IA32中断与异常研究

Liuzhi, Nov 2010

liuzhizhiyi@163.com

IA32一共255个中断源

- 在保护模式下面是IDT，在实模式下面是从0开始的内存指定的interrupt service routine的地址。
- 0 ~ 31 reversed for IA32 architecture.
- 31 ~ 255 can be used by other software/hardware.
- 软件对中断的处理
  - 中断向量表。（IVT）是一张表，指明每一个中断的处理函数入口地址。

reserved中断向量号

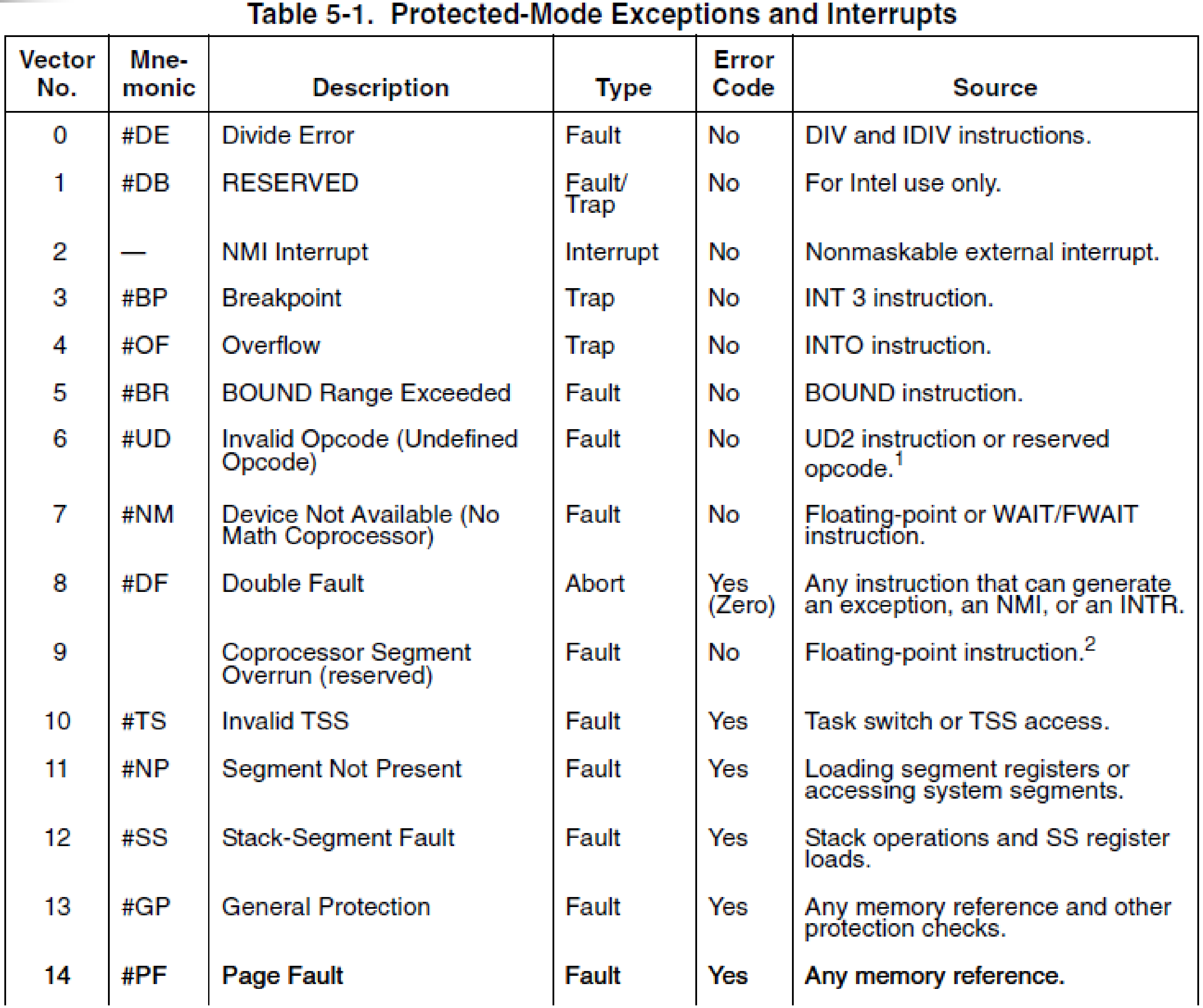
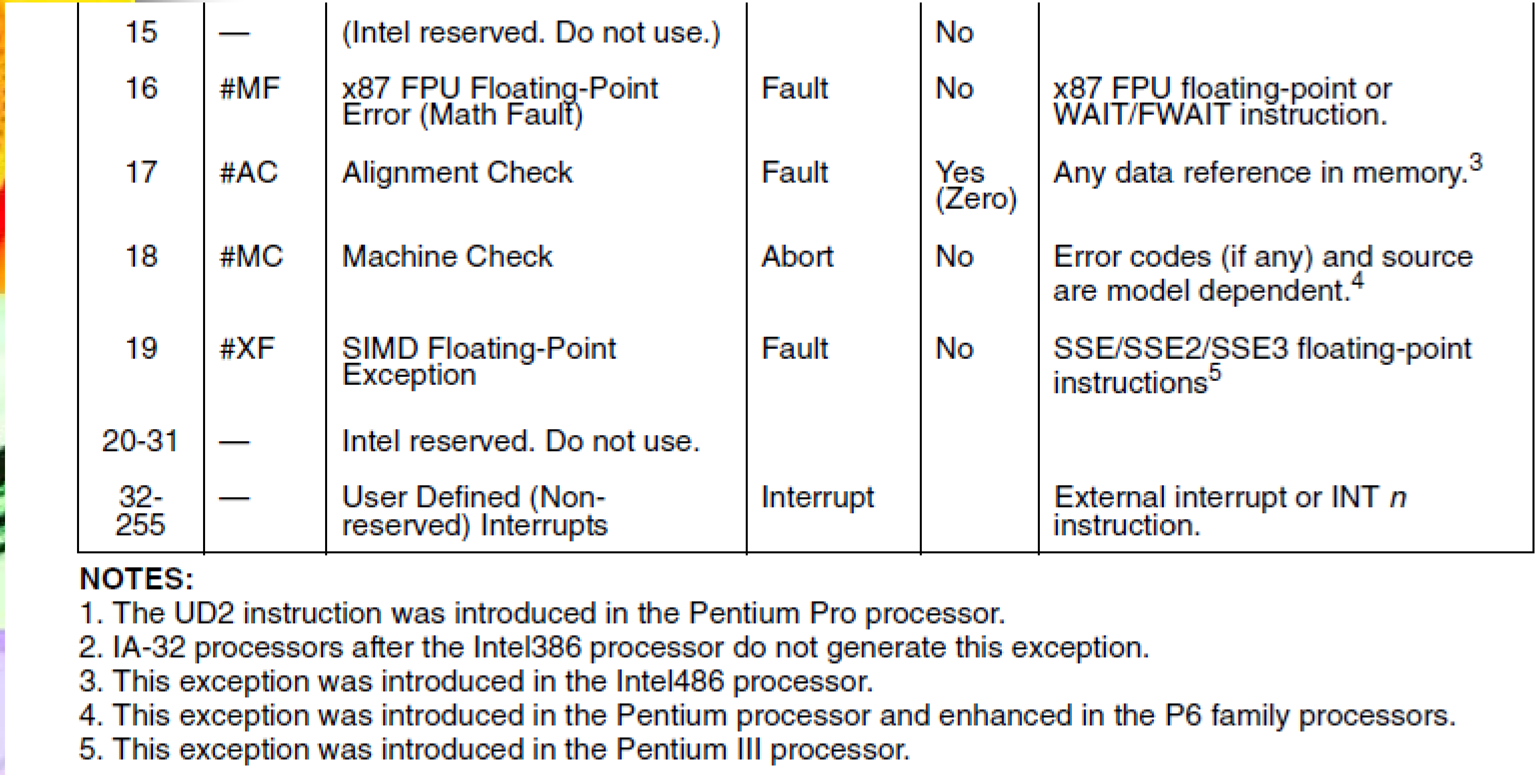


Table 5-1. Protected-Mode Exceptions and Interrupts

Vector No.	Mne-monic	Description	Type	Error Code	Source
0	#DE	Divide Error	Fault	No	DIV and IDIV instructions.
1	#DB	RESERVED	Fault/ Trap	No	For Intel use only.
2	—	NMI Interrupt	Interrupt	No	Nonmaskable external interrupt.
3	#BP	Breakpoint	Trap	No	INT 3 instruction.
4	#OF	Overflow	Trap	No	INTO instruction.
5	#BR	BOUND Range Exceeded	Fault	No	BOUND instruction.
6	#UD	Invalid Opcode (Undefined Opcode)	Fault	No	UD2 instruction or reserved opcode.[1]
7	#NM	Device Not Available (No Math Coprocessor)	Fault	No	Floating-point or WAIT/FWAIT instruction.
8	#DF	Double Fault	Abort	Yes (Zero)	Any instruction that can generate an exception, an NMI, or an INTR.
9		Coprocessor Segment Overrun (reserved)	Fault	No	Floating-point instruction.[2]
10	#TS	Invalid TSS	Fault	Yes	Task switch or TSS access.
11	#NP	Segment Not Present	Fault	Yes	Loading segment registers or accessing system segments.
12	#SS	Stack-Segment Fault	Fault	Yes	Stack operations and SS register loads.
13	#GP	General Protection	Fault	Yes	Any memory reference and other protection checks.
14	#PF	Page Fault	Fault	Yes	Any memory reference.

reserved中断向量号



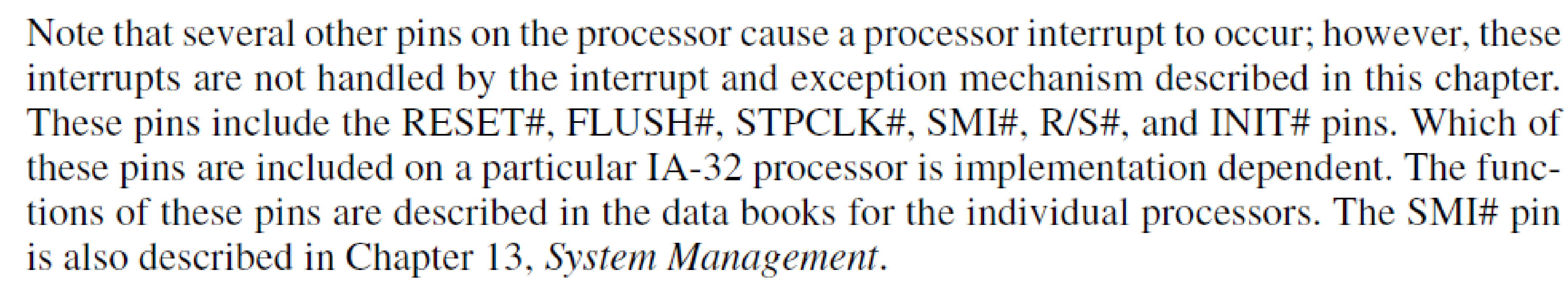
15	—	(Intel reserved. Do not use.)		No	
16	#MF	x87 FPU Floating-Point Error (Math Fault)	Fault	No	x87 FPU floating-point or WAIT/FWAIT instruction.
17	#AC	Alignment Check	Fault	Yes (Zero)	Any data reference in memory.[3]
18	#MC	Machine Check	Abort	No	Error codes (if any) and source are model dependent.[4]
19	#XF	SIMD Floating-Point Exception	Fault	No	SSE/SSE2/SSE3 floating-point instructions[5]
20-31	—	Intel reserved. Do not use.			
32-255	—	User Defined (Non-reserved) Interrupts	Interrupt		External interrupt or INT *n* instruction.

