1052 Elwell Court, Palo Alto, CA 94303 Tel: 800.683.5943 www.zfmicro.com

Routing ZFx86 Interrupts

The designer must keep various considerations in mind when using the ZFx86 and its available interrupts. Interrupts are a limited commodity in any design and burdened with many years of x86 legacy. This document summarizes interrupt usage within the historical context of the x86 processor as well as within the new ZFx86 paradigm.

Legacy Development

The original IBM PC defined sixteen interrupts. In that early definition, some interrupts were reserved or assigned to particular uses. Table 1 shows these standard definitions.

Table 1. Standard x86 Interrupt Assignments

IRQ Number	x86 Recommended Assignment	Other Common Uses
0	System timer	
1	Keyboard controller	
2	Second PIC cascade. The occurrence of this interrupt tells the processor that an interrupt has been recorded by the second PIC, and that it can go interrogate the second PIC to find out what interrupt actually happened. IRQ 2 cannot be generated by a device – the signal line on the ISA bus called IRQ 2 actually routes to IRQ 9.	
3	Serial port two (COM2:)	Network card, mouse.
4	Serial port one (COM1:)	Network card, mouse
5	Parallel (printer) port two (LPT2:) and others	Sound card, mouse. This was the fixed disk interrupt prior to IDE and SCSI.
6	Floppy disk controller	
7	Parallel (printer) port one (LPT1:)	Sound card
8	Real-time clock (RTC)	
9	Unassigned This is the actual assignment forced by the motherboard for devices generating IRQ 2. Video cards are often capable of generating IRQ 2, but the interrupt is not processed by Windows.	
10	Unassigned	
11	Unassigned	
12	Unassigned	InPort (motherboard) mouse
13	Math Coprocessor	
14	Unassigned	Primary IDE
15	Unassigned	Secondary IDE

Further complicating the matter, the ZFx86, with its very high level of integration, contains many devices on board that automatically take control of an interrupt. See Table 2.

Table 2. ZFx86 Interrupt Assignments

IRQ Number	ZFx86 Assignment
0	System timer
1	Keyboard controller
2	Second PIC cascade
3	Serial port two (COM2:)
4	Serial port one (COM1:)
5	Unassigned
6	Floppy disk controller
7	Parallel (printer) port one (LPT1:)
8	Real-time clock (RTC)
9	Unassigned
10	Unassigned
11	USB
12	PS/2 Mouse
13	Math Coprocessor
14	Primary IDE
15	Secondary IDE

Looking at Table 2, you notice that out of the 16 interrupts available only three interrupts are actually free (IRQ5, 9, 10). Disabling the devices within the ZFx86 device frees up the resources.

PCI Architecture Develops

Over time, the ISA bus was found to be too slow for some applications. Therefore, in 1993, the industry formed a consortium tasked to resolve some of the deficiencies of the ISA architecture while still maintaining compatibility with the older (legacy) environment. This group defined the PCI architecture and, to the point of this paper, the usage of interrupts within the architecture. The use of the PCI architecture is very well explained in 4th edition of the *"PCI Hardware and Software: Architecture and Design"* written by Edward Solari and George Willse.An extraction from that book follows:

> There are several protocol requirements for implementing the INTx# signal lines:
>
> The three device types defined on a PCI bus are the PCI bus master, the target, and the BRIDGE; collectively they are called PCI resources.
>
> - Several devices (each with individual configuration registers) can be attached to each INTx# signal line; this is defined as interrupt sharing.
> - A device is wire-ORed to an INTx# signal line.
> - Once a device has asserted an INTx# signal line, it must keep it asserted until instructed to deassert by the appropriate interrupt service routine.

- A “single-device” integrated circuit chip on the platform or add-in card...that requires only one interrupt must use the INTA# signal line. Each integrated circuit chip or add-in card with at least one device that requires an interrupt service must use the INTA# signal line before any of the other interrupt signal lines can be used. The additional devices in an integrated circuit chip or an add-in card can also use INTA# or use any one of the other INTx# signal lines (INTB#, INTC#, and INTD#).
- Each individual device can only request service (assert) on one of the INTx# single lines. It can never request service on two or more INTX# signal lines.

The interrupt acknowledge cycle protocol on the PCI bus is only one cycle that returns the interrupt vector. It is the responsibility of the HOST/PCI BRIDGE or similar circuitry to convert the dual cycle 8259 compatible interrupt acknowledge cycle on the HOST bus to a SINGLE cycle PCI interrupt acknowledge cycle. It also is the responsibility of the PCI resource that contains the 8259 to convert the SINGLE cycle PCI interrupt acknowledge cycle into the dual cycle required by the 8259.

Additional reference material gleaned from *Interrupt-Driven PC System Design* written by Joseph McGivern, published by Annabooks, San Diego, California, USA, and *PCI System Architecture, Third Edition*, written by Tom Shanley and Don Anderson, published by Addison-Westley Publishing Company, International.

The ZFx86 BIOS implementation offers a number of choices to the designer who needs to integrate one or more interrupt driven embedded PCI device using the PCI specification. One interpretion of the specification leads to the wiring shown below. Since the guideline states that each single functioning PCI device always asserts INTA#, this results in a much higher activity load (resulting in greater system latency) on Route 0 than on Route 3. See Figure 1.

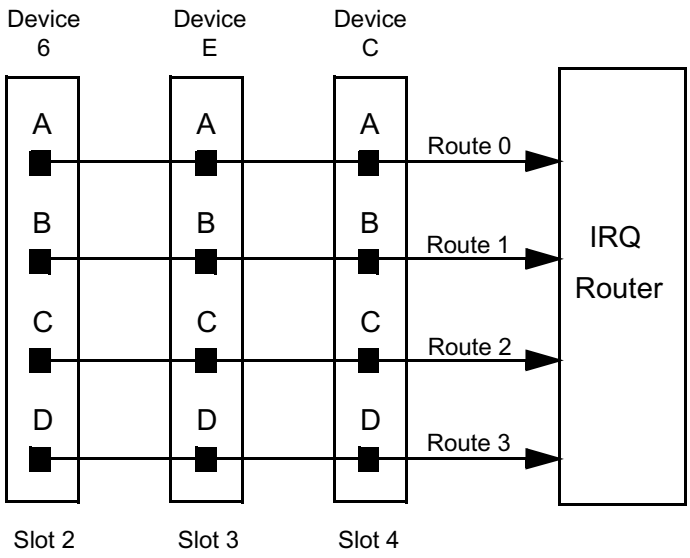

Figure 1. PCI Device Using IRQ Router

Therefore, alternative wiring schemes may be used to evenly spread the activity across more than one interrupt route. This is what was implemented in the Integrated Development System. See Figure 2.

