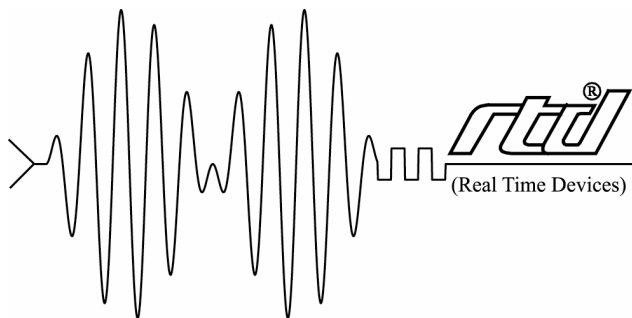


Directly Accessing the SSD Control Registers on RTD cpuModules



RTD Embedded Technologies, Inc.

"Accessing the Analog World"®

SWM-64000007
Rev. A

ISO9001 and AS9100 Certified

Page 1 of 8

Revision History

Rev. A	10/06/2003	New manual naming method
Rev. B	04/30/2007	Added register map and description for the 786-series (VIA Chipset). Re-wrote the section for the 686-series (Geode Chipset) to improve clarity. Re-worded the Introduction section to better explain the purpose/scope of the document. Removed references to the now-discontinued 386-series. Removed the obsolete ANCxxx numbering convention. Title changed to more accurately reflect the content.

Published by:

RTD Embedded Technologies, Inc.
103 Innovation Blvd.
State College, PA 16803
<http://www.rtd.com>

Copyright 2007 by RTD Embedded Technologies, Inc.
All rights reserved.

The RTD Logo is a registered trademark of RTD Embedded Technologies. cpuModule and utilityModule are trademarks of RTD Embedded Technologies. PC/104 is a registered trademark of PC/104 Consortium. All other trademarks appearing in this document are the property of their respective owners.

Introduction

Some RTD cpuModules feature a 32-pin DIP Solid State Disk (SSD) socket. Some cpuModules feature two SSD sockets. The SSD socket, coupled with special routines in the RTD Enhanced BIOS, allows you to install a memory chip, and use it as a floppy. This allows you to use boot the cpuModule without an external storage device (floppy, hard drive, etc).

The SSD socket can be configured for two different modes of operation: SSD mode, and BIOS Extension mode. In SSD mode, the socket works as a paged memory disk, and relies on the BIOS disk routines to virtualize a floppy disk. In BIOS extension mode, the device is memory mapped to a set address (e.g. D000:0000). At that address, a BIOS Extension device, such as an M-Systems DiskOnChip may be used.

SSD mode is only directly supported under operating systems that use BIOS routines for disk access, such as DOS. SSD mode is not supported under other operating systems. In the case of an unsupported operating system, the user must directly access the SSD registers.

This document lists the SSD control registers of various RTD cpuModules. A user may use the information provided below to access the SSD socket directly. Note that this document does not cover programming algorithms for the devices that may be installed in those sockets. For information on how to actually read/write each device, contact the device maker.

486-series cpuModules

The following sections explain how to access the SSD Sockets of a 486-series RTD cpuModule. The SSD device is page-mapped 16KB at a time into low memory.

Initialization

Before accessing the SSD, the cpuModule must be initialized with the device type for each SSD socket and the memory window. This is normally done by performing the following steps:

Select the device type for U1 by writing bits DEVICEA[2..0] of I/O address 010h.

Select device type for U2 by writing bits DEVICEB[2..0] of I/O address 011h.

Select the memory window by writing I/O address 017h.

Accessing the SSD

After the CPU is configured, you may access the SSD for read and write operations as follows:

Write to I/O address 014h to select Device A or B (U1 or U2), and enable the SSD.

Select the 16KB page by writing bits ADD[14..16] to I/O address 012h and bits ADD[17..19] to I/O address 013h. This directly determines address bits 14 - 19 to the SSD device. Note that address lines A0 - A13 go directly to the SSD devices. Each socket can access up to 1MB, or 64 x 16KB pages. On smaller devices, only the lower pages are used: 512KB devices use 32 x 16KB pages; 256KB uses 16 x 16KB pages, and 128KB uses 8 x 16KB pages.

Read or write the device at the window address plus the offset into the 16KB page.

When finished, write to I/O address 014h to disable the SSD.

I/O Map

The registers which control the SSD are summarized and described below. These registers are only 4 bits wide. When written to as byte locations, the upper 5 bits written are ignored. Note that these registers are write-only. The contents may not be read back.

