Tulip Driver User's Guide

Jeff Garzik

jgarzik@mandrakesoft.com

Tulip Driver User's Guide

by Jeff Garzik

Table of Contents

Chapter 1. Introduction

The Tulip Ethernet Card Driver is maintained by Jeff Garzik (<`jgarzik@mandrakesoft.com`>).

The Tulip driver was developed by Donald Becker and changed by Jeff Garzik, Takashi Manabe and a cast of thousands.

For 2.4.x and later kernels, the Linux Tulip driver is available at http://sourceforge.net/projects/tulip/

This driver is for the Digital "Tulip" Ethernet adapter interface. It should work with most DEC 21*4*-based chips/ethercards, as well as with work-alike chips from Lite-On (PNIC) and Macronix (MXIC) and ASIX.

The original author may be reached as becker@scyld.com, or C/O Scyld Computing Corporation, 410 Severn Ave., Suite 210, Annapolis MD 21403

Additional information on Donald Becker's tulip.c is available at http://www.scyld.com/network/tulip.html

Chapter 2. Driver Compatibility

This device driver is designed for the DECchip "Tulip", Digital's single-chip ethernet controllers for PCI (now owned by Intel). Supported members of the family are the 21040, 21041, 21140, 21140A, 21142, and 21143. Similar work-alike chips from Lite-On, Macronics, ASIX, Compex and other listed below are also supported.

These chips are used on at least 140 unique PCI board designs. The great number of chips and board designs supported is the reason for the driver size and complexity. Almost of the increasing complexity is in the board configuration and media selection code. There is very little increasing in the operational critical path length.

Chapter 3. Board-specific Settings

PCI bus devices are configured by the system at boot time, so no jumpers need to be set on the board. The system BIOS preferably should assign the PCI INTA signal to an otherwise unused system IRQ line.

Some boards have EEPROMs tables with default media entry. The factory default is usually "autoselect". This should only be overridden when using transceiver connections without link beat e.g. 10base2 or AUI, or (rarely!) for forcing full-duplex when used with old link partners that do not do autonegotiation.

Chapter 4. Driver Operation

4.1. Ring buffers

The Tulip can use either ring buffers or lists of Tx and Rx descriptors. This driver uses statically allocated rings of Rx and Tx descriptors, set at compile time by RX/TX_RING_SIZE. This version of the driver allocates skbuffs for the Rx ring buffers at open() time and passes the skb->data field to the Tulip as receive data buffers. When an incoming frame is less than RX_COPYBREAK bytes long, a fresh skbuff is allocated and the frame is copied to the new skbuff. When the incoming frame is larger, the skbuff is passed directly up the protocol stack and replaced by a newly allocated skbuff.

The RX_COPYBREAK value is chosen to trade-off the memory wasted by using a full-sized skbuff for small frames vs. the copying costs of larger frames. For small frames the copying cost is negligible (esp. considering that we are pre-loading the cache with immediately useful header information). For large frames the copying cost is non-trivial, and the larger copy might flush the cache of useful data. A subtle aspect of this choice is that the Tulip only receives into longword aligned buffers, thus the IP header at offset 14 isn't longword aligned for further processing. Copied frames are put into the new skbuff at an offset of "+2", thus copying has the beneficial effect of aligning the IP header and preloading the cache.

4.2. Synchronization

The driver runs as two independent, single-threaded flows of control. One is the send-packet routine, which enforces single-threaded use by the dev->tbusy flag. The other thread is the interrupt handler, which is single threaded by the hardware and other software.

The send packet thread has partial control over the Tx ring and 'dev->tbusy' flag. It sets

the tbusy flag whenever it's queuing a Tx packet. If the next queue slot is empty, it clears the tbusy flag when finished otherwise it sets the 'tp->tx_full' flag.

