Kernel driver i2c-ocores

Supported adapters:

- OpenCores.org I2C controller by Richard Herveille (see datasheet link) https://opencores.org/project/i2c/overview

Author: Peter Korsgaard <peter@korsgaard.com>

Description

i2c-ocores is an i2c bus driver for the OpenCores.org I2C controller IP core by Richard Herveille.

Usage

i2c-ocores uses the platform bus, so you need to provide a struct platform_device with the base address and interrupt number. The dev.platform_data of the device should also point to a struct ocores_i2c_platform_data (see linux/platform_data/i2c-ocores.h) describing the distance between registers and the input clock speed. There is also a possibility to attach a list of i2c_board_info which the i2c-ocores driver will add to the bus upon creation.

E.G. something like:

```
static struct resource ocores_resources[] = {
	[0] = {
		.start  = MYI2C_BASEADDR,
		.end    = MYI2C_BASEADDR + 8,
		.flags  = IORESOURCE_MEM,
	},
	[1] = {
		.start  = MYI2C_IRQ,
		.end    = MYI2C_IRQ,
		.flags  = IORESOURCE_IRQ,
	},
};

/* optional board info */
struct i2c_board_info ocores_i2c_board_info[] = {
	{
		I2C_BOARD_INFO("tsc2003", 0x48),
		.platform_data = &tsc2003_platform_data,
		.irq = TSC_IRQ
	},
	{
		I2C_BOARD_INFO("adv7180", 0x42 >> 1),
		.irq = ADV_IRQ
	}
};

static struct ocores_i2c_platform_data myi2c_data = {
	.regstep        = 2,            /* two bytes between registers */
	.clock_khz      = 50000,        /* input clock of 50MHz */
	.devices        = ocores_i2c_board_info, /* optional table of devices */
	.num_devices    = ARRAY_SIZE(ocores_i2c_board_info), /* table size */
};

static struct platform_device myi2c = {
	.name                   = "ocores-i2c",
	.dev = {
		.platform_data  = &myi2c_data,
	},
	.num_resources          = ARRAY_SIZE(ocores_resources),
	.resource               = ocores_resources,
};
```