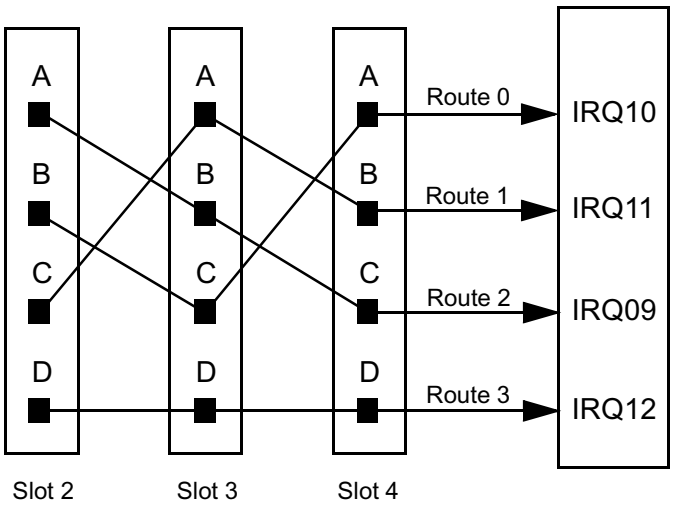

Figure 2. Preferred PCI Device HW Interrupt Wiring Scheme

Please note that both schemes are correct from the specification perspective, and that many more alternatives can be envisioned.

Plug and Play Operation

The PCI consortium's second goal was to create the ability to allow operating systems (through the use of their device drivers) to perform Plug and Play tasks. Plug and Play is the ability of any system to automatically detect and configure new system hardware after a reset. This lead to the creation of a PCI interrupt table in the BIOS. The table implemented in the ZFx86 BIOS is shown below:

```
  PCI_SLOT_DESC BUS0, SLOT_1_IDSEL, \
                SLOT_1_LABEL, \
                PIRQ0, PIRQ1, PIRQ2, PIRQ3, \
                ZFx86_ROUTING, ZFx86_ROUTING, \
                ZFx86_ROUTING, ZFx86_ROUTING
; Route            0     1      2      3
; PCI Slot Pin     A6    B7     A7     B8
; ZFx86 Ball       D02   E04    D01    E03

  PCI_SLOT_DESC BUS0, SLOT_2_IDSEL, \
                SLOT_2_LABEL, \
                PIRQ2, PIRQ0, PIRQ1, PIRQ3, \
                ZFx86_ROUTING, ZFx86_ROUTING, \
                ZFx86_ROUTING, ZFx86_ROUTING
; Route            2     0      1      3
; PCI Slot Pin     A6    B7     A7     B8
; ZFx86 Ball       D01   D02    E04    E03
```

```
PCI_SLOT_DESC BUS0, SLOT_3_IDSEL, \
              SLOT_3_LABEL, \
              PIRQ1, PIRQ2, PIRQ0, PIRQ3, \
              ZFx86_ROUTING, ZFx86_ROUTING, \
              ZFx86_ROUTING, ZFx86_ROUTING
; Route           1     2      0      3
; PCI Slot Pin    A6    B7     A7     B8
; ZFx86 Ball      E04   D01    D02    E03

  PCI_SLOT_DESC BUS0, SLOT_4_IDSEL, \
              SLOT_4_LABEL, \
              PIRQ3, PIRQ2, PIRQ1, PIRQ0, \
              ZFx86_ROUTING, ZFx86_ROUTING, \
              ZFx86_ROUTING, ZFx86_ROUTING
; Route           3     2      1      0
; PCI Slot Pin    A6    B7     A7     B8
; ZFx86 Ball      E03   D01    E04    D02

  PCI_SLOT_DESC BUS0, SLOT_5_IDSEL, \
              SLOT_5_LABEL, \
              PIRQ0, PIRQ3, PIRQ2, PIRQ1, \
              ZFx86_ROUTING, ZFx86_ROUTING, \
              ZFx86_ROUTING, ZFx86_ROUTING
; Route           0     3      2      1
; PCI Slot Pin    A6    B7     A7     B8
; ZFx86 Ball      D02   E03    D01    E04

  PCI_SLOT_DESC BUS0, SLOT_6_IDSEL, \
              SLOT_6_LABEL, \
              PIRQ1, PIRQ0, PIRQ3, PIRQ2, \
              ZFx86_ROUTING, ZFx86_ROUTING, \
              ZFx86_ROUTING, ZFx86_ROUTING
; Route           1     0      3      2
; PCI Slot Pin    A6    B7     A7     B8
; ZFx86 Ball      E04   D02    E03    D01

  PCI_SLOT_DESC BUS0, SLOT_7_IDSEL, \
              SLOT_7_LABEL, \
              PIRQ2, PIRQ1, PIRQ0, PIRQ3, \
              ZFx86_ROUTING, ZFx86_ROUTING, \
              ZFx86_ROUTING, ZFx86_ROUTING
; Route           2     1      0      3
; PCI Slot Pin    A6    B7     A7     B8
; ZFx86 Ball      D01   E04    D02    E03

  PCI_SLOT_DESC BUS0, SLOT_8_IDSEL, \
              SLOT_8_LABEL, \
              PIRQ3, PIRQ0, PIRQ1, PIRQ2, \
              ZFx86_ROUTING, ZFx86_ROUTING, \
              ZFx86_ROUTING, ZFx86_ROUTING
; Route           3     0      1      2
; PCI Slot Pin    A6    B7     A7     B8
; ZFx86 Ball      E03   D02    E04    D01
```

```
PCI_SLOT_DESC BUS0, SLOT_9_IDSEL, \
              SLOT_9_LABEL, \
              PIRQ2, PIRQ3, PIRQ0, PIRQ1, \
              ZFx86_ROUTING, ZFx86_ROUTING, \
              ZFx86_ROUTING, ZFx86_ROUTING
; Route           2     3      0      1
; PCI Slot Pin    A6    B7     A7     B8
; ZFx86 Ball      D01   E03    D02    E04

  PCI_SLOT_DESC BUS0, SLOT_10_IDSEL, \
              SLOT_10_LABEL, \
              PIRQ1, PIRQ2, PIRQ3, PIRQ0, \
              ZFx86_ROUTING, ZFx86_ROUTING, \
              ZFx86_ROUTING, ZFx86_ROUTING
; Route           1     2      3      0
; PCI Slot Pin    A6    B7     A7     B8
; ZFx86 Ball      E04   D01    E03    D02
```

