

# [PATCH 07/16] cell: update cell be register definitions

---

*Source:* <http://linux.derkeiler.com/Mailing-Lists/Kernel/2006-10/msg08453.html>

---

- *From:* [arnd@xxxxxxxx](mailto:arnd@xxxxxxxx)
  - *Date:* Tue, 24 Oct 2006 18:31:20 +0200
- 

From: David Erb <djerb@xxxxxxxx>

There are a few definitions that are required by subsequent patches, so add them here.

The original patch is from David Erb, but is significantly cleaned up by Kevon Corry.

Cc: Kevin Corry <kevcorry@xxxxxxxx>

Signed-off-by: Arnd Bergmann <arnd.bergmann@xxxxxxxx>

Index: linux-2.6/arch/powerpc/platforms/cell/cbe\_regs.h

```
----- linux-2.6.orig/arch/powerpc/platforms/cell/cbe_regs.h
+++ linux-2.6/arch/powerpc/platforms/cell/cbe_regs.h
@@ -4,6 +4,11 @@
* This file is intended to hold the various register definitions for CBE
* on-chip system devices (memory controller, IO controller, etc...)
*
+ * (C) Copyright IBM Corporation 2001,2006
+ *
+ * Authors: Maximino Aguilar (maguilar@xxxxxxxx)
+ * David J. Erb (djerb@xxxxxxxx)
+ *
* (c) 2006 Benjamin Herrenschmidt <benh@xxxxxxxxxxxxxxxxxxxx>, IBM Corp.
*/

@@ -22,6 +27,7 @@
#define HID0_CBE_THERM_INT_EN 0x0000000400000000ul
#define HID0_CBE_SYSERR_INT_EN 0x0000000200000000ul

+#define MAX_CBE 2

/*
*
@@ -29,45 +35,86 @@
*
*/
```

[PATCH 07/16] cell: update cell be register definitions

```
+union spe_reg {
+ u64 val;
+ u8 spe[8];
+};
+
+union ppe_spe_reg {
+ u64 val;
+ struct {
+ u32 ppe;
+ u32 spe;
+ };
+};
+
+
+struct cbe_pmd_regs {
- u8 pad_0x0000_0x0800[0x0800 - 0x0000]; /* 0x0000 */
+ /* Debug Bus Control */
+ u64 pad_0x0000; /* 0x0000 */
+
+ u64 group_control; /* 0x0008 */
+
+ u8 pad_0x0010_0x00a8 [0x00a8 - 0x0010]; /* 0x0010 */
+
+ u64 debug_bus_control; /* 0x00a8 */
+
+ u8 pad_0x00b0_0x0100 [0x0100 - 0x00b0]; /* 0x00b0 */
+
+ u64 trace_aux_data; /* 0x0100 */
+ u64 trace_buffer_0_63; /* 0x0108 */
+ u64 trace_buffer_64_127; /* 0x0110 */
+ u64 trace_address; /* 0x0118 */
+ u64 ext_tr_timer; /* 0x0120 */
+
+ u8 pad_0x0128_0x0400 [0x0400 - 0x0128]; /* 0x0128 */
+
+ /* Performance Monitor */
+ u64 pm_status; /* 0x0400 */
+ u64 pm_control; /* 0x0408 */
+ u64 pm_interval; /* 0x0410 */
+ u64 pm_ctr[4]; /* 0x0418 */
+ u64 pm_start_stop; /* 0x0438 */
+ u64 pm07_control[8]; /* 0x0440 */
+
+ u8 pad_0x0480_0x0800 [0x0800 - 0x0480]; /* 0x0480 */
+
+ /* Thermal Sensor Registers */
- u64 ts_ctr1; /* 0x0800 */
- u64 ts_ctr2; /* 0x0808 */
- u64 ts_mtsr1; /* 0x0810 */
- u64 ts_mtsr2; /* 0x0818 */
- u64 ts_itr1; /* 0x0820 */
```

