes the register with new value.