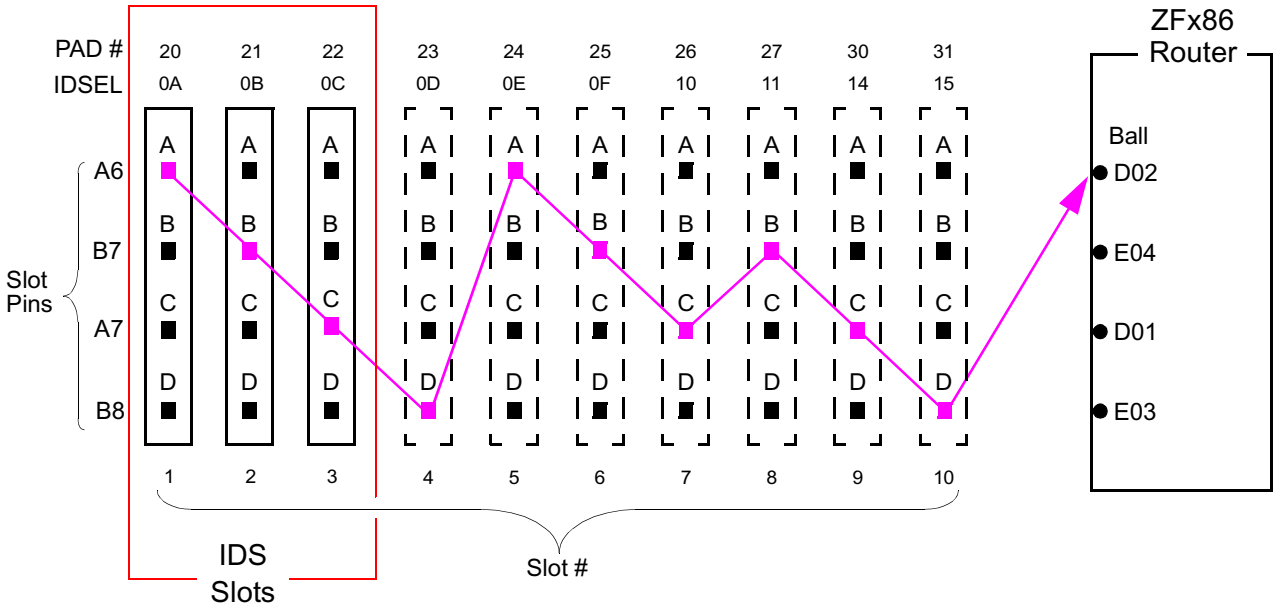

Figure 3. Route 0

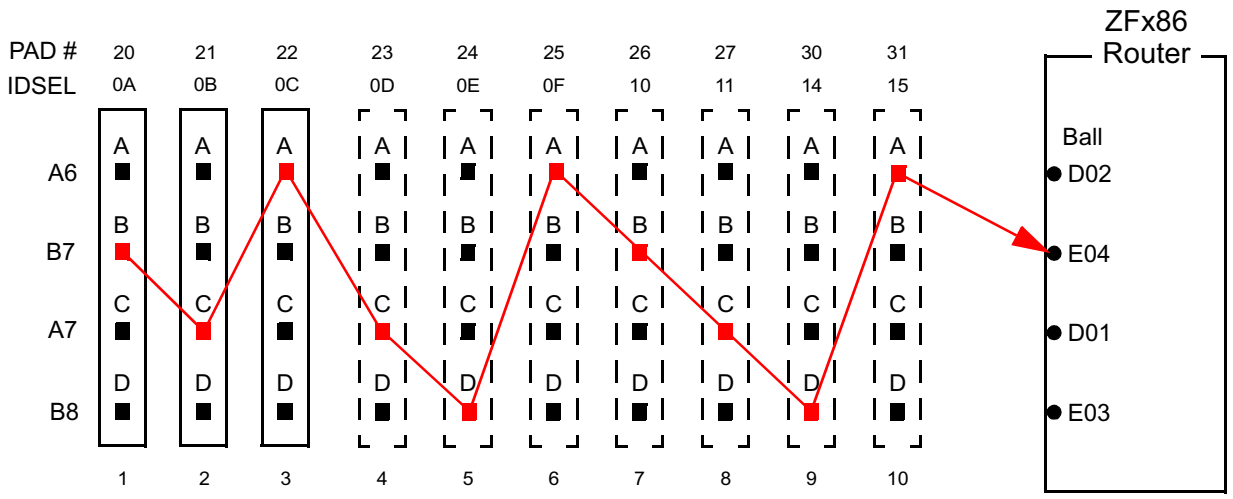

Figure 4. Route 1

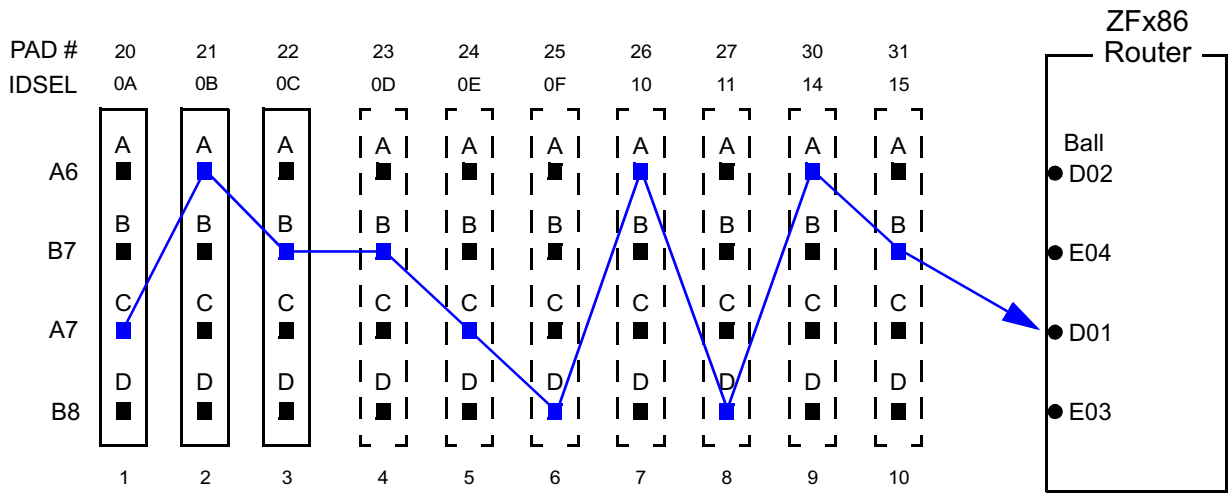

Figure 5. Route 2

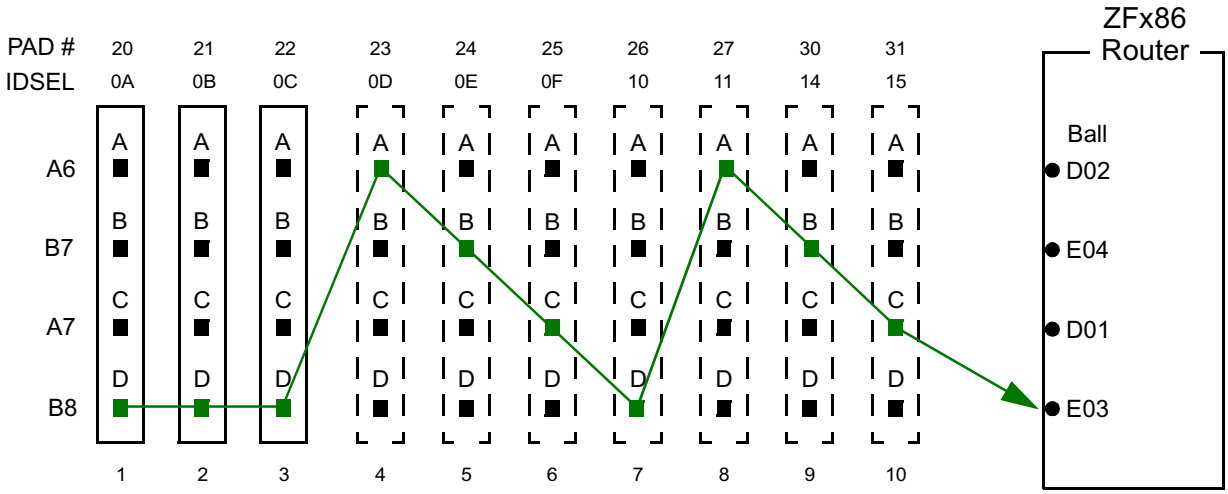

Figure 6. Route 3

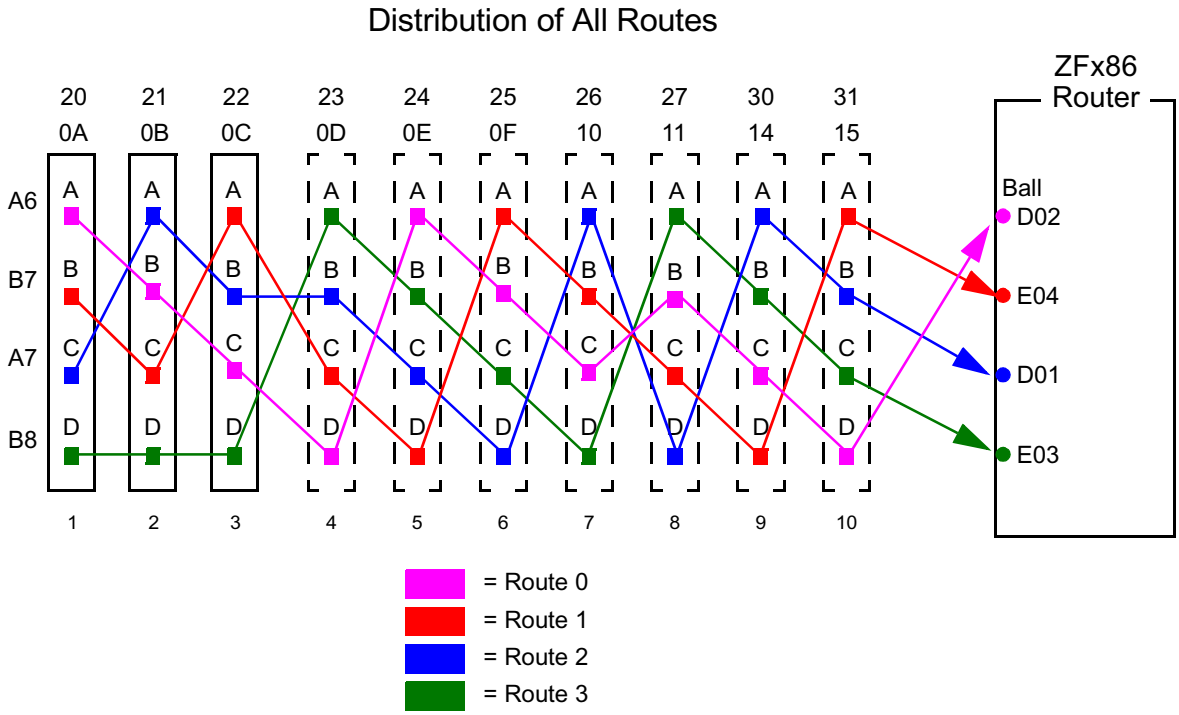

Figure 7. All Route Distribution

The BIOS provides services that allow the designer to view the settings. The routing options are returned through the INT 1Ah function PCI Services (AH = B1h), called Get PCI IRQ routing Options (AL= 0Eh) routine. This service returns a data table showing IRQ routing table entries for each device or slot. Each entry describes how the ZFx86 based platform routes hardware interrupt lines to individual integrated PCI devices and slots. Figure 8 contains a sample ZFx86 BIOS PCI IRQ Routing Table where the BIOS stores the device or slot information.

```
PCI Interrupt Routing Information
------------------------------------------------------------------------

BusNum : 00h       DevNum : 15h    SlotNum : 0ah
Interrupt A - Link Value : 02h IRQ Bit Map : d6f8h
Interrupt B - Link Value : 03h IRQ Bit Map : d6f8h
Interrupt C - Link Value : 04h IRQ Bit Map : d6f8h
Interrupt D - Link Value : 01h IRQ Bit Map : 0800h

BusNum : 00h       DevNum : 14h    SlotNum : 09h
Interrupt A - Link Value : 03h IRQ Bit Map : d6f8h
Interrupt B - Link Value : 04h IRQ Bit Map : d6f8h
Interrupt C - Link Value : 01h IRQ Bit Map : 0800h
Interrupt D - Link Value : 02h IRQ Bit Map : d6f8h

BusNum : 00h       DevNum : 11h    SlotNum : 08h
Interrupt A - Link Value : 04h IRQ Bit Map : d6f8h
Interrupt B - Link Value : 01h IRQ Bit Map : 0800h
Interrupt C - Link Value : 02h IRQ Bit Map : d6f8h
Interrupt D - Link Value : 03h IRQ Bit Map : d6f8h

BusNum : 00h       DevNum : 10h    SlotNum : 07h
Interrupt A - Link Value : 03h IRQ Bit Map : d6f8h
Interrupt B - Link Value : 02h IRQ Bit Map : d6f8h
Interrupt C - Link Value : 01h IRQ Bit Map : 0800h
