Software Configuration of National Semiconductor's LPC SuperI/O Family

National Semiconductor
Application Note AN-1129
Ilan Margalit
May 1999

National Semiconductor's Low Pin Count (LPC) SuperI/O family offers a wide range of single-chip solutions for the most commonly used I/O peripherals. The first generation of these devices includes the PC87360, PC87363, PC87364, PC87365 and PC87366. All five devices are available in 128-in PQFP packages, and their common functions (e.g., power supply, Legacy functions and the LPC bus interface) are pin-compatible. In addition, all these devices have a common configuration programming model, which makes it easier to port configuration firmware from one device to another.

This application note describes the configuration programming model for National's LPC SuperI/O family, and provides several examples that can be used to compose the complete SuperI/O configuration BIOS.

DEFINITIONS

- Logical device
  A functional module of the SuperI/O, e.g., the Floppy Disk Controller, that has a dedicated set of configuration registers for setting its base address, assigned IRQ number, etc.
- Register bank
  A group of registers, usually associated with a specific function. Such a group is usually part of a larger structure containing several such groups, or banks, that is used to reduce the address space occupied by the registers. Consequently, a bank selection operation must be performed prior to accessing any of the banked registers.
- Index-Data register pair
  Two 1-byte I/O ports that enable indirect access to all SuperI/O configuration registers. These ports occupy only a single pair of bytes in the I/O address space, and, theoretically, allow access to up to 256 byte-registers.

OVERVIEW

The configuration register structure can be divided into two major groups:

- Global SuperI/O configuration/control registers
- Logical device-specific configuration/control registers

The global registers control functions which affect more than one specific logical device, e.g., wait-state number selection, or control SuperI/O functions which are not part of a particular logical device, such as pin function multiplexing.

The logical device-specific registers control functions associated with a specific logical device, or the resources assigned to that logical device.

While the global SuperI/O configuration registers can be accessed directly, and are not banked, the logical device-specific configuration registers are grouped into banks, where each bank is dedicated to a particular logical device.

All SuperI/O configuration registers can be accessed via a single Index-Data register pair. The Index register is located at the SuperI/O configuration base address, an I/O byte-address, and the Data register is located at offset 1 from this base address (base + 1). Thus, the entire SuperI/O configuration register structure occupies only two I/O byte-addresses.

The configuration base address of the SuperI/O, and thus the addresses of the Index-Data register pair, are determined by the hardware strap option on the BADDR pin (pin 101 of the SuperI/O devices listed above).

SELECTING THE SUPERI/O BASE ADDRESS

To select the SuperI/O configuration base address give the BADDR strap pin the required value during a hardware reset.

If BADDR is sampled low on the trailing (rising) edge of the hardware reset signal ($\overline{\text{LRESET}}$), the configuration base address of the SuperI/O is 2Eh. Consequently, the address of the Index register is 2Eh, and the address of the Data register is 2Fh. This option does not require any external circuitry, as the SuperI/O has an internal static pull-down resistor on the BADDR pin. This pull-down is enabled only during the hardware reset process, and thus causes no unnecessary current consumption during normal operation.

If BADDR is sampled high on the trailing edge of the hardware reset signal ($\overline{\text{LRESET}}$), the configuration base address of the SuperI/O is 4Eh. Consequently, the address of the Index register is 4Eh, and the address of the Data register is 4Fh. This option requires an external pull-up resistor of 10 KOhm on the BADDR pin.

ACCESSING THE INDEX-DATA REGISTER PAIR

The SuperI/O Index and Data registers are the only registers that need to be accessed during Super I/O configuration. These two registers can be accessed at all times. There is no need for any kind of preceding operation.

When the Index register is written, the written data is latched in it. When it is read, the data latched in it is returned. When the Data register is written, the data is latched in the target configuration register pointed to by the Index register. Thus, these registers form two ports that enable indirect access to the whole SuperI/O configuration structure.

If a logical device-specific configuration register is accessed, the target register is selected according to the currently selected register bank, associated with the currently selected logical device. Therefore, when the Data register is accessed, the software must be certain about the values currently latched in the Index and Logical Device Number registers. This may not always be the case when different pieces of code are written independently, without knowing in advance the order of their execution. It is the responsibility of the software designer to ensure that the Data register is written only after the target register is appropriately selected.

In general, when accessing SuperI/O configuration registers of any kind, the read-modify-write policy is recommended. This prevents unwanted alteration of configuration bits, and makes the configuration software portable for future generations of these SuperI/O devices.

