Thank you for purchasing this Factory Service Manual CD/DVD from servicemanuals4u.com.

Please check out our eBay auctions for more great deals on Factory Service Manuals:

servicemanuals4u

ThinkPad 560Z
Technical Reference

xxxxxxx
xxxx-xxxx-xx

Note

Before using this information and the product it supports, be sure to read the general information under “Notices” on page B-1.

First Edition (October 1998)

Contents

Preface

This technical reference contains hardware and software interface information specific to the IBM ThinkPad 560Z computer. This technical reference is intended for those who develop hardware and software products for the computer. Users should understand computer architecture and programming concepts.

This publication consists of the following sections and appendixes:

- Section 1, “System Overview,” describes the system, features, and specifications.
- Section 2, “System Board,” describes the system-specific hardware implementations.
- Section 3, “Subsystems,” describes the hardware functions specific to the ThinkPad 560Z computers.
- Appendix A, “System Management API (SMAPI) BIOS Overview,” describes the system software interface built into the system, called the System Management Application Program Interface (SMAPI) BIOS, which controls the system information, system configuration, and power management features of the ThinkPad system.
- Appendix B, “Notices,” contains special notices and trademark information.

An index is also included.

This technical reference should be used with the following publications:

IBM Personal System/2 Hardware Interface Technical Reference

IBM Personal System/2 and Personal Computer BIOS Interface

These publications contain additional information on many of the subjects discussed in this technical reference. Information about diskette drives, hard disk drives, adapters, and external options are in separate technical references.

Attention

The term *Reserved* describes certain signals, bits, and registers that should not be changed. Use of reserved areas can cause compatibility problems, loss of data, or permanent damage to the hardware. When the contents of a register are changed, the state of the reserved bits must be preserved. If possible, read the register first and change only the bits that must be changed.

Section 1. System Overview

Description

The *IBM Personal System/2 Hardware Interface Technical Reference* describes devices common to the PS/2 AT-bus system family.

The IBM ThinkPad 560Z computer (hereafter called the 560Z, *ThinkPad computer*, or *computer*) is a notebook-size computer that features the AT* bus architecture. Each computer supports one external diskette drive and one internal hard disk drive.

Programs can distinguish the foregoing models of computers from other ThinkPad models by reading the system ID: Interrupt X'15', function code (AH)=X'23', (AL)=X'10', returns (AL)=X'37' for the 560Z.

The system microprocessor contains an internal cache and cache controller.

Figure 1-1 lists the model bytes, submodel bytes, and system clock speed of the system board.

Model	Model Byte (Hex)	Submodel Byte (Hex)	System Clock
560Z	FC	01	66 MHz

Figure 1-1. Model and Submodel Bytes

For a listing of the other systems, refer to the *IBM Personal System/2 and Personal Computer BIOS Interface*.

System Board Devices and Features

Figure 1-2 lists the system board devices and their features. The *IBM Personal System/2 Hardware Interface Technical Reference* describes devices common to PS/2 products by type number.

Device	Type	Features
Microprocessor	–	Intel® Mobile Pentium® II Processor • 233/300 MHz • 32 KB on-chip cache
Level 2 cache	–	512 KB
System timers	1	Channel 0: system timer Channel 1: refresh generation Channel 2: tone generator for speaker
ROM subsystem	–	128 KB by 4 banks (1 KB equals 1024 bytes)
RAM subsystem	–	32 MB. Expandable up to 96 MB. 64 MB. Expandable up to 128 MB.
CMOS RAM subsystem	–	128 bytes CMOS RAM with real-time clock/calendar

Figure 1-2 (Part 1 of 2). System Board Devices and Features

Device	Type	Features
Video subsystem	–	SVGA video functions: • Up to 16,777,216 colors on the TFT LCD and an external display See “Video Subsystem” on page 3-2 for more details of the video subsystem.
DMA controller	1	Seven DMA channels (AT compatible) Four 8-bit channels and three 16-bit channels
Interrupt controller	1	15 levels of system interrupts (interrupts are edge-triggered)
Keyboard/auxiliary device controller	1	Internal keyboard TrackPoint Press-to-Select Auxiliary device connector Password security
Diskette drive controller	2	Supports: • 3.5-in. diskette (1.44 MB) • 3.5-in. diskette (1.2 MB) (Japan-unique) • 3.5-in. diskette (720 KB)
Hard disk controller	–	Supports IDE controller
Serial controller port	2	EIA-232-E interface (16550 compatible) Programmable as serial port 1, 2, 3, or 4 One 9-pin, D-sub connector
Parallel controller port	1	Programmable as parallel port 1, 2, or 3 IEEE P1284-A compatible Supports bidirectional input and output Enhanced Parallel Port (EPP) compatible Extended Capabilities Port (ECP) compatible
Expansion bus adapter	–	Supports externally attached devices: • Port replicator • Port replicator with Advanced EtherJet Feature
PC Card[1] slots	–	Conforms to the standards and specifications listed in Figure 3-3 on page 3-8 • Two Type I or II PC cards • One Type III PC card
Audio subsystem	–	Sound Blaster**-Pro compatible
Infrared subsystem	–	Supports: • IrDA 1.1
Universal serial bus (USB)	–	Supports: • USB input and output devices

Figure 1-2 (Part 2 of 2). System Board Devices and Features

System Board I/O Address Map

Figure 1-3 shows the I/O address map.

Address (Hex)	Device
0000–001F	DMA controller 1
0020, 0021	Interrupt controller 1 (Master)
002E, 002F [a]	Super I/O configuration registers
0040, 0043	Timer counter 1
0060	Keyboard, auxiliary device
0061	System control port B
0062	Power management controller
0064	Keyboard, auxiliary device
0066	Power management controller
0070	NMI mask and RTC address
0071	RTC data
0072, 0073	RTC/CMOS Extended
0080–008F	DMA page registers
0092	System control port A
00A0, 00A1	Interrupt controller 2 (slave)
00B2, 00B3 [b]	Advanced power management port
00C0–00DF	DMA controller 2
00F0	Coprocessor error
0170–0177	IDE Secondary
01F0–01F7	IDE Primary
0201	Game port (Joy Stick)
0220–022F	Audio port - Sound Blaster 1
0230–023F	Audio port - Sound Blaster 2
0240–024F	Audio port - Sound Blaster 3
0250–025F	Audio port - Sound Blaster 4
0278–027F	Parallel port 3 - LPT3 or EPP/ECP
02E8–02EF	Serial port 4 - COM4 (IR)
02F8–02FF	Serial port 2 - COM2 (IR)
0300–0301	Audio port - MIDI (MPU-401) 1
0310–0311	Audio port - MIDI (MPU-401) 2
0320–0321	Audio port - MIDI (MPU-401) 3
0330–0331	Audio port - MIDI (MPU-401) 4
0370–0375	FDC secondary
0376	IDE secondary
0377	FDC secondary
0378–037F	Parallel port 2 - LPT2 or EPP/ECP
0388–038B	Audio port - Adlib (FM synthesizer)

Figure 1-3 (Part 1 of 2). System Board I/O Address Map

Address (Hex)	Device
03B4, 03B5, 03BA	Video subsystem
03BC–03BF	Parallel port 1 - LPT1 or ECP
03C0–03C5	Video subsystem
03C6–03C9	Video DAC
03D4, 03D5, 03DA,	Video subsystem
03E0–03E3	PC Card interface
03E8–03EF	Serial port 3 - COM3
03F0–03F5	FDC primary
03F6	IDE primary
03F7	FDC primary
03F8–03FF	Serial port 1 - COM1
04D0, 04D1 [b]	Interrupt controller edge/level control
0534–0537	Audio port - WSS-1
0538–053F	Audio port - Control port-1
0608–060B	Audio port - WSS-2
0678–067A	Parallel port 3 - ECP
0778–077A	Parallel port 2 - ECP
07BC–07BE	Parallel port 1 - ECP
0CF8–0CFB	PCI configuration address register
0CF9 [b]	Reset control register
0CFC–0CFF	PCI configuration data register
0D38–0D3F	Audio port - control port-2
0E84–0E87	Audio port - WSS-3
0E88–0E8F	Audio port - control port-3
0F44–0F47	Audio port - WSS-4
0FF0–0FF7	Audio port - control port-4
15E0–15EF	Power management register
8000–801F	USB I/O space register
EF00–EF37	Power management I/O register
EF00–EF3F	Power management register
EFA0–EFAD	SMBus I/O space register
EFA0–EFAF	SMBus I/O space register

[a] NS PC97338 unique registers; [b] Intel PIIX4E unique registers

Figure 1-3 (Part 2 of 2). System Board I/O Address Map

Specifications

Figure 1-4 to Figure 1-7 on page 1-9 list the specifications for the computer.

Performance Specifications

Device		Cycle Time (ns)
Microprocessor (66 MHz–15 ns clock)		
Access to RAM:[1]		
Memory read	Page hit, burst	135 ns
	Page miss, burst	255 ns
Memory write	Page miss, burst	45 ns
Access to ROM:		1000
Refresh rate (typically performed every 15.6 μs)		750 (minimum)
DMA controller (4 MHz–250 μs clock):		1250
Bus cycles (AT):		
8 bit		1000
16 bit		625

[1] The cycle times shown for access to system-board RAM are based on 60-ns EDO memory.

Figure 1-4. Performance Specifications for the ThinkPad 560Z

Physical Specifications

Size

Width: 297 mm (11.7 in.)
Depth: 222 mm (8.7 in.)
Height: 31.0 mm (1.22 in.)

Weight[1] (approximate value)

2640-9xx model:
1.86 kg (4.10 lb)
2640-Bxx model:
1.90 kg (4.20 lb)

Air Temperature

System on (without diskette)
5.0°C to 35.0°C (41°F to 95°F)

System on (with diskette)
10.0°C to 35.0°C (50°F to 95°F)

System off
5.0°C to 43.0°C (41°F to 110°F)

Humidity

System (without diskette)
8% to 95%

System (with diskette)
8% to 80%

Maximum altitude[2]: 3048 m (10 000 ft) in unpressurized conditions

Heat output: 35 W (119.4 BTUs/hour) at maximum configuration

Acoustical readings (see Figure 1-7 on page 1-9)

Electrical (see Figure 1-6 on page 1-9)

Electromagnetic compatibility: FCC class B

[1] With battery pack installed.
[2] This is the maximum altitude at which the specified air temperatures apply. At higher altitudes, the maximum air temperatures are lower than those specified.

Figure 1-5. Physical Specifications for the ThinkPad 560Z

Electrical Specifications

	(56 W)
Input Voltage[1] (V ac)	100–240
Frequency (Hz)	50/60
Input[2] (kVA)	0.132

[1] Range is automatically selected; sine wave input is required.
[2] At maximum configuration.

Figure 1-6. Electrical Specifications for the ThinkPad 560Z

Acoustical Readings

L_{WAd} in bels		L_{pAm} in dB		$<L_{pA}>_m$ in dB	
Operate	**Idle**	**Operate**	**Idle**	**Operate**	**Idle**
3.7	3.5	30	28	27	26

Notes:

L_{WAd}	Is the declared sound power level for the random sample of machines.
L_{pAm}	Is the mean value of the A-weighted sound pressure levels at the operator position (if any) for the random sample of machines.
$<L_{pA}>_m$	Is the mean value of the A-weighted sound pressure levels at the one-meter position for the random sample of machines.
Operate	Shows the value while using the hard disk drive.

All measurements made in accordance with ANSI S12.10 and reported in conformance with ISO 9296.

Figure 1-7. Acoustical Readings for the ThinkPad 560Z

Power Supply

The power supply converts the ac voltage to dc voltage and provides power for the following:

- System board set
- Diskette drive
- Hard disk drive
- Auxiliary devices
- Keyboard
- LCD panel
- PC Card cards

Voltages

The power supply generates five different dc voltages: VCCCPU, VCC3A, VCC5M, VCCSW, and VCC12M. Figure 1-8 shows the maximum current for each voltage.

Output	Voltage (V dc)	Current (A)
VCCCPU	+1.7 or +1.6	2.20
VCC3A	+3.3	2.00
VCC5M	+5.0	3.00
VCCSW	+5.0	0.01
VCC12M	+12.0	0.11

Figure 1-8. Power Supply Maximum Current

Output Protection

A short circuit placed on any dc output (between outputs or between an output and a dc return) latches all dc outputs into a shutdown state, with no hazardous condition to the power supply.

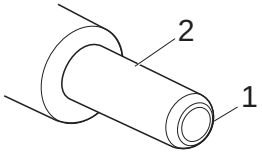
If an overvoltage fault occurs in the power supply, the power supply latches all dc outputs into a shutdown state before any output exceeds 135% of the nominal value of the power supply.

Voltage Sequencing

When power is turned on, the output voltages reach their operational voltages within 2 seconds.

Power Supply Connector

The following connector is used with the AC Adapter. The total power capacity of this connector must not exceed 4.0 A.



Refer to Figure 1-9 for the appropriate adapter pin assignments.

Pin	Voltage
1	+7.0 V dc to +16.0 V dc (depending on charging conditions)
2	Ground

Figure 1-9. Voltage Pin Assignments for 56-W AC Adapter

Battery Pack

The ThinkPad computer uses a lithium-ion (Li-ion) battery pack that meets the following electrical specifications:

Nominal Voltage	+10.8 V dc
Capacity (average)	2.2 ampere hours (AH)
Protection	Overcurrent protection Overvoltage protection Overdischarge protection Thermal protection

Figure 1-10. Battery Pack Specifications

Section 2. System Board

Description

This section describes the microprocessor, connectors, memory subsystems, and miscellaneous system functions and ports for the ThinkPad computers. You can find additional information about these topics in *IBM Personal System/2 Hardware Interface Technical Reference–AT-Bus Subsystems*.

Microprocessor

The ThinkPad 560Z computer uses the Intel® Mobile Pentium® II 233-MHz or 300-MHz processor with MMX™ technology.

The processor has a 32-bit address bus and a 64-bit data bus. It is software-compatible with all previous microprocessors. The processor has an internal, split data and instruction, 32-KB write-back cache. It includes pipelined math coprocessor functions and superscalar architecture (two execution units).

Cache Memory Operation

The cache memory in the Intel Pentium microprocessor enables the microprocessor to read instructions and data much faster than if the microprocessor had to access system memory. When an instruction is first used or data is first read or written, it is transferred to the cache memory from main memory. This enables future accesses to the instructions or data to occur much faster.

The cache is disabled and empty when the microprocessor comes out of the reset state. The cache is tested and enabled during the power-on self-test (POST).

The cache memory in the Intel Pentium microprocessor is loaded from system memory in 32-byte increments, each referred to as a *cache line*. A cache line is aligned on a paragraph boundary. A reference to any byte contained in a cache line results in the entire line being read into the cache memory (if the data was not already in the cache). When the microprocessor gives up control of the system bus, the cache memory enters “snoop” mode and monitors all write and read operations. If memory data is written to a location in the cache and the cache line is in the “modified” state, the corresponding cache line is written back to system memory and is invalidated.