Interrupt D - Link Value : 04h IRQ Bit Map : d6f8h

BusNum : 00h       DevNum : 0fh    SlotNum : 06h
Interrupt A - Link Value : 02h IRQ Bit Map : d6f8h
Interrupt B - Link Value : 01h IRQ Bit Map : 0800h
Interrupt C - Link Value : 04h IRQ Bit Map : d6f8h
Interrupt D - Link Value : 03h IRQ Bit Map : d6f8h

BusNum : 00h       DevNum : 0eh    SlotNum : 05h
Interrupt A - Link Value : 01h IRQ Bit Map : 0800h
Interrupt B - Link Value : 04h IRQ Bit Map : d6f8h
Interrupt C - Link Value : 03h IRQ Bit Map : d6f8h
Interrupt D - Link Value : 02h IRQ Bit Map : d6f8h

BusNum : 00h       DevNum : 0dh    SlotNum : 04h
Interrupt A - Link Value : 04h IRQ Bit Map : d6f8h
Interrupt B - Link Value : 03h IRQ Bit Map : d6f8h
Interrupt C - Link Value : 02h IRQ Bit Map : d6f8h
Interrupt D - Link Value : 01h IRQ Bit Map : 0800h
```

```
BusNum : 00h       DevNum : 0ch    SlotNum : 03h
Interrupt A - Link Value : 02h IRQ Bit Map : d6f8h
Interrupt B - Link Value : 03h IRQ Bit Map : d6f8h
Interrupt C - Link Value : 01h IRQ Bit Map : 0800h
Interrupt D - Link Value : 04h IRQ Bit Map : d6f8h

BusNum : 00h       DevNum : 0bh    SlotNum : 02h
Interrupt A - Link Value : 03h IRQ Bit Map : d6f8h
Interrupt B - Link Value : 01h IRQ Bit Map : 0800h
Interrupt C - Link Value : 02h IRQ Bit Map : d6f8h
Interrupt D - Link Value : 04h IRQ Bit Map : d6f8h

BusNum : 00h       DevNum : 0ah    SlotNum : 01h
Interrupt A - Link Value : 01h IRQ Bit Map : 0800h
Interrupt B - Link Value : 02h IRQ Bit Map : d6f8h
Interrupt C - Link Value : 03h IRQ Bit Map : d6f8h
Interrupt D - Link Value : 04h IRQ Bit Map : d6f8h

BusNum : 00h       DevNum : 13h    SlotNum : 00h
Interrupt A - Link Value : 01h IRQ Bit Map : 0800h
Interrupt B - Link Value : 00h IRQ Bit Map : 0800h
Interrupt C - Link Value : 00h IRQ Bit Map : 0800h
Interrupt D - Link Value : 00h IRQ Bit Map : 0800h
---------------------------------------------------------------------
```

Figure 8. ZFx86 BIOS PCI IRQ Routing Table Example

