# Open Source Tooling for PHY Management and Testing

Lee Trager
FBNIC Software Engineer
Netdev 0x19
Zagreb, Croatia
March 2025

Meta

# Agenda

01 Overview

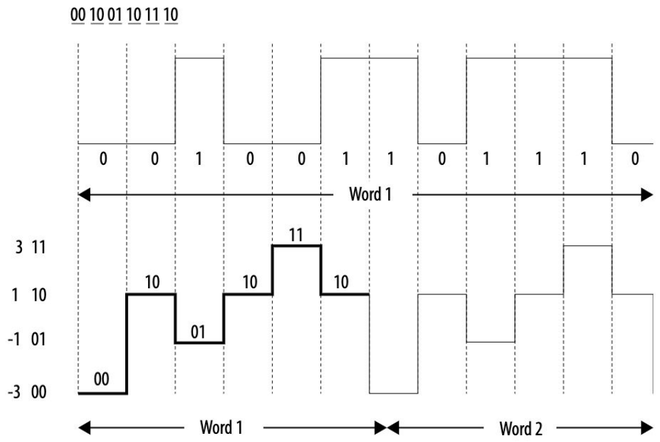
02 Vendor tools today

03 Netdev 0x18 Recap

04 ethtool integration

# TX Coefficients

- Modern Ethernet uses pulse-amplitude modulation(PAM) to achieve a high rate of speed
- Allows increasing data being sent by defining ranges for different combinations of bits.
- Ranges differ based on mode(PAM4/NRZ)
- Values are typically static and seldom change
- Hardware interoperability testing requires values to be evaluated, tested, and potentially modified




# PRBS Testing

- Pseudo random binary sequence(PRBS) testing sends or receives a random binary sequence to the device
- Used to perform data integrity tests on the link
- Error counters are used to determine link stability, PRBS test itself performs no checks
- Users determines when to start and stop a testing

PRBS7 = $x^7 + x^6 + 1$

PRBS9 = $x^9 + x^5 + 1$

PRBS11 = $x^{11} + x^9 + 1$

PRBS13 = $x^{13} + x^{12} + x^2 + x + 1$

PRBS15 = $x^{15} + x^{14} + 1$

PRBS20 = $x^{20} + x^3 + 1$

PRBS23 = $x^{23} + x^{18} + 1$

PRBS31 = $x^{31} + x^{28} + 1$

# NVIDIA – Getting TX Coefficients

```
[root@localhost ~]# mstlink -d 19:00.0 --show_serdes_tx --advance

Operational Info
----------------
State                       : Active
Physical state                  : ETH_AN_FSM_ENABLE
Speed                        : 400G
Width                        : 8x
FEC                          : Standard_RS-FEC - (544,514)
Loopback Mode                   : No Loopback
Auto Negotiation                : ON

Supported Info
--------------
Enabled Link Speed (Ext.)       : 0x000097f2 (400G_8X,200G_4X,100G_2X,100G_4X,50G_1X,50G_2X,40G,25G,10G,1G)
Supported Cable Speed (Ext.)    : 0x000097fe
(400G_8X,200G_4X,100G_2X,100G_4X,50G_1X,50G_2X,40G,25G,10G,5G,2.5G,1G)

Troubleshooting Info
--------------------
Status Opcode                   : 0
Group Opcode                    : N/A
Recommendation                     : No issue was observed.

Tool Information
----------------
Firmware Version                : 28.33.2018
amBER Version                   : 2.00
MSTFLINT Version                   : mstflint 4.20.1-1.fb1

Serdes Tuning Transmitter Info
------------------------------
Serdes TX parameters               : fir_pre2,fir_pre1,fir_main,fir_post1
Lane 0                       : 2,-8,53,0
Lane 1                       : 2,-8,53,0
Lane 2                       : 2,-8,53,0
Lane 3                       : 2,-8,53,0
Lane 4                       : 2,-8,53,0
Lane 5                       : 2,-8,53,0
Lane 6                       : 2,-8,53,0
Lane 7                       : 2,-8,53,0
```

# NVIDIA – Setting TX Coefficients

```
[root@localhost ~]# mstlink -d 19:00.0 --serdes_tx 2,-9,49,-2 --advance

Operational Info
----------------
State                       : Active
Physical state                  : ETH_AN_FSM_ENABLE
Speed                        : 400G
Width                        : 8x
FEC                          : Standard_RS-FEC - (544,514)
Loopback Mode                    : No Loopback
Auto Negotiation                 : ON

Supported Info
--------------
Enabled Link Speed (Ext.)       : 0x000097f2 (400G_8X,200G_4X,100G_2X,100G_4X,50G_1X,50G_2X,40G,25G,10G,1G)
Supported Cable Speed (Ext.)    : 0x000097fe
(400G_8X,200G_4X,100G_2X,100G_4X,50G_1X,50G_2X,40G,25G,10G,5G,2.5G,1G)

Troubleshooting Info
--------------------
Status Opcode                   : 0
Group Opcode                    : N/A
Recommendation                     : No issue was observed.

Tool Information
----------------
Firmware Version                : 28.33.2018
amBER Version                   : 2.00
MSTFLINT Version                   : mstflint 4.20.1-1.fb1

Configuring Port Transmitter Parameters...
```

