

Re: [patch 1/5] MMC OMAP driver

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2006-01/msg11443.html>

- *From:* Pierre Ossman <drzeus-list@xxxxxxxxxx>
 - *Date:* Tue, 31 Jan 2006 16:29:27 +0100
-

Anderson Briglia wrote:

> Adds OMAP MMC driver.

>

> Signed-off-by: Juha Yrjölä <juha.yrjola@xxxxxxxxxx>

> Signed-off-by: Tony Lindgren <tony@xxxxxxxxxx>

>

> +

> + /* Our hardware needs to know exact type */

> + switch (cmd->flags & MMC_RSP_MASK) {

> + case MMC_RSP_NONE:

> + /* resp 0 */

> + break;

> + case MMC_RSP_SHORT:

> + /* resp 1, resp 1b */

> + /* OR resp 3!! (assume this if bus is set opendrain) */

> + if (host->bus_mode == MMC_BUSMODE_OPENDRAIN)

> + resptype = 3;

> + else

> + resptype = 1;

> + break;

> + case MMC_RSP_LONG:

> + /* resp 2 */

> + resptype = 2;

> + break;

> + }

Don't do this. If you cannot figure out how the hardware handles the different types then at least have a simple switch statement over the types (and of course a failure case for unknown types). Don't try to deduce the correct mode based on things that aren't strictly related.

> +

> + /* Any data transfer means adtc type (but that information is not
> + * in command structure, so we flagged it into host struct.)

> + * However, telling bc, bcr and ac apart based on response is

> + * not foolproof:

> + * CMD0 = bc = resp0 CMD15 = ac = resp0

> + * CMD2 = bcr = resp2 CMD10 = ac = resp2

> + *

Re: [patch 1/5] MMC OMAP driver

```
> + * Resolve to best guess with some exception testing:
> + * resp0 -> bc, except CMD15 = ac
> + * rest are ac, except if opendrain
> + */
> + if (host->data) {
> + cmdtype = OMAP_MMC_CMDTYPE_ADTC;
> + } else if (resptype == 0 && cmd->opcode != 15) {
> + cmdtype = OMAP_MMC_CMDTYPE_BC;
> + } else if (host->bus_mode == MMC_BUSMODE_OPENDRAIN) {
> + cmdtype = OMAP_MMC_CMDTYPE_BCR;
> + } else {
> + cmdtype = OMAP_MMC_CMDTYPE_AC;
> + }
```

Same thing here. If you need the command types then extend the MMC layer to include those. It's not like it's some closed proprietary black box. :)

```
> +/* PIO only */
> +static void
> +mmc_omap_sg_to_buf(struct mmc_omap_host *host)
> +{
> + struct scatterlist *sg;
> +
> + sg = host->data->sg + host->sg_idx;
> + host->buffer_bytes_left = sg->length;
> + host->buffer = page_address(sg->page) + sg->offset;
> + if (host->buffer_bytes_left > host->total_bytes_left)
> + host->buffer_bytes_left = host->total_bytes_left;
> +}
> +
```

Just so you know, this isn't highmem safe. Perhaps that will never be an issue on OMAP, but still. The solution is not obvious, but it's being discussed under the thread named "How to map high memory for block io".

```
> +static irqreturn_t mmc_omap_irq(int irq, void *dev_id, struct pt_regs *regs)
> +{
> + struct mmc_omap_host * host = (struct mmc_omap_host *)dev_id;
> + u16 status;
> + int end_command;
> + int end_transfer;
> + int transfer_error;
> +
> + if (host->cmd == NULL && host->data == NULL) {
> + status = OMAP_MMC_READ(host->base, STAT);
> + printk(KERN_INFO "MMC%d: Spurious interrupt 0x%04x\n", host->id, status);
> + if (status != 0) {
> + OMAP_MMC_WRITE(host->base, STAT, status);
> + OMAP_MMC_WRITE(host->base, IE, 0);
> + }
> + return IRQ_HANDLED;
```

Re: [patch 1/5] MMC OMAP driver

```
> + }  
> +
```

Why don't you let the kernel handle spurious interrupts? You sure that OMAP will never get the ability to share the MMC interrupt with something else? :)

```
> + if (status & OMAP_MMC_STAT_CMD_TOUT) {  
> + /* Timeouts are routine with some commands */  
> + if (host->cmd) {  
> + if (host->cmd->opcode != MMC_ALL_SEND_CID &&  
> + host->cmd->opcode != MMC_SEND_OP_COND &&  
> + host->cmd->opcode != MMC_APP_CMD &&  
> + !mmc_omap_cover_is_open(host))  
> + printk(KERN_ERR "MMC%d: Command timeout, CMD%d\n",  
> + host->id, host->cmd->opcode);  
> + host->cmd->error |= MMC_ERR_TIMEOUT;  
> + end_command = 1;  
> + }  
> + }
```

Bad! This is a text book example of something that should be done in the MMC layer, not the drivers. Same thing for the other error messages.

```
> + if (status & OMAP_MMC_STAT_CARD_ERR) {  
> + if (host->cmd && host->cmd->opcode == MMC_STOP_TRANSMISSION) {  
> + u32 response = OMAP_MMC_READ(host->base, RSP6)  
> + | (OMAP_MMC_READ(host->base, RSP7) << 16);  
> + /* STOP sometimes sets must-ignore bits */  
> + if (!(response & (R1_CC_ERROR  
> + | R1_ILLEGAL_COMMAND  
> + | R1_COM_CRC_ERROR))) {  
> + end_command = 1;  
> + continue;  
> + }  
> + }
```

Are you sure this only happens for MMC_STOP_TRANSMISSION or for any stop command?

```
> +static inline int is_broken_card(struct mmc_card *card)  
> +{  
> + int i;  
> + struct mmc_cid *c = &card->cid;  
> + static const struct broken_card_cid {  
> + unsigned int manfid;  
> + char prod_name[8];  
> + unsigned char hwrev;  
> + unsigned char fwrev;  
> + } broken_cards[] = {  
> + { 0x00150000, "\x30\x30\x30\x30\x30\x30\x15\x00", 0x06, 0x03 },
```

Re: [patch 1/5] MMC OMAP driver

Re: [patch 1/5] MMC OMAP driver

```
> + };
> +
> + for (i = 0; i < sizeof(broken_cards)/sizeof(broken_cards[0]); i++) {
> + const struct broken_card_cid *b = broken_cards + i;
> +
> + if (b->manfid != c->manfid)
> + continue;
> + if (memcmp(b->prod_name, c->prod_name, sizeof(b->prod_name)) != 0)
> + continue;
> + if (b->hwrev != c->hwrev || b->fwrev != c->fwrev)
> + continue;
> + return 1;
> + }
> + return 0;
> + }
```

Again, not something you do in the driver.

```
> +
> + /* Some cards (vendor left unnamed to protect the guilty) seem to
> + * require this delay after power-up. Otherwise we'll get mysterious
> + * data timeouts.
> + */
```

Same here.

Rgds
Pierre

-

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

• **References:**

- ◆ [\[patch 1/5\] MMC OMAP driver](#)
◇ From: Anderson Briglia

- Prev by Date: [\[PATCH\] OOM kill: children accounting](#)
- Next by Date: [Re: \[patch 5/5\] MMC OMAP driver](#)
- Previous by thread: [\[patch 1/5\] MMC OMAP driver](#)
- Next by thread: [\[patch 5/5\] MMC OMAP driver](#)
- Index(es):
 - ◆ [Date](#)
 - ◆ [Thread](#)

Re: [patch 1/5] MMC OMAP driver