The routing must be associated with a legacy interrupt using the PCI Steering mechanism.This design choice, in theory, does not have a correct solution. In the ZFx86 case, the default configuration chosen is found in Table 3.

Table 3. ZFx86 Default PCI Interrupts

PCI Interrupt	ISA Interrupt	ZFx86 Ball
INT A	IRQ 9	D02
INT B	IRQ 10	E04
INT C	IRQ 11	D01
INT D	IRQ 12	E03

This absorbs most of the remaining system IRQ’s; however, it also creates a conflict with two devices:

- USB (IRQ11), and
- the mouse (IRQ12).

Therefore, enabling the USB in the BIOS setup screen takes over IRQ11 conversely if IRQ11 is removed as a valid option using the BIOS CMOS settings the internal USB controller will not be available.

Mapping PCI Interrupts To Any IRQ

The final complication that arises in the system is that the OS through its device driver may want to change the interrupt that is associated with the device. This is accomplished using the PCI interrupt steering register. This mechanism allows PCI interrupts to be mapped to most of the IRQs of the internal 8259s using PCI Interrupt Steering Registers 1 and 2 (found at F0 Index 5Ch and 5Dh). See Table 4.

Table 4. F0 Index 5C and 5D – PCI Interrupt Steering Register 1 and 2

Index 5Ch	PCI Interrupt Steering Register 1 (R/W)			Reset Value = 00h
7:4	**INTB# Target Interrupt**			
	0000 = Disable	0100 = IRQ4	1000 = RSVD	1100 = IRQ12
	0001 = IRQ1	0101 = IRQ5	1001 = IRQ9	1101 = RSVD
	0010 = RSVD	0110 = IRQ6	1010 = IRQ10	1110 = IRQ14
	0011 = IRQ3	0111 = IRQ7	1011 = IRQ11	1111 = IRQ15
3:0	**INTA# Target Interrupt**			
	0000 = Disable	0100 = IRQ4	1000 = RSVD	1100 = IRQ12
	0001 = IRQ1	0101 = IRQ5	1001 = IRQ9	1101 = RSVD
	0010 = RSVD	0110 = IRQ6	1010 = IRQ10	1110 = IRQ14
	0011 = IRQ3	0111 = IRQ7	1011 = IRQ11	1111 = IRQ15
Note: The target interrupt must first be configured as level sensitive via I/O Port 4D0h and 4D1h in order to maintain PCI interrupt compatibility				

