

[PATCH 11/13] cell: split out board specific files

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2006-04/msg06635.html>

- *From:* Arnd Bergmann <arnd@xxxxxxxx>
 - *Date:* Sun, 30 Apr 2006 01:28:23 +0200
-

From: Geoff Levand <geoffrey.levand@xxxxxxxxxxxx>

Split the Cell BE support into generic and platform dependant parts.

Creates a new config variable CONFIG_PPC_IBM_CELL_BLADE. The existing CONFIG_PPC_CELL is now used to denote the generic Cell processor support. Also renames spu_priv1.c to spu_priv1_mmio.c.

Signed-off-by: Geoff Levand <geoffrey.levand@xxxxxxxxxxxx>

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

Index: linux-2.6/arch/powerpc/Kconfig

=====

--- linux-2.6.orig/arch/powerpc/Kconfig 2006-04-29 22:53:50.000000000 +0200

+++ linux-2.6/arch/powerpc/Kconfig 2006-04-29 22:54:42.000000000 +0200

@@ -391,11 +391,17 @@

For more informations, refer to <<http://www.970eval.com>>

config PPC_CELL

- bool " Cell Broadband Processor Architecture"

+ bool

+ default n

+

+config PPC_IBM_CELL_BLADE

+ bool " IBM Cell Blade"

depends on PPC_MULTIPLATFORM && PPC64

+ select PPC_CELL

select PPC_RTAS

select MMIO_NVRAM

select PPC_UDBG_16550

+ select SPUFS_PRIV1_MMIO

config XICS

depends on PPC_PSERIES

@@ -440,7 +446,7 @@

default y

[PATCH 11/13] cell: split out board specific files

config CELL_IIC
- depends on PPC_CELL
+ depends on PPC_IBM_CELL_BLADE
bool
default y

Index: linux-2.6/arch/powerpc/configs/cell_defconfig

```
=====  
--- linux-2.6.orig/arch/powerpc/configs/cell_defconfig 2006-04-29 22:53:50.000000000 +0200  
+++ linux-2.6/arch/powerpc/configs/cell_defconfig 2006-04-29 22:54:42.000000000 +0200  
@@ -116,6 +116,7 @@  
# CONFIG_PPC_PMAC is not set  
# CONFIG_PPC_MAPLE is not set  
CONFIG_PPC_CELL=y  
+CONFIG_PPC_IBM_CELL_BLADE=y  
# CONFIG_U3_DART is not set  
CONFIG_PPC_RTAS=y  
# CONFIG_RTAS_ERROR_LOGGING is not set  
@@ -132,6 +133,8 @@  
# Cell Broadband Engine options  
#  
CONFIG_SPU_FS=m  
+CONFIG_SPU_BASE=y  
+CONFIG_SPUFS_PRIV1_MMIO=y  
CONFIG_SPUFS_MMAP=y
```


Index: linux-2.6/arch/powerpc/platforms/cell/Kconfig

```
=====  
--- linux-2.6.orig/arch/powerpc/platforms/cell/Kconfig 2006-04-29 22:53:51.000000000 +0200  
+++ linux-2.6/arch/powerpc/platforms/cell/Kconfig 2006-04-29 22:54:42.000000000 +0200  
@@ -5,11 +5,20 @@  
tristate "SPU file system"  
default m  
depends on PPC_CELL  
+ select SPU_BASE  
help  
The SPU file system is used to access Synergistic Processing  
Units on machines implementing the Broadband Processor  
Architecture.  
  
+config SPU_BASE  
+ bool  
+ default n  
+  
+config SPUFS_PRIV1_MMIO  
+ bool  
+ default n  
+  
config SPUFS_MMAP  
bool
```

