Advanced Programmable Interrupt Controller

In computing, Intel's **Advanced Programmable Interrupt Controller** (**APIC**) is a family of interrupt controllers. As its name suggests, the APIC is more advanced than Intel's 8259 Programmable Interrupt Controller (PIC), particularly enabling the construction of multiprocessor systems. It is one of several architectural designs intended to solve interrupt routing efficiency issues in multiprocessor computer systems.

The APIC is a split architecture design, with a local component (LAPIC) usually integrated into the processor itself, and an optional I/O APIC on a system bus. The first APIC was the 82489DX – it was a discrete chip that functioned both as local and I/O APIC. The 82489DX enabled construction of symmetric multiprocessor (SMP) systems with the Intel 486 and early Pentium processors; for example, the reference two-way 486 SMP system used three 82489DX chips, two as local APICs and one as I/O APIC. Starting with the P54C processor, the local APIC functionality was integrated into the Intel processors' silicon. The first dedicated I/O APIC was the Intel 82093AA, which was intended for PIIX3-based systems.

Contents

Overview

There are two components in the Intel APIC system, the *local APIC* (LAPIC) and the *I/O APIC*. There is one LAPIC in each CPU in the system. In the very first implementation (**82489DX**), the LAPIC was a discrete circuit opposed to its thereafter implementation in Intel processors' silicon. There is typically one I/O APIC for each peripheral bus in the system. In original system designs, LAPICs and I/O APICs were connected by a dedicated APIC bus. Newer systems use the system bus for communication between all APIC components.

Each APIC, whether a discrete chip or integrated in a CPU, has a version register containing a four-bit version number for its specific APIC implementation. For example, the 82489DX has an APIC version number of 0, while version 1 was assigned to the first generation of local APICs integrated in the Pentium 90 and 100 processors.[1]

In systems containing an 8259 PIC, the 8259 may be connected to the LAPIC in the system's bootstrap processor (BSP), or to one of the system's I/O APICs, or both. Logically, however, the 8259 is only connected once at any given time.

Discrete APIC

The first-generation Intel APIC chip, the 82489DX, which was meant to be used with Intel 80486 and early Pentium processors, is actually an external local and I/O APIC in one circuit. The Intel MP 1.4 specification refers to it as "discrete APIC" in contrast with the "integrated APIC" found in most of the Pentium processors.[2] The 82489DX had 16 interrupt lines;[3] it also had a quirk that it could lose some ISA interrupts.[4]

In a multiprocessor 486 system, each CPU had to be paired with its own 82489DX; additionally a supplementary 82489DX had to be used as I/O APIC. The 82489DX could not emulate the 8259A (XT-PIC) so these also had to be included as physical chips for backwards compatibility.[5] The 82489DX was a packaged as a 132-pin PQFP.[3]

Integrated local APICs

Local APICs (LAPICs) manage all external interrupts for some specific processor in an SMP system. In addition, they are able to accept and generate inter-processor interrupts (IPIs) between LAPICs. LAPICs may support up to 224 usable interrupt vectors from an I/O APIC. Vector numbers 0 to 31, out of 0 to 255, are reserved for exception handling by x86 processors.

All Intel processors starting with the P54C have a built-in local APIC.[6][7] However, if the local APIC is disabled in a P5 processor, it cannot be re-enabled by software; this limitation no longer exists in the P6 processors and later ones.[7] In single-processor systems, the major advantage of the local APIC is that PCI cards no longer need to have their interrupts mapped to ISA interrupts, but can use virtual PCI IRQs above 15, resulting in fewer conflicts and better performance.[6]