# NVIDIA – Starting and Stopping PRBS Testing

```
[root@localhost ~]# mlxlink -d /dev/mst/mt53104_pciconf0 --port 3 --cable --prbs_select HOST --prbs_mode EN --checker_pattern
PRBS13 --invert_checker --generator_pattern PRBS31 --swap_generator --lane_rate HDR

[root@localhost ~]# mlxlink -d /dev/mst/mt53104_pciconf0 --port 3 --cable --prbs_select HOST --prbs_mode DS
```

# Broadcom – Getting and Setting TX Coefficients

```
[root@localhost ~] niccli -pci 0000:03:00.0 txfir -get -modtype PAM4 -lane 0

--------------------------------------------------------------------------------
NIC CLI v232.0.153.0 - Broadcom Inc. (c) 2024 (Bld-106.52.39.138.16.0)
--------------------------------------------------------------------------------

Transmit Finite Impulse Response (TxFIR):
    Pre1   : -20
    Pre2   : 0
    Main   : 132
    Post1  : -16
    Post2  : 0
    Post3  : 0
    Amp    : 0

[root@localhost ~] niccli -pci 0000:03:00.0 txfir -set -modtype PAM4 -lane 1 -pre1 1 -pre2 -2 -main 12 -amp 10 -post1 -10 -post2 15
-post3 10
```

# Broadcom – Starting and Stopping PRBS Testing

```
[root@localhost ~] niccli -pci 0000:03:00.0 prbs_test -enable -mode PRBS31 -rxlanemask 255 -txlanemask 255 -duration 10
[root@localhost ~] niccli -pci 0000:03:00.0 prbs_test -enable -disable
```

# fbnic – Getting TX Coefficients

Note: This is a proof of concept not intended for upstream

```
[root@localhost ~]# cd /sys/kernel/debug/fbnic/0000\:01\:00.0/fbn/tx_fir/
[root@localhost tx_fir]# echo reset > ctrl
[root@localhost tx_fir]# for i in */*; do echo "$i - $(cat $i)"; done
global/main_tap - 34
global/post_tap - 0
global/pre_tap_1 - 3
global/pre_tap_2 - 0
global/pre_tap_3 - 0
lane0/main_tap - 34
lane0/post_tap - 0
lane0/pre_tap_1 - 3
lane0/pre_tap_2 - 0
lane0/pre_tap_3 - 0
lane1/main_tap - 34
lane1/post_tap - 0
lane1/pre_tap_1 - 3
lane1/pre_tap_2 - 0
lane1/pre_tap_3 - 0
```

