Phylink and SFP: Going Beyond 1G Copper

Andrew Lunn

andrew@lunn.ch

LPC 2018
Purpose of this Talk

To raise awareness of MAC driver writers of the Phylink and SFP subsystems, and what problems they solve.

Anybody writing a MAC driver for >1Gbps, or making use of an SFP should use it.
Recent new MAC drivers

- Intel IGC: 2.5G
- Freescale DPAA: 10G
- Aquantia AQC111 USB dongle: 2.5G, 5G.
- DEC TURBOchannel FDDI, 100Mbps
Solidrun Clearfog
Solidrun MACCHIATObin

- DDR4 DIMM
- TDM Header
- UART 1
- MCI
- DC Jack
- ATX Power
- Boot Selection
- Freq Selection
- 12V GND Pair
- SATA Ports
- SD Card
- MicroUSB
- USB 3.0
- RJ45 1GbE
- SFP+ 10GbE / RJ45
- SFP 2.5GbE
- PCIe 4x
- UART 2
- JTAG Header
- USB 2.0 Header
- FAN Header
- Reset Button
- Reset Header
New to Embedded Systems – 10G and SFP

Russell King was asked to add mainline support for these two boards

- Clearfog: Maybe first embedded Linux with an SFP, controlled by Linux?
- MACCHIATObin: Maybe first embedded Linux with 10G and SFP+, controlled by Linux?

Clearly not the first 10G or SFP Linux board. But controlled by Linux, not firmware?

Linux had no core support for SFPs, or 10G PHYs
SFP- Small Form Factor, Pluggable

Cage and Module for fiber or copper RJ45.

SERDES data plane

i2c control plane, similar to AT24 EEPROM

GPIO controls:
  - LOS, TX disable, TX Fault, Module present
SFP block diagram, Fiber
When is an SFP Up?

RS - Reconciliation Sublayer – Glue between MAC and PCS
PCS – Physical Coding Subsystem – AKA SERDES

Link up = !LOS && PCS SYNC
SFP SERDES Configuration

- SFP EEPROM contains max baudrate, eg 4.2Gbps
- SFP driver determines 1000Base-X, 2500Base-X
- MAC needs to validate it can actually do this
- No Auto-neg. MAC needs to be configured via ethtool to 1000Base-X or 2500Base-X.
When is a Multi-G Link Up?

RS - Reconciliation Sublayer – Glue between MAC and PCS
PCS – Physical Coding Sybsystem – AKA SERDES
PMA – Physical Medium Attachment
PMD – Physical Medium Dependent
When is a Multi-G Link Up?

1) Auto-neg Completes, 2500Base-T decided upon
2) PHY PCS configured to 2500Base-X
3) MAC PCS configured to 2500Base-X
4) MAC PCS Syncs

=> Link is up.
Phylib API

Classic API between MAC and PHY

- `struct phy_device`
- `phy_connect()`, `of_phy_connect()`, `phy_disconnect()`
- `phy_start()`, `phy_stop()`
- `adjust_link()` callback for link up/down, auto-neg

Works great for 10/100/1000 Half/Full Copper PHYs
Limitations of phylib

Only supports Copper PHYs using MDIO

Copper PHYs are assumed to be cold plug

Little dynamic behavior:
- Link up, link down
- Speed, duplex, Pause, EEE

MAC is not really involved
Dynamic behavior of SPFs and PHYs

Module can be hot-plugged into the cage

MAC-SFP/PHY connection depends on Module and link partner, MAC and PHY need to negotiate

- 1000Base-X for 1Gbps Fiber
- SGMII for 1Gbps Copper
- 2500Base-X for 2.5Gbps Fiber or Copper
- 10GBase-X for 10Gbps Fiber or Copper
Phylink API 1/2

struct phylink
phylink_create(), phylink_destroy()
phylink_connect_phy(),
phylink_of_connect_phy(),
phylink_disconnect()
phylink_start(), phylink_stop()

Very similar to phylib
phylink_mac_change()
struct phylink_mac_ops {
    void (*validate)(struct net_device *ndev,
                     unsigned long *supported,
                     struct phylink_link_state *state);
    int (*mac_link_state)(struct net_device *ndev,
                           struct phylink_link_state *state);
    void (*mac_config)(struct net_device *ndev,
                        unsigned int mode,
                        const struct phylink_link_state *state);
    void (*mac_an_restart)(struct net_device *ndev);
    void (*mac_link_down)(struct net_device *ndev,
                          unsigned int mode,
                          phy_interface_t interface);
    void (*mac_link_up)(struct net_device *ndev,
                        unsigned int mode,
                        phy_interface_t interface,
                        struct phy_device *phy);
};
Good examples, etc

- Marvell MVNETA
- DSA and mv88e6xxx, bcm_sf2
- mvpp2 – still WIP

SFP Freebies

```bash
# ethtool --module-info sff2

Identifier                        : 0x02 (module soldered to motherboard)
Extended identifier               : 0x04 (GBIC/SFP defined by 2-wire interface ID)
Connector                         : 0x07 (LC)
Transceiver codes                 : 0x04 0x00 0x00 0x02 0x12 0x00 0x01 0xf5
Transceiver type                  : Infiniband: 1X LX
Encoding                          : 0x01 (8B/10B)
BR, Nominal                       : 1200MBd
Rate identifier                   : 0x00 (unspecified)
Length (SMF,km)                   : 25km
Length (SMF)                      : 25000m
Length (50um)                     : 0m
Length (62.5um)                   : 1000m
Laser wavelength                  : 1550nm
Vendor name                       : COTSWORKS
Vendor OUI                        : 00:00:00
Vendor PN                         : SFBG53DRAP
Laser bias current                : 12.264 mA
Laser output power                : 0.2760 mW / -5.59 dBm
Module temperature                : 30.62 degrees C / 87.12 degrees F
Module voltage                    : 3.2304 V
```
SFP Freebies

HWMON Sensors

in0: +3.29 V (crit min = +2.90 V, min = +3.00 V) (max = +3.60 V, crit max = +3.70 V)
temp1: +33.0°C (low = -5.0°C, high = +80.0°C) (crit low = -10.0°C, crit = +85.0°C)
power1: 1000.00 nW (max = 794.00 uW, min = 50.00 uW) ALARM (LCRIT) (lcrit = 40.00 uW, crit = 1000.00 uW)
curr1: +0.00 A (crit min = +0.00 A, min = +0.00 A) ALARM (LCRIT, MIN) (max = +0.01 A, crit max = +0.01 A)
Go out there and use it

- Please submit MAC drivers using Phylink, not firmware.
- Please submit more 10G PHY drivers

And ask me questions

(now or over a beer later)