[PATCH 07/16] cell: update cell be register definitions

```
- u64 ts_itr2; /* 0x0828 */
- u64 ts_gitr; /* 0x0830 */
- u64 ts_isr; /* 0x0838 */
- u64 ts_imr; /* 0x0840 */
- u64 tm_cr1; /* 0x0848 */
- u64 tm_cr2; /* 0x0850 */
- u64 tm_simr; /* 0x0858 */
- u64 tm_tpr; /* 0x0860 */
- u64 tm_str1; /* 0x0868 */
- u64 tm_str2; /* 0x0870 */
- u64 tm_tsr; /* 0x0878 */
+ union spe_reg ts_ctsr1; /* 0x0800 */
+ u64 ts_ctsr2; /* 0x0808 */
+ union spe_reg ts_mtsr1; /* 0x0810 */
+ u64 ts_mtsr2; /* 0x0818 */
+ union spe_reg ts_itr1; /* 0x0820 */
+ u64 ts_itr2; /* 0x0828 */
+ u64 ts_gitr; /* 0x0830 */
+ u64 ts_isr; /* 0x0838 */
+ u64 ts_imr; /* 0x0840 */
+ union spe_reg tm_cr1; /* 0x0848 */
+ u64 tm_cr2; /* 0x0850 */
+ u64 tm_simr; /* 0x0858 */
+ union ppe_spe_reg tm_tpr; /* 0x0860 */
+ union spe_reg tm_str1; /* 0x0868 */
+ u64 tm_str2; /* 0x0870 */
+ union ppe_spe_reg tm_tsr; /* 0x0878 */

/* Power Management */
- u64 pm_control; /* 0x0880 */
-#define CBE_PMD_PAUSE_ZERO_CONTROL 0x10000
- u64 pm_status; /* 0x0888 */
+ u64 pmcr; /* 0x0880 */
+#define CBE_PMD_PAUSE_ZERO_CONTROL 0x10000
+ u64 pmsr; /* 0x0888 */

/* Time Base Register */
- u64 tbr; /* 0x0890 */
+ u64 tbr; /* 0x0890 */

- u8 pad_0x0898_0x0c00 [0x0c00 - 0x0898]; /* 0x0898 */
+ u8 pad_0x0898_0x0c00 [0x0c00 - 0x0898]; /* 0x0898 */

/* Fault Isolation Registers */
- u64 checkstop_fir; /* 0x0c00 */
- u64 recoverable_fir;
- u64 spec_att_mchk_fir;
- u64 fir_mode_reg;
- u64 fir_enable_mask;
+ u64 checkstop_fir; /* 0x0c00 */
+ u64 recoverable_fir; /* 0x0c08 */
```

[PATCH 07/16] cell: update cell be register definitions

```
+ u64 spec_att_mchk_fir; /* 0x0c10 */
+ u64 fir_mode_reg; /* 0x0c18 */
+ u64 fir_enable_mask; /* 0x0c20 */

- u8 pad_0x0c28_0x1000 [0x1000 - 0x0c28]; /* 0x0c28 */
+ u8 pad_0x0c28_0x1000 [0x1000 - 0x0c28]; /* 0x0c28 */
};

extern struct cbe_pmd_regs __iomem *cbe_get_pmd_regs(struct device_node *np);
@@ -102,18 +149,20 @@ struct cbe_iic_regs {

/* IIC interrupt registers */
struct cbe_iic_thread_regs thread[2]; /* 0x0400 */
- u64 iic_ir; /* 0x0440 */
- u64 iic_is; /* 0x0448 */
+
+ u64 iic_ir; /* 0x0440 */
+ u64 iic_is; /* 0x0448 */
+#define CBE_IIC_IS_PMI 0x2

u8 pad_0x0450_0x0500[0x0500 - 0x0450]; /* 0x0450 */

/* IOC FIR */
u64 ioc_fir_reset; /* 0x0500 */
- u64 ioc_fir_set;
- u64 ioc_checkstop_enable;
- u64 ioc_fir_error_mask;
- u64 ioc_syserr_enable;
- u64 ioc_fir;
+ u64 ioc_fir_set; /* 0x0508 */
+ u64 ioc_checkstop_enable; /* 0x0510 */
+ u64 ioc_fir_error_mask; /* 0x0518 */
+ u64 ioc_syserr_enable; /* 0x0520 */
+ u64 ioc_fir; /* 0x0528 */

u8 pad_0x0530_0x1000[0x1000 - 0x0530]; /* 0x0530 */
};
@@ -122,6 +171,48 @@ extern struct cbe_iic_regs __iomem *cbe_
extern struct cbe_iic_regs __iomem *cbe_get_cpu_iic_regs(int cpu);

+struct cbe_mic_tm_regs {
+ u8 pad_0x0000_0x0040[0x0040 - 0x0000]; /* 0x0000 */
+
+ u64 mic_ctl_cnfg2; /* 0x0040 */
+#define CBE_MIC_ENABLE_AUX_TRC 0x8000000000000000LL
+#define CBE_MIC_DISABLE_PWR_SAV_2 0x0200000000000000LL
+#define CBE_MIC_DISABLE_AUX_TRC_WRAP 0x0100000000000000LL
+#define CBE_MIC_ENABLE_AUX_TRC_INT 0x0080000000000000LL
+
+ u64 pad_0x0048; /* 0x0048 */
```