In the Microsoft family of operating systems, Windows XP was the first to properly make use of virtual IRQs provided by the local APIC, with some partial support present in Windows 2000.[6] In Windows 2000, the spreading of PCI card interrupts to virtual IRQ happens only if the machine also has an I/O APIC. Microsoft blamed their Windows 2000 implementation on information they had received from Intel supposedly promising that all future systems would contain an I/O APIC as well, something that in retrospect did not happen.[8] (A Microsoft document from that era even claimed that "without an I/O APIC in the system, the local APICs are useless. In such a situation, Windows 2000 has to revert to using the 8259 PIC."[9]) The sudden widespread use of the local APIC with Windows XP did expose a number of bugs in various BIOS implementations. In particular, Microsoft was forced to retain the PCI IRQ stacking behavior for any machine with a CardBus controller (and without an I/O APIC).[8]

The Message Signaled Interrupts (MSI) feature of the PCI 2.2 and later specifications cannot be used without the local APIC being enabled.[8] Use of MSI obviates the need for an I/O APIC. Additionally, up to 224 interrupts are supported in MSI mode, and IRQ sharing is not allowed.[10]

APIC timer

Another advantage of the local APIC is that it also provides a high-resolution (on the order of one microsecond or better) timer that can be used in both interval and one-shot mode.[7]

The APIC timer had its initial acceptance woes. A Microsoft document from 2002 (which advocated for the adoption of High Precision Event Timer instead) criticized the LAPIC timer for having "poor resolution" and stating that "the clocks silicon is sometimes very buggy".[11] Nevertheless, the APIC timer is used for example by Windows 7 when profiling is enabled, and by Windows 8 in all circumstances. (Before Windows 8 claimed exclusive rights to this timer, it was also used by some programs like CPU-Z.) Under Microsoft Windows the APIC timer is not a shareable resource.[12]

The aperiodic interrupts offered by the APIC timer are used by the Linux kernel from 2.6.18 onwards to implement its tickless kernel feature; the legacy 8253 Programmable Interval Timer is no longer used by tickless kernels.[13] A VMware document notes that "software does not have a reliable way to determine its frequency. Generally, the only way to determine the local APIC timer's frequency is to measure it using the PIT or CMOS timer, which yields only an approximate result."[14]

I/O APICs

I/O APICs contain a redirection table, which is used to route the interrupts it receives from peripheral buses to one or more local APICs. The first-generation dedicated I/O APIC, the 82093AA, had support for 24 interrupt lines.[10] It was packaged as a 64-Pin PQFP.[15] The 82093AA normally connected to the PIIX3 and used its integrated legacy 8259 PICs.[15]

According to a 2009 Intel benchmark using Linux, the I/O APIC reduced interrupt latency by a factor of almost three relative to the 8259 emulation (XT-PIC), while using MSI reduced the latency even more, by a factor of nearly seven relative to the XT-PIC baseline.[16]

Variants

The *xAPIC* was introduced with the Pentium 4, while the *x2APIC* is the most recent generation of the Intel's programmable interrupt controller, introduced with the Nehalem microarchitecture.[17] The major improvements of the x2APIC address the number of supported CPUs and performance of the interface.

The x2APIC now uses 32 bits to address CPUs, allowing to address up to $2^{32} - 1$ CPUs using the physical destination mode. The logical destination mode now works differently and introduces clusters; using this mode, one can address up to $2^{20} - 16$ processors. The x2APIC architecture also provides backward compatibility modes to the original Intel APIC Architecture (introduced with the Pentium/P6) and with the xAPIC architecture (introduced with the Pentium 4).

The improved interface reduces the number of needed APIC register access for sending Inter-processor interrupts (IPIs). Because of this advantage, QEMU can and does emulate x2APIC for older processors that do not physically support it, going back to Conroe and even for AMD Opteron G-series processors (neither of which natively support x2APIC).[18][19]

APICv is the Intel's brand name for hardware virtualization support aimed at reducing interrupt overhead in guests. APICv was introduced in the Ivy Bridge-EP processor series, which is sold as Xeon E5-26xx v2 (launched in late 2013) and as Xeon E5-46xx v2 (launched in early 2014).[20][21][22] AMD announced a similar technology called AVIC,[23][24] it is available family 15h models 6Xh (Carrizo) processors and newer.[25]

Issues

There are a number of known bugs in implementations of APIC systems, especially with concern to how the 8254 is connected. Defective BIOSes may not set up interrupt routing properly, or provide incorrect ACPI tables and Intel MultiProcessor Specification (MPS) tables.