When the microprocessor performs a memory read, the data address is used to find the data in the cache. If the data is found (a hit), it is read from the cache memory and no external bus cycle occurs. If the data is not found (a miss), an external bus cycle is used to read the data from system memory. If the address of the missed data is in a cacheable address space, the data is stored in the cache memory and the remainder of the cache line is read.

When the microprocessor performs a memory write, the data address is used to search the cache. If the address is found (a hit), the data is written to the cache and no external bus cycle is used to write the data to system memory. (If the address of the write operation was not in the cache memory but was in cacheable address space, the data is read back into the cache memory and the remainder of the cache line is read.)

Cacheable Address Space

Cacheable address space is defined as system memory that resides on the system board (0–640 KB and 1 MB–96 MB or 128 MB). Cacheability of system memory is up to 512 MB in the L2 cache, and is up to 4 GB in the on-chip L1 cache. Nothing in address range X'A0000'–X'BFFFF', I/O address space, or memory in any AT slot is cached.

ROM address space (X'C0000'–X'C9FFF' and X'F0000'–X'FFFFF') is cacheable for *code read operations only*. If data in this address range is already in cache memory and the address range is written to, the cached line is invalidated and is read again from RAM (in which the BIOS is shadowed in).

Keyboard/Mouse Connector

Each ThinkPad computer has a keyboard/mouse connector where the IBM mouse, keyboard, or numeric keypad is connected.

Signals

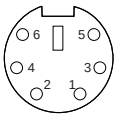
The keyboard and mouse signals are driven by open-collector drivers pulled to 5 V dc through a pull-up resistor. Figure 2-1 lists the signals.

Sink current	20 mA	Maximum
High-level output voltage	5.0 V dc minus pullup	Minimum
Low-level output voltage	0.5 V dc	Maximum
High-level input voltage	2.0 V dc	Minimum
Low-level input voltage	0.8 V dc	Maximum

Figure 2-1. Keyboard and Mouse Signals

Connector

The keyboard/mouse connector uses a 6-pin, miniature DIN connector.



Pin	I/O	Signal Name
1	I/O	Mouse Data
2	I/O	Keyboard Data
3	–	Ground
4	–	+5 V dc
5	I/O	Mouse Clock
6	I/O	Keyboard Clock

Figure 2-2. Keyboard/Mouse Connector Pin Assignments

Note: The maximum current for +5 V dc (pin 4) is 0.5 A for both the mouse and the numeric keypad.

Scan Codes

Figure 2-3 shows the key numbers assigned to keys on the 84-key keyboard (for the U.S. and Japan). Figure 2-4 on page 2-6 shows the key numbers assigned to keys on the 85-key keyboard (for countries other than the U.S. and Japan). For scan codes assigned to each numbered key, refer to the *IBM Personal System/2 Hardware Interface Technical Reference*.

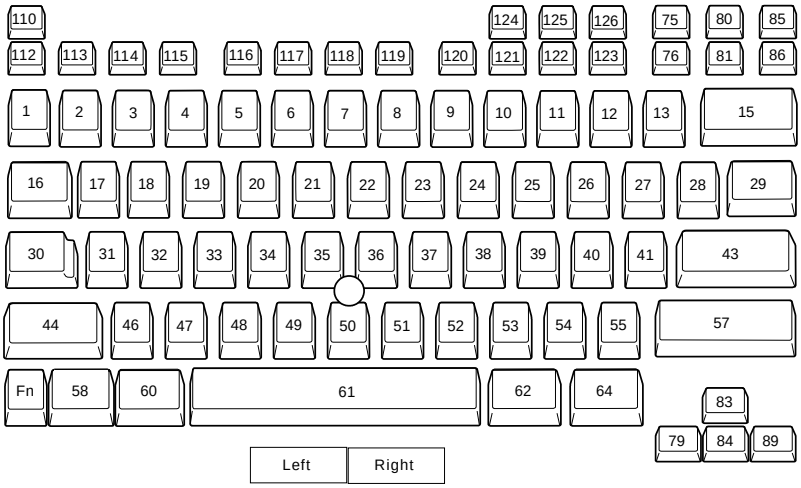


Figure 2-3. Key Numbers for the 84-Key Keyboard

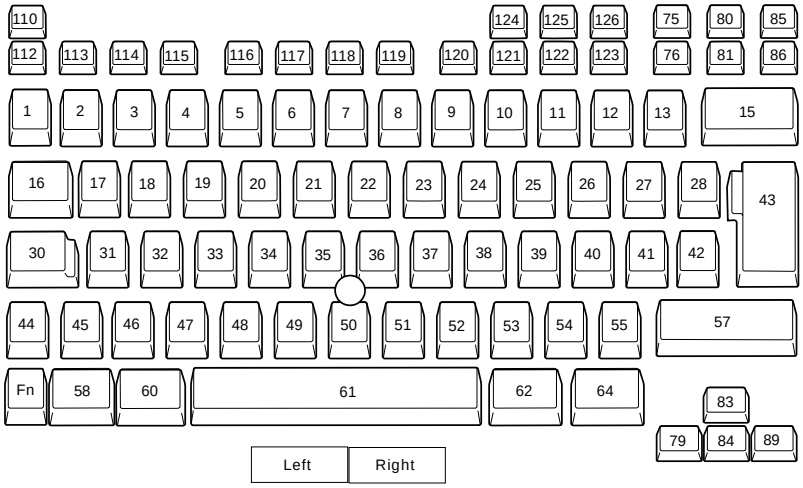


Figure 2-4. Key Numbers for the 85-Key Keyboard

Keyboard ID

The keyboard ID consists of 2 bytes: X'83AB' (the built-in keyboard with the external numeric keypad) or X'84AB' (the built-in keyboard only). Interrupt X'16', function code (AH)=X'0A', returns the keyboard ID.

Figure 2-5 shows the key numbers assigned to keys on the external numeric keypad. For scan codes assigned to each numbered key, refer to the *IBM Personal System/2 Hardware Interface Technical Reference*.

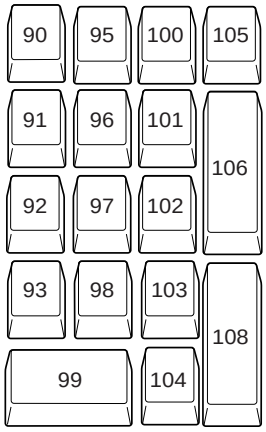


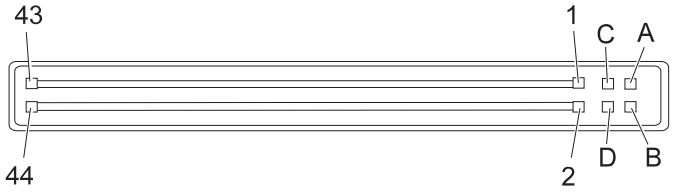
Figure 2-5. Key Numbers for the External Numeric Keypad

Displayable Characters and Symbols

For displayable characters and symbols that are keyable from the keyboard, refer to the *IBM Personal System/2 Hardware Interface Technical Reference*.

Hard Disk Drive Connector

The hard disk drive connected to the system board is removable. The following shows the pin assignments for the connector on the hard disk drive.



Note: The pin position A, B, C, and D are not used by the ThinkPad 560Z computer.

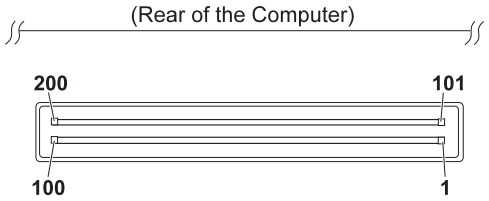
Pin	Signal	Description	Pin	Signal	Description
1	–RESET	Reset	23	–DIOW	I/O Write
2	GND	Ground	24	GND	Ground
3	DD07	Data 7	25	–DIOR	I/O Read
4	DD08	Data 8	26	GND	Ground
5	DD06	Data 6	27	IORDY	I/O Ready
6	DD09	Data 9	28	CSEL	Cable Select
7	DD05	Data 5	29	–DMACK	DMA Acknowledgment
8	DD10	Data 10	30	GND	Ground
9	DD04	Data 4	31	INTRQ	Interrupt Request
10	DD11	Data 11	32	–	Not connected
11	DD03	Data 3	33	DA01	Address 1
12	DD12	Data 12	34	–PDIAG*	Passed diagnostics (slave drive)
13	DD02	Data 2	35	DA00	Address 0
14	DD13	Data 13	36	DA02	Address 2
15	DD01	Data 1	37	–CS0	Chip Select 0
16	DD14	Data 14	38	–CS1	Chip Select 1
17	DD00	Data 0	39	–DASP	Drive Activity /Slave drive present
18	DD15	Data 15	40	GND	Ground
19	GND	Ground	41	+5V	+5V dc
(20)	Key	Not connected	42	+5V	+5V dc
21	DMARQ	DMA request	43	GND	Ground
22	GND	Ground	44	–	Not connected

Note: * –PDIAG is not used by the ThinkPad 560Z computer.

Figure 2-6. Hard Disk Drive Connector Pin Assignments

External Connector

The Port Replicator is connected through the 200-pin external connector at the bottom of the computer. This connector is installed on the system board and has the following pin assignments:



Type Legend:

A: Audio signal
F: Diskette drive signal
G: Ground
I: Interrupt request signal
J: Joy stick signal
K: Keyboard/Mouse signal
PB: Serial port signal
PC: Power control signal
PM: Power Management signal
PR: Parallel port signal
PW: Power line
S: Serial port signal
U: USB signal
V: Video signal

Pin	Signal	Type	Pin	Signal	Type
1	Ground	G	23	Docking Type Select	PM
2	Suspend Power Good	PC	24	–External Power Good	PC
3	ac/dc Power	PW	25	Mouse Data	K
4	ac/dc Power	PW	26	Mouse Clock	K
5	ac/dc Power	PW	27	Ground	G
6	ac/dc Power	PW	28	–PME	PM
7	–Suspend Status	PM	29	USB_OC1	U
8	Ground	G	30	Ground	G
9	Line In Left	A	31	–Ring Indicator	S
10	+5V	PW	32	Clear to Send	S
11	Line Out Left	A	33	Request to Send	S
12	Analog Ground	A	34	Data Set Ready	S
13	Ground	G	35	Ground	G
14	Data Rate Select 1	F	36	Ground	G
15	–Drive Select 1	F	37	–AUTO FD XT	PR
16	–Docking SMI	PM	38	–ERROR	PR
17	–MOTOR ENABLE 0	G	39	–INIT	PR
18	–Direction In	G	40	–SLCT IN	PR
19	–Step	G	41	Data Bit 4	PR
20	Write Data	G	42	Data Bit 6	PR
21	–Write Enable	G	43	–ACK	PR
22	–Head 1 Select	G	44	PE	PR

Figure 2-7 (Part 1 of 3). 200-Pin External Connector Pin Assignments

Pin	Signal	Type	Pin	Signal	Type
45	Ground	G	85	Ground	G
46	Red	V	86	–STROBE	PR
47	Blue	V	87	Data Bit 0	PR
48	Hsync	V	88	Data Bit 1	PR
49	Vsync	V	89	Data Bit 2	PR
50	Ground	G	90	Data Bit 3	PR
51	Ground	G	91	Data Bit 5	PR
52	Power Good	PC	92	Data Bit 7	PR
53	ac/dc Power	PW	93	BUSY	PR
54	ac/dc Power	PW	94	SLCT	PR
55	ac/dc Power	PW	95	Ground	G
56	ac/dc Power	PW	96	–Video Powerdown	V
57	–Power On	PW	97	Green	V
58	Ground	G	98	DDC Data	V
59	Line In Right	A	99	DDC Clock	V
60	+5V	PW	100	Ground	G
61	Line Out Right	A	101	Ground	G
62	Ground	G	102	–PCIRST	PB
63	Monitor ID 2	V	103	–PREQ	PB
64	–INDEX	F	104	Ground	G
65	SMB_CLK	PM	105	AD:31	PB
66	SMB_DATA	PM	106	AD:29	PB
67	–Track 0	F	107	AD:27	PB
68	Monitor ID 0	V	108	AD:25	PB
69	–Write Protect	F	109	Ground	G
70	Read Data	F	110	C_BE:3	PB
71	–DRV2	F	111	AD:23	PB
72	–Diskette Change	F	112	AD:21	PB
73	Ground	G	113	AD:19	PB
74	+5V	PW	114	AD:17	PB
75	Keyboard Data	K	115	Ground	G
76	Keyboard Clock	K	116	–IRDY	PB
77	Ground	G	117	Ground	G
78	USBP1+	U	118	–DEVSEL	PB
79	USBP1–	U	119	Ground	G
80	Ground	G	120	–LOCK	PB
81	Data Terminal Ready	S	121	NC	–
82	Transmit Data	S	122	–SERR	PB
83	Receive Data	S	123	Ground	G
84	Data Carrier Detect	S	124	C_BE:1	PB

Figure 2-7 (Part 2 of 3). 200-Pin External Connector Pin Assignments

Pin	Signal	Type	Pin	Signal	Type
125	AD:15	PB	163	AD:18	PB
126	AD:13	PB	164	AD:16	PB
127	AD:11	PB	165	C_BE:2	PB
128	AD:9	PB	166	Ground	G
129	Ground	G	167	+FRAME	PB
130	AD:7	PB	168	Ground	G
131	AD:5	PB	169	–TRDY	PB
132	AD:3	PB	170	Ground	G
133	AD:1	PB	171	–STOP	PB
134	Ground	G	172	Ground	G
135	–PMERI	PM	173	PAR	PB
136	Docking Speaker	A	174	Ground	G
137	NC	–	175	AD:14	PB
138	–Suspend Not Reset	PC	176	AD:12	PB
139	Ground	G	177	AD:10Bit 0	PB
140	IRQ:4	I	178	AD:8	PB
141	IRQ:7	I	179	C_BE:0	PB
142	IRQ:10	I	180	Ground	G
143	IRQ:12	I	181	AD:6	PB
144	–INTA	I	182	AD:4	PB
145	JAB1	J	183	AD:2	PB
146	JBB1	J	184	AD:0	PB
147	JACX	J	185	Ground	G
148	JBCX	J	186	–CLKRUN	PB
149	MIDI_IN	A	187	–Docking Enable	PC
150	Ground	G	188	Ground	G
151	Ground	G	189	IRQ:3	I
152	PCICLK	PB	190	IRQ:5	I
153	Ground	G	191	IRQ:9	I
154	–GNT	PB	192	IRQ:11	I
155	Ground	G	193	IRQ:15	I
156	AD:30	PB	194	–INTB	I
157	AD:28	PB	195	JAB2	J
158	AD:26	PB	196	JBB2	J
159	AD:24	PB	197	JACY	J
160	Ground	G	198	JBCY	J
161	AD:22	PB	199	MIDI_OUT	A
162	AD:20	PB	200	Docking Status	PM

Figure 2-7 (Part 3 of 3). 200-Pin External Connector Pin Assignments

Diskette Drive and Controller

Figure 2-8 shows the read, write, and format capabilities of the diskette drive for the ThinkPad computer.