[PATCH 07/16] cell: update cell be register definitions

[PATCH 07/16] cell: update cell be register definitions

```
+
+ u64 mic_aux_trc_base; /* 0x0050 */
+ u64 mic_aux_trc_max_addr; /* 0x0058 */
+ u64 mic_aux_trc_cur_addr; /* 0x0060 */
+ u64 mic_aux_trc_grf_addr; /* 0x0068 */
+ u64 mic_aux_trc_grf_data; /* 0x0070 */
+
+ u64 pad_0x0078; /* 0x0078 */
+
+ u64 mic_ctl_cnfg_0; /* 0x0080 */
+#define CBE_MIC_DISABLE_PWR_SAV_0 0x8000000000000000LL
+
+ u64 pad_0x0088; /* 0x0088 */
+
+ u64 slow_fast_timer_0; /* 0x0090 */
+ u64 slow_next_timer_0; /* 0x0098 */
+
+ u8 pad_0x00a0_0x01c0[0x01c0 - 0x0a0]; /* 0x00a0 */
+
+ u64 mic_ctl_cnfg_1; /* 0x01c0 */
+#define CBE_MIC_DISABLE_PWR_SAV_1 0x8000000000000000LL
+ u64 pad_0x01c8; /* 0x01c8 */
+
+ u64 slow_fast_timer_1; /* 0x01d0 */
+ u64 slow_next_timer_1; /* 0x01d8 */
+
+ u8 pad_0x01e0_0x1000[0x1000 - 0x01e0]; /* 0x01e0 */
+};
+
+extern struct cbe_mic_tm_regs __iomem *cbe_get_mic_tm_regs(struct device_node *np);
+extern struct cbe_mic_tm_regs __iomem *cbe_get_cpu_mic_tm_regs(int cpu);
+
+/* Init this module early */
extern void cbe_regs_init(void);
```

Index: linux-2.6/arch/powerpc/platforms/cell/cbe\_regs.c

```
=====
--- linux-2.6.orig/arch/powerpc/platforms/cell/cbe_regs.c
+++ linux-2.6/arch/powerpc/platforms/cell/cbe_regs.c
@@ -8,6 +8,7 @@
```

```
#include <linux/percpu.h>
#include <linux/types.h>
+#include <linux/module.h>
```

```
#include <asm/io.h>
#include <asm/pgtable.h>
@@ -16,8 +17,6 @@
```

```
#include "cbe_regs.h"
```