[PATCH 11/13] cell: split out board specific files

depends on SPU_FS && SPARSEMEM && !PPC_64K_PAGES

Index: linux-2.6/arch/powerpc/platforms/cell/Makefile

```
=====
--- linux-2.6.orig/arch/powerpc/platforms/cell/Makefile 2006-04-29 22:54:25.000000000 +0200
+++ linux-2.6/arch/powerpc/platforms/cell/Makefile 2006-04-29 22:54:42.000000000 +0200
@@ -1,14 +1,13 @@
-obj-y += interrupt.o iommu.o setup.o spider-pic.o
-obj-y += pervasive.o
-
-obj-$(CONFIG_SMP) += smp.o
+obj-$(CONFIG_PPC_IBM_CELL_BLADE) += interrupt.o iommu.o setup.o \
+ spider-pic.o pervasive.o
+ifeq ($(CONFIG_SMP),y)
+obj-$(CONFIG_PPC_IBM_CELL_BLADE) += smp.o
+endif

# needed only when building loadable spufs.ko
-spufs-modular-$(CONFIG_SPU_FS) += spu_syscalls.o
-obj-y += $(spufs-modular-m)
-
-# always needed in kernel
-spufs-builtin-$(CONFIG_SPU_FS) += spu_callbacks.o spu_base.o spu_priv1.o
-obj-y += $(spufs-builtin-y) $(spufs-builtin-m)
+spufs-modular-$(CONFIG_SPU_FS) += spu_syscalls.o

-obj-$(CONFIG_SPU_FS) += spufs/
+obj-$(CONFIG_SPU_BASE) += spu_callbacks.o spu_base.o \
+ $(spufs-modular-m)
+obj-$(CONFIG_SPUFS_PRIV1_MMIO) += spu_priv1_mmio.o
+obj-$(CONFIG_SPU_FS) += spufs/
```

Index: linux-2.6/arch/powerpc/platforms/cell/spu_priv1.c

```
=====
--- linux-2.6.orig/arch/powerpc/platforms/cell/spu_priv1.c 2006-04-29 22:53:50.000000000 +0200
+++ /dev/null 1970-01-01 00:00:00.000000000 +0000
@@ -1,133 +0,0 @@
-/*
- * access to SPU privileged registers
- */
-#include <linux/module.h>
-
-#include <asm/io.h>
-#include <asm/spu.h>
-
-void spu_int_mask_and(struct spu *spu, int class, u64 mask)
- {
- u64 old_mask;
-
- old_mask = in_be64(&spu->priv1->int_mask_RW[class]);
- out_be64(&spu->priv1->int_mask_RW[class], old_mask & mask);
- }
-EXPORT_SYMBOL_GPL(spu_int_mask_and);
```

[PATCH 11/13] cell: split out board specific files

[PATCH 11/13] cell: split out board specific files

```
-  
-void spu_int_mask_or(struct spu *spu, int class, u64 mask)  
-  
-{  
- u64 old_mask;  
-  
- old_mask = in_be64(&spu->priv1->int_mask_RW[class]);  
- out_be64(&spu->priv1->int_mask_RW[class], old_mask | mask);  
-}  
-EXPORT_SYMBOL_GPL(spu_int_mask_or);  
-  
-void spu_int_mask_set(struct spu *spu, int class, u64 mask)  
-  
-{  
- out_be64(&spu->priv1->int_mask_RW[class], mask);  
-}  
-EXPORT_SYMBOL_GPL(spu_int_mask_set);  
-  
-u64 spu_int_mask_get(struct spu *spu, int class)  
-  
-{  
- return in_be64(&spu->priv1->int_mask_RW[class]);  
-}  
-EXPORT_SYMBOL_GPL(spu_int_mask_get);  
-  
-void spu_int_stat_clear(struct spu *spu, int class, u64 stat)  
-  
-{  
- out_be64(&spu->priv1->int_stat_RW[class], stat);  
-}  
-EXPORT_SYMBOL_GPL(spu_int_stat_clear);  
-  
-u64 spu_int_stat_get(struct spu *spu, int class)  
-  
-{  
- return in_be64(&spu->priv1->int_stat_RW[class]);  
-}  
-EXPORT_SYMBOL_GPL(spu_int_stat_get);  
-  
-void spu_int_route_set(struct spu *spu, u64 route)  
-  
-{  
- out_be64(&spu->priv1->int_route_RW, route);  
-}  
-EXPORT_SYMBOL_GPL(spu_int_route_set);  
-  
-u64 spu_mfc_dar_get(struct spu *spu)  
-  
-{  
- return in_be64(&spu->priv1->mfc_dar_RW);  
-}  
-EXPORT_SYMBOL_GPL(spu_mfc_dar_get);  
-  
-u64 spu_mfc_dsisr_get(struct spu *spu)  
-  
-{  
- return in_be64(&spu->priv1->mfc_dsisr_RW);  
-}  
-EXPORT_SYMBOL_GPL(spu_mfc_dsisr_get);
```