Diskette Type	Format Size		
	720 KB	**1.2 MB**	**1.44 MB**
3.5-inch 1.0 MB Diskette	RWF	–	–
3.5-inch 2.0 MB Diskette	–	RWF	RWF

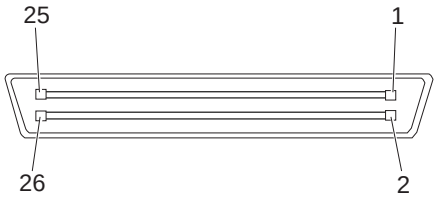
Legend:

1 KB (kilobyte)	1024 bytes
1 MB (megabyte)	1,048,576 bytes
R	Read
W	Write
F	Format

Figure 2-8. Diskette Drive Read, Write, and Format Capabilities

Diskette Drive Connector

The external diskette drive is connected through the diskette drive connector, on the left side of the computer. Figure 2-9 shows the pin assignments of the connector:



Pin	Signal	Type
1	GND	Ground
2	DRATE1	Data Rate Select 1
3	VCC5B	+5V dc
4	–	Reserved
5	GND	Ground
6	–	Reserved
7	GND	Ground
8	–INDEX	Index
9	–	Reserved
10	–	Reserved
11	–DRVSEL0	Drive Select 0
12	DRATE0	Data Rate Select 0
13	–MOTEN0	Motor Enable 0
14	–	Reserved
15	–FDIR	Direction In
16	–FSTEP	Step
17	WRDATA	Write Data
18	–FWREN	Write Enable
19	GND	Ground
20	–TRAK0	Track 0
21	–	Reserved
22	–FWPROTECT	Write Protect
23	RDDATA	Read Data
24	–FSIDE1SEL	Side 1 Select
25	–	Reserved
26	–DISKCHG	Disk Change

Figure 2-9. Diskette Drive Connector Pin Assignments

Memory

The ThinkPad computers use the following types of memory:

- Read-only memory (ROM)
- Random access memory (RAM)
- Real-time clock/complementary metal-oxide semiconductor RAM (RT/CMOS RAM)

ROM Subsystem

The ROM subsystem consists of four banks of 128-KB memory. ROM is active when power is turned on and is assigned to the top of the first and last 1MB of address space (X'000F0000'–X'000FFFFF' and X'FFFF0000'–X'FFFFFFFF'). After POST checks that system memory is operating correctly, the ROM code is copied to RAM at the same address space, and ROM is disabled.

RAM Subsystem

The RAM subsystem on the system board starts at address X'00000000' of the address space. The RAM subsystem for the ThinkPad 560Z is 64 bits wide.

The 32-MB or 64-MB base memory is on the system board. One 144-pin 8-byte dual inline memory module (DIMM) connector is provided on the system board. This connector accepts a 16-MB, 32-MB, or 64-MB EDO type DIMM. The memory capacity can be increased up to 96 MB or 128 MB when a DIMM is used (see “System Board Memory Connector for DIMM” on page 2-15).

The total amount of usable memory is less than the amount of memory installed because of ROM-to-RAM remapping and power management.

System Memory Map

Memory is mapped by the memory controller registers.

Figure 2-10 shows the memory map for a correctly functioning system. Memory can be mapped differently if POST detects an error in system board memory or RT/CMOS RAM. In the figure, the variable *x* represents the number of 1MB blocks of system board memory starting at or above the X'100000' boundary.

Hex Address Range	Function
00000000 to 0009FFFF	640-KB system board RAM
000A0000 to 000BFFFF	Video RAM
000C0000 to 000C9FFF	System board video BIOS ROM mapped to RAM
000CA000 to 000EFFFF	Channel ROM
000F0000 to 000FFFFF	64-KB system board ROM mapped to RAM
00100000 to (00100000 + *x* MB)	*x*-MB system board RAM
FFFF0000 to FFFFFFFF	64-KB system board ROM (same as X'000F0000' to X'000FFFFF')

Figure 2-10. System Memory Map

System Board Memory Connector for DIMM

The system board of ThinkPad 560Z has one DIMM connector that directly accepts one 144-pin DIMM of one of the following three different capacities: 16-MB, 32-MB, or 64-MB.

Figure 2-11 on page 2-16 shows the pin assignments for the DIMM connector.

Pin	Signal	Pin	Signal	Pin	Signal
1	Ground	49	MD42	97	MD25
2	Ground	50	MD21	98	MD38
3	MD15	51	MD41	99	MD24
4	MD48	52	MD22	100	MD39
5	MD14	53	MD40	101	+3.3V dc
6	MD49	54	MD23	102	+3.3V dc
7	MD13	55	Ground	103	MA6
8	MD50	56	Ground	104	MA7
9	MD12	57	Ground	105	MA8
10	MD51	58	Ground	106	MA11
11	+3.3V dc	59	Ground	107	Ground
12	+3.3V dc	60	Ground	108	Ground
13	MD11	61	Not connected	109	MA9
14	MD52	62	Not connected	110	Ground
15	MD10	63	+3.3V dc	111	MA10
16	MD53	64	+3.3V dc	112	Ground
17	MD9	65	Not connected	113	+3.3V dc
18	MD54	66	Not connected	114	+3.3V dc
19	MD8	67	–WE	115	–CAS3
20	MD55	68	Not connected	116	–CAS4
21	Ground	69	–RAS2	117	–CAS7
22	Ground	70	Not connected	118	–CAS0
23	–CAS1	71	–RAS3	119	Ground
24	–CAS6	72	Not connected	120	Ground
25	–CAS5	73	Ground	121	MD56
26	–CAS2	74	Not connected	122	MD7
27	+3.3V dc	75	Ground	123	MD57
28	+3.3V dc	76	Ground	124	MD6
29	MA0	77	Ground	125	MD58
30	MA3	78	Ground	126	MD5
31	MA1	79	Ground	127	MD59
32	MA4	80	Ground	128	MD4
33	MA2	81	+3.3V dc	129	+3.3V dc
34	MA5	82	+3.3V dc	130	+3.3V dc
35	Ground	83	MD31	131	MD60
36	Ground	84	MD32	132	MD3
37	MD47	85	MD30	133	MD61
38	MD16	86	MD33	134	MD2
39	MD46	87	MD29	135	MD62
40	MD17	88	MD34	136	MD1
41	MD45	89	MD28	137	MD63
42	MD18	90	MD35	138	MD0
43	MD44	91	Ground	139	Ground
44	MD19	92	Ground	140	Ground
45	+3.3V dc	93	MD27	141	I²C Data
46	+3.3V dc	94	MD36	142	I²C Clock
47	MD43	95	MD26	143	+3.3V dc
48	MD20	96	MD37	144	+3.3V dc

Figure 2-11. DIMM Connector Pin Assignments

RT/CMOS RAM

The RT/CMOS RAM (real-time clock/complementary metal-oxide semiconductor RAM) module contains the real-time clock and 128 bytes of CMOS RAM. The clock circuitry uses 14 bytes of this memory; the remainder is allocated to configuration and system-status information. A battery is built into the module to keep the RT/CMOS RAM active when the power supply is not turned on.

Figure 2-12 lists the RT/CMOS RAM bytes and their addresses.

Address (Hex)	RT/CMOS RAM Bytes
000–00D	Real-time clock
00E	Diagnostic status
00F	Shutdown status
010	Diskette drive type
011	Hard disk 2 and 3 drive type
012	Hard disk 0 and 1 drive type
013	Reserved
014	Equipment
015, 016	Low and high base memory
017, 018	Low and high expansion memory
019	Hard disk 0 extended byte
01A	Hard disk 1 extended byte
01B	Hard disk 2 extended byte
01C	Hard disk 3 extended byte
01D–02D	Reserved
02E, 02F	Checksum
030, 031	Low and high usable memory above 1 MB
032	Date-century
033–07F	Reserved

Figure 2-12. RT/CMOS RAM Address Map

RT/CMOS Address and NMI Mask Register (X'0070')

The NMI mask register is used with the RT/CMOS data register (X'0071') to read from and write to the RT/CMOS RAM bytes.

Attention

The operation following a write to X'0070' should access X'0071'; otherwise, intermittent failures of the RT/CMOS RAM can occur.

Bit	Function
7	NMI mask
6–0	RT/CMOS RAM address

Figure 2-13. RT/CMOS Address and NMI Mask Register (X'0070')

Bit 7 When this write-only bit is set to 1, the NMI is masked (disabled). This bit is set to 1 by a power-on reset.

Bits 6–0 These bits are used to select RT/CMOS RAM addresses.

RT/CMOS Data Register (X'0071')

The RT/CMOS data register is used with the RT/CMOS address and NMI mask register (X'0070') to read from and write to the RT/CMOS RAM bytes.

Bit	Function
7–0	RT/CMOS data

Figure 2-14. RT/CMOS Data Register (X'0071')

RT/CMOS RAM I/O Operations

During I/O operations to the RT/CMOS RAM addresses, you should mask interrupts to prevent other interrupt routines from changing the RT/CMOS address register before data is read or written. After I/O operations, you should leave the RT/CMOS address and NMI mask register (X'0070') pointing to status register D (X'00D').

Attention

The operation following a write to X'0070' should access X'0071'; otherwise, intermittent failures of the RT/CMOS RAM can occur.

Writing to the RT/CMOS RAM requires the following:

1. Write the RT/CMOS RAM address to the RT/CMOS address and NMI mask register (X'0070').
2. Write the data to the RT/CMOS data register (X'0071').
3. Write the address, X'0F', to the RT/CMOS and NMI mask register; this leaves X'0070' pointing to the shutdown status byte (X'0F').
4. Read address X'0071' to restore the RT/CMOS.

Reading from the RT/CMOS RAM requires the following steps:

1. Write the RT/CMOS RAM address to the RT/CMOS and NMI mask register (X'0070').
2. Read the data from the RT/CMOS data register (X'0071').
3. Write the address, X'0F', to the RT/CMOS and NMI mask register; this leaves X'0070' pointing to the shutdown status byte (X'0F').
4. Read address X'0071' to restore the RT/CMOS.

Real-Time Clock Bytes (X'000'–X'00D'): Bit definitions and addresses for the real-time clock bytes are shown in Figure 2-15.

Address (Hex)	Function	Byte Number
000	Seconds	0
001	Second alarm	1
002	Minutes	2
003	Minute alarm	3
004	Hours	4
005	Hour alarm	5
006	Day of week	6
007	Date of month	7
008	Month	8
009	Year	9
00A	Status register A	10
00B	Status register B	11
00C	Status register C	12
00D	Status register D	13

Figure 2-15. Real-Time Clock Bytes (X'000'–X'00D')

Note: The setup program initializes status registers A and B when the time and date are set. Interrupt 1AH is the BIOS interface to read and set the time and date; it initializes the registers in the same way that the setup program does.

Status Register A (X'00A')

Bit	Function
7	Update in progress
6–4	22-stage divider
3–0	Rate-selection bits

Figure 2-16. Status Register A (X'00A')

Bit 7 If set to 1, this bit indicates that the time-update cycle is in progress. If set to 0, it indicates that the current date and time can be read.

Bits 6–4 These bits identify which time-base frequency is being used. The system initializes these bits to B'010', which selects a 32.768-kHz time base. This is the only value supported by the system for proper timekeeping.

Bits 3–0 These bits allow the selection of a divider output frequency. The system initializes the rate-selection bits to B'0110', which selects a 1.024-kHz square-wave

output frequency and a 976.562-microsecond periodic interrupt rate.

Status Register B (X'00B')

Bit	Function
7	Set
6	Enable periodic interrupt
5	Enable alarm interrupt
4	Enable update-ended interrupt
3	Enable square wave
2	Date mode
1	24-hour mode
0	Enable daylight-saving time

Figure 2-17. Status Register B (X'00B')

Bit 7 If set to 0, this bit updates the cycle, normally by advancing the count at a rate of one cycle per second. If set to 1, it immediately ends any update cycle in progress, and the program can initialize the 14 time bytes without any further updates occurring until this bit is set to 0.

Bit 6 This is a read/write bit that allows an interrupt to occur at a rate specified by the rate and divider bits in status register A. If set to 1, this bit enables the interrupt. The system initializes this bit to 0.

Bit 5 If set to 1, this bit enables the alarm interrupt. The system initializes this bit to 0.

Bit 4 If set to 1, this bit enables the update-ended interrupt. The system initializes this bit to 0.

Bit 3 If set to 1, this bit enables the square-wave frequency as set by the rate-selection bits in status register A. The system initializes this bit to 0.

Bit 2 This bit indicates whether the binary-coded-decimal (BCD) or binary format is used for time-and-date calendar updates. If set to 1, this bit indicates the binary format. The system initializes this bit to 0.

Bit 1 This bit indicates whether the hours byte is in 12-hour or 24-hour mode. If set to 1, this bit indicates the 24-hour mode. The system initializes this bit to 1.

Bit 0 If set to 1, this bit enables the daylight-saving-time mode. If set to 0, this bit disables the daylight-saving-time mode, and the clock reverts to standard time. The system initializes this bit to 0.

Status Register C (X'00C')

Bit	Function
7	Interrupt request flag
6	Periodic interrupt flag
5	Alarm interrupt flag
4	Update-ended interrupt flag
3–0	Reserved

Figure 2-18. Status Register C (X'00C')

Note: Interrupts are enabled by bits 6, 5, and 4 in status register B.

Bit 7 If set to 1, this bit indicates that an interrupt has occurred; bits 6, 5, and 4 indicate the type of interrupt.

Bit 6 If set to 1, this bit indicates that a periodic interrupt has occurred.

Bit 5 If set to 1, this bit indicates that an alarm interrupt has occurred.

Bit 4 If set to 1, this bit indicates that an update-ended interrupt has occurred.

Bits 3–0 These bits are reserved.

Status Register D (X'00D')

Bit	Function
7	Valid RAM
6–0	Reserved

Figure 2-19. Status Register D (X'00D')

Bit 7 This read-only bit monitors the internal battery. If set to 1, this bit indicates that the real-time clock has power. If set to 0, it indicates that the real-time clock has lost power and the data in CMOS is no longer valid.

Bits 6–0 These bits are reserved.

CMOS RAM Configuration

Figure 2-20 shows the bit definitions for the CMOS RAM configuration bytes.

Diagnostic Status Byte (X'00E')

Bit	Function
7	Real-time clock power
6	Configuration record and checksum status
5	Incorrect configuration
4	Memory size mismatch
3	Hard disk controller/drive C initialization status
2	Time status indicator
1, 0	Reserved

Figure 2-20. Diagnostic Status Byte (X'00E')

Bit 7 If set to 1, this bit indicates that the real-time clock has lost power.