Index 5Dh	PCI Interrupt Steering Register 2 (R/W)			Reset Value = 00h
7:4	**INTD# Target Interrupt**			
	0000 = Disable	0100 = IRQ4	1000 = RSVD	1100 = IRQ12
	0001 = IRQ1	0101 = IRQ5	1001 = IRQ9	1101 = RSVD
	0010 = RSVD	0110 = IRQ6	1010 = IRQ10	1110 = IRQ14
	0011 = IRQ3	0111 = IRQ7	1011 = IRQ11	1111 = IRQ15
3:0	**INTC# Target Interrupt**			
	0000 = Disable	0100 = IRQ4	1000 = RSVD	1100 = IRQ12
	0001 = IRQ1	0101 = IRQ5	1001 = IRQ9	1101 = RSVD
	0010 = RSVD	0110 = IRQ6	1010 = IRQ10	1110 = IRQ14
	0011 = IRQ3	0111 = IRQ7	1011 = IRQ11	1111 = IRQ15
Note: The target interrupt must first be configured as level sensitive via I/O Port 4D0h and 4D1h in order to maintain PCI interrupt compatibility				

This results in some devices (much as in the legacy world) being associated with certain interrupts. For example, networks cards typically use interrupt 10. Looking at Table 6 and Figure 9 it is clear that you can route any PCI Interrupt (INTx) to virtually any system IRQ.

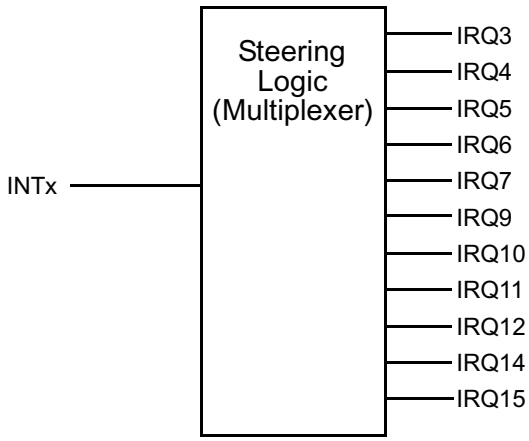

Figure 9. IRQ Steering Logic

ZFx86 PCI Interrupt Configuration Settings

The ZFx86 BIOS assigns PCI interrupts and builds the ROM based PIRQ Table at boot up. Two BIOS configuration settings allow the embedded designer to fine tune the PIRQ Table before it is written to ROM:

1. Use the Advanced > PCI/PNP ISA IRQ Resource Exclusion menu.

   This setting allows you to reserve specific ISA interrupts numbered 3, 4, 5, 7, 9, 10, and 11. Interrupts reserved under this setting will *not* be used for PCI IRQ allocation.

2. Use the Advanced > PCI IRQ [Line 1:][... Line 2:][...Line3:][...Line 4:] selection.

   This setting uses the ZFx86 steering logic and allows you to override the IRQ route with IRQ 3 through 15.

Please see *Interrupt-Driven PC System Design* for a detailed discussion of PCI based interrupt optimization.

Setting Up the IDS Interrupt Slots

Let's go through a specific IDS example:

1. Plug three network cards into the IDS' Slots 1, 2, and 3.

   Each network card uses INTA (pin A6) on the slot to activate the interrupt.

   On the ZFx86, this pin is physically routed to ball D02 for slot1, ball D01 for slot 2, and ball E04 for slot 3.

2. The device driver looks at the BIOS interrupt table and sees that the default is that IRQ9 is connected to D02, that IRQ11 is tied to D01, and that IRQ10 is associated with E04.

3. In the CMOS setting IRQ9 was disabled; therefore, the driver must steer the interrupt to an available IRQ using Index register 5C and 5D. In this case, 0A received IRQ11; device 0B received IRQ10, and device 0C (slot 3) received IRQ3.

PCI Configuration Header Text

Notice the interrupt assignments of the three network cards in the PCI Configuration in the header text below:

- Device 0A = IRQ11
- Device 0B = IRQ10
- Device 0C = IRQ3

All three slots are using INTA#.

```
PCI Registers for Bus[00], Dev[0A], Func[0]
Vendor: 3COM Corporation              Device: 9055h
Class: Ethernet Network
                    x (low nibble of Register Offset)
x :  0   1   2   3   4   5   6   7   8   9   A   B   C   D   E   F
0x: B7  10  55  90  13  00  10  02  30  00  00  02  04  50  00  00
1x: 01  10  00  00  00  E0  DF  FE  00  00  00  00  00  00  00  00
2x: 00  00  00  00  00  00  00  00  00  00  00  00  B7  10  55  90
3x: 00  00  00  00  DC  00  00  00  00  00  00  00  0B  01  0A  0A
4x: 00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
5x: 00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
6x: 00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
7x: 00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
8x: 00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
9x: 00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
Ax: 00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
Bx: 00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
Cx: 00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
Dx: 00  00  00  00  00  00  00  00  00  00  00  00  01  00  01  76
Ex: 00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
Fx: 00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00

Vendor: 10B7                          Revision: 30h
Device: 9055                          Class: Network Controller
```

```
Sub-Class: Ethernet Network             Prog Int: 00h
# of Functions: Single

I/O Slave Interface: Enabled            Memory Slave Interface: Enabled
PCI Master: No                          DEVSEL# Timing: Medium

Expansion ROM: Disabled                 Base Address: 00000000h
Address Range 1: I/O    00001000h       Address Range 2: Memory FEDFE000h
Address Range 3: None   00000000h       Address Range 4: None   00000000h
Address Range 5: None   00000000h       Address Range 6: None   00000000h
Interrupt Assignment: IRQ11             PCI Interrupt: INTA#

Minimum Grant (1/4 us): 0Ah             Maximum Latency: 0Ah
Cache Line Size: 04h Dwords             Master Latency Timer: 50h Clocks
Memory Write/Invalidate: Supported      Palette Snooping: No
Generates Fast Back to Back: No         Accepts Fast Back to Back: No
Address Stepping: Not Required          Monitor Special Cycles: No
BIST: Not Supported                     BIST Status: Passed

Parity Errors: Ignored                  Parity Error Status: Inactive
SERR# Generation: Disabled              SERR# Status: Inactive
Received Target Abort: Inactive         Signalled Target Abort: Inactive
Master Abort Status: Inactive           PERR# Status: Inactive

PCI Registers for Bus[00], Dev[0B], Func[0]
Vendor: 3COM Corporation                Device: 9055h
Class: Ethernet Network
                    x (low nibble of Register Offset)
x :  0   1   2   3   4   5   6   7   8   9   A   B   C   D   E   F
0x: B7  10  55  90  13  00  10  02  30  00  00  02  04  50  00  00
1x: 81  10  00  00  00  E4  DF  FE  00  00  00  00  00  00  00  00
2x: 00  00  00  00  00  00  00  00  00  00  00  00  B7  10  55  90
3x: 00  00  00  00  DC  00  00  00  00  00  00  00  0A  01  0A  0A
4x: 00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
5x: 00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
6x: 00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
7x: 00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
8x: 00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
9x: 00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
Ax: 00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
Bx: 00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
Cx: 00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
Dx: 00  00  00  00  00  00  00  00  00  00  00  00  01  00  01  76
Ex: 00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
Fx: 00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00

Vendor: 10B7                            Revision: 30h
Device: 9055                            Class: Network Controller
Sub-Class: Ethernet Network             Prog Int: 00h
# of Functions: Single
```

```
I/O Slave Interface: Enabled            Memory Slave Interface: Enabled
PCI Master: No                          DEVSEL# Timing: Medium

Expansion ROM: Disabled                 Base Address: 00000000h
Address Range 1: I/O    00001080h       Address Range 2: Memory FEDFE400h
Address Range 3: None   00000000h       Address Range 4: None   00000000h
Address Range 5: None   00000000h       Address Range 6: None   00000000h
Interrupt Assignment: IRQ10             PCI Interrupt: INTA#

Minimum Grant (1/4 us): 0Ah             Maximum Latency: 0Ah
Cache Line Size: 04h Dwords             Master Latency Timer: 50h Clocks
Memory Write/Invalidate: Supported      Palette Snooping: No
Generates Fast Back to Back: No         Accepts Fast Back to Back: No
Address Stepping: Not Required          Monitor Special Cycles: No
BIST: Not Supported                     BIST Status: Passed

Parity Errors: Ignored                  Parity Error Status: Inactive
SERR# Generation: Disabled              SERR# Status: Inactive
Received Target Abort: Inactive         Signalled Target Abort: Inactive
Master Abort Status: Inactive           PERR# Status: Inactive

PCI Registers for Bus[00], Dev[0C], Func[0]
Vendor: 3COM Corporation                 Device: 9055h
Class: Ethernet Network
                    x (low nibble of Register Offset)
x :  0   1   2   3   4   5   6   7   8   9   A   B   C   D   E   F
0x: B7  10  55  90  13  00  10  02  30  00  00  02  04  50  00  00
1x: 01  14  00  00  00  E8  DF  FE  00  00  00  00  00  00  00  00
2x: 00  00  00  00  00  00  00  00  00  00  00  00  B7  10  55  90
3x: 00  00  00  00  DC  00  00  00  00  00  00  00  03  01  0A  0A
4x: 00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
5x: 00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
6x: 00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
7x: 00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
8x: 00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
9x: 00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
Ax: 00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
Bx: 00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
Cx: 00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
Dx: 00  00  00  00  00  00  00  00  00  00  00  00  01  00  01  76
Ex: 00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00
Fx: 00  00  00  00  00  00  00  00  00  00  00  00  00  00  00  00

Vendor: 10B7                            Revision: 30h
Device: 9055                            Class: Network Controller
Sub-Class: Ethernet Network             Prog Int: 00h
# of Functions: Single

I/O Slave Interface: Enabled            Memory Slave Interface: Enabled
PCI Master: No                          DEVSEL# Timing: Medium

Expansion ROM: Disabled                 Base Address: 00000000h
Address Range 1: I/O    00001400h       Address Range 2: Memory FEDFE800h
```

```
Address Range 3: None   00000000h      Address Range 4: None   00000000h
Address Range 5: None   00000000h      Address Range 6: None   00000000h
Interrupt Assignment: IRQ03            PCI Interrupt: INTA#

Minimum Grant (1/4 us): 0Ah            Maximum Latency: 0Ah
Cache Line Size: 04h Dwords            Master Latency Timer: 50h Clocks
Memory Write/Invalidate: Supported     Palette Snooping: No
Generates Fast Back to Back: No        Accepts Fast Back to Back: No
Address Stepping: Not Required         Monitor Special Cycles: No
BIST: Not Supported                    BIST Status: Passed

Parity Errors: Ignored                 Parity Error Status: Inactive
SERR# Generation: Disabled             SERR# Status: Inactive
Received Target Abort: Inactive        Signalled Target Abort: Inactive
Master Abort Status: Inactive          PERR# Status: Inactive
```

ZFx86 PCI Interrupt Definitions

In summary, the HW and SW designer may access many alternate configuration available for a PCI system. To simplify this task, the ZFx86 BIOS and processor makes certain selections for these alternatives consistent with the legacy and history that exists in the x86 world. Designers may change these choices by manipulating the internal registers of the ZFx86; however, those choices may then no longer be compliant with industry practices (although 100% compliant with the specification). This results in situations were standard drivers distributed with the devices do not correctly interface in the system. Tables 5 and 6 define the default selections made by the ZFx86 processor.

Table 5. PCI Steering, Interrupt, and IRQ HW Routing

IRQ #	ZFx86 Device	ZFx86 Default PCI Steering
0	System Timer	Reserved
1	Keyboard	Reserved
2	Second PIC	Reserved
3	COM2	Allowed
4	COM1	Allowed
5	Unassigned	Allowed
6	Floppy Disk	Allowed
7	LPT1	Allowed
8	Real Time Clk	Reserved
9	Unassigned	Allowed – Default INTA
10	Unassigned	Allowed – Default INTB
11	USB	Allowed – Default INTC
12	PS/2 Mouse	Allowed – Default INTD
13	FPU	Reserved
14	Primary IDE	Allowed
15	Secondary IDE	Allowed

The slot number (device ID) is connected to the device using the IDSEL signal to one of the PCI address lines. For example, PAD 26, IDSEL 10h, Slot 7. See Figure 4 on page 7.

