

USING KEYBOARD OR MOUSE TO WAKE UP THE SYSTEM FROM SUSPEND OR SOFT OFF WITH SMSC I/O DEVICES

By Lawrence Chao, Fidel Wang and Mark Atchison

Most recent SMSC I/O chips have built in the PME (Power Management Event) functions. One example of these functions is that a PME event can be generated by pressing any key of the keyboard or moving the mouse. The PME event can wake up the system from soft off (from ACPI power state S4 to S0) or resume a system which is in standby or suspend mode (from ACPI power state S1, S2, or S3 to S0). The software procedures and hardware considerations for using the PME feature are discussed in this document.

Software Procedures

When transitioning between S0, the Active ACPI state, and a Suspend (S1, S2 or S3) state or Soft Off (S4), the PME registers in the Super I/O part enable wake sources and provide status information about the source of the wake event. These registers and their operation are described below.

I. PME Registers

Four registers within the Super I/O are associated with the PME function. They are the PME Control Register, PME Status Register, PME Wake Status Register and the PME Wake Enable Register. The PME Control Register controls the assertion of the nPME signal. If the PME_En bit is not set, the chip will not generate a PME regardless of whether the wake event has occurred and is enabled. The PME status register indicates whether a PME would have occurred if enabled in the PME Control register. The PME Wake Status register indicates which sources caused or would have caused (if enabled) a PME. Note that these bits will be set whether or not they are enabled by their corresponding bit in the PME Wake Enable register. They should be cleared by writing a 1 to any bit that is set before enabling them for wakeup in the PME Wake Enable register.

The PME Wake Enable register contains a bit to enable each wakeup source to generate a PME. Setting a bit enables the wake source to generate a PME, clearing disables it.

Following are the PME register definitions for the FDC37M70x. The registers for the other parts may differ but generally provide the same functionality.

A. PME Control Register

NAME	REG INDEX	DEFINITION
PME Control Default = 0x00 on V _{TR} POR	0xC5 (R/W)	Bit[0] PME_En = 0 nPME signal assertion is disabled (default) = 1 Enables FDC37M70x to assert nPME signal Bit[7:1] Reserved PME_En is not affected by Vcc POR, SOFT RESET or HARD RESET

B. PME Status Register

NAME	REG INDEX	DEFINITION
PME Status Default = 0x00 on POR V _{TR}	0xC6 (R/w Clear)	Bit[0] PME_Status = 0 (default) = 1 Set when FDC37M70x would normally assert the PCI nPME signal, independent of the state of the PME_En bit. Bit[7:1] Reserved PME_Status is not affected by Vcc POR, SOFT RESET or HARD RESET. Writing a "1" to PME_Status will clear it and cause the FDC37M70x to stop asserting nPME, in enabled. Writing a "0" to PME_Status has no effect.

C. PME Wake Status Register

NAME	REG INDEX	DEFINITION
PME Wake Status Default = 0x00 on V _{TR} POR	0xC7 (R/w Clear)	This register indicates the state of the individual PME wake sources, independent of the individual source enables or the PME_En bit. If the wake source has asserted a wake event, the associated PME Wake Status bit will be a "1". Bit[0] Reserved Bit[1] R12 Bit[2] R11 Bit[3] KBD Bit[4] MOUSE Bit[7:5] Reserved The PME Wake Status register is not affected by Vcc POR, SOFT RESET or HARD RESET. Writing a "1" to Bit[4:0] will clear it. Writing a "0" to any bit in PME Wake Status Register has no effect.