Bit 6 If set to 1, this bit indicates that the checksum is incorrect.

Bit 5 This bit indicates the results of a power-on check of the equipment byte (X'014'). If set to 1, this bit indicates that the configuration information is incorrect.

Bit 4 When set to 1, this bit indicates that the memory size does not match the configuration information.

Bit 3 When set to 1, this bit indicates that the controller or hard disk drive failed initialization.

Bit 2 When set to 1, this bit indicates that the time is invalid.

Bits 1, 0 These bits are reserved.

Shutdown Status Byte (X'00F'): This byte is defined by the power-on diagnostic programs.

Diskette Drive Type Byte (X'010'): This byte indicates the type of the installed diskette drive.

Bit	Drive Type
7–4	Diskette drive type
3–0	Reserved

Figure 2-21. Diskette Drive Type Byte (X'010')

Bits 7–4 These bits indicate the diskette drive type.

Bits 7–4	Description
0 1 1 0	Diskette drive (2.88 MB)
0 1 0 0	Diskette drive (1.44 MB)
Note: Combinations not shown are reserved.	

Figure 2-22. Diskette Drive Type Bits 7–4

Bits 3–0 These bits are reserved.

Hard Disk Drive Type Byte (X'011'): This byte defines the type of hard disk drive installed. X'00' indicates that no hard disk drive is installed.

Bit	Drive Type
7–4	Hard disk drive type 2
3–0	Hard disk drive type 3

Figure 2-23. Hard Disk Type Byte (X'011')

Bit 7–4	Description
0 0 0 0	No drive installed for hard disk drive 2
1 1 1 1	Use CMOS X'1B' for hard disk drive 2

Figure 2-24. Hard Disk Drive Type 2 (Bits 7–4)

Bit 3–0	Description
0 0 0 0	No drive installed for hard disk drive 3
1 1 1 1	Use CMOS X'1C' for hard disk drive 3

Figure 2-25. Hard Disk Drive Type 3 (Bits 3–0)

Hard Disk Drive Type Byte (X'012'): This byte defines the type of hard disk drive installed. X'00' indicates that no hard disk drive is installed.

Bit	Drive Type
7–4	Hard disk drive 0
3–0	Hard disk drive 1

Figure 2-26. Hard Disk Drive Type Byte

Reserved Bytes (X'013'): These bytes are reserved.

Equipment Byte (X'014'): This byte defines the basic equipment in the system for the power-on diagnostic tests.

Bit	Description
7, 6	Number of diskette drives
5, 4	Display operating mode
3, 2	Reserved
1	Coprocessor presence
0	Diskette drive 0 presence

Figure 2-27. Equipment Byte

Bits 7, 6 These bits indicate the number of installed diskette drives.

Bits 7,6	Number of Diskette Drives
0 0	One drive
0 1	Reserved
1 0	Reserved
1 1	Reserved

Figure 2-28. Installed Diskette Drive Bits

Bits 5, 4 These bits indicate the operating mode of the display attached to the video port.

Bits 5,4	Display Operating Mode
0 0	Reserved
0 1	40-column mode
1 0	80-column mode
1 1	Monochrome mode

Figure 2-29. Display Operating Mode Bits

Bits 3–2 These bits are reserved.

Bit 1 If set to 1, this bit indicates that a coprocessor is installed.

Bit 0 If set to 1, this bit indicates that physical diskette drive 0 is installed.

Low and High Base Memory Bytes (X'015' and X'016'): The low and high base memory bytes define the amount of memory below the 640-KB address space.

The value in these bytes represents the number of 1-KB blocks of base memory. For example, X'0280' indicates 640 KB. The low byte is X'015'; the high byte is X'016'.

Low and High Expansion Memory Bytes (X'017' and X'018'): The low and high expansion memory bytes define the amount of memory above the 1-MB address space.

The value in these bytes represents the number of 1-KB blocks of expansion memory. For example, X'0800' indicates 2048 KB. The low byte is X'017'; the high byte is X'018'.

Reserved Bytes (X'01D'–X'02D'): These bytes are reserved.

Configuration Checksum Bytes (X'02E' and X'02F'): The configuration checksum bytes contain the checksum character for bytes X'010' through X'02D' of the 64-byte CMOS RAM. The high byte is X'02E'; the low byte is X'02F'.

Low and High Usable Memory Bytes (X'030' and X'031'): The low and high usable memory bytes define the total amount of contiguous memory from 1 MB to 20 MB.

The hexadecimal values in these bytes represent the number of 1-KB blocks of usable memory. For example, X'0800' is equal to 2048 KB. The low byte is X'30'; the high byte is X'31'.

Date-Century Byte (X'032'): Bits 7 through 0 of the date-century byte contain the binary-coded decimal value for the century. For information about reading and setting this byte, refer to the *IBM Personal System/2 and Personal Computer BIOS Interface*.

Reserved Bytes (X'033'–X'07F'): These bytes are reserved.

Miscellaneous System Functions and Ports

This section provides information about nonmaskable interrupts (NMIs), the power-on password, and hardware compatibility.

Nonmaskable Interrupt (NMI)

The NMI signals the system microprocessor that a parity error or a channel check timeout has occurred. This situation can cause lost data or an overrun error on some I/O devices. The NMI masks all other interrupts. The interrupt return (IRET) instruction restores the interrupt flag to the state it was in before the interrupt occurred. A system reset causes a reset of the NMI.

The NMI requests from system board parity and channel check are subject to mask control with the NMI mask bit in the RT/CMOS Address register. See “RT/CMOS Address and NMI Mask Register (X'0070')” on page 2-18. The power-on default of the NMI mask is 1 (NMI disabled). Before the NMI is enabled after a power-on reset, the parity-check states are initialized by POST.

Attention

The operation following a write to X'0070' should access X'0071'; otherwise, intermittent failures of the RT/CMOS RAM can occur.

System Control Port B (X'0061')

Bit definitions for the write and read functions of this port are shown in the following figures:

Bit	Function
7–4	Reserved
3	Reserved (should be 0)
2	Enable parity check
1	Enable speaker data
0	Timer 2 gate to speaker

Figure 2-30. System Control Port B (X'0061', Write)

Bit	Function
7	Parity check
6	Channel check
5	Timer 2 output
4	Toggles with each refresh request
3	Reserved
2	Enable parity check
1	Enable speaker data
0	Timer 2 gate to speaker

Figure 2-31. System Control Port B (X'0061', Read)

Bit 7 If set to 1, this bit indicates that the PCI system error (SERR#) was pulsed active.

Bit 6 If set to 1, this bit indicates a channel check has occurred.

Bit 5 If read, this bit indicates the condition of the timer/counter 2 'output' signal.

Bit 4 If read, this bit toggles for each refresh request.

Bit 3 Reserved.

Bit 2 If set to 0, this bit enables the PCI system error (SERR#). This bit is set to 1 during a power-on reset.

Bit 1 If set to 1, this bit enables the speaker data.

Bit 0 If set to 1, this bit enables the timer 2 gate.

System Control Port A (X'0092')

Bit	Function
7–3	Reserved
2	Reserved (must be set to 0)
1	Alternate gate A20
0	Alternate hot reset

Figure 2-32. System Control Port A (X'0092')

Bits 7–3 These bits are reserved.

Bit 2 This bit is reserved.

Bit 1 This bit is used to enable the 'address 20' signal (A20) when the microprocessor is in the real address mode. If this bit is set to 0, A20 cannot be used in real mode addressing. This bit is set to 0 during a system reset.

Bit 0 This bit provides an alternative method of resetting the system microprocessor. This alternative method supports operating systems requiring faster operation than that provided on the IBM Personal Computer AT. Resetting the system microprocessor switches the microprocessor from protected mode to real address mode.

This bit is set to 0 by either a system reset or a write operation. If a write operation changes this bit from 0 to 1, the 'processor reset' signal is pulsed after the reset has occurred. While the reset is occurring, the latch remains set so that POST can read this bit. If the bit is set to 0, POST assumes that the system was just powered on. If the bit is set to 1, POST assumes that the microprocessor has been switched from protected mode to real mode.

If bit 0 is used to reset the system microprocessor to the real mode, use the following procedure:

1. Disable all maskable and nonmaskable interrupts.
2. Reset the system microprocessor by writing a 1 to bit 0.
3. Issue a Halt instruction to the system microprocessor.
4. Reenable all maskable and nonmaskable interrupts.

If you do not follow this procedure, the results are unpredictable.

Note: Whenever possible, use BIOS as an interface to reset the system microprocessor to the real mode. For more information about resetting the system microprocessor, refer to the *IBM Personal System/2 and Personal Computer BIOS Interface*.

Power-On Password

RT/CMOS RAM has 8 bytes reserved for the power-on password and the check character. The 8 bytes are initialized to X'00'. The microprocessor can access these bytes only during POST. After POST is completed, if a power-on password is installed, the password bytes are locked and cannot be accessed by any program.

During power-on password installation, the password (1 to 7 characters) is stored in the security space.

Installing the password is a function of the built-in system program *Easy-Setup*. The power-on password does not appear on the screen when it is installed, changed, or removed. After the power-on password has been installed, it can be changed or removed only during POST.

Selectable Drive-Startup Sequence

Selectable drive-startup (selectable boot) allows you to control the startup sequence of the drives in your computer. The order in which the computer looks for the drives for your operating system is the *drive-startup sequence*. If you are working with multiple operating systems, you might want to change the drive-startup sequence to load the operating system from the hard disk without first checking the diskette drive, or to do a remote program load (RPL).

Attention

When changing your startup sequence, you must be extremely careful when doing write operations (such as copying, saving, or formatting). Your data or programs can be overwritten if you select the wrong drive.

For more information about the selectable drive-startup sequence, refer to the *ThinkPad User's Guide*.

Hardware Compatibility

The computer supports most of the interfaces used by the IBM Personal Computer AT and the Personal System/2 (PS/2) products. In many cases, command and status organization of these interfaces are maintained.

The functional interfaces for the computer are compatible with the following:

- The Intel 8259 interrupt controllers (edge trigger mode).
- The Intel 8254 timers driven from 1.193 MHz (channels 0, 1, and 2).
- The Intel 8237 DMA controller-address/transfer counters, page registers, and status fields only. The command and request registers, and the rotate and mask functions, are not supported. The mode register is partially supported.
- The NS16550 serial communications controller.
- The Intel® Mobile Pentium® II microprocessor.
- The Intel 8086**, 8088**, 80286**, 80386**, and i486DX microprocessors.
- The Intel 8087**, 80287**, 80387** math coprocessors.
- The Intel 82077AA** diskette drive controller.
- The keyboard interface at addresses X'0060' and X'0064'.
- Display modes supported by the IBM Monochrome Display and Printer Adapter, the IBM Color/Graphics Monitor Adapter, and the IBM Enhanced Graphics Adapter.
- The parallel printer ports (Parallel 1, Parallel 2, and Parallel 3) in compatibility mode.

Error Codes

POST returns a three or more character code message to indicate the type of test that failed. Figure 2-33 lists the failure indicated with the associated error code.

Error Code	Description
101	Interrupt failure.
102	Timer failure.
103	Timer interrupt failure.
104	Protected mode failure.
105	Last 8042 command not accepted.
107	NMI test failure.
108	Timer bus test failure.
109	Low meg-chip select test.
110	Planar parity.
111	I/O parity.
118	Planar parity error logged.
158	A supervisor password is set, but no hard disk password is set.
159	The hard disk password is not identical to the supervisor password.
161	Dead battery.
163	Date and time are not set; clock not updated.
173	CMOS CRC error.
174	Configuration error.
175	Bad EEPROM CRC 1.
177	Bad supervisor password checksum.
178	EEPROM is not functional.
179	NVRAM error log full.
183	Supervisor password is needed.
184	Bad power-on password checksum.
185	Corrupted startup boot sequence.
186	Inconsistency between EEPROM and security lock latch 2.
188	Bad EEPROM CRC 2.
189	Too many passwords attempted.
190	Critically low battery condition detected.
191XX	PM initialization error.
192	Fan error.
195	Configuration mismatch error found during hibernation wake-up.
196	Critical error found during hibernation wake-up.
201	Memory data error.
202	Memory line error 00 through 15.
203	Memory line error 16 through 23.
215	Memory test failure on on-board memory.
221	ROM to RAM remap error.

Figure 2-33 (Part 1 of 2). Error Codes

Error Code	Description
225	Unsupported DIMM.
301	Keyboard error.
601	Diskette drive or controller error.
602	No valid boot record on diskette.
604	Invalid diskette drive error.
1101	Serial-A test failure.
1201	Serial-B test failure.
1701	Hard disk controller failure.
1780, 1790	Hard disk 0 error.
1781, 1791	Hard disk 1 error.
1801	Non supported port replicator.
2401	System board video error.
8081	PC Card presence test failure (PC Card revision number also checked).
8082	PC Card register test failure.
8601	System bus error (8042 mouse interface).
8602	External mouse error.
8603	System bus error or mouse error.
8611	System bus error (I/F between 8042 and IPDC).
8612	TrackPoint III error.
8613	System board or TrackPoint III error.
I9990301	Hard disk error.
I9990302	Invalid hard disk boot record.
I9990303	Bank-2 flash ROM checksum error.
I9990305	No bootable device.

Figure 2-33 (Part 2 of 2). Error Codes

Section 3. Subsystems

This section describes the video, DSP, IR, and PC Card subsystems of the ThinkPad computers. If also provides the Programmable Option Select (POS) information for the video, DSP, and IR subsystems.

Video Subsystem

The video subsystem consists of the XGA video controller, which inteprets the monitor buffer. The video subsystem supports an IBM thin-film transistor (TFT) or High Performance Addressing (HPA) as follows:

LCD Type	VRAM Size	Color Depth		Resolution	
		On the LCD	On the External Monitor	On the LCD	On the External Monitor
XGA TFT **XGA HPA**	2 MB	65,536	16,777,216	800×600	640×480 800×600 1024×768

The video subsystem also supports PS/2 analog displays without any additional adapters.

