

Modify LX2160ARDB SFPs from 25G to 10G

On the LX2160ARDB, there are 2 25G SFP interfaces, but no 10G SFP interface. When customer want to test the 10G SFPs to evaluate the performance, they have to change the 25G SFP interfaces into 10G by reconfiguration the SW configuration.

In this document, it will give details of configuring the LX2160ARDB to support the customer's 10G SFP interfaces requirement with LSDK2108. At the end, an image will be generated to deployed into the SD card. Because SD card is a convenient way boot up LX2160ARDB, if one wrong move could brick the system, the customer could unplug the SD card to repeat the steps below.

Total steps to highspeed rate change.

To configure the 10G SFP interfaces, it will need adjust 3 different steps of the SW.

1. Create RCW using the 25G to 10G sequence
2. Configure U-boot menuconfig to set INPHI IN112525_s03 speed to 10G
3. Adjust DPC file by adding enet_if="XFI" property to DPMAC.5/6

Now, assumption customer could build the image with flex-builder, and their boot mode is SD. Suggest customer could backup files above steps and change the default files, when will generate image with default setting. Below will list 3 steps details of LSDK2108.

1. Create RCW using the 25G to 10G sequence

Modify */flexbuild_lsdk2108_github/components/firmware/rcw/lx2160ardb_rev2/XGGFF_PP_HHHH_RR_19_5_2/rcw_2200_750_3200_19_5_2.rcw

Add: “#include <../lx2160asi/LX2160A_Protocol19_25G_to_10G_sequence.rcw>” at the end of the file.

```
rcw_2200_750_3200_19_5_2.rcw x
61 #include <../lx2160asi/a008851_PEX5.rcw>
62
63 /*SerDes Errata A-050479*/
64 #include <../lx2160asi/a050479_PEX3.rcw>
65 #include <../lx2160asi/a050479_PEX5.rcw>
66
67 #include <../lx2160asi/LX2160A_Protocol19_25G_to_10G_sequence.rcw>
```

Create a file, named "LX2160A_Protocol19_25G_to_10G_sequence.rcw" in the */flexbuild_lsdk2108_github/components/firmware/rcw/lx2160asi

```
tic@tic-Latitude-7480:~/nxp/build/flexbuild/flexbuild_lsdk2108_github/components
/firmware/rcw/lx2160asi$ ls
25g_eq_s1_lane_a.rcw  a009531_PEX6.rcw
25g_eq_s1_lane_b.rcw  a010554.rcw
25g_eq_s1_lane_c.rcw  a011270.rcw
25g_eq_s1_lane_d.rcw  a050234.rcw
25g_eq_s1_lane_e.rcw  a050479_PEX1.rcw
25g_eq_s1_lane_f.rcw  a050479_PEX2.rcw
25g_eq_s1_lane_g.rcw  a050479_PEX3.rcw
25g_eq_s1_lane_h.rcw  a050479_PEX4.rcw
a008851_PEX1.rcw      a050479_PEX5.rcw
a008851_PEX2.rcw      a050479_PEX6.rcw
a008851_PEX3.rcw      bootloctr_nor.rcw
a008851_PEX4.rcw      bootloctr_sd.rcw
a008851_PEX5.rcw      common.rcw
a008851_PEX6.rcw      flexspi_divisor_24.rcw
a009531_PEX1.rcw      flexspi_divisor_28.rcw
a009531_PEX2.rcw      flexspi_divisor_32.rcw
a009531_PEX3.rcw      LX2160A_Protocol19_25G_to_10G_sequence.rcw
a009531_PEX4.rcw      lx2160a.rcwi
a009531_PEX5.rcw      scratchrwl.rcw
```

Copy the contents below to the file:

```
.pbi
write 0x01EA10B0, 0x99999999
write 0x01EA10B4, 0x00000000

write 0x01EA0D00, 0x00000052
write 0x01EA0D24, 0x10000000
write 0x01EA0D28, 0x00000000
write 0x01EA0D30, 0x10808307
write 0x01EA0D44, 0x10000000
write 0x01EA0D48, 0x10000000
write 0x01EA0D50, 0x00000000
write 0x01EA0D58, 0x81000020
write 0x01EA0D68, 0x80000000
write 0x01EA0D74, 0x00002000
write 0x01EA0D80, 0x00408000

write 0x01EA0C00, 0x00000052
write 0x01EA0C24, 0x10000000
write 0x01EA0C28, 0x00000000
write 0x01EA0C30, 0x10808307
write 0x01EA0C44, 0x10000000
write 0x01EA0C48, 0x10000000
write 0x01EA0C50, 0x00000000
write 0x01EA0C58, 0x81000020
write 0x01EA0C68, 0x80000000
write 0x01EA0C74, 0x00002000
write 0x01EA0C80, 0x00408000

.end
```

```
1 .pbi
2 write 0x01EA10B0, 0x99999999
3 write 0x01EA10B4, 0x00000000
4
5 write 0x01EA0D00, 0x00000052
6 write 0x01EA0D24, 0x10000000
7 write 0x01EA0D28, 0x00000000
8 write 0x01EA0D30, 0x10808307
9 write 0x01EA0D44, 0x10000000
10 write 0x01EA0D48, 0x10000000
11 write 0x01EA0D50, 0x00000000
12 write 0x01EA0D58, 0x81000020
13 write 0x01EA0D68, 0x80000000
14 write 0x01EA0D74, 0x00002000
15 write 0x01EA0D80, 0x00408000
16
17 write 0x01EA0C00, 0x00000052
18 write 0x01EA0C24, 0x10000000
19 write 0x01EA0C28, 0x00000000
20 write 0x01EA0C30, 0x10808307
21 write 0x01EA0C44, 0x10000000
22 write 0x01EA0C48, 0x10000000
23 write 0x01EA0C50, 0x00000000
24 write 0x01EA0C58, 0x81000020
25 write 0x01EA0C68, 0x80000000
26 write 0x01EA0C74, 0x00002000
27 write 0x01EA0C80, 0x00408000
28
29 .end
```