D. **PME Wake Enable Register**

NAME	REG INDEX	DEFINITION
PME Wake Enable Default = 0x00 on V _{TR} POR	0xC8 (R/W)	This register is used to enable individual FDC37M70x PME wake sources onto the nPME wake bus. When the PME Wake Enable register bit for a wake source is active ("1"), if the source asserts a wake event and the PME_En bit is "1", the source will assert the PCI nPME signal. When the PME Wake Enable register bit for a wake source is inactive ("0"), the PME Wake Status register will indicate the state of the wake source but will not assert the PCI nPME signal. Bit[0] Reserved Bit[1] RI2 Bit[2] RI1 Bit[3] KBD Bit[4] MOUSE Bit[7:5] Reserved The PME Wake Enable register is not affected by Vcc POR, SOFT RESET or HARD RESET. Writing a "1" to Bit[4:0] will clear it. Writing a "0" to any bit in PME Wake Enable register has no effect (See Note*)

***Note: Data Sheet Errata (Important)**

The Note In PME Wake Enable Register description in the original FDC37M70x data sheet was **INCORRECT**. The individual sources are enabled to cause a PME by writing a 1. Write a 0 to disable a source.

II. **PME Suspend Sequence, S0 to S1**

In order to transition from the S0 to S1 state with Wake events enabled to generate a PME that resumes the system back to S0, follow the procedure below:

1. Clear Wake Status Register bits
2. Clear PME Status Register, bit 0
3. Enable Wake events in PME Wake Enable Register
4. Enable PME output in PME Control Register

III. **PME Resume Sequence, S1 to S0**

When the PME signal has caused the system to transition from the S1 state back to S0, the following procedure is followed.

1. Disable Wake events in PME Wake Enable Register
2. Clear Wake Status Register bits
3. Clear PME Status Register, bit 0
4. Disable PME output in PME Control Register

IV. **Special Consideration for AMI and Phoenix 42i BIOS**

Keyboard BIOS code in the FDC37M702 and FDC37M707 pulses the keyboard and mouse clock pins low every 120 milliseconds. This will wake the system if PME is enabled for keyboard or mouse wakeup. Sending the 2-byte D1 command to the keyboard controller stops the keyboard controller from pulsing the CLK lines low until the second byte of the command is received. ***This can be accomplished in an ACPI compliant manner by first relocating the keyboard controller to 360/364 and then sending the D1 command to the keyboard controller at 364. Since ports 360 and 364 are not run-time registers, the ACPI methods are able to read and write them without violating the ACPI spec.***

During Suspend sequence:

1. Set the msb, bit-7, in Global Configuration Register 0x2E. This relocates the keyboard controller to 360/364.
2. Send 0xD1 command to Port 364. While the keyboard controller is waiting for the second byte of this 2-byte sequence, it will not pulse the CLK lines.
3. Clear the msb, bit-7 of Global Configuration Register 0x2E to restore the Keyboard Controller base I/O address to 60/64.

During Resume sequence:

1. Set the msb, bit-7, in Global Configuration Register 0x2e to relocate the keyboard controller to 360/364.
2. Send 0xFF command to Port 364. This cancels the D1 command.
3. Clear the msb, bit-7 in Global Configuration Register 0x2E. This relocates the keyboard controller to 60/64.
4. D1 command Stops KCLK and MCLK pulsing, FF command terminates D1 command.

Note that the BIOS must claim addresses 360 and 364 as motherboard resources so that there will be no conflicts when the Keyboard Controller base address is relocated. Also, no other code must write to address 60 between the time that the D1 command is written to 364 and the time that the FF command is written.

In the 37B78x parts, the bit in Global Configuration Register 0x2e that relocates the keyboard controller also relocates the RTC to addresses 370 and 371. Therefore the configuration registers must be located at an address other than 370/371 in order for this workaround to work properly. No code must run that accesses the RTC while it is relocated to 370/371 or unpredictable results may occur. In this part, the KDAT and MDAT lines can be connected to GPIO pins on the part that can generate a PME on the falling edge. The bit that is enabled for wakeup then becomes the bit associated with the chosen pin rather than the Keyboard or Mouse bit.

Windows 98 Considerations

_PTS (Prepare to Sleep) Method was not called by some release candidate versions of Win98. This is the method that should contain the procedure to prepare the system to wake via the PME event. This may be changed in future versions.

Suspend and Resume Sequences therefore could be inserted into _SST Method, the System indicator control method that indicates the system status.

