Using the Interrupt Serializer of the PC87360, PC87364 and PC87365

National Semiconductor
Application Note 1121
Ilan Margalit
May 1999

The PC87360, PC87364 and PC87365, members of National Semiconductor's 128-pin LPC SuperI/O™ family, incorporate an Interrupt Serializer. This application note provides guidelines for using the Interrupt Serializer and for writing the software required for operating it correctly. It assumes basic familiarity with the "Serialized IRQ Support for PCI Systems" specification.

Definitions

SERIRQ bus - a synchronous bus, consisting of a clock line, usually the LPC clock (LCLK), and a serial data line, SERIRQ. Used by various devices, e.g. the SuperI/O, to transfer interrupt request information to the system's SERIRQ host (interrupt controller), in a serial format.

IRQ Frame - a sequence of SERIRQ bus clock (LCLK) cycles, associated with a specific interrupt request number, and used by devices, to which the associated interrupt request number is assigned, to indicate the state of this interrupt request. For example, IRQ Frame 1 is used by a device to which IRQ number 1 is assigned, to indicate the state of its IRQ1 signal.

Interrupt request source - a device, or module, that may assert an interrupt request to be transmitted to the SERIRQ host by the SuperI/O.

Internal interrupt request source - any of the logical devices of the SuperI/O that has an interrupt request line (e.g. the FDC).

External interrupt request source - an external device that can assert an interrupt request signal, connected to a SuperI/O PIRQn pin.

Overview

The PC87360, PC87364 and the PC87365 interface the SERIRQ bus to transmit interrupt requests to the system's interrupt controller. Since interrupt requests coming from different sources are, by definition, parallel, they must be serialized before they can be transmitted over the SERIRQ bus. For this purpose, the PC87360, PC87364 and PC87365 incorporate an Interrupt Serializer.

This Interrupt Serializer supports both the Continuous and Quiet modes of the SERIRQ bus, and 15 IRQ frames (1 to 15). Although IRQ Frames higher than 15 are not supported, once Frame 15 is completed, the Interrupt Serializer follows the activity on the SERIRQ bus and waits for a Stop frame. Thus the Interrupt Serializer can operate with any SERIRQ host that supports 16 IRQ frames (0 to 15), or above.

The Interrupt Serializer can handle interrupt requests coming from two possible sources:

- Internal sources (via internal signals)
- External devices (via the PIRQn pins)

Although Legacy interrupt request assignment is usually preferred, the internal interrupt request sources may be assigned any IRQ number between 1 and 15.

Unlike internal sources, each external source is assigned a fixed IRQ number according to the PIRQ number of the pin to which its interrupt request is connected. For example, an external device, whose interrupt request is connected to PIRQ7, is always assigned IRQ number 7. Since the PC87360, PC87364 and PC87365 only have the functions PIRQ3 - 7, 9 - 12, 14 - 15, an external interrupt request source can only be assigned IRQ numbers 3 - 7, 9 - 12 or 14 - 15.

The Interrupt Serializer can combine these two types of sources to determine the value to be transmitted over the SERIRQ bus during each IRQ Frame.

Enabling the Interrupt Serializer

Bit 6 of the SIOCF1 register controls the Interrupt Serializer.

By default, the value of this bit is 0, the PIRQn functions are deselected from their respective pins, and the Interrupt Serializer only handles interrupt requests originating from internal sources.

When bit 6 is set to 1, the PIRQn functions are selected on their respective pins, thus making them valid interrupt request sources.

Selecting the Interrupt Request Source

Once the PIRQn functions are enabled as valid interrupt request sources, two interrupt request sources (one internal and one external) can be assigned the same IRQ number. As already mentioned, this is possible only for IRQ numbers 3 - 7, 9 - 12 and 14 - 15.

To avoid conflict, the Interrupt Serializer selects one of the following as the actual interrupt request source for each IRQ Frame (associated with an IRQ number):

- The internal source
- The external source
- A combination of the two sources

The selection is made according to the values of the Interrupt Number and Wake-Up on IRQ Enable (Index 70h) and Interrupt Request Type Select (Index 71h) configuration registers of all internal logical devices, and therefore it is software controlled.

Selecting the Internal Interrupt Source

To configure the Interrupt Serializer to select the internal source, or logical device, for a certain IRQ Frame, this logical device should be assigned the IRQ Frame. Assigning an IRQ Frame to a logical device is achieved by writing the number of the IRQ Frame to bits 3 - 0 of the Interrupt Number and Wake-Up on IRQ Enable register of this logical device. This register is located at index 70h, within the configuration register space of this logical device. In addition, this logical device must be programmed to have a non-inverted interrupt request by writing 1 to bit 1 of its Interrupt Type Select register. This register is located at index 71h, within the configuration register space of this logical device.

Selecting the Interrupt Request Source (Continued)

Selecting the External Interrupt Source

To configure the Interrupt Serializer to select the external source, the IRQ Frame must be released from any logical device to which it has been assigned to. Releasing an IRQ Frame is achieved by changing bits 3 - 0 of the Interrupt Number Select register of the logical device to which the IRQ Frame is assigned. To avoid further conflict, it is recommended to set these bits to 0, which indicates no IRQ assignment, and then, if another IRQ Frame has to be assigned to the logical device, change their value to assign the desired IRQ Frame to the logical device.