The APIC can also be a cause of system failure when the operating system does not support it properly. On older operating systems, the I/O and local APICs often had to be disabled. While this is not possible anymore due to the prevalence of symmetric multiprocessor and multi-core systems, the bugs in the firmware and the operating systems are now a rare occurrence.

Competition

AMD and Cyrix once proposed as somewhat similar-in-purpose OpenPIC architecture supporting up to 32 processors;[26] it had at least declarative support from IBM and Compaq around 1995.[27] No x86 motherboard was released with OpenPIC however.[28] After the OpenPIC's failure in the x86 market, AMD licensed Intel's APIC for its AMD Athlon and later processors.

IBM however developed their MultiProcessor Interrupt Controller (MPIC) based on the OpenPIC register specifications.[29] MPIC was used in PowerPC based designs, including those of IBM, for instance in some RS/6000 systems,[30] but also by Apple, as late as their Power Mac G5s.[31][32]

See also

- Intel 8259
- Programmable Interrupt Controller (PIC)
- Inter-processor interrupt (IPI)
- Interrupt
- Interrupt handler
- Interrupt latency
- Message Signaled Interrupts (MSI)
- Non-maskable interrupt (NMI)

References

1. Intel MultiProcessor Specification (http://www.intel.com/design/archives/processors/pro/docs/242016.htm) version 1.4, page 3-5, May 1997
2. Intel MultiProcessor Specification (http://www.intel.com/design/archives/processors/pro/docs/242016.htm) version 1.4, page 1-4, May 1997
3. Badri Ram (2001). *Adv Microprocessors Interfacing* (https://books.google.com/books?id=eVcEWDIeTYcC&pg=PT314). Tata McGraw-Hill Education. p. 314. ISBN 978-0-07-043448-6
4. http://people.freebsd.org/~fsmp/SMP/papers/apicsubsystem.txt
5. Intel MultiProcessor Specification, version 1.4, page 5-3, May 1997
6. Scott M. Mueller (2011). *Upgrading and Repairing PCs* (20th ed.). Que Publishing. p. 242. ISBN 978-0-13-268218-3
7. Uwe Walter, Vincent Oberle. µ-second precision timer support for the Linux kernel (http://telematics.tm.kit.edu/publications/Files/61/walter_ibm_linux_challenge.pdf)
8. APIC-Based Interrupt Subsystems on Uniprocessor PCs (http://msdn.microsoft.com/en-us/windows/hardware/gg462964.aspx)
9. Key Benefits of the I/O APIC (http://www.microsoft.com/whdc/archive/io-apic.mspx)
10. James Coleman, Reducing Interrupt Latency Through the Use of Message Signaled Interrupts (http://www.intel.com/content/dam/www/public/us/en/documents/white-papers/msg-signaled-interrupts-paper.pdf), pp. 10-11
11. Guidelines For Providing Multimedia Timer Support (http://msdn.microsoft.com/en-us/library/windows/hardware/gg463347.aspx), September 20, 2002
12. Windows 8 and APIC timer (http://social.msdn.microsoft.com/Forums/windowsdesktop/en-US/5d075378-a45f-433b-a3f7-73f974ec962f/windows-8-and-apic-timer?forum=wdk). Archived (https://web.archive.org/web/20140222070735/http://social.msdn.microsoft.com/Forums/windowsdesktop/en-US/5d075378-a45f-433b-a3f7-73f974ec962f/windows-8-and-apic-timer?forum=wdk) February 22, 2014, at the Wayback Machine.
13. Determining and changing the rate of timer interrupts a guest operating system requests (1005802) (http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=1005802)
14. Timekeeping in VMware Virtual Machines (for VMware vSphere 5.0, Workstation 8.0, Fusion 4.0) (http://www.vmware.com/files/pdf/Timekeeping-In-VirtualMachines.pdf), page 8
15. Intel 82093AA I/O Advanced Programmable Interrupt Controller (I/O APIC) Datasheet (http://www.intel.com/design/chipsets/datashts/290566.htm)
16. James Coleman, Reducing Interrupt Latency Through the Use of Message Signaled Interrupts (http://www.intel.com/content/dam/www/public/us/en/documents/white-papers/msg-signaled-interrupts-paper.pdf), p. 19
17. "Next Generation Interrupt Architecture" (http://www.computerbase.de/bildstrecke/20069/27/)
18. https://lists.gnu.org/archive/html/qemu-devel/2013-07/msg03756.html
19. http://lists.nongnu.org/archive/html/qemu-devel/2014-01/msg02441.html
20. http://www.linuxplumbersconf.org/2012/wp-content/uploads/2012/09/2012-lpc-virt-intel-vt-feat-nakajima.pdf
21. https://software.intel.com/en-us/blogs/2013/12/17/apic-virtualization-performance-testing-and-iozone
22. http://www.intel.com/content/dam/www/public/us/en/documents/product-briefs/xeon-e5-4600-v2-brief.pdf
23. Wei Huang, Introduction of AMD Advanced Virtual Interrupt Controller (http://www.slideshare.net/xen_com_mgr/introduction-of-amd-virtual-interrupt-controller), XenSummit 2012
24. http://www.linuxplumbersconf.org/2012/wp-content/uploads/2012/09/2012-lpc-virt-interrupt-virt-kvm-roedel.pdf