I/O Address	BIT 2	BIT 1	BIT 0	Default
-------------	-------	-------	-------	---------

010h	DeviceA2	DeviceA1	DeviceA0	0 0 0
011h	DeviceB2	DeviceB1	DeviceB0	0 0 0
012h	ADD16	ADD15	ADD14	XXX
013h	ADD19	ADD18	ADD17	XXX
014h		ENSSD	SELB/A	X 0 0
017h		WIN1	WIN0	X 0 0

The bits of these registers are described below:

DEVICEA[2..0] and DEVICEB[2..0]

These bits select SSD device types as follows:

0 = NOVRAM or 512K SRAM

1 = SRAM 128K

2 = Flash EPROM 128K/256K/512K

4 = EPROM 128K/256K

6 = EPROM 512K/1MB

ADD[19..14]

Selects 16K page of device A or B of SSD

ENSSD

0 = Disable SSD access

1 = Enable SSD access

SEL B/A

0 = Enable device A (U1)

1 = Enable device B (U2)

WIN[1..0]

Selects memory window for 16K SSD bank, as follows:

x = Segment:Address

0 = C000:0000 TO C000:3FFF

1 = C800:0000 TO C800:3FFF

2 = D000:0000 TO D000:3FFF

3 = D800:0000 TO D800:3FFF

686-series Geode Chipset cpuModules (GX/HX series)

On the 686-series, the SSD Socket is mapped into a 16KB window. A page register is used to allow access to the entire chip (divided into 16KB pages).

This section only applies to GX and HX series Geode cpuModules. Newer CX-series cpuModules do not have an SSD socket.

686-series SSD Control Register Map

The SSD control functions of the cpuModule are I/O mapped in the region from 0x0010 to 0x001B.

I/O Port	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0010						BANKA2	BANKA1	BANKA0
0x0011						BANKB2	BANKB1	BANKB0
0x0012						PADDR16	PADDR15	PADDR14
0x0013						PADDR19	PADDR18	PADDR17
0x0014							ENROMDSK	SELBA
0x0015								
0x0016								
0x0017							WIN1	WIN0
0x0018								
0x0019								
0x001A							DOCWINA1	DOCWINA0
0x001B							DOCWINB1	DOCWINB0

The bits in the above table that are shaded grey should be treated as “reserved” bits. Reserved bits must not be altered by your software. The contents of those bits must be preserved by bit-wise AND/OR operations in your code. Failure to do so may prevent the cpuModule from working properly.

686-series SSD Control Register Bit Values

BANKA and BANKB bits [2, 1, 0]

These bits define the type of device installed in SSD Socket A (BANKA bits) or SSD Socket B (BANKB bits). For cpuModules with only 1 SSD socket, the BANKB bits may be ignored.

The possible values are as follows:

000 = NVRAM or 512K SRAM

001 = SRAM 128K

010 = Flash 128K/256K/512K

011 = Reserved

100 = EEPROM 128K/256K

101 = EEPROM 512K/1MB

111 = BIOS Extension 32KB (uses DOCWIN bits listed below)

PADDR bits [19 - 16]

These bits set the 16KB page which will be mapped into the SSD window. Before reading/writing a certain offset in the SSD chip, make sure the page is set correctly. For any accesses which cross the 16KB boundary, the PADDR bits must be reloaded with the new page value.

SELBA bit

This bit selects whether the SSD Window is mapped to SSD Socket A or SSD Socket B. On cpuModules with only one SSD socket, make sure this is always mapped to A.

0 = SSD Window mapped to Socket A

1 = SSD Window mapped to Socket B

ENROMDISK bit

This bit controls whether the SSD socket is mapped to the SSD Window. The socket must be mapped into the window before it can be accessed. SELBA is used to control which socket (A or B) is mapped.

0 = SSD Socket is NOT mapped into SSD Window

1 = SSD Socket is mapped into SSD Window

WIN bits [0, 1]

These bits set the location of the SSD Window. The SSD Window is memory-mapped to an address below the 1MB boundary.

Possible values are:

00 = SSD Window occupies C000:0000 to C000:3FFF

01 = SSD Window occupies C800:0000 to C800:3FFF

10 = SSD Window occupies D000:0000 to D000:3FFF

11 = SSD Window occupies D800:0000 to D800:3FFF

When selecting an address for the SSD Window, make sure to use a location which does not conflict with other memory-mapped devices (e.g. VGA BIOS, USB, etc). Consult your cpuModule manual for more information.

DOCWINA and DOCWINB bits [0, 1]

These bits set the location of the BIOS Extension window. These bits are only relevant if the device installed in the SSD Socket is to be used as a BIOS Extension (e.g. DiskOnChip). To use it as a BIOS Extension, make sure that BANKA or BANKB is set accordingly. On systems with only one SSD socket, the DOCWINB bits may be ignored.

Possible values are:

00 = BIOS Extension Window occupies C000:0000 to C000:7FFF

01 = BIOS Extension Window occupies C800:0000 to C800:7FFF

10 = BIOS Extension Window occupies D000:0000 to D000:7FFF

11 = BIOS Extension Window occupies D800:0000 to D800:7FFF

When selecting an address for the BIOS Extension Window, make sure to select a different location than the SSD Window. If the two windows conflict, the cpuModule may behave erratically. Additionally, the BIOS Extension Window must not conflict with any other memory-mapped device (e.g. VGA BIOS, USB, etc).