When suspending via mouse click or Keyboard, extra keys and mouse packets will either wake the system immediately or wake Win98 before it has fully completed its Suspend sequence. Therefore, the Resume sequence should check whether the Suspend sequence had fully completed. If the system has not gone into suspend yet, read the byte from the Keyboard Controller's output buffer and repeat the Suspend sequence. This can also be done in an ACPI compliant manner by relocating the Keyboard controller to 360 / 364 as described above in the **Special Consideration for AMI and Phoenix 42i BIOS** section.

Hardware Considerations

Do not connect the FDC37M70x PME pin to the same pin on the chipset that the PCI PME pins are connected. If the FDC37M70x and PCI devices are connected to the same pin, Win98 will only check through the PCI space and does not check ISA devices. Connect the FDC37M70x PME pin to its own pin on the chipset.

An application issue was found such that in some designs with some SMSC I/O chips, If the keyboard or mouse is enabled as a PME event to wake up the system, a PME will be generated immediately when the system is being turned off, so that the system will turn on without the keyboard or mouse being touched.

The cause was identified and a hardware resolution, which requires an external gate, has been developed. It was investigated, then determined that a software resolution is not possible. Please see the following figures and description for details.

Affected SMSC I/O Chips:

FDC37M70x Rev.B or earlier
FDC37B72x Rev.A
FDC37B77x Rev.C or earlier
FDC37B78x Rev.C or earlier
FDC37B80x Rev.A

Cause of the Issue:

Many south bridges will activate the reset to PCI and ISA bus whenever it senses the power good signal from the ATX power supply indicating the main 5V/3V power is going away. In many existing designs, the ISA RESET_DRV is connected to the Reset pin of the I/O chips. This will reset the 8042 keyboard controller in I/O chip then the keyboard and mouse clock and data lines will be driven low by the keyboard controller. While the keyboard or mouse is enabled as a PME event to wake up the system, then the I/O chip will drive nPME low to wake up the system. The details of this behavior and reaction are described below in Figure 1.