[PATCH 11/13] cell: split out board specific files

```
-
-void spu_mfc_dsisr_set(struct spu *spu, u64 dsisr)
-{
- out_be64(&spu->priv1->mfc_dsisr_RW, dsisr);
-}
-EXPORT_SYMBOL_GPL(spu_mfc_dsisr_set);
-
-void spu_mfc_sdr_set(struct spu *spu, u64 sdr)
-{
- out_be64(&spu->priv1->mfc_sdr_RW, sdr);
-}
-EXPORT_SYMBOL_GPL(spu_mfc_sdr_set);
-
-void spu_mfc_sr1_set(struct spu *spu, u64 sr1)
-{
- out_be64(&spu->priv1->mfc_sr1_RW, sr1);
-}
-EXPORT_SYMBOL_GPL(spu_mfc_sr1_set);
-
-u64 spu_mfc_sr1_get(struct spu *spu)
-{
- return in_be64(&spu->priv1->mfc_sr1_RW);
-}
-EXPORT_SYMBOL_GPL(spu_mfc_sr1_get);
-
-void spu_mfc_tclass_id_set(struct spu *spu, u64 tclass_id)
-{
- out_be64(&spu->priv1->mfc_tclass_id_RW, tclass_id);
-}
-EXPORT_SYMBOL_GPL(spu_mfc_tclass_id_set);
-
-u64 spu_mfc_tclass_id_get(struct spu *spu)
-{
- return in_be64(&spu->priv1->mfc_tclass_id_RW);
-}
-EXPORT_SYMBOL_GPL(spu_mfc_tclass_id_get);
-
-void spu_tlb_invalidate(struct spu *spu)
-{
- out_be64(&spu->priv1->tlb_invalidate_entry_W, 0ul);
-}
-EXPORT_SYMBOL_GPL(spu_tlb_invalidate);
-
-void spu_resource_allocation_groupID_set(struct spu *spu, u64 id)
-{
- out_be64(&spu->priv1->resource_allocation_groupID_RW, id);
-}
-EXPORT_SYMBOL_GPL(spu_resource_allocation_groupID_set);
-
-u64 spu_resource_allocation_groupID_get(struct spu *spu)
-{
```

[PATCH 11/13] cell: split out board specific files

```
- return in_be64(&spu->priv1->resource_allocation_groupID_RW);
-}
-EXPORT_SYMBOL_GPL(spu_resource_allocation_groupID_get);
-
-void spu_resource_allocation_enable_set(struct spu *spu, u64 enable)
- {
- out_be64(&spu->priv1->resource_allocation_enable_RW, enable);
-}
-EXPORT_SYMBOL_GPL(spu_resource_allocation_enable_set);
-
-u64 spu_resource_allocation_enable_get(struct spu *spu)
- {
- return in_be64(&spu->priv1->resource_allocation_enable_RW);
-}
-EXPORT_SYMBOL_GPL(spu_resource_allocation_enable_get);
```

Index: linux-2.6/arch/powerpc/platforms/cell/spu_priv1_mmio.c

=====

```
--- /dev/null 1970-01-01 00:00:00.000000000 +0000
+++ linux-2.6/arch/powerpc/platforms/cell/spu_priv1_mmio.c 2006-04-29 22:54:42.000000000 +0200
@@ -0,0 +1,133 @@
+/*
+ * access to SPU privileged registers
+ */
+#include <linux/module.h>
+
+#include <asm/io.h>
+#include <asm/spu.h>
+
+void spu_int_mask_and(struct spu *spu, int class, u64 mask)
+ {
+ u64 old_mask;
+
+ old_mask = in_be64(&spu->priv1->int_mask_RW[class]);
+ out_be64(&spu->priv1->int_mask_RW[class], old_mask & mask);
+}
+EXPORT_SYMBOL_GPL(spu_int_mask_and);
+
+void spu_int_mask_or(struct spu *spu, int class, u64 mask)
+ {
+ u64 old_mask;
+
+ old_mask = in_be64(&spu->priv1->int_mask_RW[class]);
+ out_be64(&spu->priv1->int_mask_RW[class], old_mask | mask);
+}
+EXPORT_SYMBOL_GPL(spu_int_mask_or);
+
+void spu_int_mask_set(struct spu *spu, int class, u64 mask)
+ {
+ out_be64(&spu->priv1->int_mask_RW[class], mask);
+}
+EXPORT_SYMBOL_GPL(spu_int_mask_set);
```