Color	Resolution
65,536 colors	640×480
	800×600
	1024×768
16,777,216 colors	640×480
	800×600

Video Modes

The video subsystem supports the modes listed in Figure 3-1 and Figure 3-2 on page 3-4:

Mode (Hex)	Type	Colors	Alpha-numeric Format	Buffer Start Address	Box Size	Max Pages	Pels
0, 1	A/N	16	40×25	B8000	8×8	8	320×200
0*, 1*	A/N	16	40×25	B8000	8×14	8	320×350
0#, 1#	A/N	16	40×25	B8000	8×16	8	320×400
2, 3	A/N	16	80×25	B8000	8×8	8	640×200
2*, 3*	A/N	16	80×25	B8000	8×14	8	640×350
2#, 3#	A/N	16	80×25	B8000	8×16	8	640×400
4, 5	APA	4	40×25	B8000	8×8	1	320×200
6	APA	2	80×25	B8000	8×8	1	640×200
7*	A/N	–	80×25	B0000	8×14	8	640×350
7#	A/N	–	80×25	B0000	8×16	8	640×400
D	APA	16	40×25	A0000	8×8	8	320×200
E	APA	16	80×25	A0000	8×8	4	640×200
F	APA	–	80×25	A0000	8×14	2	640×350
10	APA	16	80×25	A0000	8×14	2	640×350
11	APA	2	80×30	A0000	8×16	1	640×480
12	APA	16	80×30	A0000	8×16	1	640×480
13	APA	256	40×25	A0000	8×8	1	320×200

Figure 3-1. BIOS Video VGA Modes

The following shows the video BIOS extended modes for the ThinkPad 560Z computer (containing a NeoMagic NM2160 video chip, which inteprets 2-MB VRAM):

Video Mode	VESA Mode Number (Hex)	External Monitor				LCD
		60	70	75	85	SVGA
320x200x32k	10D		o			o
320x200x64k	10E		o			o
320x240x256	120	o				o
320x240x64k	121	o				o
400x300x256	122	o				o
400x300x64k	123	o				o
512x384x256	124	o				o
512x384x64k	125	o				o
640x400x256	100		o			o
640x480x256	101	o		o	o	o
640x480x32k	110	o		o	o	o
640x480x64k	111	o		o	o	o
640x480xTrue	112	o		o	o	o
800x600x16	102	o		o	o	o
800x600x256	103	o		o	o	o
800x600x32k	113	o		o	o	o
800x600x64k	114	o		o	o	o
800x600xTrue	115	o		o	o	o
1024x768x16	104	o				
1024x756x256	105	o	o	o		
1024x768x32k	116	o	o	o		
1024x768x64k	117	o	o	o		

Figure 3-2. Video BIOS Extended Modes—NeoMagic NM2160

Audio Subsystem

The crystal audio subsystem provides 16-bit stereo audio with high-quality FM music synthesis using four operators per voice.

The Sound Blaster support function provides three system settings: I/O address, IRQ level, and DMA channel.

Sound Blaster Support Function

I/O Address	IRQ Level	DMA Channel
0220–022F (Default)	IRQ 5 (Default)	DMA 0
0240–024F	IRQ 7	DMA 1 (Default)
0338–033F (FM synthesizer)	IRQ 10	–
	IRQ 11	–

Audio Port Specifications

- Audio output:
  - ⅛-inch mini-jack for headphone
  - Headphone speaker output: 22 mW (32 Ω) maximum
  - Maximum output level: 2.4 Vpp
  - Output impedance: 75 Ω
- Audio input:
  - ⅛-inch mini-jack for microphone or line input
  - Microphone gain: 26 dB minimum, 48.5 dB maximum
  - Maximum input level:
    Microphone: 125 mVpp
    Line In: 3.0 Vpp
  - Input impedance:
    Microphone: 47 k Ω
    Line In: 30 k Ω

Infrared (IR) Subsystem

The IR subsystem of ThinkPad 560Z is designed to be compatible with the IrDA** Serial Infrared Physical Layer Link Specification Version 1.0 and Data Link Specification Version 1.0.

The IR subsystem of ThinkPad 560Z is designed to be compatible with the IrDA** Serial Infrared Physical Layer Link Specification Version 1.1 and Data Link Specification Version 1.0.

The I/O address can be selected from the following with the system utility program. The IR subsystem uses one serial port address.

I/O Address	
03F8–03FF	Serial port 1 (Default)
02F8–02FF	Serial port 2
03E8–03EF	Serial port 3
02E8–02EF	Serial port 4

PC Card Subsystem

The system board has two PC Card slots that support the following types of PC Cards:

- 16 bit PC Card Type–I, II, III 5V, 3.3V
- CardBus PC Card Type–I, II, III 3.3V

DMA is not supported.

The maximum current per slot is:

- 500 mA at 5 V dc
- 500 mA at 3.3 V dc
- 50 mA at 12 V dc

The PCI1250 PCI-to-Cardbus Controller Unit[1] is used as the PC Card controller in the system unit. The available interrupt levels are IRQ 3, 4, 5, 7, 9, 10, 11, and 15.

The system unit resumes operation from suspend mode when it receives the 'RI_OUT' signal. The Type I and Type II PC Cards can be installed into either the upper or the lower slot, or into both slots at the same time. The Type III PC Card, however, must be installed only in the lower slot. The Type II PC card cannot be used in the upper slot when a Type III PC Card is used.

Either of the PC Card slots accepts a Zoomed Video Card with appropriate drivers provided by venders.

The PC Card slots are designed according to the PC Card standard released in March 1997.

[1] Manufactured by Texas Instruments Corporation.

Pin Assignments

Figure 3-3 shows the pin assignments for the PC Card slots.

Pin	16-Bit PC Card	32-Bit PC Card
1	Ground	Ground
2	D3	CAD0
3	D4	CAD1
4	D5	CAD3
5	D6	CAD5
6	D7	CAD7
7	CE1#	CC/BE0#
8	A10	CAD9
9	OE	CAD11
10	A11	CAD12
11	A9	CAD14
12	A8	CC/BE1#
13	A13	CPAR
14	A14	CPERR#
15	WE#	CGNT#
16	IRQ#	CINT#
17	Vcc	Vcc
18	Vpp	Vpp
19	A16	CCLK
20	A15	CIRDY#
21	A12	CC/BE2#
22	A7	CAD18
23	A6	CAD20
24	A5	CAD21
25	A4	CAD22
26	A3	CAD23
27	A2	CAD24
28	A1	CAD25
29	A0	CAD26
30	D0	CAD27

Figure 3-3 (Part 1 of 2). PC Card Slot Pin Assignments

Pin	16-Bit PC Card	32-Bit PC Card
31	D1	CAD29
32	D2	Reserved
33	IOIS16#	CCLKRUN#
34	Ground	Ground
35	Ground	Ground
36	CD1#	CCD1#
37	D11	CAD2
38	D12	CAD4
39	D13	CAD6
40	D14	Reserved
41	D15	CAD8
42	CE2	CAD10
43	VS1#	CVS1
44	IORD#	CAD13
45	IOWR#	CAD15
46	A17	CAD16
47	A18	Reserved
48	A19	CBLOCK#
49	A20	CSTOP#
50	A21	CDEVSEL#
51	Vcc	Vcc
52	Vpp	Vpp
53	A22	CTRDY#
54	A23	CFRAME#
55	A24	CAD17
56	A25	CAD19
57	AS2#	CVS2
58	RESET	CRST#
59	WAIT#	CSERR#
60	INPACK#	CREQ#
61	REG#	CC/BE3#
62	SPKR#	CAUDIO
63	STSCHG#	CSTSCHG
64	D8	CAD28
65	D9	CAD30
66	D10	CAD31
67	CD2#	CCD2#
68	GND	GND

Figure 3-3 (Part 2 of 2). PC Card Slot Pin Assignments

The maximum current for +12 V dc is 0.05 A for each slot for Vpp. When the computer is in suspend mode, it requires a current of 0.05 A including both slots.

Appendix A. System Management API (SMAPI) BIOS Overview

What Is SMAPI BIOS?

The ThinkPad Basic Input/Output System (BIOS) provides a special software interface, called the System Management Application Program Interface (SMAPI) BIOS, to control the following unique features of the ThinkPad system:

System information
: This BIOS provides unique ThinkPad information, such as the system identifier (system ID).

System configuration
: The ThinkPad SMAPI BIOS provides system configuration control for such features as display device selection or resource configuration for built-in devices.

Power management
: Through the SMAPI BIOS, the operating system or application software can control the ThinkPad power management features (the power mode or the suspend, hibernation, and resume options).

“Header Image” on page A-4 describes how to use the SMAPI BIOS.

Header Image

Systems that support SMAPI BIOS must provide the following header image in the F000 segment system ROM area at the 16-byte boundary. The client needs to search and find this SMAPI BIOS header image to get the entry point for the service.

Field	Offset	Length	Value
Signature	X'00'	4 bytes	'$SMB' (ASCII)
Version (Major)	X'04'	Byte	X'01'
Version (Minor)	X'05'	Byte	X'00'
Length	X'06'	Byte	X'20'
Checksum	X'07'	Byte	–
Information Word	X'08'	Word	–
Reserved 1	X'0A'	Word	–
Real mode 16-bit offset to entry point	X'0C'	Word	–
Real mode 16-bit code segment address	X'0E'	Word	–
Reserved 2	X'10'	Word	–
16-bit protected mode offset to entry point	X'12'	Word	–
16-bit protected mode code segment base address	X'14'	Doubleword	–
32-bit protected mode offset to entry point	X'18'	Doubleword	–
32-bit protected mode code segment base address	X'1C'	Doubleword	–

Signature ASCII Code '$SMB' is stored at the top of the header image.

Version (Major/Minor)
Indicates the SMAPI BIOS version.

Length The length of the header image.

Checksum Checksum byte area. The client verifies that this header image is valid by using this checksum; the client should check all header image bytes, and the result will be zero bytes.

Information Word

This area identifies the BIOS service level defined below.

```
Information Word
        Bit 0    : Real/V86 mode interface support
        Bit 1    : 16-bit protected mode support
        Bit 2    : 32-bit protected mode support
        Bits 3-15: Reserved
```

Real Mode Entry Point

The entry point is specified in segment, offset format. Clients using Real/V86 mode can use this area for the far-call value.

16-bit or 32-bit Protected Mode Entry Point

The code base code address specifies the physical address for this BIOS, and the client must prepare the selector for this BIOS. The length should be 64 KB.

Calling Convention

The client can invoke the SMAPI BIOS with a far-call to the entry point that is specified in the header file. All parameters for the BIOS and other results are stored in the client data area; the client needs to prepare an input parameter and output parameter area in its data area, and informs this area by pushing those pointers onto its stack before the far-calls.

The SMAPI BIOS uses the stack/data area directly with the selector when the BIOS is invoked. Therefore, the caller needs to define the same privilege level as the BIOS.

Parameter Structure

The memory allocation for the input/output field should be prepared by the caller. The input field specifies the function request to the SMAPI BIOS, and the BIOS fills in the return value to the output field.

Input Field

Field	Offset	Length
Major function number	X'00'	Byte
Minor function number	X'01'	Byte
Parameter 1	X'02'	Word
Parameter 2	X'04'	Word
Parameter 3	X'06'	Word
Parameter 4	X'08'	Doubleword
Parameter 5	X'0C'	Doubleword

Output Field

Field	Offset	Length
Return code	X'00'	Byte
Auxiliary return code	X'01'	Byte
Parameter 1	X'02'	Word
Parameter 2	X'04'	Word
Parameter 3	X'06'	Word
Parameter 4	X'08'	Doubleword
Parameter 5	X'0C'	Doubleword

Sample in Assembler Language

```
;
; Input Parameter Structure
;
SMB_INPARM              STRUC
 @SMBIN_FUNC            DB      ?
 @SMBIN_SUB_FUNC        DB      ?
 @SMBIN_PARM_1          DW      ?
 @SMBIN_PARM_2          DW      ?
 @SMBIN_PARM_3          DW      ?
 @SMBIN_PARM_4          DD      ?
 @SMBIN_PARM_5          DD      ?
SMB_INPARM              ENDS


;
; Output Parameter Structure
;

SMB_OUTPARM             STRUC
 @SMBOUT_RC             DB      ?
 @SMBOUT_SUB_RC         DB      ?
 @SMBOUT_PARM_1         DW      ?
 @SMBOUT_PARM_2         DW      ?
 @SMBOUT_PARM_3         DW      ?
 @SMBOUT_PARM_4         DD      ?
 @SMBOUT_PARM_5         DD      ?
SMB_OUTPARM             ENDS
```

Sample in C Language

```
//
// Input Parameter Structure
//
typedef struct {
   BYTE     SMBIN_FUNC     ;
   BYTE     SMBIN_SUB_FUNC ;
   WORD     SMBIN_PARM_1   ;
   WORD     SMBIN_PARM_2   ;
   WORD     SMBIN_PARM_3   ;
   DWORD    SMBIN_PARM_4   ;
   DWORD    SMBIN_PARM_5   ;
} INPARM, *PINPARM ;


//
// Output Parameter Structure
//
typedef struct {
   BYTE     SMBOUT_RC      ;
   BYTE     SMBOUT_SUB_RC  ;
   WORD     SMBOUT_PARM_1  ;
   WORD     SMBOUT_PARM_2  ;
   WORD     SMBOUT_PARM_3  ;
   DWORD    SMBOUT_PARM_4  ;
   DWORD    SMBOUT_PARM_5  ;
} OUTPARM, *POUTPARM ;


typedef INPARM    far * FPINPARM ;
typedef OUTPARM   far * FPOUTPARM ;
```

Calling Convention Pseudo Code

The following describes the calling convention using pseudo code.

Assembler Language

```
InputParm       SMB_INPARM      < >
OutputParm      SMB_OUTPARM     < >

16-bit

        push    ds
        mov     ax, offset OutputParm
        push    ax
        push    ds
        mov     ax, offset InputParm
        push    ax
        call    dword ptr SmapiBios
        add     sp, 8


32-bit
        push    ds
        mov     eax, offset OutputParm
        push    eax
        push    ds
        mov     eax, offset InputParm
        push    eax
        call    fword ptr SmapiBios
        add     sp, 16
```

C Language

```
typedef WORD (far * SMB)(FPINPARM, FPOUTPARM) ;




SMB         SmapiBios ;
INPARM      InputParm ;
OUTPARM     OutputParm ;
WORD        RC ;

RC = SmapiBios(&InputParm, &OutputParm) ;
```

Return Codes

The following return codes are stored in both the AL (AX) register and the return code field of the output parameter.

`X'00'` No Error
`X'53'` SMAPI function is not available
`X'81'` Invalid parameter
`X'86'` Function is not supported
`X'90'` System error
`X'91'` System is invalid
`X'92'` System is busy
`X'A0'` Device error (disk read error)
`X'A1'` Device is busy
`X'A2'` Device is not attached
`X'A3'` Device is disabled
`X'A4'` Request parameter is out of range
`X'A5'` Request parameter is not accepted

All other values are reserved.