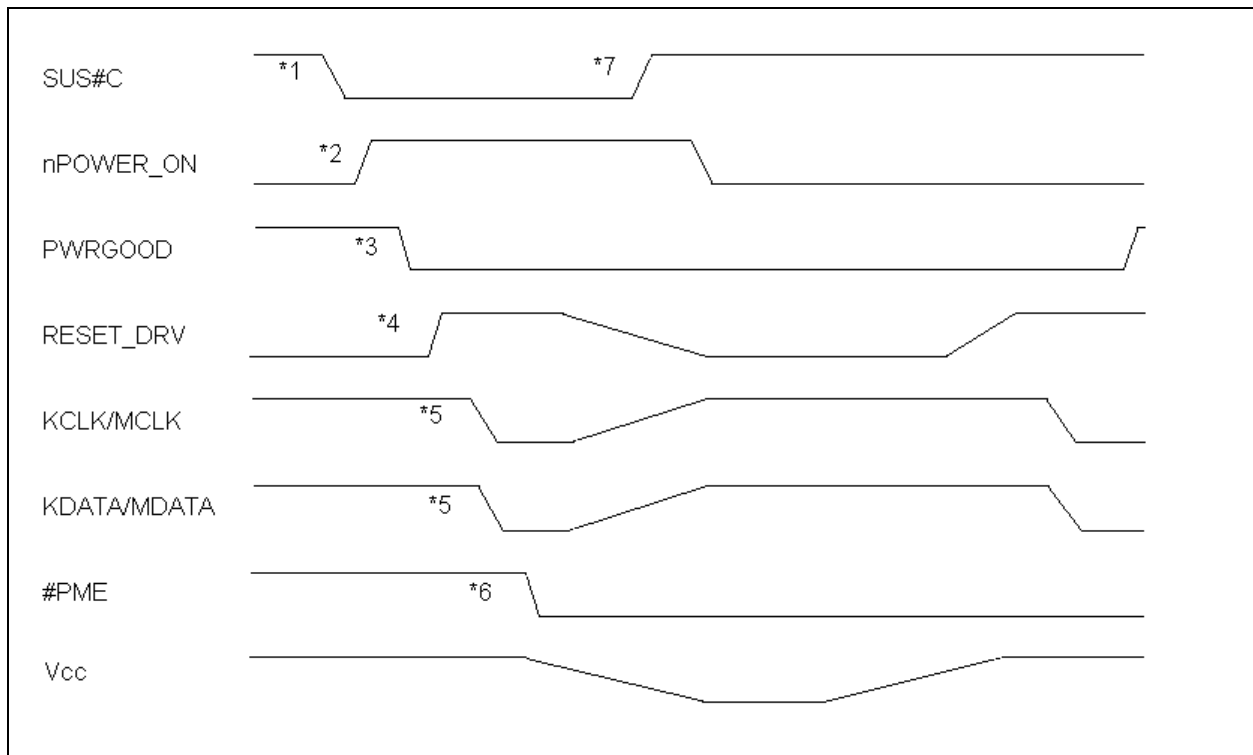


FIGURE 1 - ORIGINAL POWER OFF SEQUENCE

- *1: When the ACPI OS or BIOS wants to enter S4 state (Soft Off) by setting the SLP_TYP Register in PIIX4, the nSUSC pin of PIIX4 is driven low to turn off the main power.
- *2: nSUSC is inverted to generate nPower_On to ATX power supply.
- *3: PowerGood is driven low by ATX power supply to indicate the main 5V/3V power is going away.
- *4: PIIX4 drives RESET to PCI and ISA buses when it senses the Power Good is low. This is intended to reset every block in the system while the power is being removed so that all blocks in the system will not conduct undesired behavior and/or corrupt the data in mass storage units.
- *5: The SMSC I/O chip is reset and the 8042 keyboard controller in I/O chip drives the Clock and Data lines of both the keyboard and mouse low.
- *6: If the keyboard or mouse is enabled as a PME event to wake up the system, the high to low transition on keyboard/mouse data pin (for FDC37B80x) or keyboard/mouse clock pin (for listed I/O chips except FDC37B80x) will trigger a wake up event and the I/O chip will drive nPME low.
- *7: While the nPME pin of I/O chip is connected to GPI[1] or other resume pin of PIIX4, PIIX4 senses the wake up event and drives the nSUSC high to turn the system back on.

The Resolution:

The above issue can be fixed by adding an external AND gate which is powered by Vtr (5V Trickle Power, also known as 5Vsb, 5V Standby Power) between ISA RESET_DRV and the RESET_DRV input for I/O chips (pin 46 for the FDC37M70x, FDC37B77x, or FDC37B80x ; Pin 53 for the FDC37B72x or FDC37B78x). Connect the other input of the AND gate to nSUSC Intel PIIX4 south bridge (pin U18, or nSUSA or nSUSB, depending on which pin is used to control the power plane for I/O chip), or its 5V buffered equivalent. The circuit is very simple as illustrated in Figure 2.

