Waking the PC87363/4/5/6 SuperI/O from ACPI Sleep States 1 and 2

National Semiconductor
Application Note AN-1131
Limor Levy-Kendler
May 1999

The PC87363/4/5/6 are new members of National Semiconductor's LPC SuperI/O family of integrated PC peripherals. They include the System Wake-Up Control (SWC) logical device, which enables wake-up support for a wide range of wake-up events and enhances the power supply control of the system. The SWC enables the PC to power-up under various conditions (e.g., a keyboard event), and to power-down in an orderly, controlled manner. This functionality is required for meeting green environment standards for minimum power consumption and maximum operating efficiency. The SWC module in the PC87363/4/5/6 conforms to the "Advanced Configuration and Power Interface (ACPI) Specification" Version 1.0a, dated July 1, 1998.

This document describes the programming of the SWC Keyboard and Mouse wake-up events and the configuration required to wake up the system from ACPI sleep states S1 and S2. In addition, it includes configuration examples. It assumes familiarity with the PC87363/4/5/6 datasheets, specifically with the description of the SWC logical device.

ACPI SYSTEM SLEEP STATES

System sleep states are categorized into six groups according to the ACPI specification, as follows:

- Work State (S0) - The CPU is fully up and running, devices are powered up and down as needed.
- Sleep State 1 (S1) - The CPU is stopped, the RAM is refreshed, no system content is lost and all system hardware maintains context.
- Sleep State 2 (S2) - The CPU and system cache context is lost, the RAM is refreshed and the system is running in a low power mode similar to S1.
- Sleep State 3 (S3) - The CPU cache and chipset context is lost, the RAM is in slow refresh and the power supply is in a generally reduced power mode. Hardware maintains system memory and restores some CPU and L2 configuration context.
- Sleep State 4 (S4) - The hardware power is reduced to minimum. Platform context is maintained.
- Soft Off (S5) - The hardware is completely off, the operating system has shut down, nothing has been saved. A complete boot is required when awakened.

In sleep states S0, S1 and S2, the main V_{DD} voltage powers the system. In sleep states S3, S4 and S5, the main V_{DD} voltage is not present. In the latter cases, only functions powered by V_{SB} (the standby voltage; i.e., the trickle current from the external AC power supply) are active.

GENERAL WAKE-UP EVENT DESCRIPTION

Each wake-up event has a status bit, enable bits and routing bits. The status bit indicates the event was detected by the SWC. The enable bits determine the conditions for which the event may be detected. The routing bits determine which output reflects the event detection.

The status bit is defined in the WK_STSn register. The enable bits (named also extension-enable bits) are defined in the WK_X1ENn, WK_X2ENn and WK_X3ENn registers. The routing bits (named also routing-enable bits) are defined in the WK_ENn, WK_SMIENn and WK_IRQENn registers. The "n" suffix in these registers stands for a number, either 0 or 1.

When an event occurs, the enable bits are checked. If at least one enable bit is set (i.e., the event detection conditions are set), the status bit is set. In addition, the event is routed to one of the SWC outputs (i.e., to $\overline{SMI}$, IRQ, $\overline{PWUREQ}$ or $\overline{PWBTOUT}$) if the corresponding routing bit is set. An event may be routed to more than one output if more than one routing bit is set.

PS/2 KEYBOARD EVENT CONFIGURATION

The PS/2 keyboard wake-up detection configuration is determined by the PS2CTL register (located at bank 0, offset 13h) bits 3-0. The keyboard wake-up detection mechanism can be programmed to detect events according to three modes, as specified in Table 1. Eight registers, PS2KEY0 - PS2KEY7 (located at bank 0, offsets 18h-1Fh) store the key data which is compared to the incoming sequence.

Table 1. Keyboard Event Configuration

Event	Configuration
Any keystroke	PS2CTL[3:0] = 0001b PS2KEY0 = 00h PS2KEY1 = 00h
Password mode: A specific programmable sequence of up to 8 alphanumeric keystrokes	PS2CTL[3:0] = <number of keystrokes in sequence + 7> PS2KEYn = <scan code data n>
Special Key Sequence mode: Any programmable sequence of up to 8 data bytes received from the keyboard	PS2CTL[3:0] = <number of keystrokes in sequence - 1> PS2KEYn = <data byte n in sequence>

Password mode can detect a minimum of 1 key and a maximum of 8 keys, while Special Key Sequence mode can detect a minimum of 2 keys and a maximum of 8 keys.

In Password mode, each keystroke in the programming sequence is composed of at least three byte transmissions from the keyboard (including the Make and Break code bytes). On the other hand, in Special Key Sequence mode, each programming sequence is composed of exactly the number of keystrokes in the sequence as configured in the PS2CTL register. In other words, the Make and Break code bytes should be programmed in the PS2CTL register for the Special Key Sequence mode, and the programming sequence should be composed of at least two byte transmissions.