NOTES:

1. The UD2 instruction was introduced in the Pentium Pro processor.
2. IA-32 processors after the Intel386 processor do not generate this exception.
3. This exception was introduced in the Intel486 processor.
4. This exception was introduced in the Pentium processor and enhanced in the P6 family processors.
5. This exception was introduced in the Pentium III processor.

不能够被IVT覆盖的中断源

- 有些类类型的中断时不能被IVT的机制cover的，有以下一些信号。见IA32手册的以下语句：

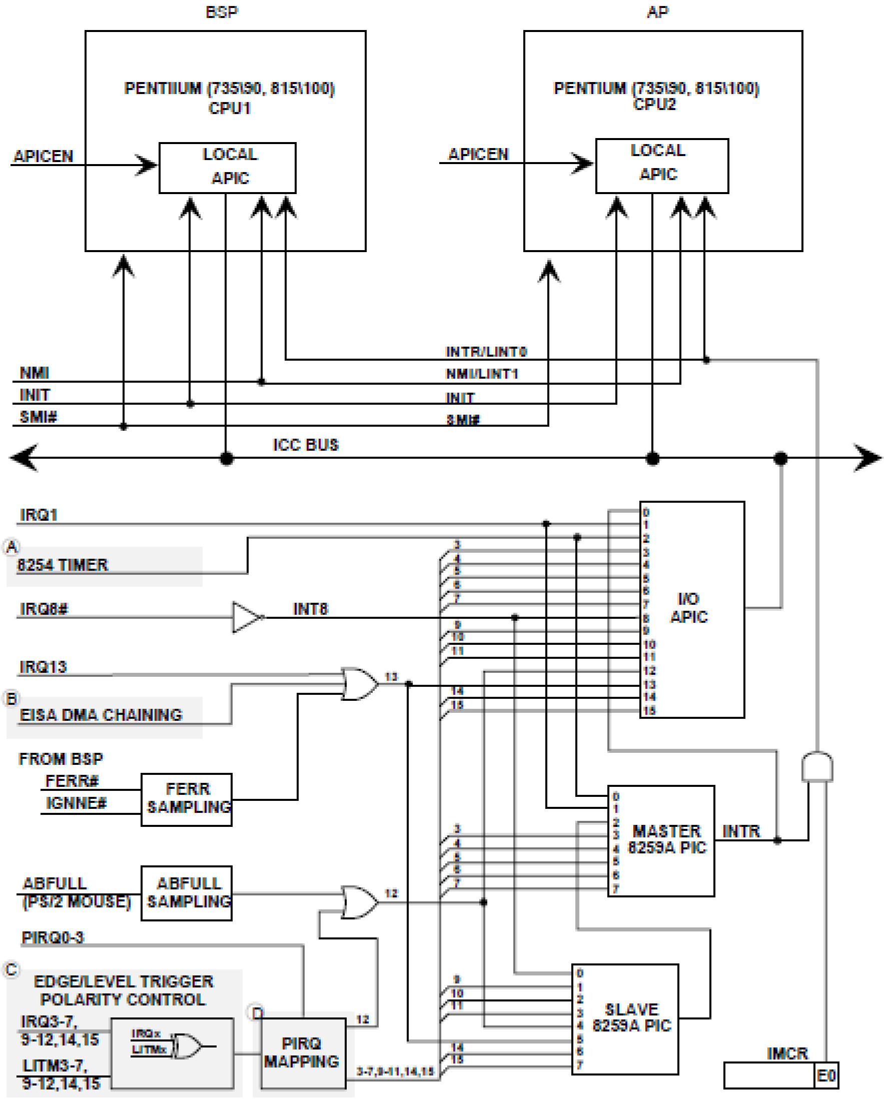


Note that several other pins on the processor cause a processor interrupt to occur; however, these interrupts are not handled by the interrupt and exception mechanism described in this chapter. These pins include the RESET#, FLUSH#, STPCLK#, SMI#, R/S#, and INIT# pins. Which of these pins are included on a particular IA-32 processor is implementation dependent. The functions of these pins are described in the data books for the individual processors. The SMI# pin is also described in Chapter 13, *System Management*.

CPU连接外部中断的方式：PIC和APIC

- PIC: 8259 IO mode。IO模式操作。
  - IO端口为20~ 21 A0 ~ A1 4D0 ~ 4D1
  - ICMR (0x22,0x23)
    - 供PIC/APIC模式切换用。
  - PIC模式下 IRQ0 ~ 7 对应INT8 ~ F IRQ8 ~ F对应INT70 ~ 77
    - 其实从软件来说，公用一个IRQ并不是问题，只要能把ISR (interrupt service routine)级联就可以了。
  - PIC模式，中断只会送到CPU0。
- APIC: MMIO方式操作。只有在Pentium之后才有的模式。

一个PIC/APIC混用硬件连线图

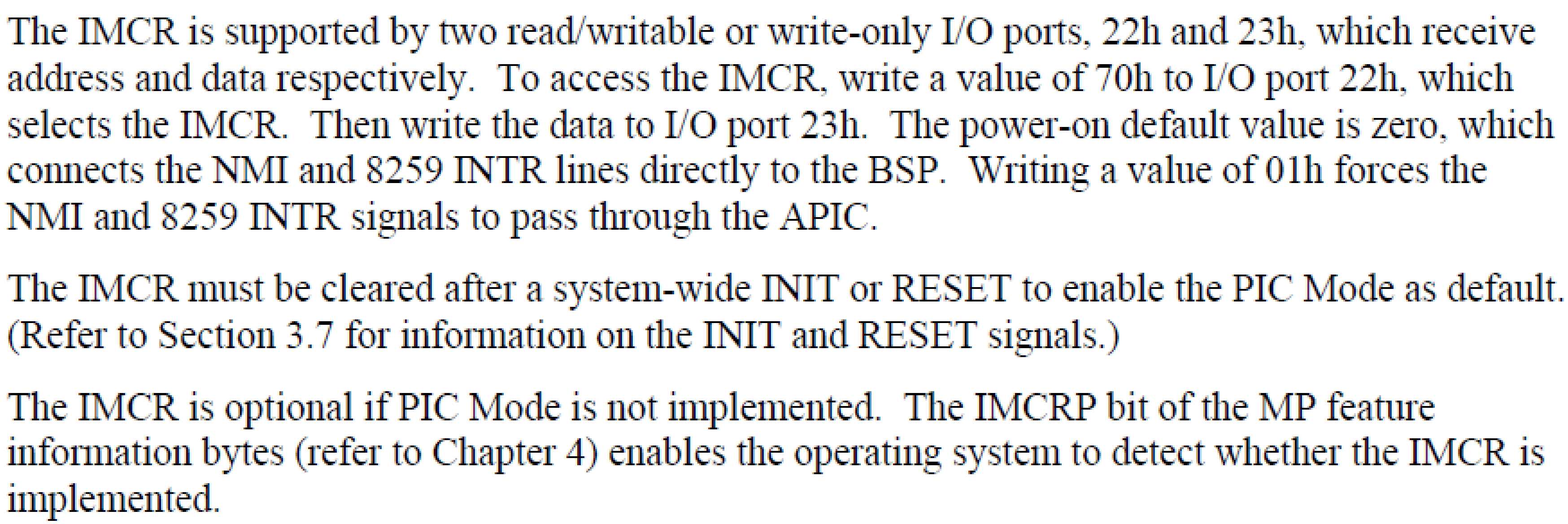


PIC/APIC都存在情况下中断实现的方法

- PIC mode: 只有一个CPU得到中断，PC/AT兼容模式。
  - PIC的总开关连线到CPU的INTR pin上面。
- Virtual Wire Mode：把IOAPIC作为一个虚拟的硬件连线连接到LAPIC，其他的和PIC模式相同。
  - PIC的总开关盒LAPIC的LINTIN0相连，完全跳过IOAPIC。
  - 或者PIC的总开关盒IOAPIC的IRQ0相连。然后IOAPIC和CPU的INTR pin相连。
- IOAPIC模式。完全利用SMP优势。
- 硬件必须实现前两种方式的一种，在OS load之前供BIOS配置整个系统。然后在切换到IOAPIC模式。

PIC/APIC模式的切换

- 其关键点在IMCR寄存器，port22 和port 23。
  - 0x70 -> 0x22, 0x01 -> 0x23就把INTR pin切换到了LAPIC上面。
- IMCR存在不存在在MP Table里面有IMCRP位说明，如果不存在，说明系统用的是Virtual wire mode之一。不用切换。



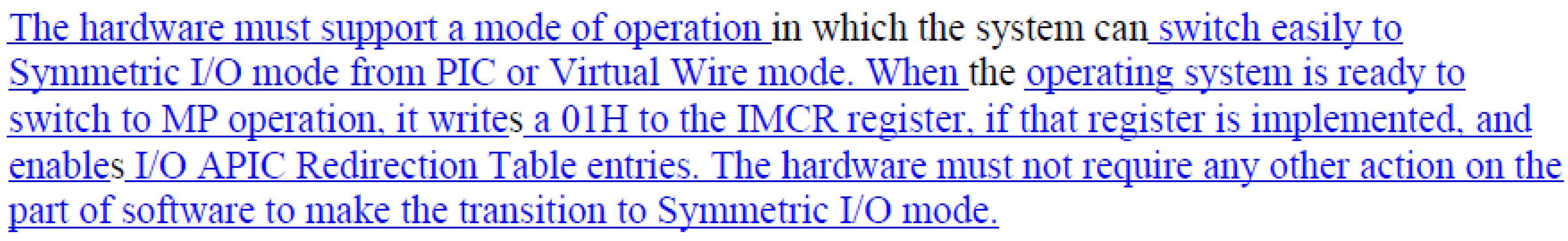
The IMCR is supported by two read/writable or write-only I/O ports, 22h and 23h, which receive address and data respectively. To access the IMCR, write a value of 70h to I/O port 22h, which selects the IMCR. Then write the data to I/O port 23h. The power-on default value is zero, which connects the NMI and 8259 INTR lines directly to the BSP. Writing a value of 01h forces the NMI and 8259 INTR signals to pass through the APIC.

The IMCR must be cleared after a system-wide INIT or RESET to enable the PIC Mode as default. (Refer to Section 3.7 for information on the INIT and RESET signals.)

The IMCR is optional if PIC Mode is not implemented. The IMCRP bit of the MP feature information bytes (refer to Chapter 4) enables the operating system to detect whether the IMCR is implemented.

