AN-BELK-006: Enabling dual Gigabit Ethernet support on BoraEVB
Table of Contents
1 Introduction................................................................................................................3
2 Block diagram...........................................................................................................4
3 Vivado Design...........................................................................................................5
4 Enabling dual Ethernet configuration in linux kernel.............................0
  4.1 Performance tests..........................................................................................2
    4.1.1 Test #1...............................................................................................3
    4.1.2 Test #2...............................................................................................4
1 Introduction

Thanks to the migration to linux kernel 3.10.17, BELK 2.2.0 allows to cleanly support dual Gigabit Ethernet configuration on BoraEVB. This application note describes how to implement such configuration, providing a reference design for Vivado 2014.4 and linux kernel configuration instructions.
## Block diagram

Simplified block diagram of dual Ethernet configuration is shown in the following picture.

First Ethernet port refers to J8 connector of BoraEVB carrier board and is based on Zynq's Gigabit Ethernet Controller 0 (Gem0). This controller is mapped at physical address 0xE000B000.

Second Ethernet port refers to J9 connector of BoraEVB carrier board and is based on Zynq's Gigabit Ethernet Controller 1 (Gem1). This controller is mapped at physical address 0xE000C000.
The fundamental difference between the two interfaces is the PHY interfacing. In case of Gem0, PHY is mounted on Bora SoM and it is interfaced directly to Processor Subsystem (PS) via MIO pads. In case of Gem1 instead, PHY is populated on BoraEVB (U9) and it is interfaced to Programmable Logic (PL) pads that belong to bank #34. Thus it is necessary to enable EMIO routing and to instantiate a GMII to RGMII bridge in PL as per PHY's interface requirement. Since bank #34 is powered at 3.3V (High Range I/O mode), RGMII duty cycle distortion specification is not matched. In case of carrier board designed for production environments, it is recommended to use a lower voltage levels and thus a different PL bank.

For more details please see section I/O Standard and Placement of PG160 GMII to RGMII LogiCORE IP Product Guide and this page.

3 Vivado Design

The following picture shows the block diagram of the Vivado project:
The project archive can be downloaded here.

4 Enabling dual Ethernet configuration in linux kernel

To enable dual Ethernet user needs to get the pre-built binaries from this link.

Alternatively kernel and device tree can be built from sources with the following procedure:

- update Bora kernel repository (as described here)
- apply the patch 0001-dts-bora-add-ETH1-phy-support.patch included in the pre-build binaries package
- build the updated kernel source as usual.

Here is the patch to enable dual Ethernet:

---

1 This link refers to DAVE Embedded Systems' RESERVED AREA. In order to access to the area, please be free to contact sales@dave.eu

2 This link refers to DAVE Embedded Systems' RESERVED AREA. In order to access to the area, please be free to contact sales@dave.eu
Put the binaries on the first (FAT32) partition of your BELK 2.2.0 SD card, overwriting the original one if needed. Please note that you need the following files:

- boot.bin
- bora.dtb
- uImage
- fpga.bin
- uEnv.txt

Insert the SD card into BoraEVB and turn on the board.

During kernel boot, user can check if the second ethernet
interface has been loaded succesfully:

```
root@bora:~# ifconfig -a

  can0  Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
         NOARP  MTU:16  Metric:1
         RX packets:0 errors:0 dropped:0 overruns:0 frame:0
         TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:10
         RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
         Interrupt:60

  eth0  Link encap:Ethernet  HWaddr 00:50:C2:B9:CF:82
         inet addr:192.168.0.209  Bcast:192.168.0.255  Mask:255.255.255.0
         UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
         RX packets:8424 errors:5 dropped:418 overruns:0 frame:0
         TX packets:5964 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:1000
         RX bytes:7562500 (7.2 MiB)  TX bytes:922668 (901.0 KiB)
         Interrupt:54 Base address:0xb000

         BROADCAST MULTICAST  MTU:1500  Metric:1
         RX packets:0 errors:0 dropped:0 overruns:0 frame:0
         TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:1000
         RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
         Interrupt:77 Base address:0xc000

  lo   Link encap:Local Loopback
       inet addr:127.0.0.1  Mask:255.0.0.0
       UP LOOPBACK RUNNING  MTU:65536  Metric:1
       RX packets:11 errors:0 dropped:0 overruns:0 frame:0
       TX packets:11 errors:0 dropped:0 overruns:0 carrier:0
       collisions:0 txqueuelen:0
       RX bytes:788 (788.0 B)  TX bytes:788 (788.0 B)
```

root@bora:~# ifconfig eth1 192.168.12.209
root@bora:~# [  228.492886] xemacps e000c000.ethernet: Set clk to 0 Hz
[  228.492886] xemacps e000c000.ethernet: Set clk to 0 Hz
root@bora:~#

4.1 Performance tests

To test the performances of the second Ethernet interface iperf based tests have been executed.
4.1.1 Test #1

In this test bed the two interfaces are connected on the same host machine (a PC running Linux) via a Gigabit switch but are associated to two different subnets:

- ETH0: 192.168.0.xxx
- ETH1: 192.168.12.xxx

On the Linux host machine `iperf` application is run in server mode:

```
bash# iperf -s
```

Here are the results of the test on the two Ethernet interfaces launched sequentially:

```
root@bora:~# iperf -c 192.168.0.210 && iperf -c 192.168.12.210
Client connecting to 192.168.0.210, TCP port 5001
TCP window size: 43.8 KByte (default)
------------------------------------------------------------
[  3] local 192.168.0.209 port 59288 connected with 192.168.0.210 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.0 sec   899 MBytes   754 Mbits/sec
------------------------------------------------------------
Client connecting to 192.168.12.210, TCP port 5001
TCP window size: 43.8 KByte (default)
------------------------------------------------------------
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.0 sec   889 MBytes   745 Mbits/sec
```

In case two `iperf` clients are run simultaneously on target, each instance's bandwidth is roughly halved:

```
root@bora:~# iperf -c 192.168.0.210
Client connecting to 192.168.0.210, TCP port 5001
TCP window size: 43.8 KByte (default)
```

---

**Application Note**

**D A V E S.r.l.** 33080 Porcia (PN), Italy
Tel. +39 0434 921215
Fax +39 04341994030
E-mail: info@dave.eu
http://www.dave.eu
2015 DAVE S.r.l. All Rights Reserved. Printed in Italy.
4.1.2 Test #2

ETH0 set up is the same used in previous test. ETH1 is point-to-point connected to a second linux host machine instead.

This configuration allows to achieve better performance on the single ETH1 iperf test (780-820 Mb/s):

```
root@bora:~# iperf -c 192.168.12.208 -i 1 -t 6000
------------------------------------------------------------
Client connecting to 192.168.12.208, TCP port 5001
TCP window size: 43.8 KByte (default)
------------------------------------------------------------
[ ID] Interval       Transfer     Bandwidth
[  3] 368.0-369.0 sec  94.6 MBytes  794 Mbits/sec
[  3] 369.0-370.0 sec  92.4 MBytes  775 Mbits/sec
[  3] 370.0-371.0 sec  97.1 MBytes  815 Mbits/sec
[  3] 371.0-372.0 sec  97.1 MBytes  815 Mbits/sec
[  3] 372.0-373.0 sec  96.1 MBytes  806 Mbits/sec
[  3] 373.0-374.0 sec  94.1 MBytes  790 Mbits/sec
[  3] 374.0-375.0 sec  96.6 MBytes  811 Mbits/sec
[  3] 375.0-376.0 sec  96.8 MBytes  812 Mbits/sec
[  3] 376.0-377.0 sec  97.8 MBytes  820 Mbits/sec
[  3] 377.0-378.0 sec  93.9 MBytes  787 Mbits/sec
[  3] 378.0-379.0 sec  95.9 MBytes  804 Mbits/sec
[  3] 379.0-380.0 sec  98.2 MBytes  824 Mbits/sec
[  3] 380.0-381.0 sec  93.2 MBytes  782 Mbits/sec
[  3] 381.0-382.0 sec  96.9 MBytes  813 Mbits/sec
[  3] 382.0-383.0 sec  92.0 MBytes  772 Mbits/sec
```
In case of simultaneous `iperf` instances on ETH0 and ETH1 ports, ETH1 bandwidth drops down to 330-350 Mb/s:

<table>
<thead>
<tr>
<th>Time Range</th>
<th>Bytes</th>
<th>Bandwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>383.0-384.0 sec</td>
<td>42.6 MBytes</td>
<td>358 Mbits/sec</td>
</tr>
<tr>
<td>384.0-385.0 sec</td>
<td>41.4 MBytes</td>
<td>347 Mbits/sec</td>
</tr>
<tr>
<td>385.0-386.0 sec</td>
<td>40.9 MBytes</td>
<td>343 Mbits/sec</td>
</tr>
<tr>
<td>386.0-387.0 sec</td>
<td>40.9 MBytes</td>
<td>343 Mbits/sec</td>
</tr>
<tr>
<td>387.0-388.0 sec</td>
<td>40.9 MBytes</td>
<td>343 Mbits/sec</td>
</tr>
<tr>
<td>388.0-389.0 sec</td>
<td>41.2 MBytes</td>
<td>346 Mbits/sec</td>
</tr>
<tr>
<td>389.0-390.0 sec</td>
<td>41.4 MBytes</td>
<td>347 Mbits/sec</td>
</tr>
<tr>
<td>390.0-391.0 sec</td>
<td>39.8 MBytes</td>
<td>333 Mbits/sec</td>
</tr>
<tr>
<td>391.0-392.0 sec</td>
<td>40.8 MBytes</td>
<td>342 Mbits/sec</td>
</tr>
<tr>
<td>392.0-393.0 sec</td>
<td>39.9 MBytes</td>
<td>334 Mbits/sec</td>
</tr>
<tr>
<td>393.0-394.0 sec</td>
<td>42.2 MBytes</td>
<td>354 Mbits/sec</td>
</tr>
</tbody>
</table>