If the sequence monitored on the KBDAT pin (while the KBCLK is toggled) matches a pre-programmed sequence (stored in PS2KEYn registers), a Keyboard event is detected. A Keyboard event can be used as a wake-up event if the event is enabled. Note that the event can be detected only one second after V_{SB} is applied to the chip.

MOUSE EVENT

The Mouse wake-up detection configuration is determined by the PS2CTL register (located at bank 0, offset 13h), bits 6-4. The Mouse wake-up detection mechanism can be programmed to detect either any mouse click or movement, a specific programmable click (left or right) or a double click. The Mouse Data Shift register (MDSR, located at bank 0, offset 17h) stores the mouse data shifted in from the mouse during transmission, if the mouse wake-up detection is enabled.

If the sequence monitored on the MDAT pin (while the MCLK is toggled) matches the configured wake-up event, a Mouse event is detected. A Mouse event can be used as a wake-up event if it is enabled.

KEYBOARD OR MOUSE WAKE-UP FROM S1 OR S2

If the system is in sleep state S1 or S2 and the keyboard controller is enabled, the first data byte is received by two logical devices: the System Wake-up Control (SWC) and the Keyboard Controller (KBC). The KBC then holds the CLK pin low (KBCLK for keyboard transmissions and MCLK for mouse transmissions) to prevent the keyboard from sending more data before the CPU has read the byte stored in the KBC buffer. However, since the CPU is in a sleep state, it cannot read the data until it wakes up. A keyboard wake-up event requires at least two or three bytes of data (depending on the configuration mode), and a mouse wake-up event also requires more than a single byte transmission. Therefore, neither a Keyboard nor a Mouse wake-up event can be generated. This causes a deadlock situation.

Several system bypass options can overcome the deadlock:

1. Route the Keyboard IRQ or the Mouse IRQ (using Module IRQ) to the $\overline{\text{PWUREQ}}$ output, as shown in Figure 1. This requires BIOS support to enable the event when leaving sleep state S0. The event must be disabled after the first Keyboard or Mouse IRQ event is received, otherwise an event is generated on each byte transmission. This solution is quite simple and does not require any special hardware connections.

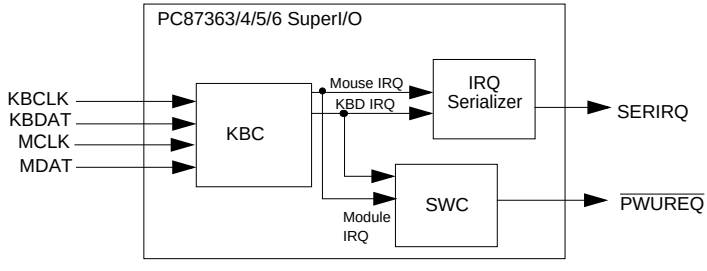

Figure 1. Route KBC IRQ to $\overline{\text{PWUREQ}}$

2. Route the Keyboard IRQ or the Mouse IRQ to the $\overline{\text{SMI}}$ output pin ($\overline{\text{SMI}}$ is routed to $\overline{\text{PME}}$), as shown in Figure 2. This requires BIOS support to enable the event when leaving sleep state S0. The event must be disabled after the first Keyboard or Mouse IRQ event is received, otherwise an event is generated on each byte transmission. This solution is also simple and does not require any special hardware connections, unless the board does not route $\overline{\text{SMI}}$ to one of the ACPI controller's GPI pins (also known as $\overline{\text{PME}}$).

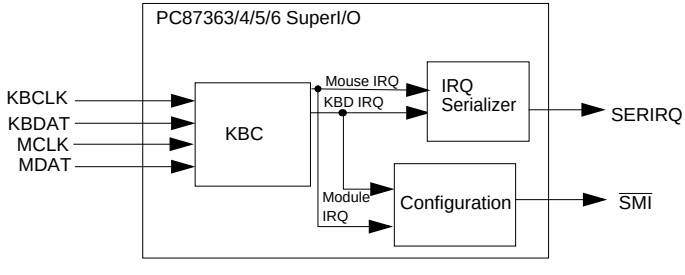

Figure 2. Route KBC IRQ to $\overline{\text{SMI}}$

3. Route the KBCLK or MCLK pin to one of the GPIE pins and configure the pin to detect a falling edge. The first falling edge of KBCLK/MCLK creates a wake-up event. This solution requires BIOS support to enable the event while leaving sleep state S0. The event must be disabled after the first wake-up event is received, otherwise an event will be generated on each bit (i.e., each toggle of the KBCLK/MCLK pin). The routing of the KBCLK/MCLK pin to one of the SuperI/O GPIE pins cannot be done by soft-

ware and therefore should be supported by the board hardware.