关于PIC/APIC切换的说明如下

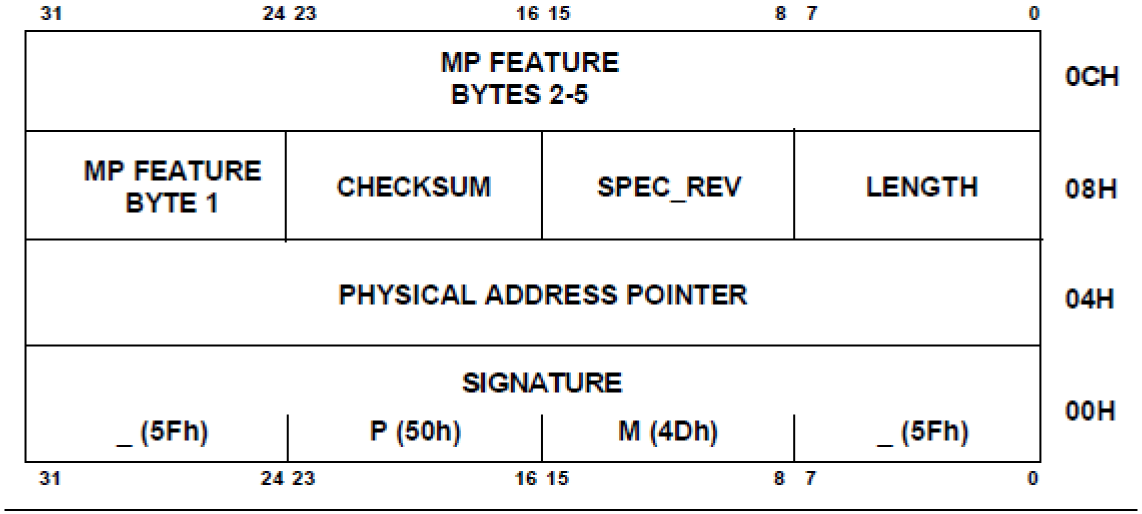
- Copy from MP Spec
- 这个时候，不需要显式切换，就把8259的所有中断源disable/mask就可以了。



The hardware must support a mode of operation in which the system can switch easily to Symmetric I/O mode from PIC or Virtual Wire mode. When the operating system is ready to switch to MP operation, it writes a 01H to the IMCR register, if that register is implemented, and enables I/O APIC Redirection Table entries. The hardware must not require any other action on the part of software to make the transition to Symmetric I/O mode.

MP Table和ACPI里面的APIC Table是支持APIC的关键

- MP Table索引：
- APIC Table见ACPI的规定。



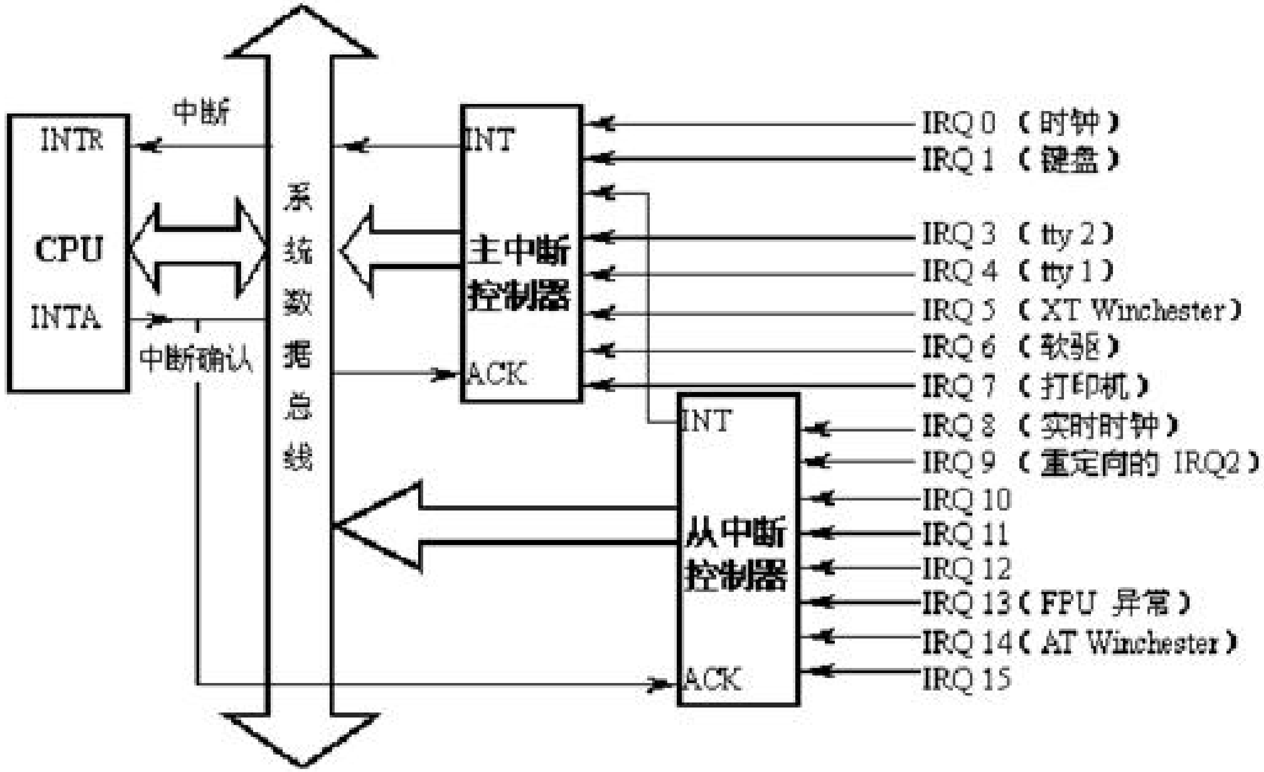
31 24	23 16	15 8	7 0	
MP FEATURE BYTES 2-5				0CH
MP FEATURE BYTE 1	CHECKSUM	SPEC_REV	LENGTH	08H
PHYSICAL ADDRESS POINTER				04H
SIGNATURE _ (5Fh)	P (50h)	M (4Dh)	_ (5Fh)	00H
31 24	23 16	15 8	7 0	

Figure 4-2. MP Floating Pointer Structure

一般情况下PIC模式的IRQ表

IRQ0	System timer
IRQ1	keyboard
IRQ2	级联
IRQ3	COM2
IRQ4	COM1
IRQ5	
IRQ6	
IRQ7	PRT1
IRQ8	CMOS timer
IRQ9	一般可以作SCI
IRQ10	
IRQ11	
IRQ12	PS2 mouse
IRQ13	Math processor exception
IRQ14	Hard disk
IRQ15	

一般情况下PIC模式的IRQ图



Local APIC和IO APIC

- APIC系统由Local APIC和IOAPIC组成。
- Local APIC供cpu之间通信。
- IOAPIC供系统和外部通信。
- IOAPIC一般是内嵌入LPC Bridge里面。也就是说，中断的BDF号码都已经定好了。
- 南桥可能会支持外部IOAPIC，比如Calpella的Ibexpeak就会支持PCIE bridge下面的IOApic，因为EOI信息可以通过PCIE Bridge。
- LAPIC和IOAPIC的项的优先级靠INT vector (0 ~ FF)的头一个数字决定，越大越高。
  - 比如INT Vector 0x2?的优先级就是2

IOAPIC的routing

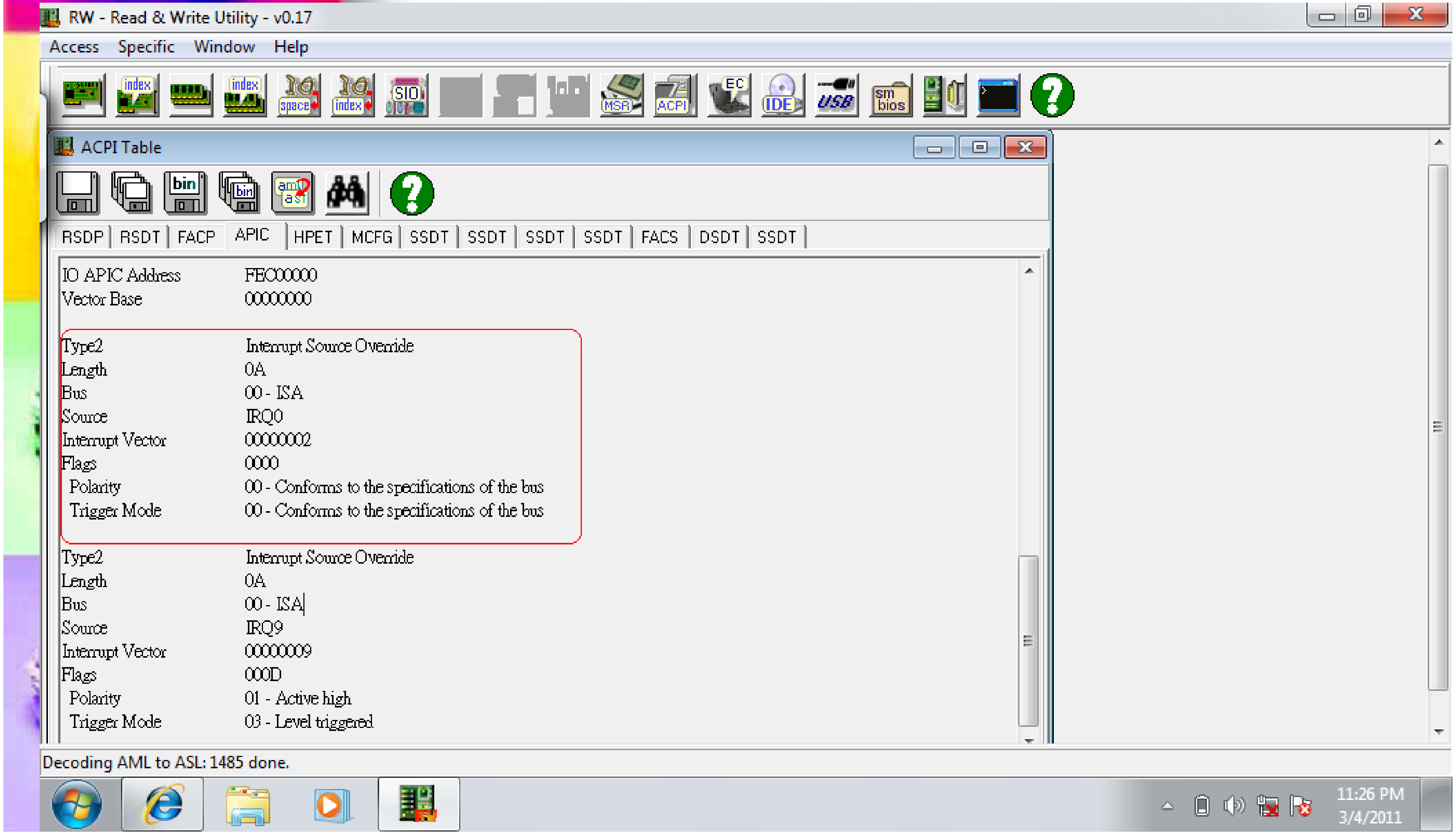
- 每一个南桥的实现可以有不同，一般情况下，南桥spec会详细说明。下一页只是一个例子予以说明。
- IOAPIC是辅助CPU的，软件要根据硬件的连线，建立IVT。

IRQ0	在virtual wire mode B里面，可以和8259级联。
IRQ1	
IRQ2	System timer
IRQ3	
IRQ4	
IRQ5	
IRQ6	
IRQ7	
IRQ8	CMOS timer
IRQ9	一般SCI
IRQ10	
IRQ11	
IRQ12	PS2 mouse
IRQ13	Math processor exception
IRQ14	Hard disk
IRQ15	Hard disk
IRQ16	PIRQA#
IRQ17	PIRQB#
IRQ18	PIRQC#
IRQ19	PIRQD#
IRQ20	PIRQE#
IRQ21	PIRQF#
IRQ22	PIRQG#
IRQ23	PIRQH#

