ETHER_DC21041_LITEON

The ETHER_DC21041_LITEON module is a driver for a PCIbus ethernet card using the LiteOn clone of the DC21041 ‘Tulip’ chip (the PNIC 82c168). The actual board which this code drives is the Kingston KNE110TX ethernet interface card.

Process Information

Prototype Name	ether_dc21041
Process Priority	driver-level
Process Name	does not matter as long as it matches the URLs used within the system to open the device.

Module Options

ETHER_BCAST_TRACE

If this symbol is defined, *in addition to* the packet-trace option, then all broadcast packets received by the driver will also be dsplayed.

ETHER_PKT_TRACE

If this symbol is defined, then all packets sent, and all non-broadcast packets received by the driver will be displayed on the polled-I/O serial line using *rome_kprintf*.

Target File Definitions

ETHER_DC21041_MAX_CARDS

The maximum number of dc21041 ethernet cards that might be in the system. The default value is 1.

Process Operation

The initialisation routine locates all the LiteOn cards in the system and initialises the driver data structures for them. The station (MAC) address is set from the board’s EEPROM stable storage and the auto-negotiation mode is used to detect the link speed and mode. Only CAT-5 compatible modes will be installed at the PHY layer (10/100 Mbps half or full duplex). The routine sets up the receiver ring for the tulip chip, installs the interrupt handler and starts the transmitter and receiver process on the chip. Finally, the initialisation routine formats and transmits the ‘setup’ packet to configure and prepare the board for normal operation.

The module has a queue handler and main process. and accepts the *NETADDRESS* message and the messages defined in the *Standard* messageset for dataflows into and out of the driver with the following processing:

CLOSE messages are handled by the main process. Any outstanding queued read requests are returned to the sender with error codes, as are all event requests sent to the driver.

EVENT messages are added to the file's event queue. Currently, the driver does not generate any events (for example for link failure), so the messages are just held until the file is closed.

FETMBLK messages are added to the file's read queue in the queue handler. The interrupt handler responds to packets in the receive ring by passing them upwards on a file open for that ethertype as replies to FETMBLK (or GETMBLK) messages.

FLUSH messages are replied to immediately from the main process, because the drvier does not buffer data internally.

GETMBLK messages are added to the file's read queue in the queue handler, then treated as FETMBLK messages above.

NEWMBLK messages return a 4k DMA-able page to the caller from the queue handler.

NETADDRESS messages return the MAC address of the interface identified by the *dest_context* field of the message (i.e. the message must be sent following an *OPEN* to that device). The *events_generated* field is set *FALSE* as the driver does not generate any events.

OPEN messages are handled in the main process. The *port* field of the URL contains the ethernet packet type in the lower 16 bits and the card number in the upper 16 bits. The process returns a value in the *dest_context* field which must be supplied on subsequent messages.

OUTMBLK messages are added to the tail of the output ring in the queue handler if there is space, and the transmit process enabled if required, otherwise they are added to the queue of messages awaiting transmission. The *b_rptr* field of the message should point to the start of the ethernet header. The interrupt handler checks for successful transmission, and requeues failed packets, before freeing PUTMBLK buffers and sending the reply. Messages are moved from the output queue to the output ring as the ring empties.

PUTMBLK messages with zero length cause the message's buffer to be returned to the buffer pool in the queue handler, otherwise the message is processed as with OUTMBLK above. When all the contents have been transmitted, the interrupt handler frees the message buffer.

RETMBLK messages free the allocated page within the queue handler and are replied to immediately.

Debug Support

ether_dc21041_stats

void *ether_dc21041_stats*(**void**)

The *ether_dc21041_stats* routine prints the traffic and error statistics for each interface onto the polled-mode I/O interface.