Initializing the 686-series SSD Socket

Some of the aforementioned control registers are initialized by the BIOS during POST. The ones which are initialized by the BIOS are:

- BANKA and BANKB
- WIN
- DOCWINA and DOCWINB

The initialization values for these registers may be set in the BIOS under the Integrated Peripherals screen. You should only need to reprogram these registers if you need to override how the BIOS has initialized them.

786-series VIA Chipset cpuModules (HX series)

On the 786-series, the SSD Socket is mapped into a 16KB window. A page register is used to allow access to the entire chip (divided into 16KB pages).

This section only applies to HX series VIA chipset cpuModules. Newer CX-series cpuModules do not have an SSD socket.

786-series SSD Control Register Map

The SSD control functions of the cpuModule are I/O mapped in the region from 0x0010 to 0x0014.

I/O Port	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0010	BEXT_SIZEA1	BEXT_SIZEA0	SSD_BANKA2	SSD_BANKA1	SSD_BANKA0	BEXT_WINA2	BEXT_WINA1	BEXT_WINA0
0x0011								
0x0012	SSD_PAGE25	SSD_PAGE24	SSD_PAGE23	SSD_PAGE22			SSD_MODE1	SSD_MODE0
0x0013	SSD_PAGE21	SSD_PAGE20	SSD_PAGE19	SSD_PAGE18	SSD_PAGE17	SSD_PAGE16	SSD_PAGE15	SSD_PAGE14
0x0014						SSD_WIN2	SSD_WIN1	SSD_WIN0

The bits in the above table that are shaded grey should be treated as “reserved” bits. Reserved bits must not be altered by your software. The contents of those bits must be preserved by bit-wise AND/OR operations in your code. Failure to do so may prevent the cpuModule from working properly.

786-series SSD Control Register Bit Values

BEXT_SIZEA bits [1, 0]

These bits define the size of the BIOS Extension window for the device in the SSD Socket. These bits are only relevant if the device installed in the SSD Socket is to be used as a BIOS Extension (e.g. DiskOnChip). To use it as a BIOS Extension, make sure that SSD_BANKA bits are set accordingly.

The possible values are as follows:

- 00 = 8KB
- 01 = 16KB
- 10 = 32KB
- 11 = 64KB

SSD_BANKA bits [2, 1, 0]

These bits define the type of device installed in SSD Socket.

The possible values are as follows:

- 000 = NVRAM or 512K SRAM
- 001 = SRAM 128K
- 010 = Flash 128K/256K/512K
- 011 = Reserved
- 100 = EEPROM 128K/256K
- 110 = EEPROM 512K/1MB
- 111 = BIOS Extension

BEXT_WINA bits [2, 1, 0]

These bits set the location of the BIOS Extension Window. They are only relevant when the SSD Socket is configured for BIOS Extension mode. The size of this window is controlled by the BEXT_SIZEA bits.

Possible Values Are:

- 000 = BIOS Extension Window Starts at C000:0000
- 001 = BIOS Extension Window Starts at C400:0000
- 010 = BIOS Extension Window Starts at C800:0000
- 011 = BIOS Extension Window Starts at CC00:0000
- 100 = BIOS Extension Window Starts at D000:0000
- 101 = BIOS Extension Window Starts at D400:0000
- 110 = BIOS Extension Window Starts at D800:0000
- 111 = BIOS Extension Window Starts at DC00:0000

Note that you must select a value which matches a boundary of its size. For example, if the size is set to 32KB, the boundary must be C000:0000, C800:0000, D800:0000, or D800:0000.

When selecting an address for the BIOS Extension Window, make sure to select a different location than the SSD Window. If the two windows conflict, the cpuModule may behave erratically. Additionally, the BIOS Extension Window must not conflict with any other memory-mapped device (e.g. VGA BIOS, USB, etc).

SSD_PAGE bits [25 - 14]

These bits set the 16KB page which will be mapped into the SSD window. Before reading/writing a certain offset in the SSD chip, make sure the page is set correctly. For any accesses which cross the 16KB boundary, the SSD_PAGE bits must be reloaded with the new page value.

SSD_MODE bits [1, 0]

These bits control whether or not the SSD Window is mapped to the SSD Socket. To access the device in the SSD socket, this mapping must be enabled.

Possible values are:

- 00 = Disabled
- 01 = Reserved
- 10 = Mapped to SSD Socket
- 11 = Reserved

Initializing the 786-series SSD Socket

Some of the aforementioned control registers are initialized by the BIOS during POST. The ones which are initialized by the BIOS are:

- BEXT_SIZEA
- SSD_BANKA
- BEXT_WINA

The initialization values for these registers may be set in the BIOS under the Integrated Peripherals screen. You should only need to reprogram these registers if you need to override how the BIOS has initialized them.