[PATCH 07/16] cell: update cell be register definitions

```

-#define MAX_CBE 2
-
/*
 * Current implementation uses "cpu" nodes. We build our own mapping
 * array of cpu numbers to cpu nodes locally for now to allow interrupt
@@ -30,6 +29,7 @@ static struct cbe_regs_map
struct device_node *cpu_node;
struct cbe_pmd_regs __iomem *pmd_regs;
struct cbe_iic_regs __iomem *iic_regs;
+ struct cbe_mic_tm_regs __iomem *mic_tm_regs;
} cbe_regs_maps[MAX_CBE];
static int cbe_regs_map_count;

@@ -42,6 +42,19 @@ static struct cbe_thread_map
static struct cbe_regs_map *cbe_find_map(struct device_node *np)
{
int i;
+ struct device_node *tmp_np;
+
+ if (strcasecmp(np->type, "spe") == 0) {
+ if (np->data == NULL) {
+ /* walk up path until cpu node was found */
+ tmp_np = np->parent;
+ while (tmp_np != NULL && strcmp(tmp_np->type, "cpu") != 0)
+ tmp_np = tmp_np->parent;
+
+ np->data = cbe_find_map(tmp_np);
+ }
+ return np->data;
+ }

for (i = 0; i < cbe_regs_map_count; i++)
if (cbe_regs_maps[i].cpu_node == np)
@@ -56,6 +69,7 @@ struct cbe_pmd_regs __iomem *cbe_get_pmd
return NULL;
return map->pmd_regs;
}
+EXPORT_SYMBOL_GPL(cbe_get_pmd_regs);

struct cbe_pmd_regs __iomem *cbe_get_cpu_pmd_regs(int cpu)
{
@@ -64,7 +78,7 @@ struct cbe_pmd_regs __iomem *cbe_get_cpu
return NULL;
return map->pmd_regs;
}
-
+EXPORT_SYMBOL_GPL(cbe_get_cpu_pmd_regs);

struct cbe_iic_regs __iomem *cbe_get_iic_regs(struct device_node *np)
{
@@ -73,6 +87,7 @@ struct cbe_iic_regs __iomem *cbe_get_iic
```

[PATCH 07/16] cell: update cell be register definitions

```
return NULL;
return map->iic_regs;
}
+
struct cbe_iic_regs __iomem *cbe_get_cpu_iic_regs(int cpu)
{
struct cbe_regs_map *map = cbe_thread_map[cpu].regs;
@@ -81,6 +96,24 @@ struct cbe_iic_regs __iomem *cbe_get_cpu
return map->iic_regs;
}

+struct cbe_mic_tm_regs __iomem *cbe_get_mic_tm_regs(struct device_node *np)
+{
+ struct cbe_regs_map *map = cbe_find_map(np);
+ if (map == NULL)
+ return NULL;
+ return map->mic_tm_regs;
+}
+
+struct cbe_mic_tm_regs __iomem *cbe_get_cpu_mic_tm_regs(int cpu)
+{
+ struct cbe_regs_map *map = cbe_thread_map[cpu].regs;
+ if (map == NULL)
+ return NULL;
+ return map->mic_tm_regs;
+}
+EXPORT_SYMBOL_GPL(cbe_get_cpu_mic_tm_regs);
+
+
void __init cbe_regs_init(void)
{
int i;
@@ -119,6 +152,11 @@ void __init cbe_regs_init(void)
prop = get_property(cpu, "iic", NULL);
if (prop != NULL)
map->iic_regs = ioremap(prop->address, prop->len);
+
+ prop = (struct address_prop *)get_property(cpu, "mic-tm",
+ NULL);
+ if (prop != NULL)
+ map->mic_tm_regs = ioremap(prop->address, prop->len);
}
}
```

Index: linux-2.6/arch/powerpc/platforms/cell/pervasive.c

```
----- linux-2.6.orig/arch/powerpc/platforms/cell/pervasive.c
+++ linux-2.6/arch/powerpc/platforms/cell/pervasive.c
@@ -54,9 +54,9 @@ static void __init cbe_enable_pause_zero
pr_debug("Power Management: CPU %d\n", smp_processor_id());
```

[PATCH 07/16] cell: update cell be register definitions

```
/* Enable Pause(0) control bit */
- temp_register = in_be64(&pregs->pm_control);
+ temp_register = in_be64(&pregs->pmcr);

- out_be64(&pregs->pm_control,
+ out_be64(&pregs->pmcr,
temp_register | CBE_PMD_PAUSE_ZERO_CONTROL);

/* Enable DEC and EE interrupt request */
@@ -87,7 +87,7 @@ static void cbe_idle(void)
unsigned long ctrl;

/* Why do we do that on every idle ? Couldn't that be done once for
- * all or do we lose the state some way ? Also, the pm_control
+ * all or do we lose the state some way ? Also, the pmcr
* register setting, that can't be set once at boot ? We really want
* to move that away in order to implement a simple powersave
*/
```

--

-

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/>