Function Description

System Information Service

Get System Identification

Input Field

```
Major function number - X'00'
Minor function number - X'00'
Parameter 1           - Reserved
Parameter 2           - Reserved
Parameter 3           - Reserved
Parameter 4           - Reserved
Parameter 5           - Reserved
```

Output Field

```
Return code           - Error status
Auxiliary return code - Return value format
                        = X'00': ASCII format
                        = X'01': Binary format
Parameter 1           - System ID
Parameter 2           - Country code
Parameter 3           - System BIOS revision
Parameter 4           - (Bits 31-16): Reserved
                      - (Bits 15-0): System management BIOS revision
                        (= X'0FFFF'): Not valid
Parameter 5           - (Bits 31-16): Reserved
                      - (Bits 15-0): SMAPI BIOS interface revision
```

Get CPU Information

Input Field

```
Major function number - X'00'
Minor function number - X'01'
Parameter 1           - Reserved
Parameter 2           - Reserved
Parameter 3           - Reserved
Parameter 4           - Reserved
Parameter 5           - Reserved
```

Output Field

```
Return code           - Error status
Auxiliary return code - Reserved
Parameter 1           - (Bits 15-8): Reserved
                        (Bits 7-0): CPU maker
                         X'01': Intel
                         X'02': AMD
Parameter 2           - CPU ID
                        (Bits 15-8): Microprocessor type
                        (Bits 7-0): Microprocessor stepping level
                        = X'FFFF': Unknown
Parameter 3           - Clock Information
                        (Bits 15-8): CPU clock (units: MHz)
                        = X'FE': CPU clock is over 254 MHz
                                 (Parameter 4 is valid.)
                        = X'FF': Unknown
                        (Bits 7-0): Internal clock (units: MHz)
                        = X'FE': Internal clock is over 254 MHz
                                 (Parameter 5 is valid.)
                        = X'FF': Unknown
Parameter 4           - (Bits 31-16): Reserved
                        (Bits 15-0): CPU clock (units: MHz)
Parameter 5           - (Bits 31-16): Reserved
                        (Bits 15-0): Internal clock (units: MHz)
```

Get Display Device Information

Input Field

```
Major function number - X'00'
Minor function number - X'02'
Parameter 1           - (Bits 15-8): Request type
                          Bit 8: LCD information
                          Bit 9: External CRT information
                          Bits 15-10: Reserved
                        (Bits 7-0): Reserved
Parameter 2           - Reserved
Parameter 3           - Reserved
Parameter 4           - Reserved
Parameter 5           - Reserved
```

Output Field

```
Return code           - Error status
Auxiliary return code - Reserved
Parameter 1           - (Bits 15-8):
                        Built-in display device (panel)
                        information 1
                        = X'00': Monochrome STN LCD
                        = X'01': Monochrome TFT LCD
                        = X'02': Color STN LCD
                        = X'03': Color TFT LCD
                        = X'FF': Unknown
                        (Bits 7-0):
                        Built-in display device (panel)
                        information 2
                        = X'00': 640x480
                        = X'01': 800x600
                        = X'02': 1024x768
                        = X'03': 1024x1024
                        = X'FF': Unknown
Parameter 2           - (Bits 15-8):
                        External CRT monitor
                        information
                        = X'00': External CRT is not attached
                        = X'10': Color monitor
                        = X'20': Monochrome monitor
                        = X'FF': Unknown
                        (Bits 7-0):
                        External CRT monitor
                        information 2
                         Bit 0: The CRT has DDC1 capability.
                         Bit 1: The CRT has DDC2 capability.
                                (2b, 2b+ or 2ab)
                         Bits 7-2: Reserved
Parameter 3           - Reserved
Parameter 4           - Reserved
Parameter 5           - Reserved
```

Get Power Management Module Information

Input Field

```
Major function number - X'00'
Minor function number - X'06'
Parameter 1           - Reserved
Parameter 2           - Reserved
Parameter 3           - Reserved
Parameter 4           - Reserved
Parameter 5           - Reserved
```

Output Field

```
Return code           - Error status
Auxiliary return code - Return value format
                        = X'00': ASCII format
                        = X'01': Binary format
Parameter 1           - Reserved
Parameter 2           - Slave controller revision
                        (= X'0FFFF'): Not valid
Parameter 3           - Reserved
Parameter 4           - Reserved
Parameter 5           - Reserved
```

Get System Current Status

Input Field

```
Major function number - X'00'
Minor function number - X'07'
Parameter 1           - Reserved
Parameter 2           - Reserved
Parameter 3           - Reserved
Parameter 4           - Reserved
Parameter 5           - Reserved
```

Output Field

```
Return code           - Error status
Auxiliary return code - Reserved
Parameter 1           - Reserved
Parameter 2           - (Bits 15-8): Current status
                          Bit 8: LID Status
                                 = 0: Open
                                 = 1: Close
                          Bit 9: Keyboard status
                                 = 0: Close
                                 = 1: Open
                          Bit 10: AC Adapter
                                 = 0: Not attached
                                 = 1: Attached
                          Bits 15-11: Reserved
                        (Bits 7-0): Reserved
Parameter 3           - Reserved
Parameter 4           - Reserved
Parameter 5           - Reserved
```

Get Video Information

Input Field

```
Major function number - X'00'
Minor function number - X'08'
Parameter 1           - Reserved
Parameter 2           - Reserved
Parameter 3           - Reserved
Parameter 4           - Reserved
Parameter 5           - Reserved
```

Output Field

```
Return code           - Error status
Auxiliary return code - Reserved
Parameter 1           - Video BIOS revision
Parameter 2           - Reserved
Parameter 3           - Reserved
Parameter 4           - Reserved
Parameter 5           - Reserved
```

Get Refresh Rate Capability

Input Field

```
Major function number - X'00'
Minor function number - X'09'
Parameter 1           - Mode
                        = X'00xx': VGA modes.
                        (Bits 7-0 are ignored.)
                        = X'0100': 640x400x256
                        = X'0101': 640x480x256
                        = X'0110': 640x480x32K
                        = X'0111': 640x480x64K
                        = X'0112': 640x480x16M
                        = X'0102': 800x600x16
                        = X'0103': 800x600x256
                        = X'0113': 800x600x32K
                        = X'0114': 800x600x64K
                        = X'0115': 800x600x16M
                        = X'0104': 1024x768x16
                        = X'0105': 1024x768x256
                        = X'0116': 1024x768x32K
                        = X'0117': 1024x768x64K
                        = X'0118': 1024x768x16M
                        = X'0106': 1280x1024x16
                        = X'0107': 1280x1024x256
                        = X'0119': 1280x1024x32K
                        = X'011A': 1280x1024x64K
                        = X'011B': 1280x1024x16M
                        = X'0A00': 1600x1200x16
                        = X'0A01': 1600x1200x256
                        = X'0A02': 1600x1200x32K
                        = X'0A03': 1600x1200x64K
                        = X'0A04': 1600x1200x16M
                        = X'0109': 1056x350x16
                        = X'010A': 1056x473x16
                        = X'010C': 1056x480x16
                        = Others: Reserved
Parameter 2           - Reserved
Parameter 3           - Reserved
Parameter 4           - Reserved
Parameter 5           - Reserved
```

Output Field

```
Return code           - Error status
Auxiliary return code - Reserved
Parameter 1           - Reserved
Parameter 2           - Refresh rate capability for
                        specified mode:
                         Bit 0: 60 Hz available.
                         Bit 1: 72 Hz available.
                         Bit 2: 75 Hz available.
                         Bit 3: 43 Hz(I) available.
                         Bit 4: 56 Hz available.
                         Bit 5: 70 Hz available.
                         Bit 6: 85 Hz available.
                         Bit 7: 48 Hz(I) available.
                         Bits 15-8: Reserved (must be B'0').
Parameter 3           - Reserved
Parameter 4           - Reserved
Parameter 5           - Reserved
```

System Configuration Service

Get Display Device State

Input Field

```
Major function number - X'10'
Minor function number - X'00'
Parameter 1           - Request type
                        = X'0000': Current hardware
                        = X'0001': CMOS (effective after reboot)
Parameter 2           - Reserved
Parameter 3           - Reserved
Parameter 4           - Reserved
Parameter 5           - Reserved
```

Output Field

```
Return code           - Error status
Auxiliary return code - Reserved
Parameter 1           - (Bits 15-8): Reserved
                        (Bits 7-0):
                        Display device function capability
                         Bit 0: Display function type
                                = 0: Not supported
                                = 1: Supported
                         Bits 7-1: Reserved
Parameter 2           - (Bits 15-8): Display current status
                         Bit 8: Built-in display (panel) status
                                = 0: Disable
                                = 1: Enable
                         Bit 9: CRT status
                                = 0: Disable
                                = 1: Enable
                         Bit 10: TV status
                                = 0: Disable
                                = 1: Enable
                         Bits 14-11: Reserved
                         Bit 15: Dual enable flag
                                = 0: Disable
                                = 1: Enable
                        (Bits 7-0): Display function type
                        = X'00': Model with no TV out
                        = X'01': Model with no simultaneous
                                 display of TV and CRT
                                 (Parameter 4 is valid.)
Parameter 3           - Reserved
Parameter 4           - (Bits 32-16): Reserved
                        (Bits 15-0): Display selection mode
                         Bit 0: Display selection mode
                                = 0: LCD-CRT selection mode
                                = 0: LCD-TV selection mode
                         Bits 15-1: Reserved
Parameter 5           - Reserved
```

Set Display Device State

Input Field

```
Major function number - X'10'
Minor function number - X'01'
Parameter 1           - (Bits 15-8):
                        Request display status
                         Bit 8: Built-in display (panel) status
                                = 0: Disable
                                = 1: Enable
                         Bit 9: CRT status
                                = 0: Disable
                                = 1: Enable
                         Bit 10: TV status
                                = 0: Disable
                                = 1: Enable
                         Bits 13-11: Reserved
                         Bit 14: Monitor detection ignore
                                = 0: Do not ignore (should be)
                                = 1: Ignore
                        Bit 15: Dual enable flag
                                = 0: Disable
                                = 1: Enable
                        (Bits 7-0): Reserved
Parameter 2           - Reserved
Parameter 3           - Reserved
Parameter 4           - Reserved
Parameter 5           - Reserved
```

Output Field

```
Return code           - Error status
Auxiliary return code - Reserved
Parameter 1           - Reserved
Parameter 2           - Reserved
Parameter 3           - Reserved
Parameter 4           - Reserved
Parameter 5           - Reserved
```

Get Pointing Device State

Input Field

```
Major function number - X'11'
Minor function number - X'02'
Parameter 1           - (Bits 15-8): Request type
                        = X'00': Current hardware
                        = X'01': CMOS (effective after reboot)
                        (Bits 7-0): Reserved
Parameter 2           - Reserved
Parameter 3           - Reserved
Parameter 4           - Reserved
Parameter 5           - Reserved
```

Output Field

```
Return code           - Error status
Auxiliary return code - Reserved
Parameter 1           - Reserved
Parameter 2           - (Bits 15-8): Pointing device current status
                          Bits 10, 8: Built-in pointing device control
                                = 00: Disable
                                = 01: Enable
                                = 10: Auto
                                = 11: Reserved
                          Bit 9: External pointing device status
                                = 0: Disable
                                = 1: Enable
                          Bits 15-11: Reserved
                         (Bits 7-0): Pointing device capability
                          Bit 0: Built-in pointing device status
                                = 0: Status is not controllable
                                = 1: Status is controllable
                          Bit 1: External pointing device status
                                = 0: Status is not controllable
                                = 1: Status is controllable
                          Bit 2: Built-in pointing device auto control
                                = 0: Not supported
                                = 1: Supported
                          Bits 7-3: Reserved
Parameter 3           - Reserved
Parameter 4           - Reserved
Parameter 5           - Reserved
```

Set Pointing Device State

Input Field

```
Major function number - X'11'
Minor function number - X'03'
Parameter 1           - (Bits 15-8):
                        Request pointing device current status
                         Bits 10, 8: Built-in pointing device control
                               = 00: Disable
                               = 01: Enable
                               = 10: Auto
                               = 11: Reserved
                         Bit 9: External pointing device status
                               = 0: Disable
                               = 1: Enable
                         Bits 15-11: Reserved
                        (Bits 7-0): Request type
                         Bit 0: Current hardware
                         Bit 1: CMOS (effective after reboot)
                         Bits 7-2: Reserved
Parameter 2           - Reserved
Parameter 3           - Reserved
Parameter 4           - Reserved
Parameter 5           - Reserved
```

Output Field

```
Return code           - Error status
Auxiliary return code - Reserved
Parameter 1           - Reserved
Parameter 2           - Reserved
                        Reserved
Parameter 3           - Reserved
Parameter 4           - Reserved
Parameter 5           - Reserved
```

Get Hotkey Sticky/Lock

Input Field

```
Major function number - X'13'
Minor function number - X'02'
Parameter 1           - Reserved
Parameter 2           - Reserved
Parameter 3           - Reserved
Parameter 4           - Reserved
Parameter 5           - Reserved
```

Output Field

```
Return code           - Error status
Auxiliary return code - Reserved
Parameter 1           - Reserved
Parameter 2           - (Bits 15-8): Capability
                           Bit 8: Sticky Fn key support
                           Bit 9: Sticky and Lock Fn key support
                                  = 0: Not supported
                                  = 1: Supported
                         Bits 15-10: Reserved
                        (Bits 7-0): Current status
                        = X'00': Function is disabled
                        = X'01': Sticky Fn key
                        = X'03': Sticky and Lock Fn key
Parameter 3           - Reserved
Parameter 4           - Reserved
Parameter 5           - Reserved
```

Set Hotkey Sticky/Lock Support

Input Field

```
Major function number - X'13'
Minor function number - X'03'
Parameter 1           - (Bits 15-8): Reserved
                        (Bits 7-0): Request status
                        = X'00': Function is disabled
                        = X'01': Sticky Fn key
                        = X'03': Sticky and Lock Fn key
Parameter 2           - Reserved
Parameter 3           - Reserved
Parameter 4           - Reserved
Parameter 5           - Reserved
```

Output Field

```
Return code           - Error status
Auxiliary return code - Reserved
Parameter 1           - Reserved
Parameter 2           - Reserved
Parameter 3           - Reserved
Parameter 4           - Reserved
Parameter 5           - Reserved
```

Power Management Service

Get Power Management Mode

Input Field

```
Major function number - X'22'
Minor function number - X'00'
Parameter 1           - Reserved
Parameter 2           - Reserved
Parameter 3           - Reserved
Parameter 4           - Reserved
Parameter 5           - Reserved
```

Output Field

```
Return code           - Error status
Auxiliary return code - Reserved
Parameter 1           - Reserved
Parameter 2           - (Bits 15-8): Power management mode
                             battery operation
                        = X'00': High performance mode
                        = X'01': Auto power management mode
                        = X'02': Manual power management mode
                        (Bits 7-0): Power management mode
                             AC operation
                        = X'00': High performance mode
                        = X'01': Auto power management mode
                        = X'02': Manual power management mode
Parameter 3           - Reserved
Parameter 4           - Reserved
Parameter 5           - Reserved
```