IOAPIC和PIC的兼容性的保证

- 从上图可以看出，IOAPIC的IRQ2接了PIC的IRQ0，是PIT Timer。
- 这是由ACPI里面的“APIC”表指明了的，其中有type2，就是Interrupt Source Override.见下图。说明把IRQ0重定向到了IRQ2。
- 除了这个表规定的，其他的IRQ0～15一定要和PIC的保持一致。

IOAPIC和PIC的兼容性的保证

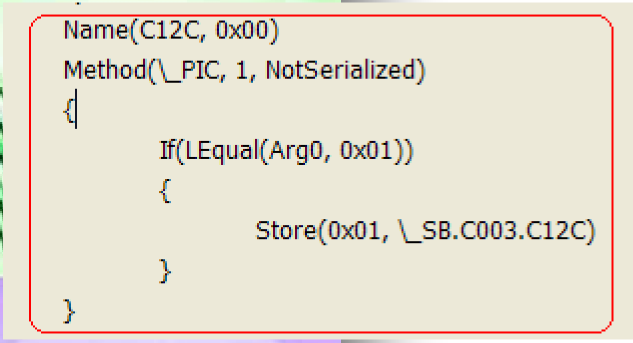

RW - Read & Write Utility - v0.17
Access Specific Window Help
ACPI Table
RSDP RSDT FACP APIC HPET MCFG SSDT SSDT SSDT SSDT FACS DSDT SSDT
11:26 PM
3/4/2011

```
IO APIC Address        FEC00000
Vector Base            00000000

Type2                  Interrupt Source Override
Length                 0A
Bus                    00 - ISA
Source                 IRQ0
Interrupt Vector       00000002
Flags                  0000
 Polarity              00 - Conforms to the specifications of the bus
 Trigger Mode          00 - Conforms to the specifications of the bus

Type2                  Interrupt Source Override
Length                 0A
Bus                    00 - ISA
Source                 IRQ9
Interrupt Vector       00000009
Flags                  000D
 Polarity              01 - Active high
 Trigger Mode          03 - Level triggered
```

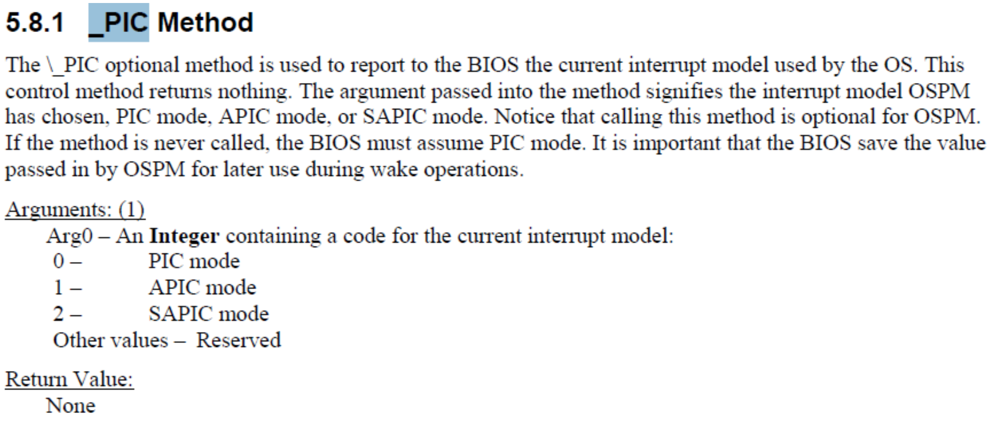
Decoding AML to ASL: 1485 done.

IOAPIC和PIC模式的选择

- ACPI1.0曾经规定FACP的44 byte指明是何种模式，从2.0开始就不用了。是OS来决定的。见ACPI Spec和样例ASL代码。所以OS不会用这个位再来判断中断模式。
- _PIC由OS执行，通知BIOS它的选择。



```
Name(C12C, 0x00)
Method(\_PIC, 1, NotSerialized)
{
        If(LEqual(Arg0, 0x01))
        {
                Store(0x01, \_SB.C003.C12C)
        }
}
```



5.8.1 _PIC Method

The _PIC optional method is used to report to the BIOS the current interrupt model used by the OS. This control method returns nothing. The argument passed into the method signifies the interrupt model OSPM has chosen, PIC mode, APIC mode, or SAPIC mode. Notice that calling this method is optional for OSPM. If the method is never called, the BIOS must assume PIC mode. It is important that the BIOS save the value passed in by OSPM for later use during wake operations.

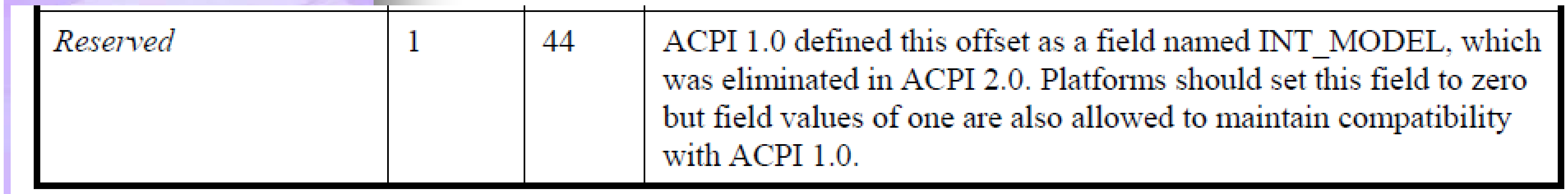
Arguments: (1)

Arg0 – An **Integer** containing a code for the current interrupt model:

0 –	PIC mode
1 –	APIC mode
2 –	SAPIC mode
Other values –	Reserved

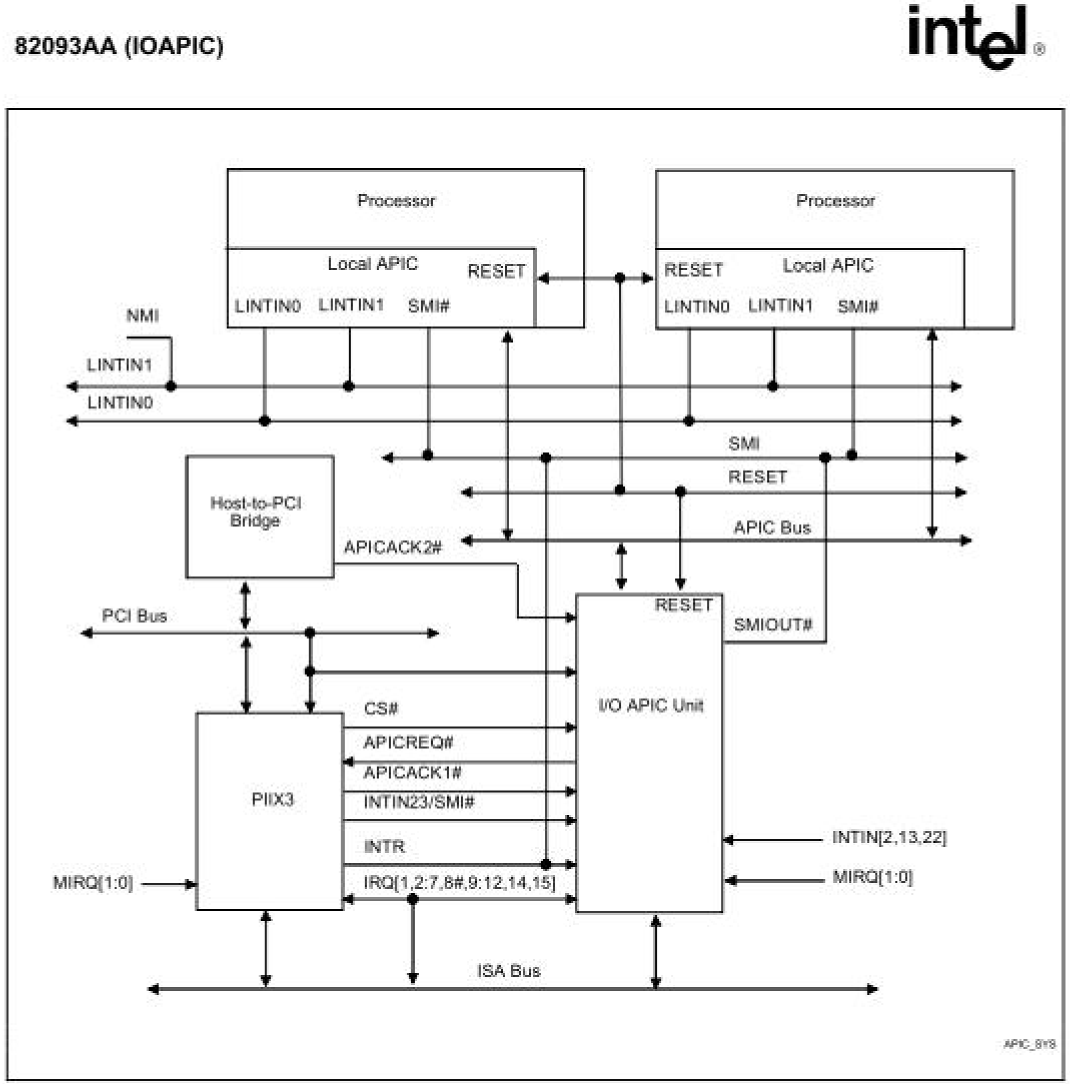
Return Value:

None



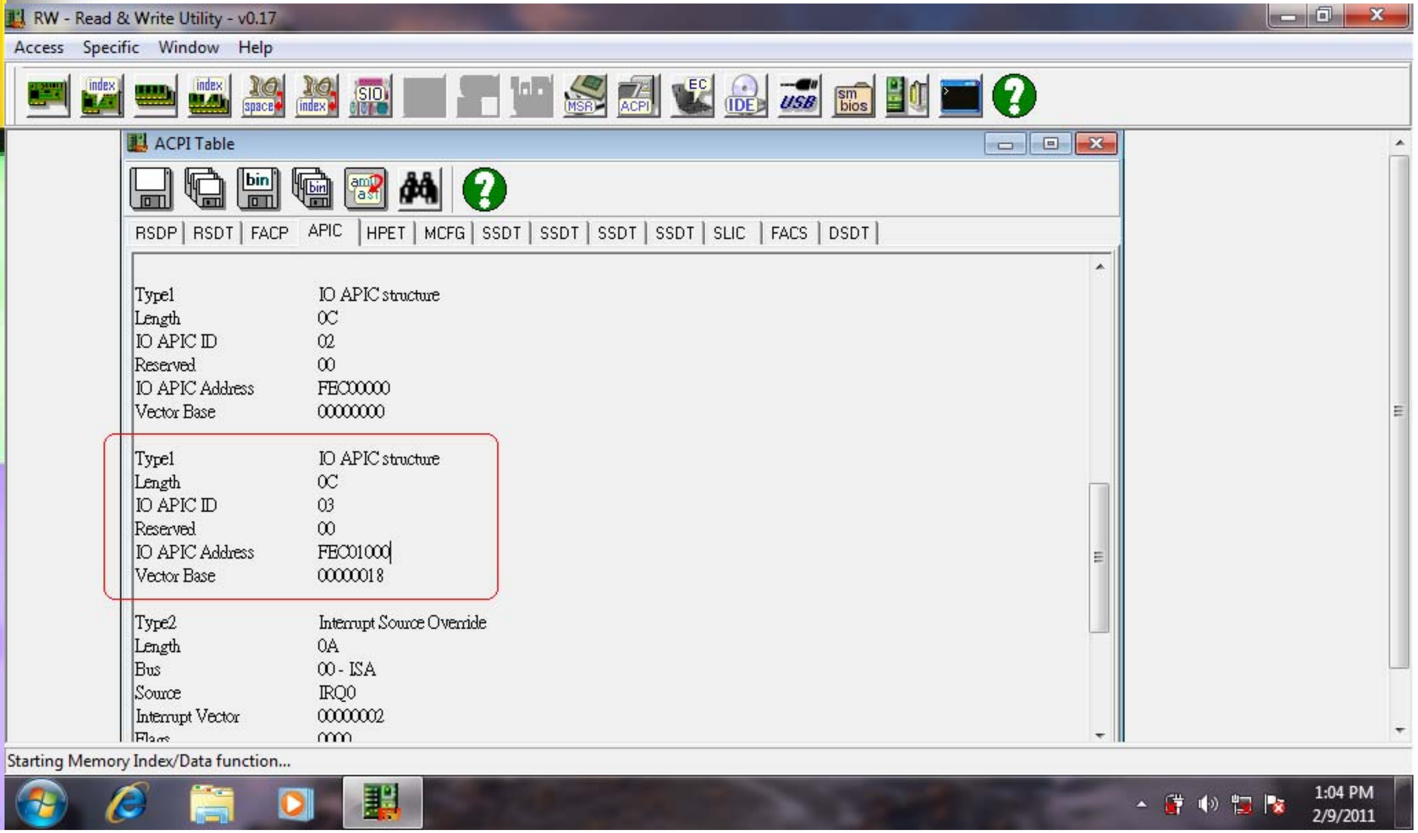
Reserved	1	44	ACPI 1.0 defined this offset as a field named INT_MODEL, which was eliminated in ACPI 2.0. Platforms should set this field to zero but field values of one are also allowed to maintain compatibility with ACPI 1.0.

IOAPIC示意图


82093AA (IOAPIC)
Processor
Local APIC
RESET
LINTIN0
LINTIN1
SMI#
Processor
RESET
Local APIC
LINTIN0
LINTIN1
SMI#
NMI
LINTIN1
LINTIN0
SMI
RESET
APIC Bus
Host-to-PCI Bridge
APICACK2#
PCI Bus
RESET
SMIOUT#
I/O APIC Unit
CS#
APICREQ#
APICACK1#
INTIN23/SMI#
INTR
IRQ[1,2:7,8#,9:12,14,15]
PIIX3
MIRQ[1:0]
INTIN[2,13,22]
MIRQ[1:0]
ISA Bus
APIC_SYS

多IOAPIC的情况

- 有的情况下，系统可以连接多个IOAPIC，需要在mptable和acpi的apic table里面加相应的项目，然后OS就可以认到。
- 多个IOAPIC的情况下，IRQ number是连续的，第二个IOAPIC的IRQ0与系统中上一个IOAPIC的最后一个IRQ是连续的。
  - 一个例子见下图。第二个IOAPIC将从0x18开始。


RW - Read & Write Utility - v0.17
Access Specific Window Help
ACPI Table
RSDP | RSDT | FACP | APIC | HPET | MCFG | SSDT | SSDT | SSDT | SSDT | SLIC | FACS | DSDT |

```
Type1               IO APIC structure
Length              0C
IO APIC ID          02
Reserved            00
IO APIC Address     FEC00000
Vector Base         00000000

Type1               IO APIC structure
Length              0C
IO APIC ID          03
Reserved            00
IO APIC Address     FEC01000
Vector Base         00000018

Type2               Interrupt Source Override
Length              0A
Bus                 00 - ISA
Source              IRQ0
Interrupt Vector    00000002
Flags               0000
```

Starting Memory Index/Data function...

Local APIC

- Local APIC在不在由CPUID指令查到。
- 默认位置是FEE00000, 4K大小。
- MSR IA32_APIC_BASE指明基地址。
- Local APIC是MMIO，不像IOAPIC是MMIO Index/Data这样子的。

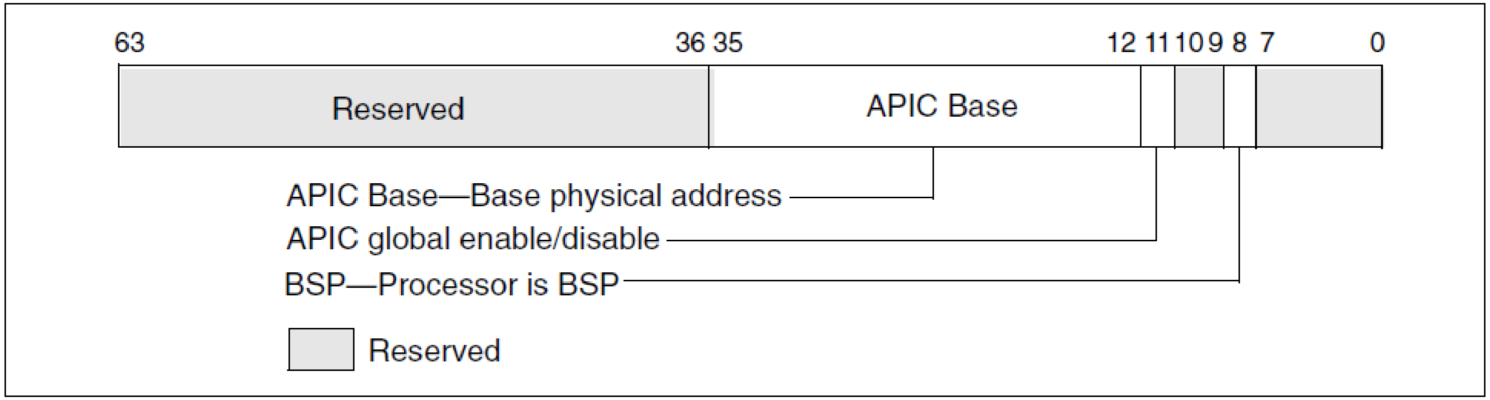
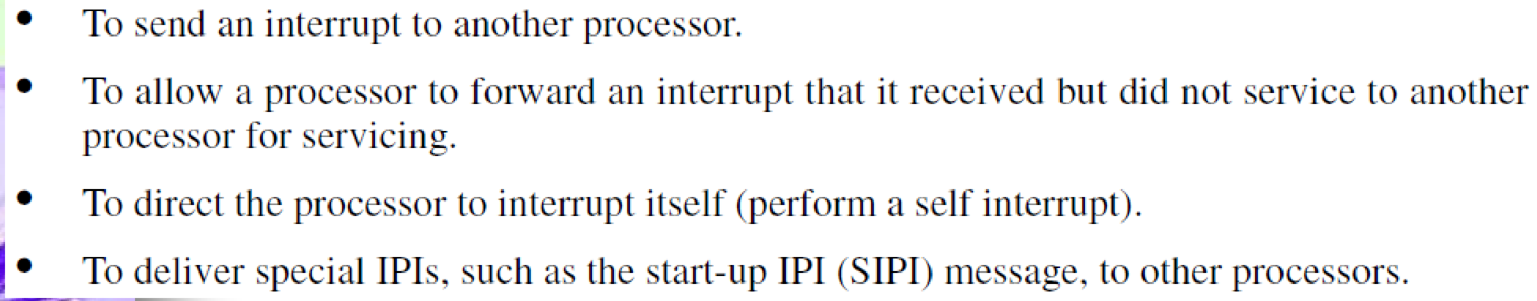


Figure 8-5. IA32_APIC_BASE MSR

IPI的基本作用

- IA32 Manual有详细解释：
  - INIT IPI 和 STARTUP IPI



- To send an interrupt to another processor.
- To allow a processor to forward an interrupt that it received but did not service to another processor for servicing.
- To direct the processor to interrupt itself (perform a self interrupt).
- To deliver special IPIs, such as the start-up IPI (SIPI) message, to other processors.

LAPIC的一些寄存器

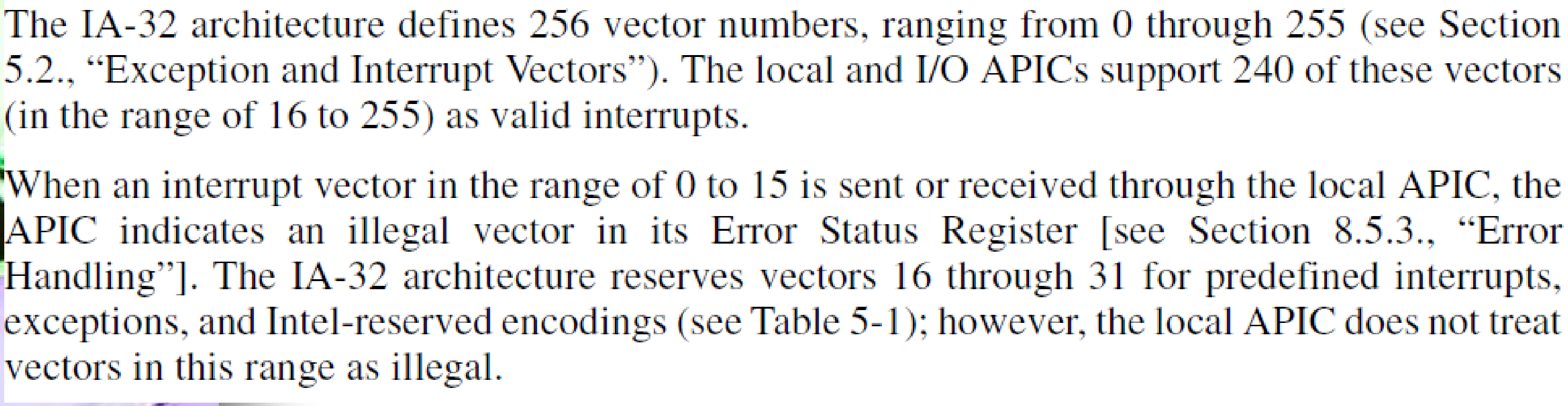
- task priority (FEE00080)
- timer interrupt vector (FEE00320)
- performance counter interrupt (FEE00340)
- local interrupt 0 (FEE00350): normal external interrupts
- local interrupt 1 (FEE00360) normal NMI processing
- error interrupt (FEE00370)
- spurious interrupt (FEE000F0)

IOAPIC

- MMIO FEC00000 to FEC00040
- IND FEC00000 (Index)
- DATA FEC00010 (Data)
  - 通过Index/Data一共指向一个大小为256 dword的空间。
- EOI FEC00040，是供cpu通知南桥，interrupt已经处理完毕。
- 其中有指定每一个int对应的vector的设定。Interrupt vector指定了interrupt service的地址。
  - 这是中断向量表和中断联系起来的核心。ISR和IRQ number的联系。
- IOAPIC是控制那些中断发往CPU的关键所在。是辅助CPU的，因为CPU只有一个ExtInt连接到外部。
- 可以用Windbg的!ioapic命令来查看。
- IOAPIC中的一个个项是RTE：register redirection table entries

