Improving MII PHY

2014/03/16
Masanobu SAITO
msaitoh@netbsd.org
MII PHY problem

- IEEE 802.3 clause 45 defines MDIO access interface.

- Defined in
  - 802.3 2009
  - 802.3at
  - 802.3av
  - 802.3az

- A lot of registers have important information
  - Status and statistics
  - Gold mine!
EEE and clause 45

• Nowadays’s new Ethernet chips have IEEE 802.3az Energy Efficiency Ethernet (EEE).
• EEE related registers are accessible by clause 45’s interface
• Almost all Ethernet drivers which have EEE function have their own definitions and access functions.
  – A lot of code duplication!
Definitions and functions

• [http://cvsweb.netbsd.org/bsdweb.cgi/src/sys/dev/mii/mdio.h](http://cvsweb.netbsd.org/bsdweb.cgi/src/sys/dev/mii/mdio.h)

• Written by me 1 year ago.

• Only each register names and the number are written. Total number of register is more than 300 😞

• Not written yet
  – The bit definitions...
  – functions to help accessing a register
Atomic access problem

• Some registers are accessed with two read/write
  – e.g. Page select and access

• Those two accesses should be done atomically.

• Some Ethernet controller has a semaphore for BMC and CPU.

• If a register is accessed by both BMC and CPU without acquiring the simultaneously, the accesses might fail.

Dangerous!
Dirty solution

• For some drivers, PHY access method for each different type of PHY are defined and used with semaphore access.
  – e.g. number of variations of read/write access functions for e1000 in FreeBSD are fifteen.
  – Please check by yourself why so many functions are provided 😊
Better solutions?

- Provide lock/unlock entry point in MII layer.
- A driver sets functions to lock/unlock PHY.
Another problem (not MII)

- If_media Options word is likely to be fully used.

```c
/*
 * if_media Options word:
 *   Bits  Use
 *   ----  ------
 *  0-4   Media subtype       MAX SUBTYPE == 31!
 *  5-7   Media type
 *  8-15  Type specific options
 * 16-18  Mode (for multi-mode devices)
 * 19     RFU
 *20-27  Shared (global) options
 *28-31  Instance
*/
#define IFM_ETHER       0x00000020
#define IFM_10_T        3               /* 10BaseT - RJ45 */
// (snip)
#define IFM_40G_LR4     30              /* 40GBase-LR4 */
```

How should we solve this problem?