Set Power Management Mode

Input Field

```
Major function number - X'22'
Minor function number - X'01'
Parameter 1           - (Bits 15-8): Power management mode
                             battery operation
                        = X'00': High performance mode
                        = X'01': Auto power management mode
                        = X'02': Manual power management mode
                        (Bits 7-0): Power management mode
                             AC operation
                        = X'00': High performance mode
                        = X'01': Auto power management mode
                        = X'02': Manual power management mode
Parameter 2           - Reserved
Parameter 3           - Reserved
Parameter 4           - Reserved
Parameter 5           - Reserved
```

Output Field

```
Return code           - Error status
Auxiliary return code - Reserved
Parameter 1           - Reserved
Parameter 2           - Reserved
                        Reserved
Parameter 3           - Reserved
Parameter 4           - Reserved
Parameter 5           - Reserved
```

Get Timer Control

Input Field

```
Major function number - X'22'
Minor function number - X'02'
Parameter 1           - Reserved
Parameter 2           - Reserved
Parameter 3           - Reserved
Parameter 4           - Reserved
Parameter 5           - Reserved
```

Output Field

```
Return code           - Error status
Auxiliary return code - Reserved
Parameter 1           - Reserved
Parameter 2           - (Bits 15-8): Capability of timer control
                          Bit 8: System (hibernation/suspend) timer
                                 = 0: Not supported
                                 = 1: Supported
                          Bit 9: Standby timer
                                 = 0: Not supported
                                 = 1: Supported
                          Bit 10: LCD off timer
                                 = 0: Not supported
                                 = 1: Supported
                          Bit 11: HDD off timer
                                 = 0: Not supported
                                 = 1: Supported
                        Bits 15-12: Reserved
                       (Bits 7-0): Timer control
                        Bit 0: System (hibernation/suspend) timer
                                 = 0: Disable
                                 = 1: Enable
                        Bit 1: Standby timer
                                 = 0: Disable
                                 = 1: Enable
                        Bit 2: LCD off timer
                                 = 0: Disable
                                 = 1: Enable
                        Bit 3: HDD off timer
                                 = 0: Disable
                                 = 1: Enable
                        Bits 7-4: Reserved
Parameter 3           - Reserved
Parameter 4           - Reserved
Parameter 5           - Reserved
```

Set Timer Control

Input Field

```
Major function number - X'22'
Minor function number - X'03'
Parameter 1           - (Bits 15-8): Reserved
                        (Bits 7-0): Timer control
                         Bit 0: System (hibernation/suspend) timer
                               = 0: Disable
                               = 1: Enable
                         Bit 1: Standby timer
                               = 0: Disable
                               = 1: Enable
                         Bit 2: LCD off timer
                               = 0: Disable
                               = 1: Enable
                         Bit 3: HDD off timer
                               = 0: Disable
                               = 1: Enable
                         Bits 7-4: Reserved
Parameter 2           - Reserved
Parameter 3           - Reserved
Parameter 4           - Reserved
Parameter 5           - Reserved
```

Output Field

```
Return code           - Error status
Auxiliary return code - Reserved
Parameter 1           - Reserved
Parameter 2           - Reserved
                        Reserved
Parameter 3           - Reserved
Parameter 4           - Reserved
Parameter 5           - Reserved
```

Event Bit Definition

```
Bits 2-0 - Reserved
Bit 3    - Standby
Bit 4    - Suspend
Bit 5    - Safe suspend
Bit 6    - Hibernation
Bit 7    - Power off
```

Note: If bits are duplicated, the highest bit is available.

Get System Event Global Condition

Input Field

```
Major function number - X'30'
Minor function number - X'00'
Parameter 1           - Reserved
Parameter 2           - Reserved
Parameter 3           - Reserved
Parameter 4           - Reserved
Parameter 5           - Reserved
```

Output Field

```
Return code           - Error status
Auxiliary return code - Reserved
Parameter 1           - Reserved
Parameter 2           - (Bits 15-8): Capability for event
                          Bit 8: Safe suspend is
                               controlled by global conditions.
                               (Safe suspend bit is ignored
                               in each event condition.)
                                = 0: Not supported
                                = 1: Supported
                        (Bits 7-0): Global condition
                          Bit 0: Enable safe suspend
                               if suspend is selected.
                                = 0: Disable
                                = 1: Enable
Parameter 3           - Reserved
Parameter 4           - Reserved
Parameter 5           - Reserved
```

Set System Event Global Condition

Input Field

```
Major function number - X'30'
Minor function number - X'01'
Parameter 1           - (Bits 15-8): Reserved
                        (Bits 7-0): Global condition for event
                         Bit 0: Enable safe suspend if suspend
                         is selected.
                               = 0: Disable
                               = 1: Enable
Parameter 2           - Reserved
Parameter 3           - Reserved
Parameter 4           - Reserved
Parameter 5           - Reserved
```

Output Field

```
Return code           - Error status
Auxiliary return code - Reserved
Parameter 1           - Reserved
Parameter 2           - Reserved
                        Reserved
Parameter 3           - Reserved
Parameter 4           - Reserved
Parameter 5           - Reserved
```

Get System Event 1 Condition

Input Field

```
Major function number - X'31'
Minor function number - X'00'
Parameter 1           - Reserved
Parameter 2           - Reserved
Parameter 3           - Reserved
Parameter 4           - Reserved
Parameter 5           - Reserved
```

Output Field

```
Return code           - Error status
Auxiliary return code - Reserved
Parameter 1           - Reserved
Parameter 2           - Hardware and software
                        event definition
                         Bits 15-8: Capability (see page A-33)
                         Bits 7-0: Condition (see page A-33)
Parameter 3           - Reserved
Parameter 4           - (Bits 31-16): Reserved
                        (Bits 15-0): Power switch detection
                                     event definition
                         Bits 15-8: Capability (see page A-33)
                         Bits 7-0: Condition (see page A-33)
Parameter 5           - (Bits 31-16): Reserved
                        (Bits 15-0): LID close detection
                                     event definition
                         Bits 15-8: Capability (see page A-33)
                         Bits 7-0: Condition (see page A-33)
```

Set System Event 1 Condition

Input Field

```
Major function number - X'31'
Minor function number - X'01'
Parameter 1           - (Bits 15-8): Reserved
                      - (Bits 7-0):
                        Condition for hardware and software event
                          (see page A-33)
Parameter 2           - Reserved
Parameter 3           - Reserved
Parameter 4           - (Bits 31-8): Reserved
                        (Bits 7-0):
                        Condition for power switch detection
                          (See page A-33)
Parameter 5           - (Bits 31-8): Reserved
                        (Bits 7-0):
                        Condition for LID close detection
                          (See page A-33)
```

Output Field

```
Return code           - Error status
Auxiliary return code - Reserved
Parameter 1           - Reserved
Parameter 2           - Reserved
Parameter 3           - Reserved
Parameter 4           - Reserved
Parameter 5           - Reserved
```

Get System Event 2 Condition

Input Field

```
Major function number - X'32'
Minor function number - X'00'
Parameter 1           - Reserved
Parameter 2           - Reserved
Parameter 3           - Reserved
Parameter 4           - Reserved
Parameter 5           - Reserved
```

Output Field

```
Return code           - Error status
Auxiliary return code - Reserved
Parameter 1           - Reserved
Parameter 2           - System timer expiry
                        event definition
                         Bits 15-8: Capability (see page A-33)
                         Bits 7-0: Condition (see page A-33)
Parameter 3           - Reserved
Parameter 4           - (Bits 31-16): Reserved
                        (Bits 15-0): Standby timer expiry
                        event definition
                         Bits 15-8: Capability (see page A-33)
                         Bits 7-0: Condition (see page A-33)
Parameter 5           - (Bits 31-16): Reserved
                        (Bits 15-0):
                         Hibernation timer during suspend
                         mode expiry event definition
                         Bits 15-8: Capability (see page A-33)
                         Bits 7-0: Condition (see page A-33)
```

Set System Event 2 Condition

Input Field

```
Major function number - X'32'
Minor function number - X'01'
Parameter 1           - (Bits 15-8): Reserved
                      - (Bits 7-0): Condition for system timer expiry
                          (see page A-33)
Parameter 2           - Reserved
Parameter 3           - Reserved
Parameter 4           - (Bits 31-8): Reserved
                        (Bits 7-0): Condition for standby
                        timer expired
                          (see page A-33)
Parameter 5           - (Bits 31-8): Reserved
                        (Bits 7-0): Condition for hibernation
                        timer during suspend mode expired
                          (see page A-33)
```

Output Field

```
Return code           - Error status
Auxiliary return code - Reserved
Parameter 1           - Reserved
Parameter 2           - Reserved
Parameter 3           - Reserved
Parameter 4           - Reserved
Parameter 5           - Reserved
```

Get System Timer

Input Field

```
Major function number - X'32'
Minor function number - X'02'
Parameter 1           - (Bits 15-8): Power mode select
                        = X'00': Reserved
                        = X'01': Manual PM mode (ac)
                        = X'02': Manual PM mode (battery)
                        = X'F3': High performance mode
                        = X'F4': Auto power management mode
                        (Bits 7-0): Reserved
Parameter 2           - Reserved
Parameter 3           - Reserved
Parameter 4           - Reserved
Parameter 5           - Reserved
```

Output Field

```
Return code           - Error status
Auxiliary return code - Reserved
Parameter 1           - (Bits 15-8): System timer capability
                         Bit 8 = 0: Timer cannot be specified
                                    in each power mode
                               = 1: Timer can be specified
                                    in each power mode
                         Bits 15- 9: Reserved
                        (Bits 7-0): Reserved
Parameter 2           - (Bits 15-8): Reserved
                        (Bits 7-0): System timer initial value
                                    (units: minutes)
                        = X'00': Disable system timer
Parameter 3           - Reserved
Parameter 4           - Reserved
Parameter 5           - Reserved
```

Set System Timer

Input Field

```
Major function number - X'32'
Minor function number - X'03'
Parameter 1           - (Bits 15-8): Power mode select
                        = X'00': All mode
                        = X'01': Manual PM mode (ac)
                        = X'02': Manual PM mode (battery)
                        = X'F3': High performance mode
                        = X'F4': Auto power management mode
                        (Bits 7-0): System timer initial
                                    value (units: minutes)
                        = X'00': Disable system timer
Parameter 2           - Reserved
Parameter 3           - Reserved
Parameter 4           - Reserved
Parameter 5           - Reserved
```

Output Field

```
Return code           - Error status
Auxiliary return code - Reserved
Parameter 1           - Reserved
Parameter 2           - Reserved
                        Reserved
Parameter 3           - Reserved
Parameter 4           - Reserved
Parameter 5           - Reserved
```

Get Hibernation Timer

Input Field

```
Major function number - X'32'
Minor function number - X'06'
Parameter 1           - Reserved
Parameter 2           - Reserved
Parameter 3           - Reserved
Parameter 4           - Reserved
Parameter 5           - Reserved
```

Output Field

```
Return code           - Error status
Auxiliary return code - Reserved
Parameter 1           - Reserved
Parameter 2           - (Bits 15-8): Reserved
                        (Bits 7-0): Hibernation timer during
                                    suspend mode initial value
                                    (units: minutes)
                          = X'00h': Disable hibernation timer
                                  during suspend mode
Parameter 3           - Reserved
Parameter 4           - Reserved
Parameter 5           - Reserved
```

Set Hibernation Timer

Input Field

```
Major function number - X'32'
Minor function number - X'07'
Parameter 1           - (Bits 15-8): Reserved
                        (Bits 7-0): Hibernation timer during
                                    suspend mode initial value
                                    (units: minutes)
                        = X'00': Disable hibernation timer
                                 during suspend mode
Parameter 2           - Reserved
Parameter 3           - Reserved
Parameter 4           - Reserved
Parameter 5           - Reserved
```

Output Field

```
Return code           - Error status
Auxiliary return code - Reserved
Parameter 1           - Reserved
Parameter 2           - Reserved
Parameter 3           - Reserved
Parameter 4           - Reserved
Parameter 5           - Reserved
```

Get System Event 3 Condition

Input Field

```
Major function number - X'33'
Minor function number - X'00'
Parameter 1           - Reserved
Parameter 2           - Reserved
Parameter 3           - Reserved
Parameter 4           - Reserved
Parameter 5           - Reserved
```

Output Field

```
Return code           - Error status
Auxiliary return code - Reserved
Parameter 1           - Reserved
Parameter 2           - Critical low battery condition
                                detection event definition
                         Bits 15-8: Capability (see page A-33)
                         Bits 7-0: Condition (see page A-33)
Parameter 3           - Reserved
Parameter 4           - (Bits 31-16): Reserved
                        (Bits 15-0): Out of environment condition
                        detection event definition
                         Bits 15-8: Capability (see page A-33)
                         Bits 7-0: Condition (see page A-33)
Parameter 5           - Reserved
```

Set System Event 3 Condition

Input Field

```
Major function number - X'33'
Minor function number - X'01'
Parameter 1           - (Bits 15-8): Reserved
                        (Bits 7-0): Condition for critical
                        low battery condition detection
                          (see page A-33)
Parameter 2           - Reserved
Parameter 3           - Reserved
Parameter 4           - (Bits 31-8): Reserved
                        (Bits 7-0): Condition for out-of-environment
                        condition detection
                          (see page A-33)
Parameter 5           - Reserved
```

Output Field

```
Return code           - Error status
Auxiliary return code - Reserved
Parameter 1           - Reserved
Parameter 2           - Reserved
Parameter 3           - Reserved
Parameter 4           - Reserved
Parameter 5           - Reserved
```

Get System Resume Condition

Input Field

```
Major function number - X'34'
Minor function number - X'00'
Parameter 1           - Reserved
Parameter 2           - Reserved
Parameter 3           - Reserved
Parameter 4           - Reserved
Parameter 5           - Reserved
```

Output Field

```
Return code           - Error status
Auxiliary return code - Reserved
Parameter 1           - Reserved
Parameter 2           - Condition for resuming trigger
                        from system suspend mode
                         Bit 0: Resume switch by hardware
                         Bit 1: LID open detection
                         Bit 2: RTC alarm (resume timer)
                                detection
                         Bit 3: RI from the serial device
                                detection
                         Bits 15-4: Reserved
Parameter 3           - Capability for resuming trigger from
                        the system suspend mode
                         Bit 0: Resume switch by hardware
                         Bit 1: LID open detection
                         Bit 2: RTC alarm (resume timer)
                                detection
                         Bit 3: RI from the serial device
                                detection
                         Bits 15-4: Reserved
Parameter 4           - Reserved
Parameter 5           - Reserved
```

Set System Resume Condition

Input Field

```
Major function number - X'34'
Minor function number - X'01'
Parameter 1           - Condition for resuming trigger
                        from the system suspend mode
                         Bit 0: Resume switch by hardware
                         Bit 1: LID open detection
                         Bit 2: RTC alarm (resume timer)
                                detection
                         Bit 3: RI from the serial device
                                detection
                         Bits 15-4: Reserved
Parameter 2           - Reserved
Parameter 3           - Reserved
Parameter 4           - Reserved
Parameter 5           - Reserved
```