[PATCH 11/13] cell: split out board specific files

```
+
+u64 spu_int_mask_get(struct spu *spu, int class)
+{
+ return in_be64(&spu->priv1->int_mask_RW[class]);
+}
+EXPORT_SYMBOL_GPL(spu_int_mask_get);
+
+void spu_int_stat_clear(struct spu *spu, int class, u64 stat)
+{
+ out_be64(&spu->priv1->int_stat_RW[class], stat);
+}
+EXPORT_SYMBOL_GPL(spu_int_stat_clear);
+
+u64 spu_int_stat_get(struct spu *spu, int class)
+{
+ return in_be64(&spu->priv1->int_stat_RW[class]);
+}
+EXPORT_SYMBOL_GPL(spu_int_stat_get);
+
+void spu_int_route_set(struct spu *spu, u64 route)
+{
+ out_be64(&spu->priv1->int_route_RW, route);
+}
+EXPORT_SYMBOL_GPL(spu_int_route_set);
+
+u64 spu_mfc_dar_get(struct spu *spu)
+{
+ return in_be64(&spu->priv1->mfc_dar_RW);
+}
+EXPORT_SYMBOL_GPL(spu_mfc_dar_get);
+
+u64 spu_mfc_dsisr_get(struct spu *spu)
+{
+ return in_be64(&spu->priv1->mfc_dsisr_RW);
+}
+EXPORT_SYMBOL_GPL(spu_mfc_dsisr_get);
+
+void spu_mfc_dsisr_set(struct spu *spu, u64 dsisr)
+{
+ out_be64(&spu->priv1->mfc_dsisr_RW, dsisr);
+}
+EXPORT_SYMBOL_GPL(spu_mfc_dsisr_set);
+
+void spu_mfc_sdr_set(struct spu *spu, u64 sdr)
+{
+ out_be64(&spu->priv1->mfc_sdr_RW, sdr);
+}
+EXPORT_SYMBOL_GPL(spu_mfc_sdr_set);
+
+void spu_mfc_sr1_set(struct spu *spu, u64 sr1)
+{
```

[PATCH 11/13] cell: split out board specific files

```
+ out_be64(&spu->priv1->mfc_sr1_RW, sr1);
+}
+EXPORT_SYMBOL_GPL(spu_mfc_sr1_set);
+
+u64 spu_mfc_sr1_get(struct spu *spu)
+{
+ return in_be64(&spu->priv1->mfc_sr1_RW);
+}
+EXPORT_SYMBOL_GPL(spu_mfc_sr1_get);
+
+void spu_mfc_tclass_id_set(struct spu *spu, u64 tclass_id)
+{
+ out_be64(&spu->priv1->mfc_tclass_id_RW, tclass_id);
+}
+EXPORT_SYMBOL_GPL(spu_mfc_tclass_id_set);
+
+u64 spu_mfc_tclass_id_get(struct spu *spu)
+{
+ return in_be64(&spu->priv1->mfc_tclass_id_RW);
+}
+EXPORT_SYMBOL_GPL(spu_mfc_tclass_id_get);
+
+void spu_tlb_invalidate(struct spu *spu)
+{
+ out_be64(&spu->priv1->tlb_invalidate_entry_W, 0ul);
+}
+EXPORT_SYMBOL_GPL(spu_tlb_invalidate);
+
+void spu_resource_allocation_groupID_set(struct spu *spu, u64 id)
+{
+ out_be64(&spu->priv1->resource_allocation_groupID_RW, id);
+}
+EXPORT_SYMBOL_GPL(spu_resource_allocation_groupID_set);
+
+u64 spu_resource_allocation_groupID_get(struct spu *spu)
+{
+ return in_be64(&spu->priv1->resource_allocation_groupID_RW);
+}
+EXPORT_SYMBOL_GPL(spu_resource_allocation_groupID_get);
+
+void spu_resource_allocation_enable_set(struct spu *spu, u64 enable)
+{
+ out_be64(&spu->priv1->resource_allocation_enable_RW, enable);
+}
+EXPORT_SYMBOL_GPL(spu_resource_allocation_enable_set);
+
+u64 spu_resource_allocation_enable_get(struct spu *spu)
+{
+ return in_be64(&spu->priv1->resource_allocation_enable_RW);
+}
+EXPORT_SYMBOL_GPL(spu_resource_allocation_enable_get);
```

[PATCH 11/13] cell: split out board specific files

Index: linux-2.6/drivers/net/Kconfig

```
=====  
--- linux-2.6.orig/drivers/net/Kconfig 2006-04-29 22:53:50.000000000 +0200  
+++ linux-2.6/drivers/net/Kconfig 2006-04-29 22:54:42.000000000 +0200  
@@ -2171,7 +2171,7 @@
```

config SPIDER_NET

tristate "Spider Gigabit Ethernet driver"

- depends on PCI && PPC_CELL

+ depends on PCI && PPC_IBM_CELL_BLADE

select FW_LOADER

help

This driver supports the Gigabit Ethernet chips present on the

--

-

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/>