Table 6. Default PCI Interrupt Routing Table

Slot #	Device (IDSEL)[a]	PCI A/D Pin[b]	INTA# BGA Pin (PCI A6)[c]	INTB# BGA Pin (PCI B7)[c]	INTC# BGA Pin (PCI A7)[c]	INTD# BGA Pin (PCI B8)[c]	Comments
1	0A	PAD20	09 – D2	10 – E4	11 – D1	12 – E3	Available
2	0B	PAD21	11 – D1	09 – D2	10 – E4	12 – E3	Available
3	0C	PAD22	10 – E4	11 – D1	09 – D2	12 – E3	Available
4	0D	PAD23	12 – E3	11 – D1	10 – E4	09 – D2	Available
5	0E	PAD24	09 – D2	12 – E3	11 – D1	10 – E4	Available
6	0F	PAD25	10 – E4	09 – D2	12 – E3	11 – D1	Available
7	10	PAD26	11 – D1	10 – E4	09 – D2	12 – E3	Available
8	11	PAD27	12 – E3	09 – D2	10 - E4	11 – D1	Available
N/A	12	Reserved	Reserved	Reserved	Reserved	Reserved	Internal South Bridge
None	13	Reserved	11 – Internal[d]	None	None	None	Internal USB
9	14	PAD30	11 – D1	12 – E3	09 – D2	10 – E4	Available
10	15	PAD31	10 – E4	11 – D1	12 – E3	09 – D2	Available

a. The device ID is in hexadecimal format.
b. The PCI A/D pin is the PCI Address/Data pin connected to the device id select line on the PCI slot (A26).
c. This interrupt designates the default interrupt before PCI steering occurs, this can change due to SW interactions. The BGA pin is the ZFx86 ball grid array physical pin. The PCI pin in parenthesis is the pin on the slot that must be wired to the ZFx86 ball grid array pin, for example, slot 1 – A6 is wired to ZFx86 – D2.
d. USB uses IRQ11 exclusively for SMI wakeup events.

Figure 10 shows the PCI interrupt routing summary. Note the following items:

- The ZFx86 internal interrupt routing is displayed in blue, and external routing is displayed in yellow.
- External ISA interrupts 3, 4, 5, 6, 7, 14, and 15 are hard wired to the respective 8259 PIC controller lines.
- If the internal device is enabled, it claims the interrupt. It will not be driven out of the chip.
- If the internal device is not enabled, the interrupt becomes available to the OS for steering of PCI IRQ's or to the ISA bus.
- ISA interrupts 9, 10, 11, and 12 are wired into the input of the steering register PIRQA, PIRQB, PIRQC, and PIRQD. In order to prevent these from being steered, they must be reserved using the BIOS Setup menu.

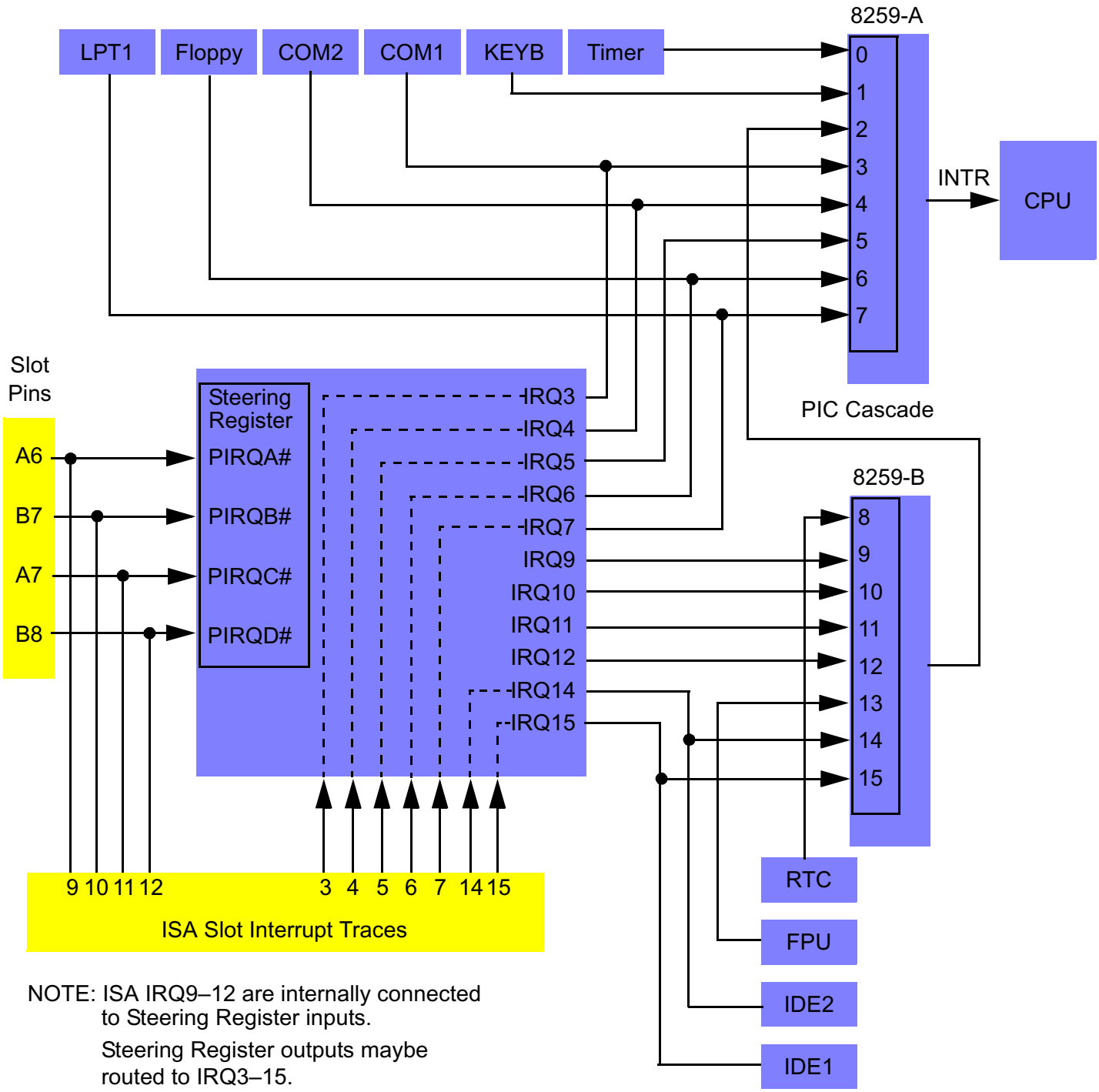

Figure 10. PCI Interrupt Routing Summary