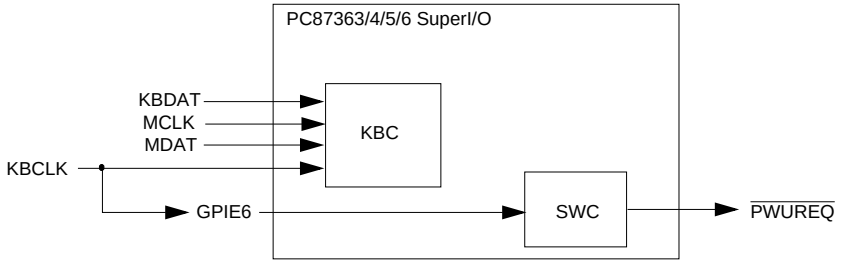

Figure 3. Route KBCLK to SuperI/O GPIE6

4. Disable the KBC logical device before entering sleep state S1. This forces the KBCLK and the MCLK pins to an inactive value, and enables the SWC logical device to receive another byte of data. This solution requires BIOS support to disable and enable the KBC when entering sleep state S1 from S0, and when exiting from S1 to S0. Data which is transmitted after wake-up but before the KBC is re-enabled is lost, and cannot be retrieved.

WAKE-UP OF OTHER EVENTS FROM S1 OR S2

Except for the Keyboard and Mouse, the following wake-up events wake up from sleep state S1 or S2 using the regular configuration:

- Modem ring - $\overline{\text{RI1}}$ and $\overline{\text{RI2}}$
- CEIR
- GPIO
- Software
- Module IRQ
- Telephone ring - $\overline{\text{RING}}$
- Standby general-purpose input event (GPIE)

For a detailed description of the operation and programming of these events, see “Configuring PC87363/4/5/6 SuperI/O System Wake-Up Control Events” application note.

CONFIGURATION EXAMPLES

All sample code sections are written in a pseudo-assembler language. This code is intended only to explain the principles of SWC programming to enable the Keyboard and Mouse wake-up events from sleep states S1 and S2.

The following three code sequences are examples of the BIOS code for SWC configuration on the PC87364. They assume that the base address of the SWC is defined as 880h, and both the SWC and KBC logical devices are active. `SIO_BASE` refers to the PC87364 device base address (which is the Index register address, and therefore `SIO_BASE+1` is the Data register address).

Please note that although not demonstrated in the examples below, it is recommended to use the read-modify-write method when writing to the registers.

Route the Keyboard IRQ (using Module IRQ) to the $\overline{\text{PWUREQ}}$ output

```
out SIO_BASE, 07h                ; Select Keyboard logical device
out SIO_BASE+1, 06h
out SIO_BASE, 70h                ; Select Keyboard interrupt number
out SIO_BASE+1, 11h              ; Select IRQ1
                                 ; Enable Keyboard wake-up on IRQ (set bit 4)
out 882h, 80h                    ; Enable Keyboard IRQ route to PWUREQ (WK_EN0[7])
```

Route the Keyboard IRQ (using Module IRQ) to the $\overline{SMI}$

```
out SIO_BASE, 07h                ; Select Keyboard logical device
out SIO_BASE+1, 06h
out SIO_BASE, 70h                ; Select Keyboard interrupt number
out SIO_BASE+1, 11h              ; Select IRQ1
                                 ; Enable Keyboard wake-up on IRQ (set bit 4)
out SIO_BASE, 23h                ; Select SMI function on pin 21 (SIOCFG3[2])
out SIO_BASE+1, 04h
out SIO_BASE, 28h                ; Enable Keyboard IRQ route to SMI (SIOCFG8[3])
out SIO_BASE+1, 08h
```

Route the Mouse IRQ (using Module IRQ) to the $\overline{SMI}$

```
out SIO_BASE, 07h                ; Select Mouse logical device
out SIO_BASE+1, 05h
out SIO_BASE, 70h                ; Select Mouse interrupt number
out SIO_BASE+1, 1Ch              ; Select IRQ12
                                 ; Enable Mouse wake-up on IRQ (set bit 4)
out SIO_BASE, 23h                ; Select SMI function on pin 21 (SIOCFG3[2])
out SIO_BASE+1, 04h
out SIO_BASE, 28h                ; Enable Mouse IRQ route to SMI (SIOCFG8[4])
out SIO_BASE+1, 10h
```

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

National Semiconductor Corporation
Americas
Tel: 1-800-272-9959
Fax: 1-800-737-7018
Email: support@nsc.com

National Semiconductor Europe
Fax: +49 (0) 1 80-530 85 86
Email: europe.support@nsc.com
Deutsch Tel: +49 (0) 1 80-530 85 85
English Tel: +49 (0) 1 80-532 78 32
Français Tel: +49 (0) 1 80-532 93 58
Italiano Tel: +49 (0) 1 80-534 16 80

National Semiconductor Asia Pacific Response Group
Tel: 65-2544466
Fax: 65-2504466
Email: sea.support@nsc.com

National Semiconductor Japan Ltd.
Tel: 81-3-5639-7560
Fax: 81-3-5639-7507

www.national.com