IOAPIC和Local APIC能使用的vector号码

- 只能用16到255
  - 如果不符合这个规则，APIC会报错。
- PIC没有这个规定。

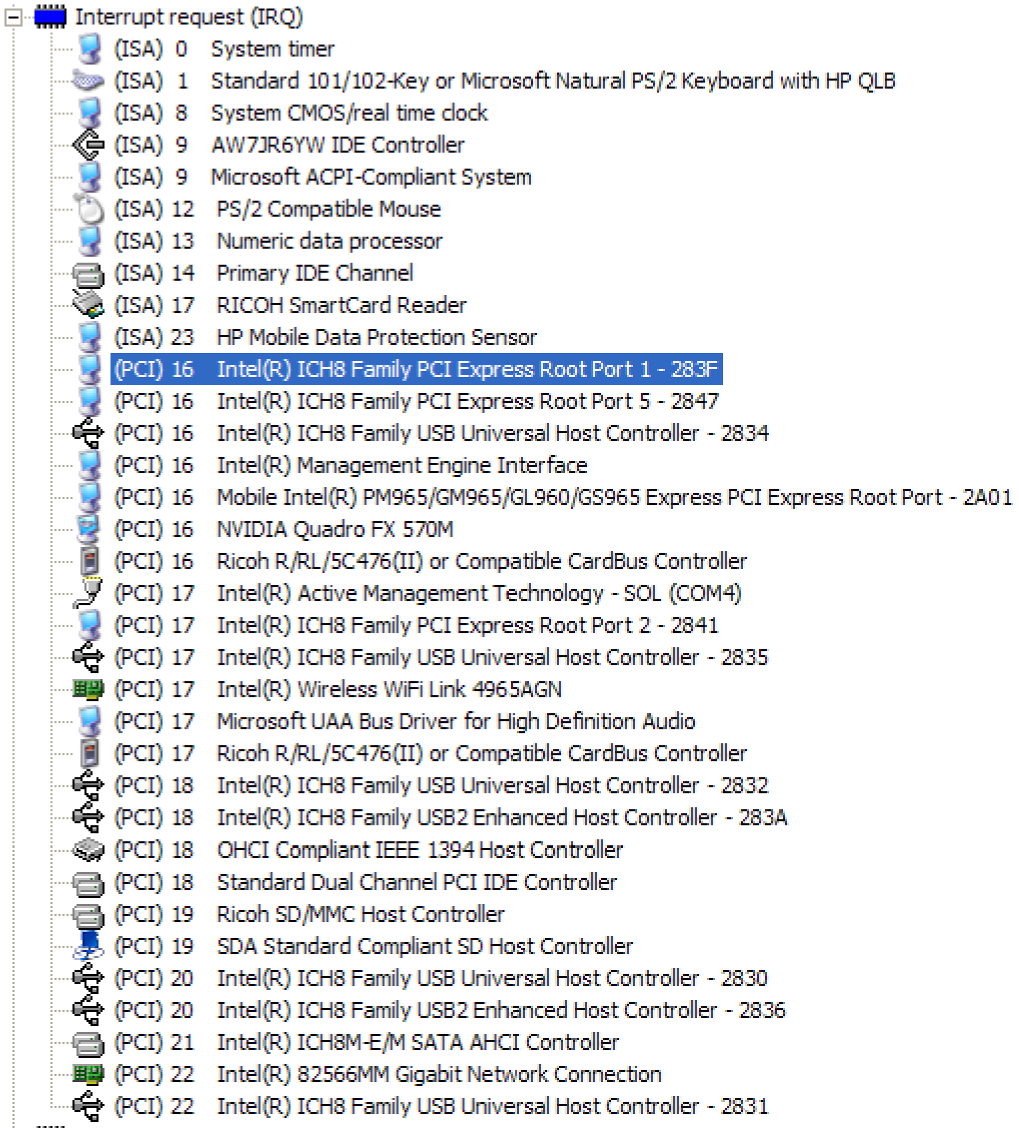


The IA-32 architecture defines 256 vector numbers, ranging from 0 through 255 (see Section 5.2., “Exception and Interrupt Vectors”). The local and I/O APICs support 240 of these vectors (in the range of 16 to 255) as valid interrupts.

When an interrupt vector in the range of 0 to 15 is sent or received through the local APIC, the APIC indicates an illegal vector in its Error Status Register [see Section 8.5.3., “Error Handling”]. The IA-32 architecture reserves vectors 16 through 31 for predefined interrupts, exceptions, and Intel-reserved encodings (see Table 5-1); however, the local APIC does not treat vectors in this range as illegal.

IOAPIC的一个实际例子

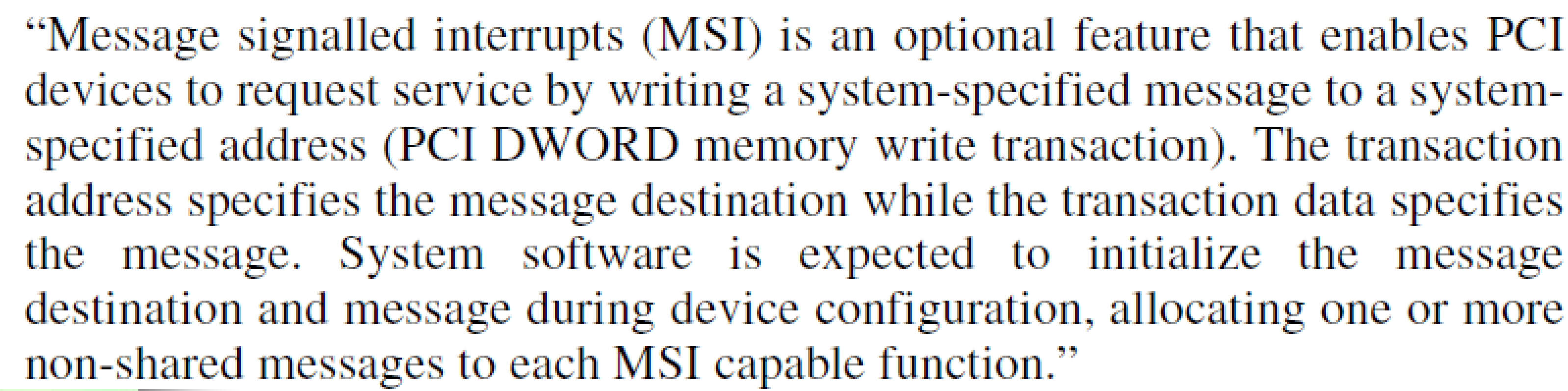
- PCI的中断用的是16以上，说明是IOAPIC模式。
- 这里所谓的ISA type指的是用ASL声明的设备作用的中断。，因此有可能数字是大于15的。



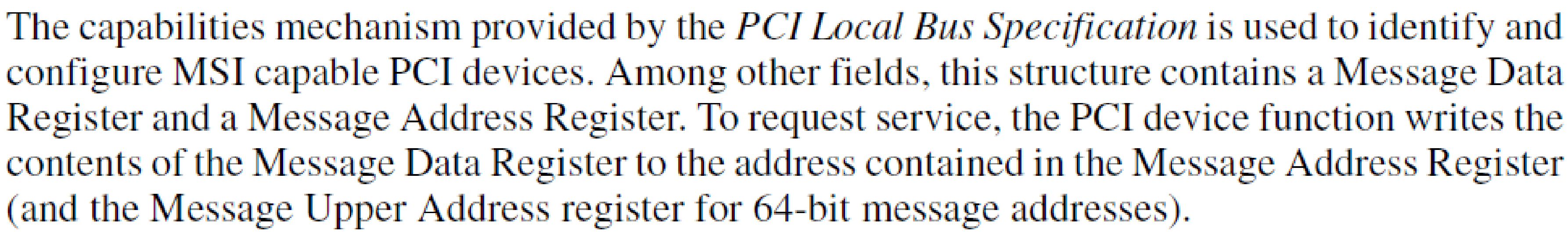
```
Interrupt request (IRQ)
    (ISA)  0   System timer
    (ISA)  1   Standard 101/102-Key or Microsoft Natural PS/2 Keyboard with HP QLB
    (ISA)  8   System CMOS/real time clock
    (ISA)  9   AW7JR6YW IDE Controller
    (ISA)  9   Microsoft ACPI-Compliant System
    (ISA) 12   PS/2 Compatible Mouse
    (ISA) 13   Numeric data processor
    (ISA) 14   Primary IDE Channel
    (ISA) 17   RICOH SmartCard Reader
    (ISA) 23   HP Mobile Data Protection Sensor
    (PCI) 16   Intel(R) ICH8 Family PCI Express Root Port 1 - 283F
    (PCI) 16   Intel(R) ICH8 Family PCI Express Root Port 5 - 2847
    (PCI) 16   Intel(R) ICH8 Family USB Universal Host Controller - 2834
    (PCI) 16   Intel(R) Management Engine Interface
    (PCI) 16   Mobile Intel(R) PM965/GM965/GL960/GS965 Express PCI Express Root Port - 2A01
    (PCI) 16   NVIDIA Quadro FX 570M
    (PCI) 16   Ricoh R/RL/5C476(II) or Compatible CardBus Controller
    (PCI) 17   Intel(R) Active Management Technology - SOL (COM4)
    (PCI) 17   Intel(R) ICH8 Family PCI Express Root Port 2 - 2841
    (PCI) 17   Intel(R) ICH8 Family USB Universal Host Controller - 2835
    (PCI) 17   Intel(R) Wireless WiFi Link 4965AGN
    (PCI) 17   Microsoft UAA Bus Driver for High Definition Audio
    (PCI) 17   Ricoh R/RL/5C476(II) or Compatible CardBus Controller
    (PCI) 18   Intel(R) ICH8 Family USB Universal Host Controller - 2832
    (PCI) 18   Intel(R) ICH8 Family USB2 Enhanced Host Controller - 283A
    (PCI) 18   OHCI Compliant IEEE 1394 Host Controller
    (PCI) 18   Standard Dual Channel PCI IDE Controller
    (PCI) 19   Ricoh SD/MMC Host Controller
    (PCI) 19   SDA Standard Compliant SD Host Controller
    (PCI) 20   Intel(R) ICH8 Family USB Universal Host Controller - 2830
    (PCI) 20   Intel(R) ICH8 Family USB2 Enhanced Host Controller - 2836
    (PCI) 21   Intel(R) ICH8M-E/M SATA AHCI Controller
    (PCI) 22   Intel(R) 82566MM Gigabit Network Connection
    (PCI) 22   Intel(R) ICH8 Family USB Universal Host Controller - 2831
```

MSI: Message Signal Interrupts

- 由PCI Spec引入
- 有这个功能的pci device会在config space里面说明地址和数据。
- 这个功能对PCI Express就是必须的了，因为PCI Express并没有独立的中断pin。
- 写的这个地址并不是直接写到了CPU，而是要经过Chipset的过滤，才决定发什么IRQ给CPU。



