

Re: [PATCH] [1/1] CPU-i386-Geode: Chipset access macros do not work as expected (2nd try)

Re: [PATCH] [1/1] CPU-i386-Geode: Chipset access macros do not work as expected (2nd try)

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2007-04/msg12022.html>

- *From:* Mikael Pettersson <mikpe@xxxxxxxx>
 - *Date:* Mon, 30 Apr 2007 17:40:34 +0200 (MEST)
-

On Mon, 30 Apr 2007 17:33:41 +0200, Juergen Beisert wrote:

Replace NSC/Cyrix specific chipset access macros by inlined functions.
With the macros a line like this fails (and does nothing):
setCx86(CX86_CCR2, getCx86(CX86_CCR2) | 0x88);
With inlined functions this line will work as expected.

Note about a side effect: Seems on Geode GX1 based systems the "suspend on halt power saving feature" was never enabled due to this wrong macro expansion. With inlined functions it will be enabled, but this will stop the TSC when the CPU runs into a HLT instruction.

Kernel outputs something like this:
Clocksource tsc unstable (delta = -472746897 ns)
Tested on a Geode GX1 system.

This is the second version with some modifications suggested by Mikael Pettersson

Signed-off-by: Juergen Beisert <juergen.beisert@xxxxxxxxxxxxxxxxxxxx>

Acked-by: Mikael Pettersson <mikpe@xxxxxxxx>

Index: linux-2.6.21/include/asm-i386/processor.h

```
=====
--- linux-2.6.21.orig/include/asm-i386/processor.h
+++ linux-2.6.21/include/asm-i386/processor.h
@@ -202,37 +202,6 @@ static inline void clear_in_cr4 (unsigned
write_cr4(cr4);
}

-/*
- * NSC/Cyrix CPU configuration register indexes
- */
-
-#define CX86_PCR0 0x20
-#define CX86_GCR 0xb8
```

Re: [PATCH] [1/1] CPU-i386-Geode: Chipset access macros do not work as expected (2nd try)

```

-#define CX86_CCR0 0xc0
-#define CX86_CCR1 0xc1
-#define CX86_CCR2 0xc2
-#define CX86_CCR3 0xc3
-#define CX86_CCR4 0xe8
-#define CX86_CCR5 0xe9
-#define CX86_CCR6 0xea
-#define CX86_CCR7 0xeb
-#define CX86_PCR1 0xf0
-#define CX86_DIR0 0xfe
-#define CX86_DIR1 0xff
-#define CX86_ARR_BASE 0xc4
-#define CX86_RCR_BASE 0xdc
-
-/*
- * NSC/Cyrix CPU indexed register access macros
- */
-
-#define getCx86(reg) ({ outb((reg), 0x22); inb(0x23); })
-
-#define setCx86(reg, data) do { \
- outb((reg), 0x22); \
- outb((data), 0x23); \
-} while (0)
-
/* Stop speculative execution */
static inline void sync_core(void)
{
Index: linux-2.6.21/include/asm-i386/processor-cyrix.h
=====
--- /dev/null
+++ linux-2.6.21/include/asm-i386/processor-cyrix.h
@@ -0,0 +1,33 @@
+/*
+ * NSC/Cyrix CPU configuration register indexes
+ */
+#define CX86_PCR0 0x20
+#define CX86_GCR 0xb8
+#define CX86_CCR0 0xc0
+#define CX86_CCR1 0xc1
+#define CX86_CCR2 0xc2
+#define CX86_CCR3 0xc3
+#define CX86_CCR4 0xe8
+#define CX86_CCR5 0xe9
+#define CX86_CCR6 0xea
+#define CX86_CCR7 0xeb
+#define CX86_PCR1 0xf0
+#define CX86_DIR0 0xfe
+#define CX86_DIR1 0xff
+#define CX86_ARR_BASE 0xc4
+#define CX86_RCR_BASE 0xdc
```

Re: [PATCH] [1/1] CPU-i386-Geode: Chipset access macros do not work as expected (2nd try) 2

Re: [PATCH] [1/1] CPU-i386-Geode: Chipset access macros do not work as expected (2nd try)

```
+
+/*
+ * NSC/Cyrix CPU indexed register access
+ */
+static inline u8 getCx86(u8 reg)
+{
+ outb(reg, 0x22);
+ return inb(0x23);
+}
+
+static inline void setCx86(u8 reg, u8 data)
+{
+ outb(reg, 0x22);
+ outb(data, 0x23);
+}
```

Index: linux-2.6.21/arch/i386/kernel/cpu/cyrix.c

```
-----
--- linux-2.6.21.orig/arch/i386/kernel/cpu/cyrix.c
+++ linux-2.6.21/arch/i386/kernel/cpu/cyrix.c
@@ -4,7 +4,7 @@
#include <linux/pci.h>
#include <asm/dma.h>
#include <asm/io.h>
-#include <asm/processor.h>
+#include <asm/processor-cyrix.h>
#include <asm/timer.h>
#include <asm/pci-direct.h>
```

Index: linux-2.6.21/arch/i386/kernel/cpu/mtrr/cyrix.c

```
-----
--- linux-2.6.21.orig/arch/i386/kernel/cpu/mtrr/cyrix.c
+++ linux-2.6.21/arch/i386/kernel/cpu/mtrr/cyrix.c
@@ -3,6 +3,7 @@
#include <asm/mtrr.h>
#include <asm/msr.h>
#include <asm/io.h>
+#include <asm/processor-cyrix.h>
#include "mtrr.h"
```

```
int arr3_protected;
```

—
To unsubscribe from this list: send the line "unsubscribe linux-kernel" in
the body of a message to majordomo@xxxxxxxxxxxxxxxxx
More majordomo info at <http://vger.kernel.org/majordomo-info.html>
Please read the FAQ at <http://www.tux.org/lkml/>