# fbnic – Setting TX Coefficients

Note: This is a proof of concept not intended for upstream

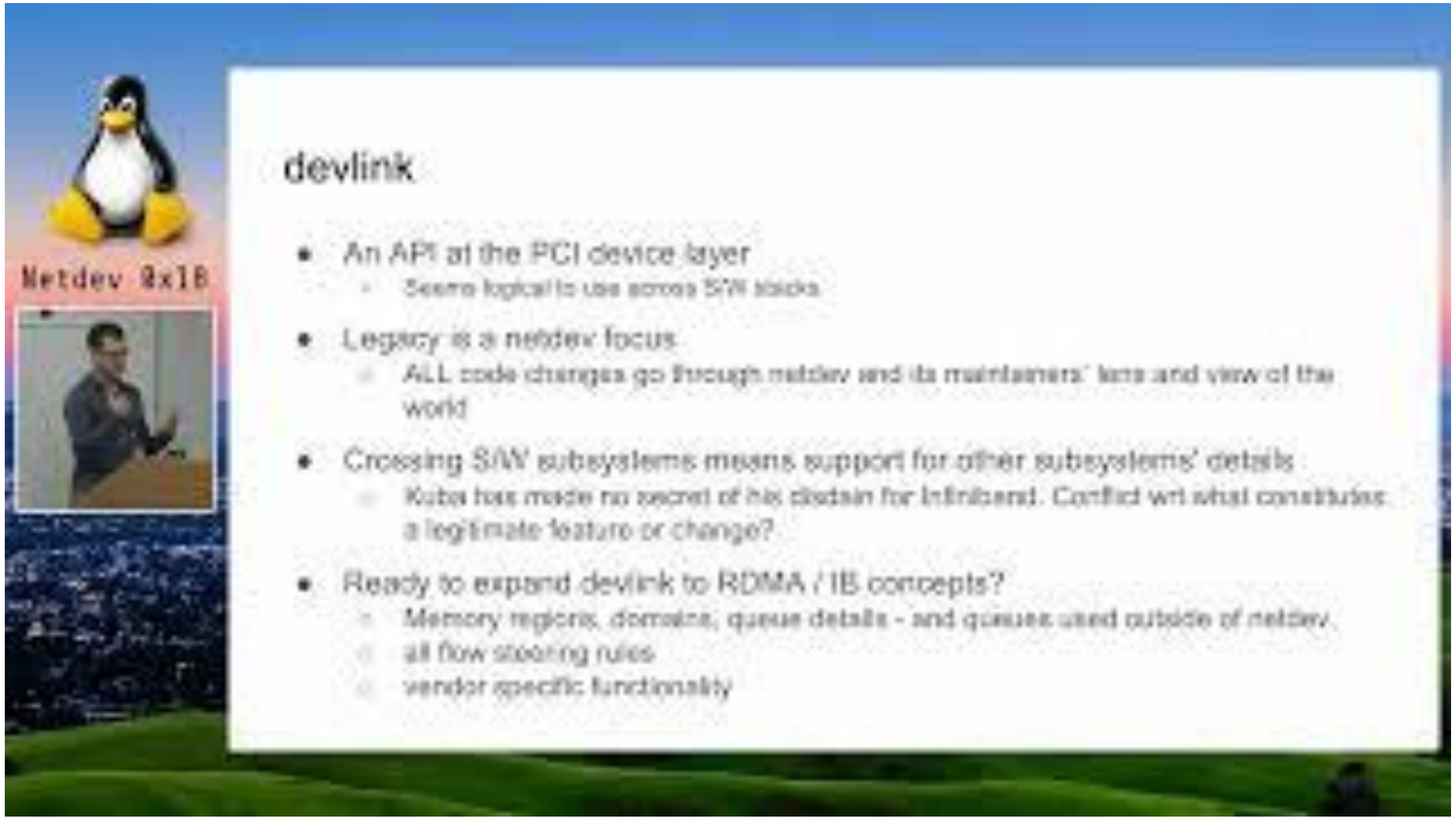
```
[root@localhost ~]# cd /sys/kernel/debug/fbnic/0000\:01\:00.0/fbn/tx_fir/
[root@localhost tx_fir]# echo reset > ctrl
[root@localhost tx_fir]# echo 35 > global/main_tap
[root@localhost tx_fir]# echo 2 > lane0/pre_tap_1
[root@localhost tx_fir]# echo 4 > lane1/pre_tap_1
[root@localhost tx_fir]# echo set > ctrl
```

# fbnic – Starting and Stopping PRBS Testing

Note: This is a proof of concept not intended for upstream

```
[root@localhost ~]# cd /sys/kernel/debug/fbnic/0000\:01\:00.0/fbn/testing
[root@localhost testing]# ip link set dev eth0 down
[root@localhost testing]# echo prbs31 > rx_test
[root@localhost testing]# echo prbs31 > tx_test
[root@localhost testing]# echo y > tx_prbs_8_10
[root@localhost testing]# echo start > ctrl
[root@localhost testing]# echo stop > ctrl
```

# Netdev 0x18 - Driver and H/W API Workshop Recap



# TX Coefficients with ethtool

- Requirements
  - Each TX coefficient from Broadcom, Mellanox, and fbnic can fit in a S8
  - Manufactures use between 4(NVIDIA) and 7(Broadcom) values per lane
  - Devices use between 2(fbnic) and 8(NVIDIA) lanes
  - Typically all lanes use the same settings but may need to be configured independently during testing
  - Must allow user to change values regardless of link state to support PRBS testing

- Open questions
  - Do we extend get/set-phy-tunable or create a new ioctl?
  - Can ethtool accept a comma separated list of values?
  - How does ethtool allow lanes to be configured independently?

# PRBS Testing with ethtool

- Requirements
  - Non-deterministic end of testing
  - Many different types of PRBS tests
  - Link may not be up during testing
  - Testing may be run on user specified lanes
  - Must allow running test with user specified TX co-efficents
  - Requires new statistics which may only be valid when used with testing
    - PRBS lock
    - PRBS Data count
    - PRBS Error count
    - PRBS BER

- Open questions
  - Do we extend cable-test or create a new ioctl?
    - Should new ioctl be PRBS specific for a generic way to run non-deterministic tests?
    - How does the user know which tests are supported?
  - Do new statics go under eth-phy or a new category?


Meta