25. https://www.mail-archive.com/xen-devel@lists.xen.org/msg81719.html
26. "OpenPIC Definition from PC Magazine Encyclopedia" (https://www.pcmag.com/encyclopedia_term/0,2542,t=OpenPIC&i=48497,00.asp). Pcmag.com. 1994-12-01. Retrieved 2011-11-03.
27. Brooke Crothers (20 March 1995). *AMD, Cyrix offer up alternative SMP spec* (https://books.google.com/books?id=lToEAAAAMBAJ&pg=PA8). InfoWorld. p. 8. ISSN 0199-6649 (https://www.worldcat.org/issn/0199-6649)
28. André D. Balsa, Note attached to "Linux Benchmarking: Part III -- Interpreting Benchmark Results" (http://linuxgazette.net/issue24/Article3e-7.html) appearing in Issue 24 of Linux Gazette, January 1998
29. IBM Multiprocessor Interrupt Controller Data Book (https://www-01.ibm.com/chips/techlib/techlib.nsf/techdocs/F27971551C9EED8E8525774A0048770A/$file/mpic_db_05_16_2011.pdf) Archived (https://web.archive.org/web/20140223012746/https://www-01.ibm.com/chips/techlib/techlib.nsf/techdocs/F27971551C9EED8E8525774A0048770A/%24file/mpic_db_05_16_2011.pdf) 2014-02-23 at the Wayback Machine.
30. Arca Systems TTAP Evaluation Facility. The IBM Corporation RS/6000 Distributed System Running AIX Version 4.3.1. TCSEC Evaluated C2 Security (http://www.ashtonlabs.com/library/FERs/CSC-FER-98-004.pdf), p. 29
31. Take a Look Inside the G5-Based Dual-Processor Power Mac (http://www.informit.com/articles/article.aspx?p=606582)
32. Power Mac G5 Developer Note (Legacy) (https://developer.apple.com/legacy/library/documentation/Hardware/Developer_Notes/Macintosh_CPUs-G5/PowerMacG5/PowerMacG5.pdf), p. 26

Further reading

- IA-32 Intel Architecture Software Developer's Manual, Volume 3A: System Programming Guide, Part 1, chapter 10

External links

- Intel 64 Architecture x2APIC Specification (PDF)
- More information on the Intel x2APIC Architecture can be found in the *Intel 64 and IA-32 Architectures Software Developer's Manuals*