Kernel driver ds2490

Supported chips:

- Maxim DS2490 based

Author: Evgeniy Polyakov <johnpol@2ka.mipt.ru>

Description

The Maxim/Dallas Semiconductor DS2490 is a chip which allows to build USB <-> W1 bridges.

DS9490(R) is a USB <-> W1 bus master device which has 0x81 family ID integrated chip and DS2490 low-level operational chip.

Notes and limitations.

- The weak pullup current is a minimum of 0.9mA and maximum of 6.0mA.
- The 5V strong pullup is supported with a minimum of 5.9mA and a maximum of 30.4 mA. (From DS2490.pdf)
- The hardware will detect when devices are attached to the bus on the next bus (reset?) operation, however only a message is printed as the core w1 code doesn't make use of the information. Connecting one device tends to give multiple new device notifications.
- The number of USB bus transactions could be reduced if w1_reset_send was added to the API. The name is just a suggestion. It would take a write buffer and a read buffer (along with sizes) as arguments. The ds2490 block I/O command supports reset, write buffer, read buffer, and strong pullup all in one command, instead of the current 1 reset bus, 2 write the match rom command and slave rom id, 3 block write and read data. The write buffer needs to have the match rom command and slave rom id prepended to the front of the requested write buffer, both of which are known to the driver.
- The hardware supports normal, flexible, and overdrive bus communication speeds, but only the normal is supported.
- The registered w1_bus_master functions don't define error conditions. If a bus search is in progress and the ds2490 is removed it can produce a good amount of error output before the bus search finishes.
- The hardware supports detecting some error conditions, such as short, alarming presence on reset, and no presence on reset, but the driver doesn't query those values.
- The ds2490 specification doesn't cover short bulk in reads in detail, but my observation is if fewer bytes are requested than are available, the bulk read will return an error and the hardware will clear the entire bulk in buffer. It would be possible to read the maximum buffer size to not run into this error condition, only extra bytes in the buffer is a logic error in the driver. The code should should match reads and writes as well as data sizes. Reads and writes are serialized and the status verifies that the chip is idle (and data is available) before the read is executed, so it should not happen.
- Running x86_64 2.6.24 UHCI under qemu 0.9.0 under x86_64 2.6.22-rc6 with a OHCI controller, ds2490 running in the guest would operate normally the first time the module was loaded after qemu attached the ds2490 hardware, but if the module was unloaded, then reloaded most of the time one of the bulk out or in, and usually the bulk in would fail. qemu sets a 50ms timeout and the bulk in would timeout even when the status shows data available. A bulk out write would show a successful completion, but the ds2490 status register would show 0 bytes written. Detaching qemu from the ds2490 hardware and reattaching would clear the problem. usbmon output in the guest and host did not explain the problem. My guess is a bug in either qemu or the host OS and more likely the host OS.

03-06-2008 David Fries <David@Fries.net>