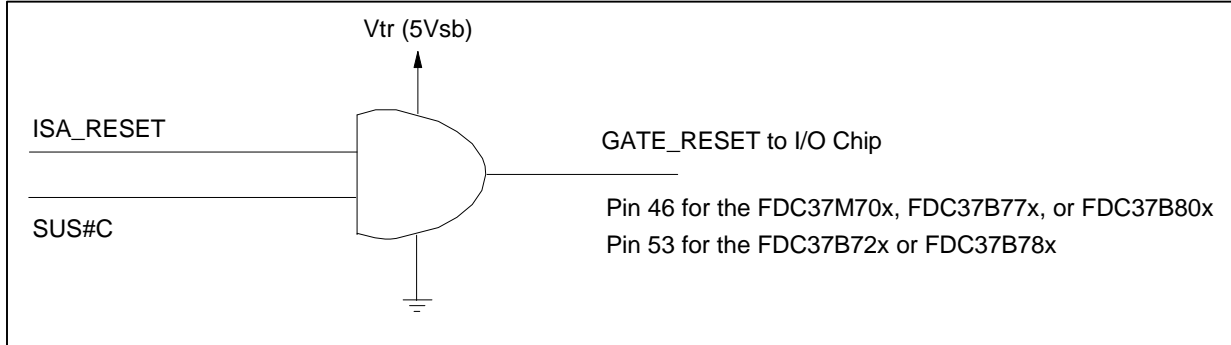


FIGURE 2 - THE HARDWARE RESOLUTION WITH AN EXTERNAL AND GATE

Because in power off sequence, nSUSC will go low before ISA RESET_DRV goes high, the RESET_DRV input to I/O chip will be blocked and not be active during the power off sequence. In the power on sequence, the nSUSC will go high before ISA RESET_DRV goes high, hence the RESET_DRV to I/O chip will not be blocked. Internal POR circuitry within the SMSC I/O detects the removal of Vcc and generates an internal POR that resets the logical blocks independently of RESET_DRV. This eliminates the possibility of undesired behavior when Vcc is removed. The power off and power on sequence with this resolution will be as described in Figure 3.

