[patch] i386: port ATI timer fix from x86_64 to i386

Source: http://linux.derkeiler.com/Mailing−Lists/Kernel/2006−02/msg10098.html

- *From*: Chuck Ebbert <76306.1226@xxxxxxxxxxxxxx>
- *Date*: Tue, 28 Feb 2006 16:17:13 −0500

Disable timer routing over 8254 when an ATI chipset is detected
(autodetect is only implemented for ACPI, but these are new systems
and should be using ACPI anyway.) Adds boot options for manually
disabling and enabling this feature. Also adds a note to the timer
error message caused by this change explaining that this error
is expected on ATI chipsets.

Signed−off−by: Chuck Ebbert <76306.1226@xxxxxxxxxxxxxx>

−−−

This would be nice for 2.6.16 but I don't recommend it because of the
diversity of i386 hardware.

Documentation/kernel−parameters.txt | 33 ++++++++++++++++++++++−−−−−−−−−−
arch/i386/kernel/acpi/earlyquirk.c | 12 ++++++++++++
arch/i386/kernel/io_apic.c | 21 ++++++++++++++++++−−
include/asm−i386/acpi.h | 1 +
4 files changed, 54 insertions(+), 13 deletions(−)

−−− 2.6.16−rc5−d2.orig/Documentation/kernel−parameters.txt
+++ 2.6.16−rc5−d2/Documentation/kernel−parameters.txt
@@ −80,6 +80,7 @@ restrictions referred to are that the re
VT Virtual terminal support is enabled.
WDT Watchdog support is enabled.
XT IBM PC/XT MFM hard disk support is enabled.
+ X86 Either X86−64 or IA−32 (i386) is enabled
X86−64 X86−64 architecture is enabled.
More X86−64 boot options can be found in
Documentation/x86_64/boot−options.txt .
@@ −167,16 +168,6 @@ running once the system is up.
override platform specific driver.
See also Documentation/acpi−hotkey.txt.

− enable_timer_pin_1 [i386,x86−64]
− Enable PIN 1 of APIC timer
− Can be useful to work around chipset bugs
− (in particular on some ATI chipsets).
− The kernel tries to set a reasonable default.

```
−
− disable_timer_pin_1 [i386,x86−64]
− Disable PIN 1 of APIC timer
− Can be useful to work around chipset bugs.
−
ad1816= [HW,OSS]
Format: <io>,<irq>,<dma>,<dma2>
See also Documentation/sound/oss/AD1816.
@@ −226,7 +217,7 @@ running once the system is up.
not play well with APC CPU idle − disable it if you have
APC and your system crashes randomly.

− apic= [APIC,i386] Change the output verbosity whilst booting
+ apic= [APIC,X86] Change the output verbosity whilst booting
Format: { quiet (default) | verbose | debug }
Change the amount of debugging information output
when initialising the APIC and IO−APIC components.
@@ −423,6 +414,16 @@ running once the system is up.
See drivers/char/README.epca and
Documentation/digiepca.txt.

+ disable_8254_timer [X86]
+ Disable interrupt 0 timer routing over the 8254
+ in addition to over the IO−APIC. The kernel tries
+ to set a sensible default.
+
+ disable_timer_pin_1 [X86]
+ Disable PIN 1 of APIC timer
+ Can be useful to work around chipset bugs
+ (in particular on some ATI chipsets).
+
dmascc= [HW,AX25,SERIAL] AX.25 Z80SCC driver with DMA
support available.
Format: <io_dev0>[,<io_dev1>[,..<io_dev32>]]
@@ −486,6 +487,16 @@ running once the system is up.
pass this option to capture kernel.
See Documentation/kdump/kdump.txt for details.

+ enable_8254_timer [X86]
+ Enable interrupt 0 timer routing over the 8254
+ in addition to over the IO−APIC. The kernel tries
+ to set a sensible default.
+
+ enable_timer_pin_1 [X86]
+ Enable PIN 1 of APIC timer
+ Can be useful to work around chipset bugs
+ (in particular on some ATI chipsets).
+
enforcing [SELINUX] Set initial enforcing status.
Format: {"0" | "1"}
See security/selinux/Kconfig help text.
```

```
−−− 2.6.16−rc5−d2.orig/arch/i386/kernel/acpi/earlyquirk.c
+++ 2.6.16−rc5−d2/arch/i386/kernel/acpi/earlyquirk.c
@@ −15,6 +15,18 @@ static int __init check_bridge(int vendo
if (vendor == PCI_VENDOR_ID_NVIDIA) {
acpi_skip_timer_override = 1;
}
+#ifdef CONFIG_X86_IO_APIC
+ /* Many ATI boards have timer problems. This fix should
+ * be harmless even on non−ATI boards, but play it safe.
+ */
+ if (vendor == PCI_VENDOR_ID_ATI) {
+ if (timer_over_8254 == 1) {
+ timer_over_8254 = 0;
+ printk(KERN_INFO "ATI board detected. "
+ "Disabling timer routing over 8254.\n"
+ }
+ }
+#endif
return 0;
}

−−− 2.6.16−rc5−d2.orig/arch/i386/kernel/io_apic.c
+++ 2.6.16−rc5−d2/arch/i386/kernel/io_apic.c
@@ −64,6 +64,22 @@ int nr_ioapic_registers[MAX_IO_APICS];

int disable_timer_pin_1 __initdata;

+int timer_over_8254 __initdata = 1;
+
+static int __init setup_disable_8254_timer(char *s)
+{
+ timer_over_8254 = −1;
+ return 1;
+}
+__setup("disable_8254_timer", setup_disable_8254_timer);
+
+static int __init setup_enable_8254_timer(char *s)
+{
+ timer_over_8254 = 2;
+ return 1;
+}
+__setup("enable_8254_timer", setup_enable_8254_timer);
+
/*
* Rough estimation of how many shared IRQs there are, can
* be changed anytime.
@@ −2267,7 +2283,8 @@ static inline void check_timer(void)
apic_write_around(APIC_LVT0, APIC_LVT_MASKED | APIC_DM_EXTINT);
init_8259A(1);
timer_ack = 1;
− enable_8259A_irq(0);
```

+ if (timer_over_8254 > 0)
+ enable_8259A_irq(0);

pin1 = find_isa_irq_pin(0, mp_INT);
apic1 = find_isa_irq_apic(0, mp_INT);
@@ −2294,7 +2311,7 @@ static inline void check_timer(void)
}
clear_IO_APIC_pin(apic1, pin1);
printk(KERN_ERR "..MP−BIOS bug: 8254 timer not connected to "
− "IO−APIC\n");
+ "IO−APIC (expected on ATI chipsets)\n");
}

printk(KERN_INFO "...trying to set up timer (IRQ0) through the 8259A ... ");
−−− 2.6.16−rc5−d2.orig/include/asm−i386/acpi.h
+++ 2.6.16−rc5−d2/include/asm−i386/acpi.h
@@ −127,6 +127,7 @@ extern int acpi_gsi_to_irq(u32 gsi, unsi
#ifdef CONFIG_X86_IO_APIC
extern int skip_ioapic_setup;
extern int acpi_skip_timer_override;
+extern int timer_over_8254;

extern void check_acpi_pci(void);

−−
Chuck
"Equations are the Devil's sentences." −−Stephen Colbert
−