Another option for releasing an IRQ Frame is to disable the logical device to which it has been assigned. This is achieved by writing 0 to bit 0 of the Activate register of the logical device. The Activate register of a logical device is located at index 30h, within the configuration register space of the logical device. Since disabling a logical device might be undesired, it is recommended to use the method mentioned in the previous paragraph.

Selecting Interrupt Source Combination

To configure the Interrupt Serializer to use the same IRQ Frame for both sources, thus sharing it between them, two conditions must be met. First, the logical device should be assigned the IRQ Frame. The process of assigning an IRQ Frame to a logical device is described above. In addition, this logical device must be programmed to have an inverted interrupt request by writing 0 to bit 1 of its Interrupt Type Select register.

The value that is driven during a shared IRQ Frame is a logic AND between the values of the internal and the external source. Consequently, to enable IRQ Frame sharing to operate correctly, the external interrupt request signal must be active low. This has to be guaranteed by the system designer.

The following table summarizes the IRQ source selection options. This selection is performed per IRQ number.

Serializing Enabled (SIOCF1 bit 6)	IRQ Internally Assigned (Note 1)	IRQ Internally Inverted (Note 2)	Selected IRQ Source
0	X	X	Internal
1	no	X	External
1	yes	no	Internal
1	yes	yes	Shared

Note 1: "IRQ Internally Assigned" means that one of the internal logical devices has been assigned the IRQ in question.

Note 2: "IRQ Internally Inverted" means that the logical device, to which the IRQ in question is assigned, has been programmed for IRQ inversion.

Examples

The following examples of code illustrate how the operations mentioned above should be performed. These examples are designed to illustrate how operations should be done. They are given in a generic language; the actual code may look different.

Enabling PIRQ Serialization

The following code enables PIRQ serialization by writing 1 to bit 6 of the SIOCF1 register. Note that a read-modify-write operation is employed to modify the value of bit 6 of the SIOCF1 register. This avoids any unwanted alteration of the other bits of this register.

```
// Enable Parallel to Serial
#define SIOCF1 0x21

ENABLE_SIO_P2S:
  out  SIO_BASE, SIOCF1
  in   al, SIO_BASE+1
  or   al, 0x40
  out  SIO_BASE+1, al
```

Assigning an IRQ Number to a Logical Device

The following code assigns IRQ number 9 to Logical Device 4 by writing the value 9h to bits 3 - 0 of the Interrupt Number and Wake-up on IRQ Enable register of logical device 4. Note that a read-modify-write operation is employed to modify the value of bits 3 - 0 of the register at index 70h. This avoids an unwanted alteration of bits 7 - 4 of this register. For further details on the functionality of bit 4 of this register, refer to the appropriate SuperI/O datasheet.

```
// Assign IRQ Number 9 to LD 4

#define  SWC_LDN      0x04
#define  SWC_IRQ_NUM  0x09
#define  LDN_INX      0x07
#define  IRQ_NUM_INX  0x70

ASSIGN_SWC_IRQ:
  out  SIO_BASE,   LDN_INX
  out  SIO_BASE+1, SWC_LDN
  out  SIO_BASE,   IRQ_NUM_INX
  in   al, SIO_BASE+1
  and  al, F0h
  or   al, SWC_IRQ_NUM
  out  SIO_BASE+1, al
```

Releasing an IRQ from All Internal Sources

The following code releases IRQ number 9 from any assigned internal logical device by going through all logical devices, one by one, and writing the value of 0h to bits 3 - 0 of the Interrupt Number and Wake-Up on IRQ Enable register of each logical device whose assigned IRQ number is 9h. If the BIOS maintains a table of IRQ assignment, this task is much simpler.

```
// Release IRQ9
#define FIRST_LDN       0x00
#define LAST_LDN        0x0A
#define IRQ_9           0x09
#define LDN_INX         0x07
#define IRQ_NUM_INX     0x70

RELEASE_IRQ9:
  mov     cl, FIRST_LDN

RELEASE_IRQ9_LDN:
  out     SIO_BASE, LDN_INX
  out     SIO_BASE+1, LDN
  out     SIO_BASE, IRQ_NUM_INX
```

Examples (Continued)

```
  in      al, SIO_BASE+1
  mov     bl, al
  and     bl, 0x0F
  cmp     bl, 0x09
  bne     RELEASE_IRQ9_NEXT_LDN
  and     al, 0xF0
  out     SIO_BASE+1, al

RELEASE_IRQ9_NEXT_LDN:
  add     cl, 0x01
  cmp     cl, LAST_LDN
  ble     RELEASE_IRQ9_LDN
```

Inverting the IRQ of a Logical Device

The following code inverts the polarity of the IRQ line assigned to the SWC module (logical device number 4). For a module with inverted IRQ polarity, the value of the IRQ is logic 0, when the module's interrupt request is active. To invert the IRQ of a logical device, set bit 1 of the relevant local device's Interrupt Request Type Select register to 0.

```
// Invert polarity of IRQ
#define   SWC_LDN       0x04
#define   LDN_INX       0x07
#define   IRQ_TYPE_INX  0x71

INVERT_SWC_IRQ:
  out  SIO_BASE, LDN_INX
  out  SIO_BASE+1, SWC_LDN
  out  SIO_BASE, IRQ_TYPE_INX
  in   al, SIO_BASE+1
  and  al, 0xFD
  out  SIO_BASE+1, al
```


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

www.national.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 Customer 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