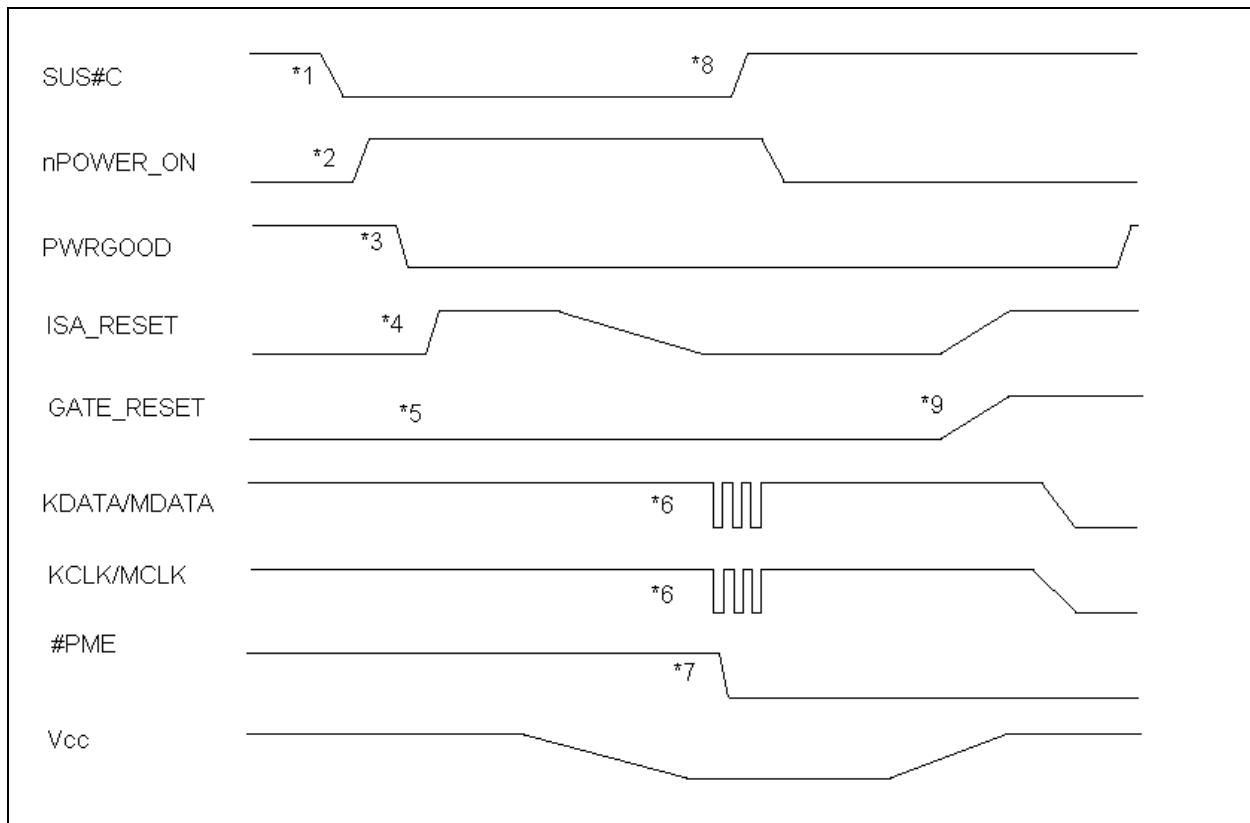


FIGURE 3 - POWER OFF AND POWER ON SEQUENCE WITH THE HARDWARE RESOLUTION

- *1~4: The normal power off procedure which is the same as the one described earlier in Figure 1.
- *5: The I/O chip is not reseted because the ISA RESET is blocked while the nSUSC is low. The data and clock signals for keyboard and mouse stays high when the Vcc is turned off without a RESET to I/O chip. The I/O chips have an internal Vcc power good which will reset FDC and other logical devices in I/O chip with or without an external RESET_DRV so that it will not caused any problem with FDD or other peripherals when the Vcc is turned off.
- * 6: A key is pressed or the mouse is moved by the user. Pulses on KDATA/KCLK are generated by the keyboard or pulses on MDATA/MCLK are generated by the mouse.
- *7~8: Same as description for *6~7 for Figure 1.
- *9: The ISA Reset generated in power on sequence can reach I/O chip as earlier because the nSUSC is high so that the Reset is not blocked.

Summary:

By adding a simple AND gate in hardware and following the software procedures listed above, it is possible to use the keyboard or mouse to resume the system from suspend or wake up the system from soft off with SMSC I/O devices. These Hardware and Software work-arounds are compliant with ACPI and WinNT.



80 Arkay Drive
Hauppauge, NY 11788
(631) 435-6000
FAX (631) 273-3123

Copyright © SMSC 2004. All rights reserved.

Circuit diagrams and other information relating to SMSC products are included as a means of illustrating typical applications. Consequently, complete information sufficient for construction purposes is not necessarily given. Although the information has been checked and is believed to be accurate, no responsibility is assumed for inaccuracies. SMSC reserves the right to make changes to specifications and product descriptions at any time without notice. Contact your local SMSC sales office to obtain the latest specifications before placing your product order. The provision of this information does not convey to the purchaser of the described semiconductor devices any licenses under any patent rights or other intellectual property rights of SMSC or others. All sales are expressly conditional on your agreement to the terms and conditions of the most recently dated version of SMSC's standard Terms of Sale Agreement dated before the date of your order (the "Terms of Sale Agreement"). The product may contain design defects or errors known as anomalies which may cause the product's functions to deviate from published specifications. Anomaly sheets are available upon request. SMSC products are not designed, intended, authorized or warranted for use in any life support or other application where product failure could cause or contribute to personal injury or severe property damage. Any and all such uses without prior written approval of an Officer of SMSC and further testing and/or modification will be fully at the risk of the customer. Copies of this document or other SMSC literature, as well as the Terms of Sale Agreement, may be obtained by visiting SMSC's website at <http://www.smcs.com>. SMSC is a registered trademark of Standard Microsystems Corporation ("SMSC"). Product names and company names are the trademarks of their respective holders.

SMSC DISCLAIMS AND EXCLUDES ANY AND ALL WARRANTIES, INCLUDING WITHOUT LIMITATION ANY AND ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE, AND AGAINST INFRINGEMENT AND THE LIKE, AND ANY AND ALL WARRANTIES ARISING FROM ANY COURSE OF DEALING OR USAGE OF TRADE.

IN NO EVENT SHALL SMSC BE LIABLE FOR ANY DIRECT, INCIDENTAL, INDIRECT, SPECIAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES; OR FOR LOST DATA, PROFITS, SAVINGS OR REVENUES OF ANY KIND; REGARDLESS OF THE FORM OF ACTION, WHETHER BASED ON CONTRACT; TORT; NEGLIGENCE OF SMSC OR OTHERS; STRICT LIABILITY; BREACH OF WARRANTY; OR OTHERWISE; WHETHER OR NOT ANY REMEDY OF BUYER IS HELD TO HAVE FAILED OF ITS ESSENTIAL PURPOSE, AND WHETHER OR NOT SMSC HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.