Intel e1000 Ethernet Controller Driver

Ivan Delalande

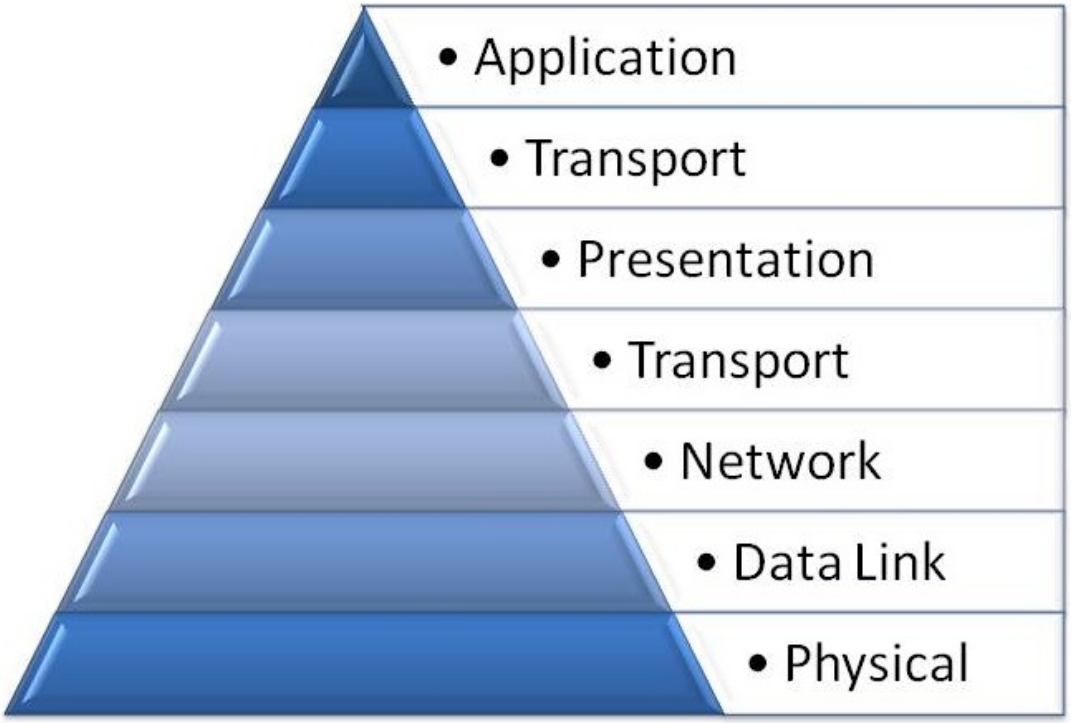
colona@lse.epita.fr
http://lse.epita.fr

February 12, 2013

Plan

Plan

Motivation



- Write a network stack for STOS.
  - Handle data link, network and transport layers.
  - Provide network syscalls: `socket`, `listen`, `connect`...

Plan

- PCI: Peripheral Component Interconnect,
- Bus to connect devices to the system.
- Hierarchical device organization:
  - Bus,
  - Device,
  - Function,
  - Register.

Device Discovery

- Bus scan to find the location of the device,
- Devices identified by their `Vendor ID` and `Device ID`
  - eg: `0x8086` ; `0x100E`

Device Structure

- Various useful information.
- Command register (bus master, address spaces, interrupt disable. . .),
- BAR: base address registers,
  - I/O address space,
  - Memory address space.

Communication with the device

- Memory-mapped I/O
  - Allow addressing devices' memory or registers as a chunk of the system memory.
- DMA: Direct Memory Access
  - Memory operations without the direct intervention of the CPU,
  - Devices that master the PCI bus can initiate memory transfer by themselves.
- PCI Interrupts
  - Four interrupt lines shared by all PCI devices.

Plan

Presentation

- Common gigabit ethernet controller by Intel with a (pretty) good documentation,
- Provide (too) many feature to reduce kernel's work:
  - Link auto-negocitation,
  - Filters IP addresses,
  - Compute IP packets' checksum,
  - Tries to limit the number of interrupts,
  - Provide effective shutdown method (a.k.a. *Packet of Death*)...
- Full customization of LEDs' blinking!

