rambat — Programing Interface rev. 0.0-rc1

Krzysztof Mazur

1 Introduction

This document describes the programing interface of the Rambat Random Access Memory controller.

The Rambat programming interface is designed to use with PCI-compatible buses, including PCI, PCI-X, PCI Express, PCI/104 and ARBus.

1.1 New revisions

This is the revision 0.0 of this specification. The first number is the major revision number. The major revision number identifies the hardware interface version. Two documentations with the same major revision define the same hardware interface. The second number is the minor revision. This number identifies the version of the documentation of the same hardware interface.

Latest revision of this document is available at `http://www.microster.pl/doc/rambat`.

1.2 Copyright

2 General description

The Rambat is a Random Access Memory (RAM) controller. The RAM is divided into power-of-two size pages. The page size is implementation specific. The page is accessible by a RAM Access Window. The visible page can be selected by the Rambat Page (`RAMBAT_PAGE`) register.

The RAM memory parameters are autoconfigured. The page size is equal to the Base Address Region 1 size. The number of pages can be detected by writting maximal value to the Rambat Page (`RAMBAT_PAGE`) register and read it back as a maximal supported page.

3 Configuration Registers

The rambat controller have 256-bytes of a PCI-compatible Configuration Address Space (see Fig. 1). Access for this address space is usually provided by a bus interface driver in the operating system. If not see the bus interface specification for information about how to access the Configuration Address Space on your system.

3.1 Vendor ID

Offset	0x00
Width	16 bit
Type	RO
Reset value	0xff00

The Vendor ID identifies the manufacturer of the device. For this device this register is equal to 0xff00.

3.2 Device ID

Offset	0x02
Width	16 bit
Type	RO
Reset value	0x0009

The Device ID identifies the device model. For this device this register is equal to 0x0009.

Offset	3	2	1	0
0x00	Device ID (0x0009)		Vendor ID (0xff00)	
0x04	Status		Command	
0x08	Base Class (0x05)	Sub-class (0x00)	ProgIF (0x00)	Revision ID
0x0c	Reserved	Header Type	Reserved	Reserved
0x10	Base Address Register 0			
0x14	Base Address Register 1			
0x18–0x2b	Reserved			
0x2c	Subsystem Device ID		Subsystem Vendor ID	
0x30–0xef	Reserved			
0xf0[1]	0x53	0x42	0x52	0x41
0xf4	Reserved			
0xf8[1]	Reserved		ARBus Command	
0xfc	Reserved			

[1]The configuration address space above 0xf0 is used only devices using the ARBus bus interface. For devices using other interfaces this area is reserved.

Figure 1: Rambat Configuration Address Space.

3.3 Command

Offset	0x04
Width	16 bit
Type	RW
Reset value	0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
`RESV`														`MEM`	`IO`
RO														RW	RO

Command Register Bit Descriptions

Bit	Name	Description
15–2	`RESV`	**Reserved**.
1	`MEM`	**Memory Space**. Set to enable decoding of Memory Regions
0	`IO`	**IO Space**. Unused, always equal to zero.

3.4 Status

Offset	0x06
Width	16 bit
Type	RO
Reset value	0x0000

The Status register is always equal to 0.

3.5 Revision ID

Offset	0x08
Width	8 bit
Type	RO
Reset value	Implementation specific

The Revision ID identifies the revision of the device.

3.6 ProgIF

Offset	0x09
Width	8 bit
Type	RO
Reset value	0x00

The ProgIF register identifies the programming interface in a specified class of the device. This device reports programming interface as 0x00.

3.7 Sub-class

Offset	0x0a
Width	8 bit
Type	RO
Reset value	0x80

The Sub-class register identifies the sub-class of the device. This device uses class 0x05 and sub-class 0x80 — RAM memory.

3.8 Base Class

Offset	0x0b
Width	8 bit
Type	RO
Reset value	0x05

The Base Class register identifies the class of the device. This device uses class 0x05 — Memory controller.

3.9 Header Type

Offset	0x0e
Width	8 bit
Type	RO
Reset value	0x00 or 0x80

The Header Type identifies the type of the Configuration Space header. This field is equal to 0x00 in single function devices or 0x80 in multifunction devices.