The interrupt handler has exclusive control over the Rx ring and records stats from the Tx ring. (The Tx-done interrupt can't be selectively turned off, so we can't avoid the interrupt overhead by having the Tx routine reap the Tx stats.) After reaping the stats, it marks the queue entry as empty by setting the 'base' to zero. Iff the 'tp->tx_full' flag is set, it clears both the tx_full and tbusy flags.

Chapter 5. Errata

The old DEC databooks were light on details. The 21040 databook claims that CSR13, CSR14, and CSR15 should each be the last register of the set CSR12-15 written. Hmmm, now how is that possible?

The DEC SROM format is very badly designed not precisely defined, leading to part of the media selection junkheap below. Some boards do not have EEPROM media tables and need to be patched up. Worse, other boards use the DEC design kit media table when it isn't correct for their board.

We cannot use MII interrupts because there is no defined GPIO pin to attach them. The MII transceiver status is polled using an kernel timer.

Chapter 6. Driver Change History

6.1. Version 0.9.14 (February 20, 2001)

- Fix PNIC problems (Manfred Spraul)
- Add new PCI id for Accton comet
- Support Davicom tulips
- Fix oops in eeprom parsing
- Enable workarounds for early PCI chipsets
- IA64, hppa csr0 support
- Support media types 5, 6
- Interpret a bit more of the 21142 SROM extended media type 3
- Add missing delay in eeprom reading

6.2. Version 0.9.11 (November 3, 2000)

- Eliminate extra bus accesses when sharing interrupts (prumpf)
- Barrier following ownership descriptor bit flip (prumpf)
- Endianness fixes for >14 addresses in setup frames (prumpf)
- Report link beat to kernel/userspace via netif_carrier_*. (kuznet)
- Better spinlocking in set_rx_mode.
- Fix I/O resource request failure error messages (DaveM catch)

- Handle DMA allocation failure.

6.3. Version 0.9.10 (September 6, 2000)

- Simple interrupt mitigation (via jamal)
- More PCI ids

6.4. Version 0.9.9 (August 11, 2000)

- More PCI ids

6.5. Version 0.9.8 (July 13, 2000)

- Correct signed/unsigned comparison for dummy frame index
- Remove outdated references to struct enet_statistics

6.6. Version 0.9.7 (June 17, 2000)

- Timer cleanups (Andrew Morton)
- Alpha compile fix (somebody?)

6.7. Version 0.9.6 (May 31, 2000)

- Revert 21143-related support flag patch
- Add HPPA/media-table debugging printk

6.8. Version 0.9.5 (May 30, 2000)

- HPPA support (willy@puffingroup)
- CSR6 bits and tulip.h cleanup (Chris Smith)
- Improve debugging messages a bit
- Add delay after CSR13 write in t21142_start_nway
- Remove unused ETHER_STATS code
- Convert 'extern inline' to 'static inline' in tulip.h (Chris Smith)
- Update DS21143 support flags in tulip_chip_info[]
- Use spin_lock_irq, not _irqsave/restore, in tulip_start_xmit()
- Add locking to set_rx_mode()
- Fix race with chip setting DescOwned bit (Hal Murray)
- Request 100% of PIO and MMIO resource space assigned to card
- Remove error message from pci_enable_device failure

6.9. Version 0.9.4.3 (April 14, 2000)

- mod_timer fix (Hal Murray)
- PNIC2 resuscitation (Chris Smith)

6.10. Version 0.9.4.2 (March 21, 2000)

- Fix 21041 CSR7, CSR13/14/15 handling
- Merge some PCI ids from tulip 0.91x
- Merge some HAS_xxx flags and flag settings from tulip 0.91x
- asm/io.h fix (submitted by many) and cleanup
- s/HAS_NWAY143/HAS_NWAY/
- Cleanup 21041 mode reporting
- Small code cleanups

6.11. Version 0.9.4.1 (March 18, 2000)

- Finish PCI DMA conversion (davem)
- Do not netif_start_queue() at end of tulip_tx_timeout() (kuznet)
- PCI DMA fix (kuznet)
- eeprom.c code cleanup
- Remove Xircom Tulip crud