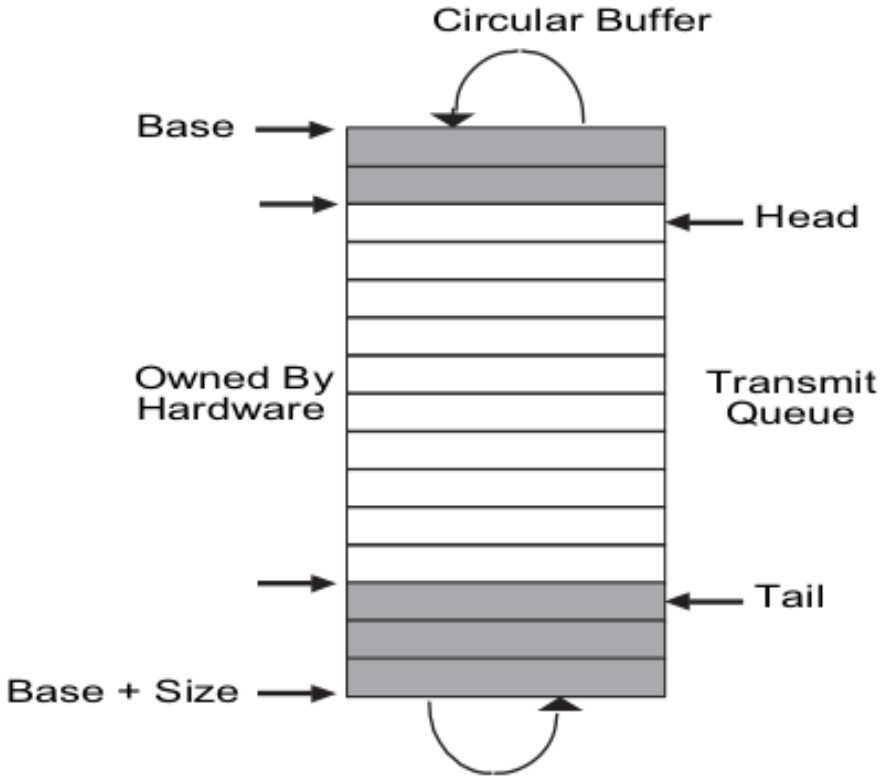
PCI Configuration

- Set the bus master bit to allow DMA from the controller,
- Retrieve the base address of the configuration register.

Controller Initialization

- Set configuration registers to appropriate values: MAC address, transmission modes. . .
- Find the proper way to read the EEPROM to get various information (MAC address, . . .),
- Allocate memory for network frames transmission and reception.

Frame transmission and reception



- Two descriptor ring structures that point to frame buffers,
- Software moves *Descriptor Tail* and hardware moves *Descriptor Head*.
- Signaling from the controller via PCI interrupts.

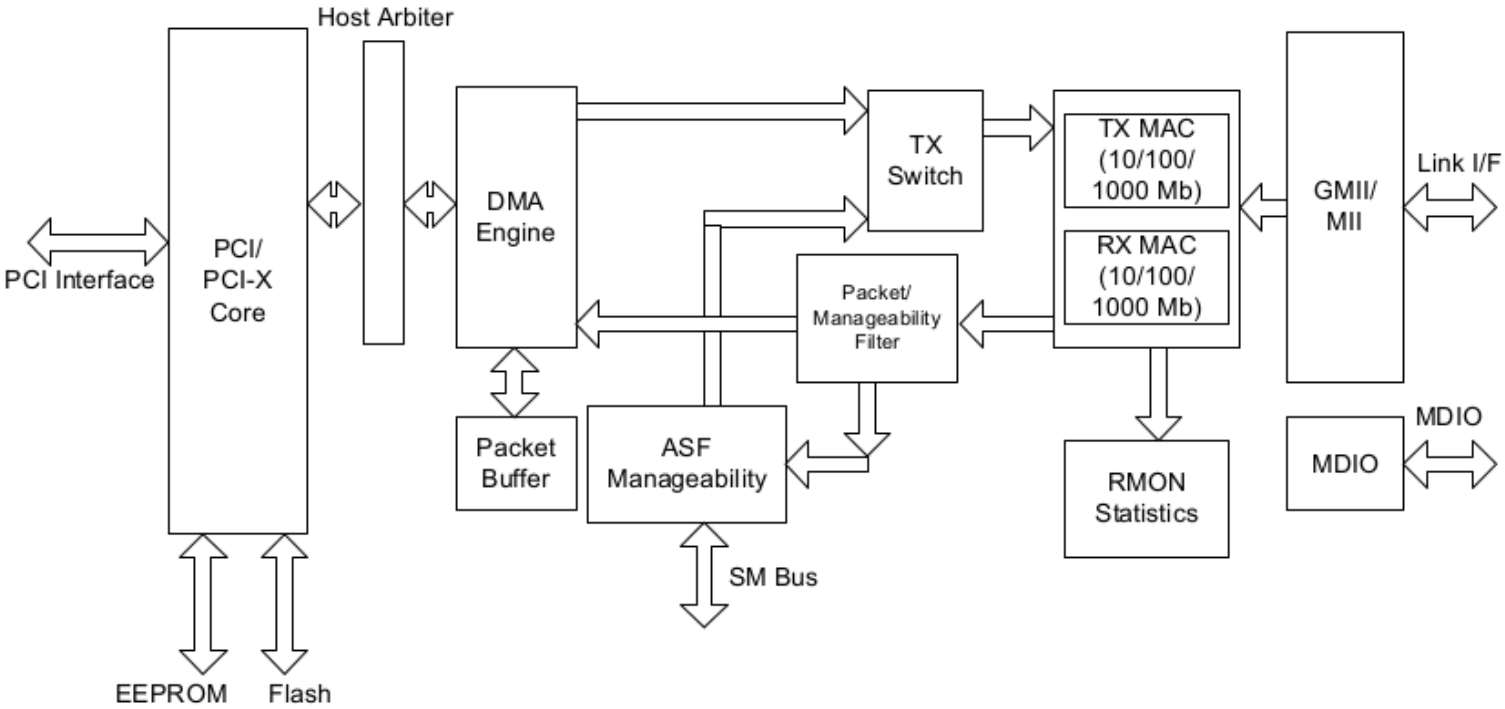
Plan

4 Conclusion

Next steps

- Process and verify the frames: check of the header, collisions detection. . .
- Proceed with the remaining layers of the network stack: network and transport.

Questions?


Host Arbiter
PCI Interface
PCI/
PCI-X
Core
DMA
Engine
TX
Switch
TX MAC
(10/100/
1000 Mb)
RX MAC
(10/100/
1000 Mb)
GMII/
MII
Link I/F
Packet/
Manageability
Filter
Packet
Buffer
ASF
Manageability
RMON
Statistics
MDIO
MDIO
SM Bus
EEPROM
Flash