2. Configure U-boot menuconfig to set speed to 10G

```
CONFIG_PHY_INPHI=y
```

```
CONFIG_IN112525_S03_10G=y
```

Modify the

*/flexbuild_lsdk2108_github/components/firmware/uboot/configs/lx2160ardb_tfa_defconfig

```
lx2160ardb_tfa_defconfig x |
88 CONFIG_WDT=y
89 CONFIG_WDT_SBSA=y
90 CONFIG_EFI_LOADER_BOUNCE_BUFFER=y
91 CONFIG_IN112525_S03_10G=y
```

```
lx2160ardb_tfa_defconfig x |
61 CONFIG_PHY_ATHEROS=y
62 CONFIG_PHY_CORTINA=y
63 CONFIG_SYS_CORTINA_NO_FW_UPLOAD=y
64 CONFIG_PHY_INPHI=y
```

3. Adjust DPC file by adding enet_if="XFI" property to DPMAC.5/6

Modify

*/flexbuild_lsdk2108_github/components/firmware/mc_utils/config/lx2160a/LX2160A-RDB/dpc-usxgmii.dts

Add MAC17/18 properties below:

```
mac@17{
    link_type = "MAC_LINK_TYPE_PHY";
};
mac@18 {
    link_type = "MAC_LINK_TYPE_PHY";
};
```

```
dpc-usxgmi.dts
79
80     mac@4 {
81         link_type = "MAC_LINK_TYPE_PHY";
82         enet_if = "USXGMII";
83     };
84     mac@5 {
85         enet_if = "XFI";
86     };
87
88     mac@6 {
89         enet_if = "XFI";
90     };
91     mac@17 {
92         link_type = "MAC_LINK_TYPE_PHY";
93     };
94     mac@18 {
95         link_type = "MAC_LINK_TYPE_PHY";
96     };
97 };
98 };
99 };
```

Generate image

Then compile the 3 steps above (modified files) follow scripts below:

In the */flexbuild_lsdk2108_github

```
$ source setup.env
$ flex-builder -i clean-firmware
$ flex-builder -c rcw -m lx2160ardb_rev2
```

Get RCW image in build/firmware/rcw/lx2160ardb_rev2/.

```
$ flex-builder -c atf -m lx2160ardb_rev2 -b sd
```

Build the changed RCW and UBOOT into ATF in build/firmware/atf/lx2160ardb_rev2/.

```
$ flex-builder -i mkfw -m lx2160ardb_rev2 -b sd
```

Build the image in the */flexbuild_lsdk2108_github/build/images

Program the firmware to SD card.

Download the image generated above into the board and flash the image into SD card.

```
=> mmc dev 0; mmc write $load_addr 8 1fff8
```

Every time reset and boot the board from SD card, when boot into U-boot

```
=> qixis_reset sd
```

Check the MAC speed

Customer could boot from Tiny Linux, and the MAC5/MAC6 has been configured as 10G.
Check the MAC5 and MAC6 on the LX2160ARDB

```
root@localhost:~# ls-addni dpmac.5
Created interface: eth2 (object:dpni.2, endpoint: dpmac.5)
root@localhost:~# ls-addni dpmac.6
Created interface: eth3 (object:dpni.3, endpoint: dpmac.6)
root@localhost:~# ifconfig eth2 192.168.65.1
root@localhost:~# ifconfig eth3 192.168.66.1 up
root@localhost:~# ifconfig eth2 up
root@localhost:~# ethtool eth2
Settings for eth2:
Supported ports: [ ]
Supported link modes:   Not reported
Supported pause frame use: No
Supports auto-negotiation: No
Supported FEC modes: Not reported
Advertised link modes:   Not reported
Advertised pause frame use: No
Advertised auto-negotiation: No
Advertised FEC modes: Not reported
Speed: 10000Mb/s
Duplex: Full
Port: Twisted Pair
PHYAD: 0
Transceiver: internal
Auto-negotiation: off
MDI-X: Unknown
Link detected: yes
root@localhost:~# ethtool eth3
Settings for eth3:
Supported ports: [ ]
Supported link modes:   Not reported
Supported pause frame use: No
Supports auto-negotiation: No
Supported FEC modes: Not reported
Advertised link modes:   Not reported
Advertised pause frame use: No
Advertised auto-negotiation: No
Advertised FEC modes: Not reported
Speed: 10000Mb/s
```

Duplex: Full
Port: Twisted Pair
PHYAD: 0
Transceiver: internal
Auto-negotiation: off
MDI-X: Unknown
Link detected: yes

Now MAC5 and MAC6 have been changed from 25G to 10G.