IMP4 — Programing Interface rev. 0.0

Krzysztof Mazur

1 Introduction

This document describes the programing interface of the IMP4 bidirectional counters controller. The IMP4 interface supports up to 255 fully independent 32-bit counters.

The IMP4 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/imp4`.

1.2 Copyright

2 General description

The IMP4 interface groups up to 255 bidirectional counters in a single PCI device. Each counter can be read and written independently. Each counter have two independent state registers: the host-accessible Counter Value (`IMP4_DATA`) register and the internal counter state. The internal counter state register is used by hardware for counting and it's not directly visible by the host.

The Counter Value (`IMP4_DATA`) register is controlled by the host. The following operations are supported:

- read — read of any part of the Counter Value (`IMP4_DATA`) register,
- write — write of any part of the Counter Value (`IMP4_DATA`) register,
- atomic register transfer from the internal counter state — read of the Counter Latch (`IMP4_LATCH`) register,
- atomic register transfer to the internal counter state — write of the Counter Set (`IMP4_SET`) register.

To read state of the internal counter the host must read the Counter Latch (`IMP4_LATCH`) register, which copies the internal counter state to the Counter Value (`IMP4_DATA`) register, and read result from the Counter Value (`IMP4_DATA`) register.

To set state of the internal counter the host must set new value to the Counter Value (`IMP4_DATA`) and write to the Counter Set (`IMP4_SET`) register. The write support in the device is, however, optional. The hardware may provide absolute read-only counters.

3 Configuration Registers

The IMP4 controller have 256-bytes of 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.

[1] This signature at 0xf0 is used only devices using ARBus bus interface. For devices using other interfaces this area is reserved.

Offset	3	2	1	0
0x00	Device ID (0x0011)		Vendor ID (0xff00)	
0x04	Status		Command	
0x08	Base Class (0x11)	Sub-class (0x80)	ProgIF (0x00)	Revision ID
0x0c	Reserved	Header Type	Reserved	Reserved
0x10	Base Address Register 0			
0x14–0x2b	Reserved			
0x2c	Subsystem Device ID		Subsystem Vendor ID	
0x30–0x3f	Reserved			
0x40	Number of Counters	Reserved		
0x44–0xef	Reserved			
0xf0[1]	0x53	0x42	0x52	0x41
0xf4-0xff	Reserved			

Figure 1: IMP4 Configuration Address Space.

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	0x0011

The Device ID identifies the device model. For this device this register is equal to 0x0011.

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 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 0x11 and sub-class 0x80 — Signal Processing Controller.

3.8 Base Class

Offset	0x0b
Width	8 bit
Type	RO
Reset value	0x11

The Base Class register identifies the class of the device. This device uses class 0x11 — Signal Processing Controller.

3.9 Header Type

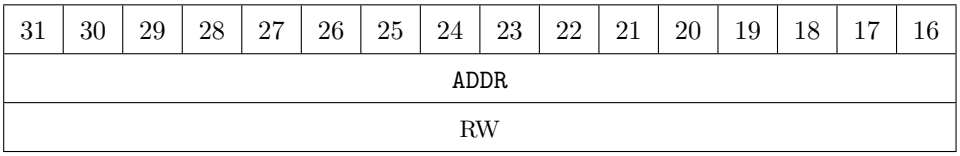
Offset	0x0e
Width	8 bit
Type	RO
Reset value	0x00 or 0x80

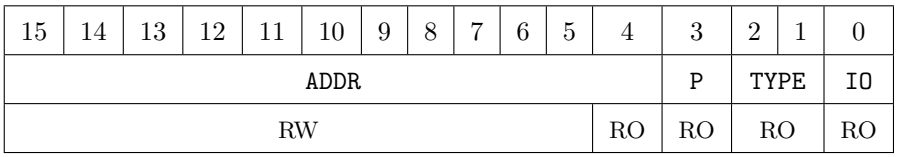
The Header Type identifies the type of Configuration Space header. This field is equal to 0x00 in single function devices or 0x80 in multifunction devices.

3.10 Base Address Register 0

Note: This subsection describes 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





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 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.

3.12 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.

3.13 Number of Counters

Offset	0x40
Width	8 bit
Type	RO
Reset value	Subsystem specific

The Number of Counters register reports number of counters implemented in the device.

4 Region 0 — Runtime registers

Each counter have separate independent set of runtime registers. The offset in Memory Region 1 for each counter is shown in following table:

Runtime Register Summary

Offset	Type	Register	Reset value	Description
0x00	—	Counter 0	—	4.1
0x08	—	Counter 1	—	4.1
0x10	—	Counter 2	—	4.1
0x18	—	Counter 3	—	4.1
⋮	—	⋮	—	4.1
0x08·i	—	Counter i	—	4.1
⋮	—	⋮	—	4.1
0x08·n	—	Counter n	—	4.1

4.1 Counter registers

Global Register Summary

Offset	Type	Register	Reset value	Description
0x00	RW	Counter Value (`IMP4_DATA`)	0x00000000	4.1.1
0x04	RO	Latch Counter (`IMP4_LATCH`)	0x00	4.1.2
0x04	WO	Set Counter (`IMP4_SET`)	—	4.1.3

4.1.1 Counter Value (`IMP4_DATA`)

Counter runtime 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
VALUE															
RW															

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

Counter Value Bit Descriptions

Bit	Name	Description
31–0	`VALUE`	**Counter Value**. The host-visible counter value. This register is not automatically updated by the hardware. The counter value must be explicitly updated by reading the Counter Latch (`IMP4_LATCH`) register.

4.1.2 Counter Latch (`IMP4_LATCH`)

Counter runtime offset	0x04
Width	8 bit
Type	RO
Reset value	0x00

7	6	5	4	3	2	1	0
RESV							
RO							

ADC Command Control (`IMP4_LATCH`) Bit Descriptions

Bit	Name	Description
7-0	`RESV`	**Reserved**.

Read from the Counter Latch (`IMP4_LATCH`) register transfers value from the internal counter value to the Counter Value register (`IMP4_DATA`).

4.1.3 Counter Set (`IMP4_SET`)

Counter runtime offset	0x04
Width	8 bit
Type	WO
Reset value	—

7	6	5	4	3	2	1	0
RESV							
WO							

Counter Set (`IMP4_SET`) Bit Descriptions

Bit	Name	Description
7-0	`RESV`	**Reserved**. Value is ignored

Write to Counter Set (`IMP4_SET`) register transfers value from from the Counter Value register (`IMP4_DATA`) to internal counter state. This operation may be ignored in case of absulute counters.