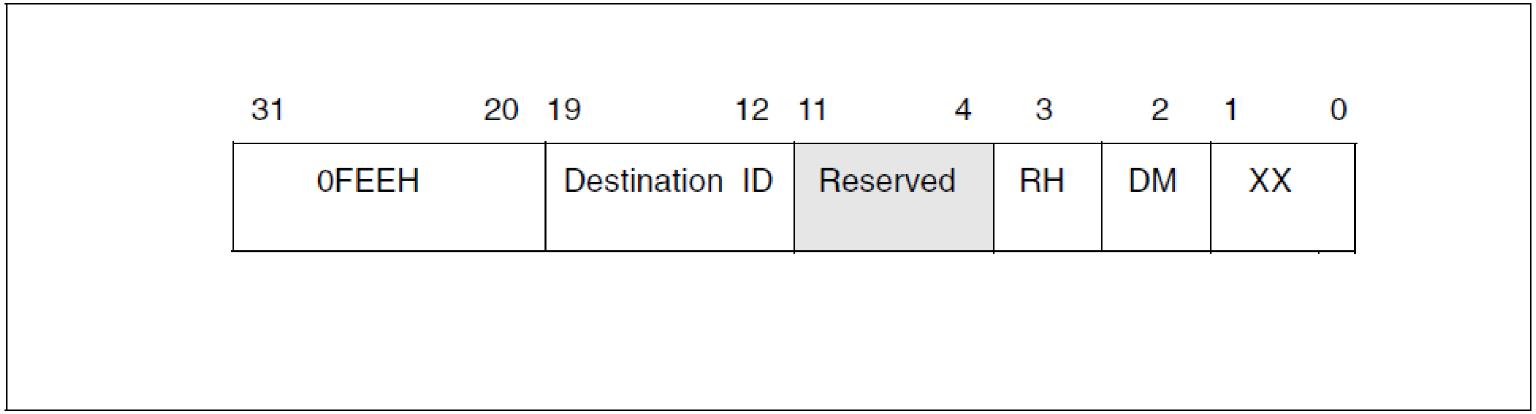
“Message signalled interrupts (MSI) is an optional feature that enables PCI devices to request service by writing a system-specified message to a system-specified address (PCI DWORD memory write transaction). The transaction address specifies the message destination while the transaction data specifies the message. System software is expected to initialize the message destination and message during device configuration, allocating one or more non-shared messages to each MSI capable function.”



The capabilities mechanism provided by the *PCI Local Bus Specification* is used to identify and configure MSI capable PCI devices. Among other fields, this structure contains a Message Data Register and a Message Address Register. To request service, the PCI device function writes the contents of the Message Data Register to the address contained in the Message Address Register (and the Message Upper Address register for 64-bit message addresses).

MSI格式

- 地址寄存器格式
  - 可以看出地址其实在LAPIC内存空间。

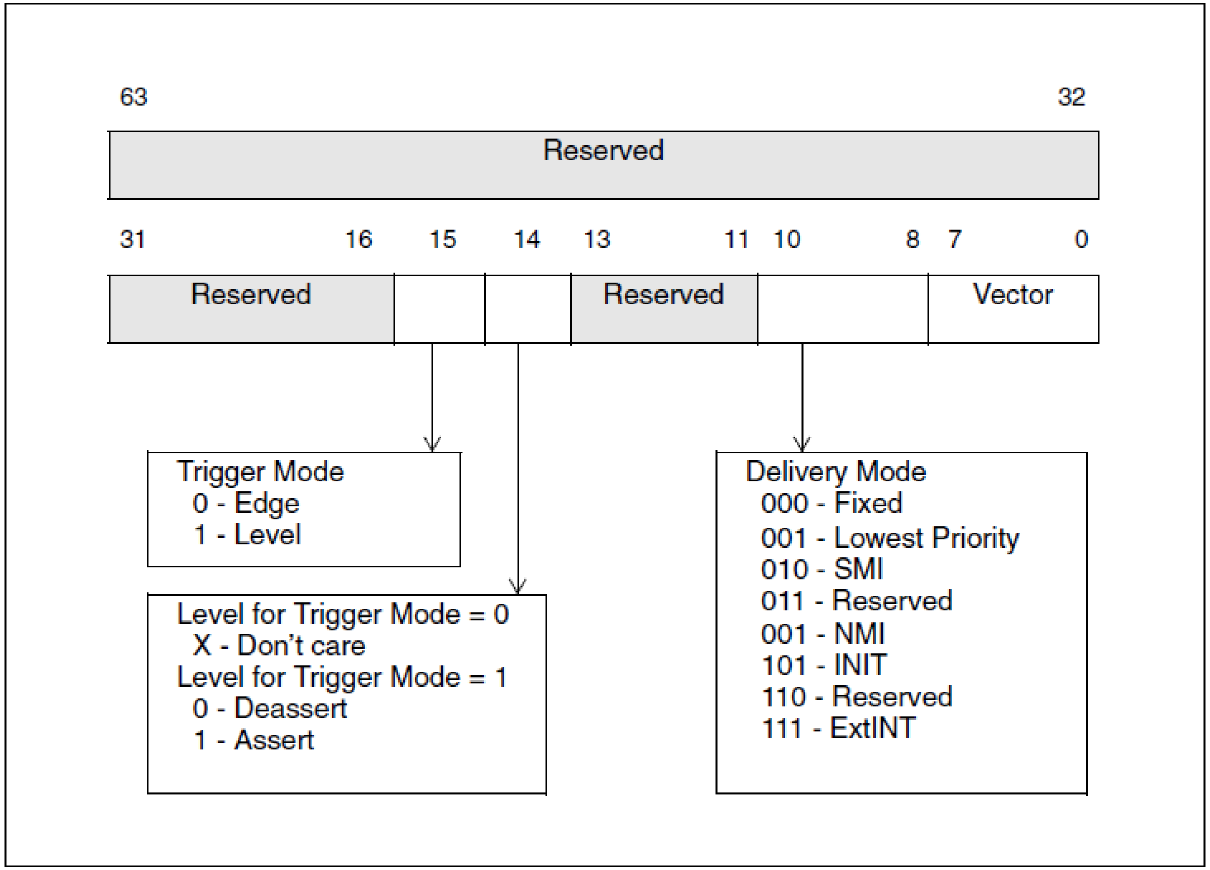


31–20	19–12	11–4	3	2	1–0
0FEEH	Destination ID	Reserved	RH	DM	XX

Figure 8-23. Layout of the MSI Message Address Register

MSI格式

- 数据寄存器格式



63 32

Reserved

31 16 15 14 13 11 10 8 7 0

Reserved | | | Reserved | | Vector

Trigger Mode
0 - Edge
1 - Level

Level for Trigger Mode = 0
X - Don’t care
Level for Trigger Mode = 1
0 - Deassert
1 - Assert

Delivery Mode
000 - Fixed
001 - Lowest Priority
010 - SMI
011 - Reserved
001 - NMI
101 - INIT
110 - Reserved
111 - ExtINT

Figure 8-24. Layout of the MSI Message Data Register

MSI和IOAPIC LAPIC的关系

- MSI只是产生中断的方式，APIC系统完全可以用MSI方式通知CPU中断产生，而不再用硬连线的方式。
- APIC系统是中断产生的物理实现。
- MSI格式的中断其实跳过了IOAPIC，直接给LAPIC发送了消息。

硬件中断与软件中断产生方式

- 硬件中断是随机产生的，可以有CPU的IF屏蔽 可屏蔽中断。
- 软件中断是用 INT XX产生的。
- 软件中断（异常）可以分为
  - Fault
  - Trap
  - Abort

再说外部中断的种类

- NMI
- SMI
- INT，一般中断，有时也写作ExtINT.（只有这种中断被IVT响应。）
  - SCI，是一个特殊的INT，一般是IRQ9。
- INIT是特殊的初始化中断，给cpu发出的。

OS拿到PCI设备中断号的方法

- PCI IRQ table，这个比较老，是win98和win95用的。
- 新的OS都用acpi里面的_PRT来拿到设备的中断号。

SMI SCI和ACPI的关系

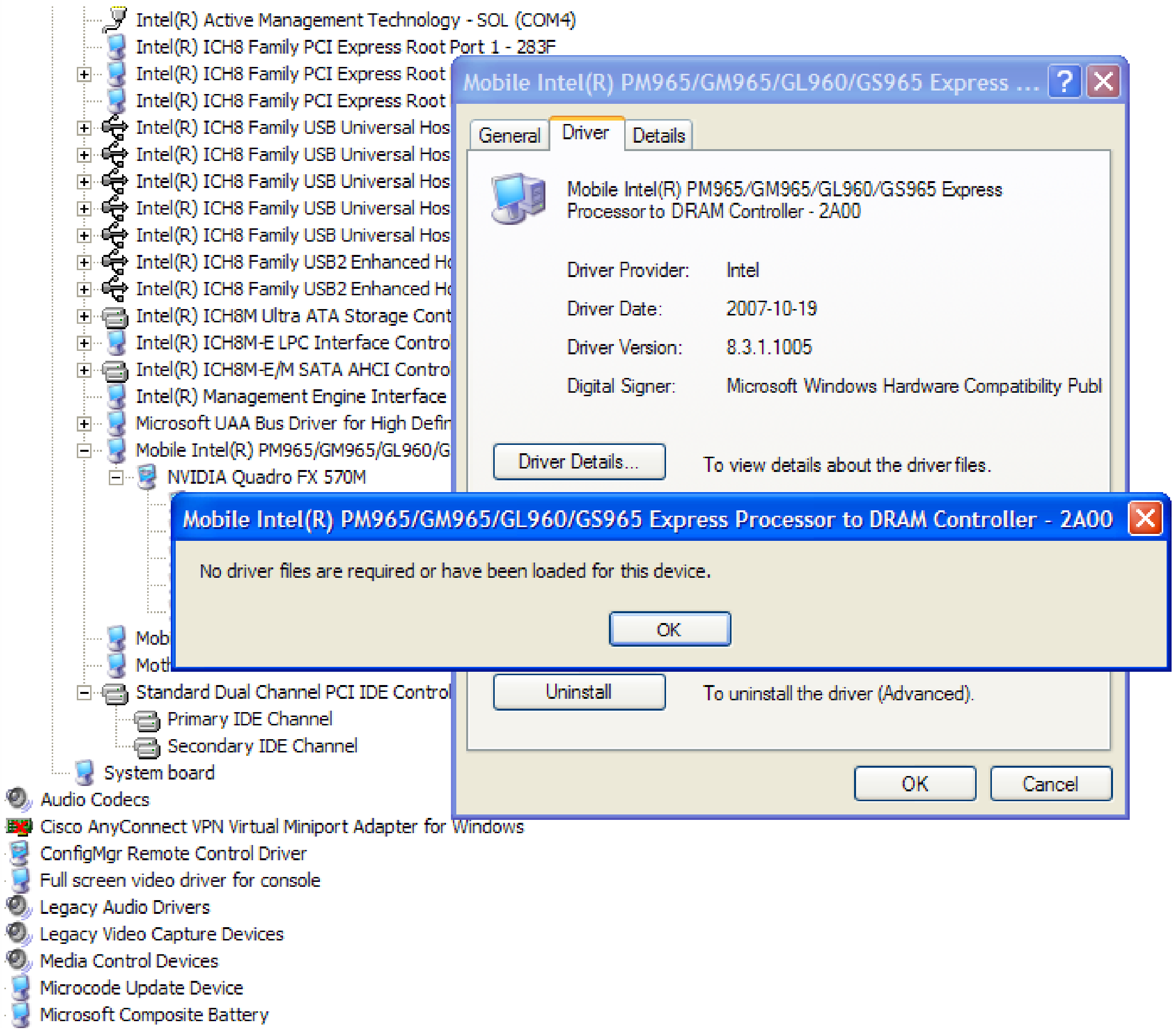
- 一般ec会把smi的输出接到南桥的一个具有smi功能的gpi上。
  - 软件同步SMI是通过写port B2来达到目的的，南桥收到B2的write之后，会拉低SMI#来让CPU进入SMM mode。
- Sci会接到南桥的具有sci功能的gpi上面。
- 这些东西都在acpi spec里面有说明。
- Smi是一个特殊类型的中断，会进入smm，而sci是一般的中断，pc上面一般是INT9，sci的响应由asl来负责，包括_gpe和ec独立的那个sci（_QXX），是一个特殊的_gpe。
- 一般情况下，设备只发IRQ，只有在wake，或者PM事件发生时，才发SMI/SCI。