Output Field

```
Return code           - Error status
Auxiliary return code - Reserved
Parameter 1           - Reserved
Parameter 2           - Reserved
Parameter 3           - Reserved
Parameter 4           - Reserved
Parameter 5           - Reserved
```

Get System Resume Timer

Input Field

```
Major function number - X'34'
Minor function number - X'02'
Parameter 1           - Reserved
Parameter 2           - Reserved
Parameter 3           - Reserved
Parameter 4           - Reserved
Parameter 5           - Reserved
```

Output Field

```
Return code           - Error status
Auxiliary return code - Reserved
Parameter 1           - Reserved
Parameter 2           - Reserved
Parameter 3           - Reserved
Parameter 4           - TOD of resume timer (BCD format)
                        Bits 7-0: Seconds    (0-59)
                        Bits 15-8: Minutes   (0-59)
                        Bits 23-16: Hours    (0-23)
                        Bits 31-24: Reserved
Parameter 5           - Date of resume timer (BCD format)
                        Bits  7-0: Day       (1-31)
                        Bits 15-8: Month     (1-12)
                        Bits 23-16: Year     (0-99)
                        Bits 30-24: Reserved
                        Bit 31: Resume date validation
                              = 0: Valid (specified day)
                              = 1: Invalid (every day)
```

Set System Resume Timer

Input Field

```
Major function number - X'34'
Minor function number - X'03'
Parameter 1           - Reserved
Parameter 2           - Reserved
Parameter 3           - Reserved
Parameter 4           - TOD of resume timer (BCD format)
                          Bits 7-0: Seconds    (0-59)
                          Bits 15-8: Minutes   (0-59)
                          Bits 23-16: Hours    (0-23)
                          Bits 31-24: Reserved
Parameter 5           - Date of resume timer (BCD format)
                          Bits 7-0: Day        (1-31)
                          Bits 15-8: Month     (1-12)
                          Bits 23-16: Year     (0-99)
                          Bits 30-24: Reserved
                          Bit 31: Resume date validation
                                = 0: Valid (specified day)
                                = 1: Invalid (every day)
```

Output Field

```
Return code           - Error status
Auxiliary return code - Reserved
Parameter 1           - Reserved
Parameter 2           - Reserved
Parameter 3           - Reserved
Parameter 4           - Reserved
Parameter 5           - Reserved
```

Request System Standby

Input Field

```
Major function number - X'70'
Minor function number - X'00'
Parameter 1           - Reserved
Parameter 2           - Reserved
Parameter 3           - Reserved
Parameter 4           - Reserved
Parameter 5           - Reserved
```

Output Field

```
Return code           - Error status
Auxiliary return code - Reserved
Parameter 1           - Reserved
Parameter 2           - Reserved
Parameter 3           - Reserved
Parameter 4           - Reserved
Parameter 5           - Reserved
```

Request System Suspend

Input Field

```
Major function number - X'70'
Minor function number - X'01'
Parameter 1           - Reserved
Parameter 2           - Reserved
Parameter 3           - Reserved
Parameter 4           - Reserved
Parameter 5           - Reserved
```

Output Field

```
Return code           - Error status
Auxiliary return code - Reserved
Parameter 1           - Reserved
Parameter 2           - Reserved
Parameter 3           - Reserved
Parameter 4           - Reserved
Parameter 5           - Reserved
```

Request System Hibernation

Input Field

```
Major function number - X'70'
Minor function number - X'02'
Parameter 1           - Reserved
Parameter 2           - Reserved
Parameter 3           - Reserved
Parameter 4           - Reserved
Parameter 5           - Reserved
```

Output Field

```
Return code           - Error status
Auxiliary return code - Reserved
Parameter 1           - Reserved
Parameter 2           - Reserved
Parameter 3           - Reserved
Parameter 4           - Reserved
Parameter 5           - Reserved
```

Request System Off

Input Field

```
Major function number - X'70'
Minor function number - X'03'
Parameter 1           - Reserved
Parameter 2           - Reserved
Parameter 3           - Reserved
Parameter 4           - Reserved
Parameter 5           - Reserved
```

Output Field

```
Return code           - Error status
Auxiliary return code - Reserved
Parameter 1           - Reserved
Parameter 2           - Reserved
Parameter 3           - Reserved
Parameter 4           - Reserved
Parameter 5           - Reserved
```

Samples

Data Structure

Assembler Language

```
;
;  Smapi BIOS Header
;
SMB_HEADER                STRUC
 @SMBHDR_SIG              DB      4 dup (?)
   ; +00 - Signature
 @SMBHDR_VER              DB      ?
   ; +04 - Major version
 @SMBHDR_VER_VER          DB      ?
   ; +05 - Minor version
 @SMBHDR_LEN              DB      ?
   ; +06 - Length
 @SMBHDR_CHKSUM           DB      ?
   ; +07 - Checksum
 @SMBHDR_INFO             DW      ?
   ; +08 - Information Word
 @SMBHDR_RSV1             DW      ?
   ; +0A - Reserve 1
 @SMBHDR_R_OFFSET         DW      ?
   ; +0C - Real mode Offset
 @SMBHDR_R_SEGMENT        DW      ?
   ; +0E - Real mode Segment
 @SMBHDR_RSV2             DW      ?
   ; +10 - Reserve 2
 @SMBHDR_P16_OFFSET       DW      ?
   ; +12 - 16-bit protected mode offset
 @SMBHDR_P16_BASE         DD      ?
   ; +14 - 16-bit protected mode base address
 @SMBHDR_P32_OFFSET       DD      ?
   ; +18 - 32-bit protected mode offset
 @SMBHDR_P32_BASE         DD      ?
   ; +1C - 32-bit protected mode base address
SMB_HEADER                ENDS
```

Parameters

```
;
;Input Parameter
;
SMB_INPARM                STRUC
 @SMBIN_FUNC              DB      ?
 @SMBIN_SUB_FUNC          DB      ?
 @SMBIN_PARM_1            DW      ?
 @SMBIN_PARM_2            DW      ?
 @SMBIN_PARM_3            DW      ?
 @SMBIN_PARM_4            DD      ?
 @SMBIN_PARM_5            DD      ?
SMB_INPARM                ENDS


;
;Output Parameter
;
SMB_OUTPARM               STRUC
 @SMBOUT_RC               DB      ?
 @SMBOUT_SUB_RC           DB      ?
 @SMBOUT_PARM_1           DW      ?
 @SMBOUT_PARM_2           DW      ?
 @SMBOUT_PARM_3           DW      ?
 @SMBOUT_PARM_4           DD      ?
 @SMBOUT_PARM_5           DD      ?
SMB_OUTPARM               ENDS
```

C Language

```
//
//  SMAPI BIOS Header
//
typedef struct {
   BYTE     SMBHDR_SIG[4]    ; // Signature
   BYTE     SMBHDR_VER          ; // Major Version
   BYTE     SMBHDR_VER_VER      ; // Minor Version
   BYTE     SMBHDR_LEN          ; // Length
   BYTE     SMBHDR_CHKSUM       ; // Checksum
   WORD     SMBHDR_INFO         ; // Information word
   WORD     SMBHDR_RSV1         ; // Reserve 1
   WORD     SMBHDR_R_OFFSET     ; // Real mode offset
   WORD     SMBHDR_R_SEGMENT    ; // Real mode segment
   WORD     SMBHDR_RSV2         ; // Reserve 2
   WORD     SMBHDR_P16_OFFSET
     ; // 16-bit Protect mode offset
   DWORD    SMBHDR_P16_BASE
     ; // 16-bit Protect mode base address
   DWORD    SMBHDR_P32_OFFSET
     ; // 32-bit Protect mode offset
   DWORD    SMBHDR_P32_BASE
     ; // 32-bit Protect mode base address
} SMB_HEADER, *PSMB_HEADER ;
```

Parameters

```
//
// Input Parameter
//
typedef struct {
   BYTE     SMBIN_FUNC       ;
   BYTE     SMBIN_SUB_FUNC   ;
   WORD     SMBIN_PARM_1     ;
   WORD     SMBIN_PARM_2     ;
   WORD     SMBIN_PARM_3     ;
   DWORD    SMBIN_PARM_4     ;
   DWORD    SMBIN_PARM_5     ;
} INPARM, *PINPARM ;

//
// Output Parameter
//
typedef struct {
   BYTE     SMBOUT_RC        ;
   BYTE     SMBOUT_SUB_RC    ;
   WORD     SMBOUT_PARM_1    ;
   WORD     SMBOUT_PARM_2    ;
   WORD     SMBOUT_PARM_3    ;
   DWORD    SMBOUT_PARM_4    ;
   DWORD    SMBOUT_PARM_5    ;
} OUTPARM, *POUTPARM ;
```

Function Declaration

C Language

```
//
// Smapi BIOS function
//
typedef WORD (far * SMB)(PINPARM, POUTPARM) ;
```

Installation Check

Assembler Language: **Real Mode**

```
;
;  FindSmapi
; -----------
;
;   On Entry : None
;   On Exit    : CF = 0 .. Find out
;                   DX - Segment
;                   BX - Pointer to Header
;
;                CF = 1 .. No Smapi BIOS
;

FindSmapi         Proc    Near

        push      eax
        push      cx
        push      si
        push      ds

        mov       ax, BIOS_SEG      ; F000 Segment
        mov       ds, ax
        mov       bx, 0             ; Start Point
        mov       cx, SMB_CAND_CNT  ; Total Check Count
        mov       eax, 'BMS$'   ; Target Strings

  @@:
        cmp       eax, dword ptr ds:[bx].@SMBHDR_SIG
        je        short @f
        add       bx, 10h      ; Next Paragraph
        loop      @b
        stc
        jmp       short FindSmapiFin
```

```
   @@:  ;         Find Smapi Head
        mov       dx, BIOS_SEG

        ;         Calculate Checksum.. next.
        pushf                ; Save Direction flag
        cld                  ; Clear it
        mov       si, bx
        xor       ax, ax
        movzx     cx,byte ptr ds:[bx].@SMBHDR_LEN
   @@:
        lodsb
        add       ah, al
        loop      @b

        popf                 ; Restore Direction flags
        cmp       ah, 1      ; Checksum is OK?
        cmc

FindSmapiFin:
        pop       ds
        pop       si
        pop       cx
        pop       eax
        ret

FindSmapi         Endp
```

C Language

```
typedef struct {
   BYTE     SMBHDR_SIG[4]         ; // Signature
   BYTE     SMBHDR_VER              ; // Major Version
   BYTE     SMBHDR_VER_VER          ; // Minor Version
   BYTE     SMBHDR_LEN              ; // Length
   BYTE     SMBHDR_CHKSUM           ; // Checksum
   WORD     SMBHDR_INFO             ; // Information Word
   WORD     SMBHDR_RSV1             ; // Reserve 1
   WORD     SMBHDR_R_OFFSET         ; // Real Mode Offset
   WORD     SMBHDR_R_SEGMENT        ; // Real Mode Segment
} SMB_HEADER_REAL, far * PFSMB_HEADER_REAL ;
```

```
BOOLEAN GetSmapiEntry(PSMB pFunc)
{
   PFSMB_HEADER_REAL        MyPtr = 0xF0000000 ;
   WORD                     cnt = 0 ;
   BYTE                     cksum = 0 ;

   //
   // 1) Search for signature first
   //
   while((cnt++ < 0x1000) &&
         !(((MyPtr->SMBHDR_SIG)[0] == '$') &&
            ((MyPtr->SMBHDR_SIG)[1] == 'S') &&
            ((MyPtr->SMBHDR_SIG)[2] == 'M') &&
            ((MyPtr->SMBHDR_SIG)[3] == 'B') )) {
      MyPtr++ ;
   }


   //
   //  2) Find the Signature?
   //
   if (cnt >= 0x1000) {
      // We cannot find it.
      return FALSE ;
      } else {
      //
      //  3) Calculate Checksum
      //
      for (cnt = 0 ; cnt < MyPtr->SMBHDR_LEN ; cnt++)
         cksum += (BYTE)((MyPtr->SMBHDR_SIG)[cnt]) ;

      if (cksum) {
         // Bad Checksum
         return FALSE ;
      } else {
         // Build Return Address
         (*pFunc) = ( (DWORD)(MyPtr->SMBHDR_R_OFFSET) +
                ((DWORD)(MyPtr->SMBHDR_R_SEGMENT)) << 16) ) ;
         return TRUE ;
      }
   }
}
```

BIOS Call

Assembler Language: **16-Bit Protected Mode**

```
;
; Build Input Parameter Field
;

mov       al, SMB_GET_SYSID
mov       [bx].@Func, al

mov       ax, offset OutputParm
push      ax
mov       ax, offset InputParm
push      ax
call      _SmapiBios
add       sp, 4

;
; Get information from Output Parm
;
or        ax, ax
jnz       Error

mov       bx, offset OutputParm
mov       al, [bx].@Parm1
```

32-Bit Protected Mode

```
;
; Build Input Parameter Field
;
mov       ebx, offset InputParm
mov       al, SMB_GET_SYSID
mov       [ebx].@Func, al

mov       eax, offset OutputParm
push      eax
mov       eax, offset InputParm
push      eax
call      _SmapiBios
add       sp, 8

;
; Get information from Output Parm
;
or        ax, ax
jnz       Error

mov       ebx, offset OutputParm
mov       ax, [ebx].@Parm1
```

C Language

```
WORD  GetSystemID()
{
   SMB         SmapiEntry ;
   INPARM      MyInput ;
   OUTPARM     MyOutput ;
   WORD        Rc = -1 ;

   if (GetSmapiEntry(&SmapiEntry)) {

      MyInput.SMBIN_FUNC        = 0 ;
      MyInput.SMBIN_SUB_FUNC    = 0 ;

      if (SmapiEntry(&MyInput, &MyOutput)) {
        // No System ID is available
      } else {
        Rc = MyOutput.SMBOUT_PARM_1 ;
      }

   } else {
      // No Smapi BIOS interface.
      // Try to use CBIOS INT 15.
   }
   return Rc ;
}
```

Appendix B. Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of the intellectual property rights of IBM may be used instead of the IBM product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, are the responsibility of the user.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk NY 10504-1784
U.S.A.

Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

AT	PS/2
IBM	ThinkPad
Personal Computer AT	TrackPoint
Personal System/2	

The following terms, denoted by a double asterisk (**) in this publication, are trademarks of other companies as follows:

Pentium®, MMX™, ProShare®, LANDesk®, and ActionMedia® are trademarks or registered trademarks of Intel Corporation in the U.S. and other countries.

IrDA	Infrared Data Association
PCMCIA	Personal Computer Memory Card International Association
Sound Blaster	Creative Labs, Inc.

Other company, product, and service names, which may be denoted by a double asterisk (**), may be trademarks or service marks of others.

Index

E

F

G

H

I

K

L

M

N

O

P

R

S

T

V

W

IBM

Part Number: xxxxxxx

Printed in the United States of America