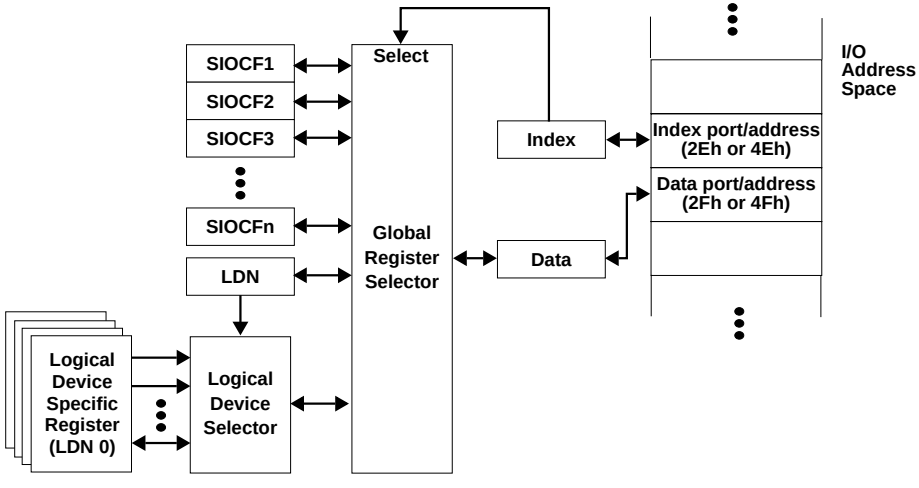

Figure 1. SuperI/O Configuration Registers Access Mechanism

ACCESSING GLOBAL CONFIGURATION REGISTERS

The global SuperI/O configuration registers are designated SIOCFn, where “n” is the register number. Although the different LPC SuperI/O devices have different sets of configuration registers, they are all accessed the same way.

These registers are always accessible, regardless of the currently selected logical device. Consequently, when a global SuperI/O configuration register is accessed, there is no need for Logical Device Number selection.

To access a global configuration register, write its index to the Index register, then write the new value to the Data register. As for any SuperI/O configuration register, the read-modify-write policy is strongly recommended.

Example

The following code sequence sets bit 0 of the SIOCF1 register to 0. SIOCF1 is one of the global SuperI/O configuration registers. This code sequence assumes that the SuperI/O configuration base address is 2Eh.

```
; Global constant definitions
#define   SIO_BASE 2Eh
#define   SIOCF1   21h

     out  SIO_BASE, SIOCF1       ; Select the SIOCF1 register
     in   al, SIO_BASE+1         ; Read the current value of SIOCF1
     and  al, FEh                ; Set bit 0 to 0
     out  SIO_BASE+1, al         ; Write the new value to the Data register
```

ACCESSING LOGICAL DEVICE SPECIFIC CONFIGURATION REGISTERS

As mentioned above, the logical device-specific registers are grouped into banks, where each bank is associated with a logical device.

To access a logical device specific register, first select its associated bank by writing the number of this bank’s associated logical device to the Logical Device Number (LDN) register. The LDN register (index 07h) is one of the SuperI/O global control registers. Once the desired bank is selected, write the index of the register to be accessed to the SuperI/O Index register. The register itself can then be accessed via the SuperI/O Data register.

Example

The following code sequence assigns IRQ number 9 to the WATCHDOG Timer. Note the read-modify-write operation employed to avoid altering the value of bits 7-4 of the Interrupt Number Select register. This is required since bit 4 of this register controls a wake-up function related to the interrupt request of the logical device, and therefore must not be unintentionally set to 0.

```
; Global constant definitions
#define   LDN_INDEX     07h.
#define   IRQ_NUM_INDEX 70H
#define   WATCHDOG_LDN  0Ah
#define   IRQ_9         09h

     out  SIO_BASE, LDN_INDEX         ; Select the LDN register
     out  SIO_BASE+1, WATCHDOG_LDN    ; Select the WATCHDOG logical device
     out  SIO_BASE, IRQ_NUM_INDEX     ; Select the Interrupt Number Select register
     in   al, SIO_BASE +1             ; Read the current IRQ assignment
     and  al, F0h                     ; Clear bits 3-0 to 0h
     or   al, IRQ_9                   ; Set bits 3-0 to 09h
     out  SIO_BASE + 1, al            ; Write the new IRQ assignment
```

VARIOUS EXAMPLES

Enabling/Disabling a Logical Device

The following code sequence activates the General Purpose I/O (GPIO) logical device.

```
; Global constant definitions
#define   GPIO_LDN           07h
#define   ACTIVATE_INDEX     30h

     mov  al, GPIO_LDN       ; Put the LDN in AL
     mov  ah, 01h            ; Put 01h in AH (put 00h to disable the LD)
     call SET_LD_STATE       ; Call the enable/disable subroutine
     ...
SET_LD_STATE:
     out  SIO_BASE, LDN_INDEX          ; Select the LDN register
     out  SIO_BASE+1, al               ; Select the desired LDN
     out  SIO_BASE, ACTIVATE_INDEX     ; Select the Activate register
     out  SIO_BASE+1, ah               ; Write the required value to the Activate register
     ret                               ; Return to the main program
```