SERIRQ和PCIIRQ, ISAIRQ

- SERIRQ
  - Serial IRQ是一根线，和pci clock对比产生一定的波形，是设备之间传递IRQ信息的一种方式。
  - 两种mode，包括continueous mode和quiet mode，详细见下页。
- PCIIRQ可以配置，一般是A到H，它是配置可变的，就是PCI IRQ Routing.
  - PCI Header里面的3C (Interrupt Line) 3D (Interrupt Pin)起最终的说明作用。
  - 在正确的driver安装之前，那个IRQ是不可信的。安装driver之后，IRQ肯定按照_PRT的建议来的。
  - 错误的driver会不管现有的IRQ配置，因为它根本不用。这时设备就会保留PIC模式下面的中断号。其实是肯定不会用到的。
- ISAIRQ一般号码不再变。是硬连线的方式。是一种约定俗成的模式。

一些没有driver的pci设备的例子

- Host bus设备就不需要driver的。



Intel(R) Active Management Technology - SOL (COM4)
Intel(R) ICH8 Family PCI Express Root Port 1 - 283F
Intel(R) ICH8 Family PCI Express Root
Intel(R) ICH8 Family PCI Express Root
Intel(R) ICH8 Family USB Universal Hos
Intel(R) ICH8 Family USB Universal Hos
Intel(R) ICH8 Family USB Universal Hos
Intel(R) ICH8 Family USB Universal Hos
Intel(R) ICH8 Family USB Universal Hos
Intel(R) ICH8 Family USB2 Enhanced H
Intel(R) ICH8 Family USB2 Enhanced H
Intel(R) ICH8M Ultra ATA Storage Cont
Intel(R) ICH8M-E LPC Interface Contro
Intel(R) ICH8M-E/M SATA AHCI Contro
Intel(R) Management Engine Interface
Microsoft UAA Bus Driver for High Defi
Mobile Intel(R) PM965/GM965/GL960/G
NVIDIA Quadro FX 570M
Mob
Mot
Standard Dual Channel PCI IDE Contro
Primary IDE Channel
Secondary IDE Channel
System board
Audio Codecs
Cisco AnyConnect VPN Virtual Miniport Adapter for Windows
ConfigMgr Remote Control Driver
Full screen video driver for console
Legacy Audio Drivers
Legacy Video Capture Devices
Media Control Devices
Microcode Update Device
Microsoft Composite Battery

Mobile Intel(R) PM965/GM965/GL960/GS965 Express ...

Mobile Intel(R) PM965/GM965/GL960/GS965 Express Processor to DRAM Controller - 2A00

Driver Provider: Intel
Driver Date: 2007-10-19
Driver Version: 8.3.1.1005
Digital Signer: Microsoft Windows Hardware Compatibility Publ

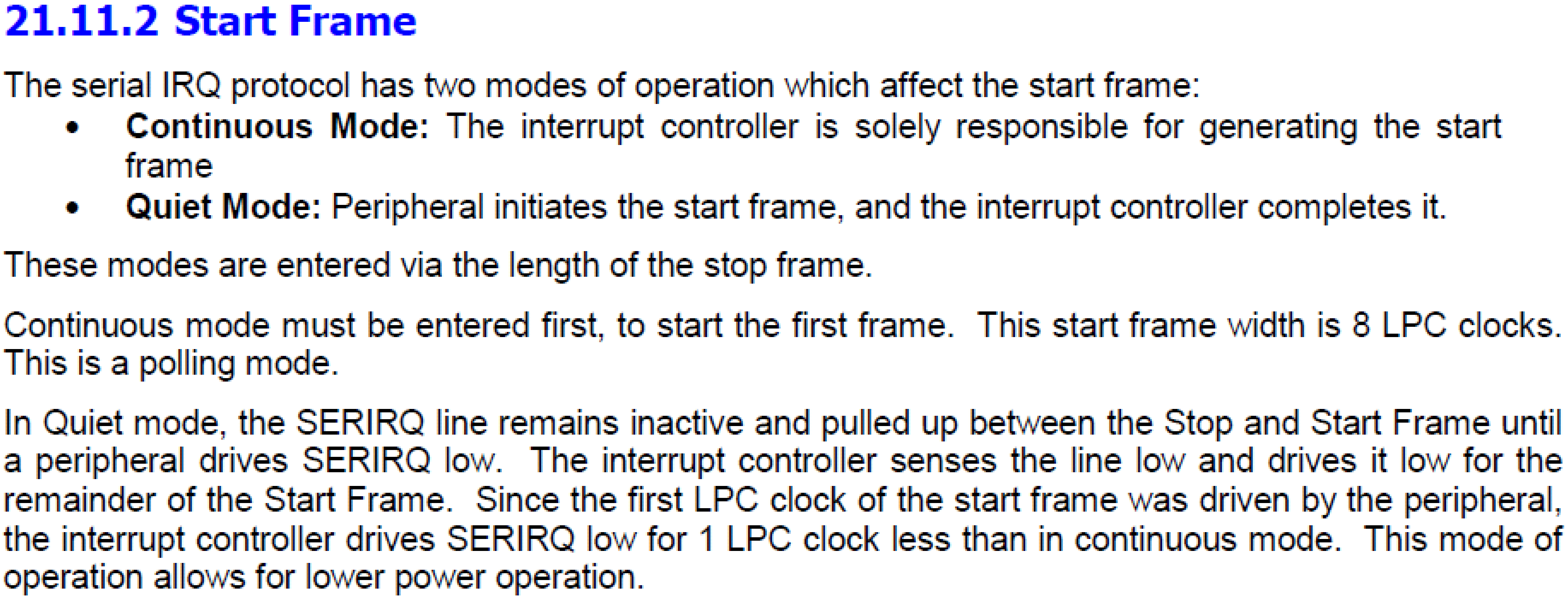
To view details about the driver files.

To uninstall the driver (Advanced).

Mobile Intel(R) PM965/GM965/GL960/GS965 Express Processor to DRAM Controller - 2A00

No driver files are required or have been loaded for this device.

SERIRQ的两种模式



21.11.2 Start Frame

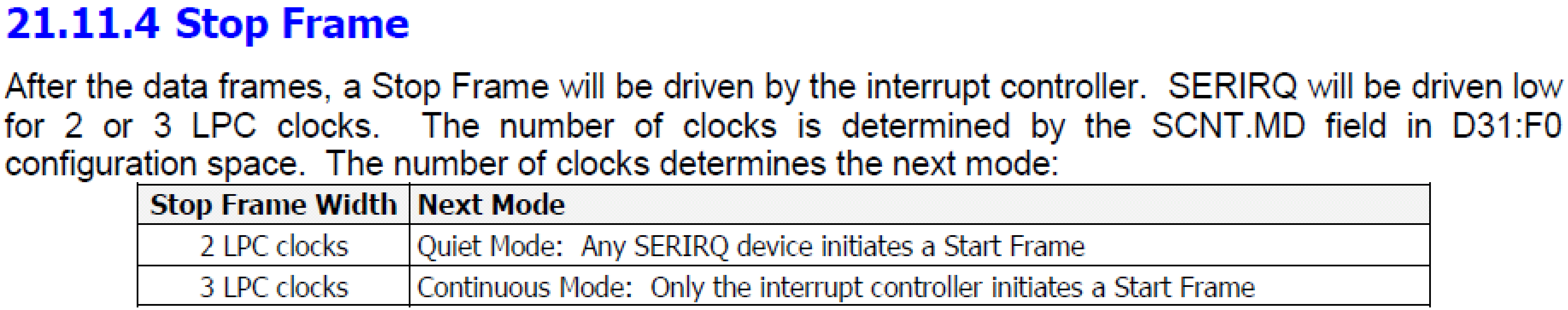
The serial IRQ protocol has two modes of operation which affect the start frame:

- **Continuous Mode:** The interrupt controller is solely responsible for generating the start frame
- **Quiet Mode:** Peripheral initiates the start frame, and the interrupt controller completes it.

These modes are entered via the length of the stop frame.

Continuous mode must be entered first, to start the first frame. This start frame width is 8 LPC clocks. This is a polling mode.

In Quiet mode, the SERIRQ line remains inactive and pulled up between the Stop and Start Frame until a peripheral drives SERIRQ low. The interrupt controller senses the line low and drives it low for the remainder of the Start Frame. Since the first LPC clock of the start frame was driven by the peripheral, the interrupt controller drives SERIRQ low for 1 LPC clock less than in continuous mode. This mode of operation allows for lower power operation.



21.11.4 Stop Frame

After the data frames, a Stop Frame will be driven by the interrupt controller. SERIRQ will be driven low for 2 or 3 LPC clocks. The number of clocks is determined by the SCNT.MD field in D31:F0 configuration space. The number of clocks determines the next mode:

Stop Frame Width	Next Mode
2 LPC clocks	Quiet Mode: Any SERIRQ device initiates a Start Frame
3 LPC clocks	Continuous Mode: Only the interrupt controller initiates a Start Frame

最后说一下物理上的中断触发方式

- 电平触发，可以共享。
  - PCI默认为此。
  - EISA是低有效电平触发。
- 边沿触发，不可共享。
  - ISA就是edge triggered.

Backup: 8254 HPET ACPI Timer的关系

- 8254
  - IO模式操作
  - Port 40 ~ 43
- HPET，想替代8254，比8254提供更多组的timer。
  - MMIO操作，控制也是南桥的MMIO直接控制。
  - 一般为FED00000基地址。
- ACPI 24-bit timer
  - 学名为PM1_TMR
  - 地址为ACPI_BASE + 08
  - 频率为14.31818Mhz / 4

Backup: 硬连线和软件模拟 (VLW)

- VLW: virtual Legacy wire，是取代南桥和cpu之间那么多连线的一种软件模拟。

Backup: 关于USB设备的中断

- 在现在的PC架构中，PCI是系统级别的总线，而USB是子总线。
- USB永远是master发起的，所以无所谓中断。
- 但是，USB控制器是有中断的。
- 这也是为什么USB鼠标等设备没有中断资源的原因。

Backup: Win7之前的系统只能安装在IA平台的原因

- 根本原因就是PC IA平台是以PCI为核心的。
- 而新的MSTN，ARM并没有PCI的架构。所以不能按照一般的Windows系统。
- 其实只要有了IRQ系统，通过MMIO就可以做成一个系统，比如现在的MSTN platform和ARM手机等等。
- Oaktrail也只是为了安装Win7才虚拟了一个PCI架构出来。

Backup: SMP架构研究

- SMP Symmetric Multi Processor的symmetric表现在：
  - 所有的CPU share同一个内存空间。
  - 所有的CPU share同一个IO系统。
- 但是在Boot up和shutdown的时候还是分BSP和AP的。