3.10 Base Address Register 0

Note: This subsection describes a 32-bit Base Address Register. Some implementations may support 64-bit Base Address Registers and such register will occupy two words. See PCI Specification for detailed description of Base Address Registers.

Offset	0x10
Width	32 bit
Type	RW
Reset value	implementation specific

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR												P	TYPE		IO
RW	RO											RO	RO		RO

Base Address Register 0 Bit Descriptions

Bit	Name	Description
31–4	ADDR	**Address**. The ADDR sets the bits 31–4 of region 0 base address. Note that on some buses like ARBus the address may be limited and higher bits are always cleared. Some least significant bits are always cleared to indicate region size.
3	P	**Prefetchable**. implementation specific
2-1	TYPE	**Type**. implementation specific, 0 (32-bit base address) is assumed in this documentation
0	IO	**IO Space indicator**. cleared to indicate Memory Space

3.11 Base Address Register 1

Note: This subsection describes a 32-bit Base Address Register. Some implementations may support 64-bit Base Address Registers and such register will occupy two words. See PCI Specification for detailed description of Base Address Registers.

Offset	0x14
Width	32 bit
Type	RW
Reset value	implementation specific

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR												P	TYPE		IO
RW												RO	RO		RO

Base Address Register 1 Bit Descriptions

Bit	Name	Description
31–4	ADDR	**Address**. The ADDR sets the bits 31–4 of region 0 base address. Note that on some buses like ARBus the address may be limited and higher bits are always cleared. Some least significant bits are always cleared to indicate region size.
3	P	**Prefetchable**. implementation specific
2-1	TYPE	**Type**. implementation specific, 0 (32-bit base address) is assumed in this documentation
0	IO	**IO Space indicator**. cleared to indicate Memory Space

3.12 Subsystem Vendor ID

Offset	0x2c
Width	16 bit
Type	RO
Reset value	Subsystem specific

The Subsystem Vendor ID is assigned by the expansion board or the subsystem vendor.

3.13 Subsystem ID

Offset	0x2e
Width	16 bit
Type	RO
Reset value	Subsystem specific

The Subsystem ID is assigned by the expansion board or the subsystem vendor.

3.14 ARBus Command

Offset	0xfa
Width	16 bit
Type	RW
Reset value	0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESV														BAR1_16	BAR0_16
RO														RW	RO

ARBus Command Register Bit Descriptions

Bit	Name	Description
15–2	RESV	**Reserved**.
1	BAR1_16	**BAR1 16-bit access**. Set to enable 16-bit access to Region 1. This bit can be hard-wired to 0 or 1 for cards that supports only 8-bit or 16-bit access.
0	BAR0_16	**BAR0 16-bit access**. Set to enable 16-bit access to Region 0. This bit can be hard-wired to 0 or 1 for cards that supports only 8-bit or 16-bit access.

4 Region 0 — Runtime registers

Runtime registers Register Summary

Offset	Type	Register	Reset value	Description
0x00	RW	Rambat Page (RAMBAT_PAGE)	0x00000000	4.1

4.1 Rambat Page (RAMBAT_PAGE)

Offset	0x00
Width	32 bit
Type	RW
Reset value	0x00000000

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

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

Rambat Page Bit Descriptions

Bit	Name	Description
31–0	PAGE	**Rambat Page**. The selected page visible in a Memory window. If the number of pages is a power of two the hardware should hard-wire unsupported bits to 0. If the number of pages is not a power of two the hardware must saturate value written to this register. Any partial write at offset 0 clears the higher bits of the register.

To detect number of pages the user sound write 0xffffffff to the Rambat Page (`RAMBAT_PAGE`) register and read-back the Rambat Page (`RAMBAT_PAGE`) register. The value is the maximal supported page. The number of pages is equal to the maximal supported page plus one. The host driver may also use just 8 bits or 16 bits of this register and use 0xff or 0xffff for probing the number of pages.

5 Region 1 — RAM Access Window

The selected page of the rambat memory is visible as the Memory Region 1. The visible page can be changed by setting the Rambat Page (`RAMBAT_PAGE`) register. The page size is equal to the memory region size.