Assigning a Base Address to a Logical Device

The following code sequence programs the GPIO logical device to have a base address of 440h.

```
; Global constant definitions
#define   GPIO_BASE          440h
#define   GPIO_LDN           07h
#define   BASE_MSB_INDEX     60h
#define   BASE_LSB_INDEX     61H

     mov  ax, GPIO_BASE                ; Put the desired base address in AX
     mov  bl, GPIO_LDN                 ; Put the required LDN in bl
     call SET_LD_BASE                  ; Call the base address setting subroutine
     ...
SET_LD_BASE:
     out SIO_BASE, LDN_INDEX           ; Select the LDN register
     out SIO_BASE +1, bl               ; Select the desired logical device
     out SIO_BASE, BASE_MSB_INDEX      ; Select the Base Address MSB register
     out SIO_BASE+1, ah                ; Write the base address MSB value
     out SIO_BASE, BASE_LSB_INDEX      ; Select the Base Address LSB register
     out SIO_BASE+1, al                ; Write the base address LSB value
     ret                               ; Return to main program
```

Selecting a DMA Channel for a Logical Device

The following code sequence assigns DMA channel number 3 to the Parallel Port logical device.

```
; Global constant definitions
#define   PPORT_DMA_CH       03h
#define   PPORT_LDN          01h
#define   DMA_CH0_SEL_INDEX  74h

     mov  al, PPORT_LDN                ; Put the LDN in al
     mov  ah, PPORT_DMA_CH             ; Put the DMA channel number in ah
     call SET_DMA_CH0                  ; Call the DMA channel setting subroutine
     ...
SET_DMA_CH0:
     out  SIO_BASE, LDN_INDEX              ; Select the LDN register
     out  SIO_BASE +1, al                  ; Select the desired logical device
     out  SIO_BASE, DMA_CH0_SEL_INDEX      ; Select the DMA Channel Select 0 register
     out  SIO_BASE+1, ah                   ; Write the DMA channel number
     ret                                   ; Return to main program
```

Routing the KBC Keyboard IRQ and Mouse IRQ to $\overline{\text{SMI}}$

The following code sequence routes the internal KBC Keyboard and Mouse IRQ signals to $\overline{\text{SMI}}$. This is done by setting bits 2 and 3 of the SIOCF8 register to 1.

```
; Global constant definitions
#define   SIOCF8        28h

KBD_MSE_IRQ2SMI:
     out  SIO_BASE, SIOCF8            ; Select the SIOCF8 register
     in   al, SIO_BASE +1             ; Read the current value of SIOCF8
     or   al, 0Ch                     ; Set bits 3-2 to 11b
     out  SIO_BASE + 1, al            ; Write the new value to the Data register
```

Enable $\overline{\text{SMI}}$ Routing to IRQ2

The following code sequence routes the internal $\overline{\text{SMI}}$ signal to IRQ Frame 2 (IRQ2) of the SERIRQ interface. This is done by setting bit 4 of the SIOCF5 register to 1.

```
; Global constant definitions
#define   SIOCF5        25h

SMI_TO_IRQ2:
     out  SIO_BASE, SIOCF5            ; Select the SIOCF5 register
     in   al, SIO_BASE +1             ; Read the current value of SIOCF5
     or   al, 10h                     ; Set bit 4 to 1b
     out  SIO_BASE + 1, al            ; Write the new value to the Data register
```

Changing the KBC Clock Source

The following sequence of operations changes the KBC clock source to a 12 MHz clock:

- Disable the KBC by writing 0 to its Activate register;
- Set bits 7-6 of the KBC Configuration register to 01b;
- Enable the KBC by writing 1 to its Activate register

These operations are implemented with the following code:

```
; Global constant definitions
#define   KBC_LDN           06h
#define   LDN_CFG0_INDEX    F0h

SET_KBC_CLK_SOURCE:
     out  SIO_BASE, LDN_INDEX         ; Select the LDN register
     out  SIO_BASE + 1, KBC_LDN       ; Select the KBC LDN
     out  SIO_BASE, ACTIVATE_INDEX    ; Select the Activate register
     out  SIO_BASE+1, 0               ; Disable the KBC
     out  SIO_BASE, LDN_CFG0_INDEX    ; Select the KBC Configuration register
     in   al, SIO_BASE +1             ; Read the current value of the register
     and  al, 3Fh                     ; Set bits 7-6 to 00b
     or   al, 40h                     ; Set bits 7-6 to 01b
     out  SIO_BASE + 1, al            ; Write the new value to the Data register
     out  SIO_BASE, ACTIVATE_INDEX    ; Select the Activate register
     out  SIO_BASE+1, 1               ; Enable the KBC
```

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

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 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