BCM63138/BCM63148/BCM4908/BCM63158

Dynamic Crossbar PHY Port Assignment

Application Note

Table of Contents

1 Overview

In BCM63138/BCM63148/BCM4908/BCM63158 devices, some Ethernet ports are not connected to PHY interface directly, instead, these ports are connected to a crossbar switch that connects to many PHY interfaces. Up to and including release 5.02L.04, the crossbar PHY-to-port assignment is defined in boardparams, and cannot be changed at run-time.

Some customers require the flexibility of moving the PHY interface from one port to another during run-time. This run-time PHY port assignment feature is implemented in release 5.02L.05.

Different devices contain different combinations of number of ports on crossbar and number of PHY interfaces. Refer to Device Crossbar Connections for details.

2 Commands

A new *ethctl* command, *phy-crossbar,* is added to support the run-time PHY Port assignment feature.

```
Usage: ethctl <interface> phy-crossbar [arguments...]
    phy-crossbar : Get/Move <interface> crossbar phys
  ethctl <interface> phy-crossbar [port <sub_port#>]
    [port <sub_port#>]: assign <sub_port#> to <interface>
  ethctl bcmsw phy-crossbar: list all port phy mapping
```

NOTE: Details of `sub_port#` for different devices is documented in Device Crossbar Connections.

2.1 Show Current Port PHY Mapping

```
// 63138REF board example
// specify bcmsw as interface list all mapping
# ethctl bcmsw phy-crossbar
eth0: on crossbar with phy endpoint: 9 10
eth4: on crossbar with phy endpoint: 13

// specify interface not on crossbar get an error and list all mapping
# ethctl eth1 phy-crossbar
eth1: not connected to crossbar!
eth0: on crossbar with phy endpoint: 9 10
eth4: on crossbar with phy endpoint: 13

// specify interface on crossbar get a list of PHYs assigned to interface
# ethctl eth4 phy-crossbar
eth4: on crossbar with phy endpoint: 13
```

2.2 Reassign PHY Interface to a Different Port

```
// 63138REF board example
# ethctl eth4 phy-crossbar port 9
Phy endpoint 9 moved from eth0 to eth4
```

NOTE: Reassigning a PHY to new port requires bringing down both the source and destination interfaces. This command will print an error if interfaces are not brought down first.

2.3 Useful Helper Commands

```
// 63138REF board example
# echo list > /proc/driver/phy/cmd
PHY: proc cmd - list
|=============================================================================|
|  Id |  State  |   Phy   |   Bus   | Addr |   Speed   | Duplex |   PHYID   |
|=============================================================================|
|  0  |  Down   |    GPHY |   GMII | 0x0c |           |        | 600d:85c0 |
|  1  |  Up     |  SERDES | SERDES | 0x06 | 1000 Mbps |  Full  | 0000:0000 |
|  2  |  Up     |    GPHY |   GMII | 0x08 | 1000 Mbps |  Full  | 600d:85c0 |
|  3  |  Down   |    GPHY |   GMII | 0x09 |           |        | 600d:85c0 |
|  4  |  Down   |    GPHY |   GMII | 0x0a |           |        | 600d:85c0 |
|  5  |  Down   |    GPHY |   GMII | 0x0b |           |        | 600d:85c0 |
|  6  |  Down   |     I2C |        | 0x00 |           |        | 0000:0000 |
|=============================================================================|

# echo crossbars > /proc/driver/phy/cmd
PHY: proc cmd - crossbars
|=========================================|
|  Id  |  active |   Phy   |   Bus  | Addr |
|=========================================|
|   0  | I2-     |         |        |      |
|      |   -E1 * |    GPHY |   GMII | 0x0c |
|=========================================|
|   1  | I0-     |         |        |      |
|      |   -E4   |    GPHY |   GMII | 0x0b |
|      |   -E0 * |  SERDES | SERDES | 0x06 |
|      |   -E0   |     I2C |        | 0x00 |
|=========================================|
```

3 Configuration Steps

1. Bring down both the source and destination interfaces (ports).
2. Reassign PHY to new interface.
3. Bring up both the source and destination interfaces (ports).

3.1 Example: on BCM963138REF board

Initial configuration:

```
eth0: <Int sw port: 0> <Logical : 00> MAC : 02:10:18:34:99:01
    Chip Physical Port 10, Cross Bar Port 1, PHY_ID <0x0007f00c:0x0c:GPHY >
    Chip Physical Port  9, Cross Bar Port 0, PHY_ID <0x06180006:0x06:SERDES >
eth1: <Ext sw port: 0> <Logical : 08> PHY_ID <0x0007f008:0x08:GPHY> MAC:02:10:18:34:99:01
eth2: <Ext sw port: 1> <Logical : 09> PHY_ID <0x0007f009:0x09:GPHY> MAC:02:10:18:34:99:01
eth3: <Ext sw port: 2> <Logical : 10> PHY_ID <0x0007f00a:0x0a:GPHY> MAC:02:10:18:34:99:01
eth4: <Ext sw port: 3> <Logical : 11> MAC : 02:10:18:34:99:01
    Chip Physical Port 13, Cross Bar Port 4, PHY_ID <0x0007f00b:0x0b:GPHY >

# ethctl bcmsw phy-crossbar
eth0: on crossbar with phy endpoint: 9 10
eth4: on crossbar with phy endpoint: 13
```

3.2 Moving Subport 9 SerDes PHY from eth0 to eth4

```
// step 1: bring down eth0 and eth4
# ifconfig eth4 down
device eth4 left promiscuous mode
br0: port 4(eth4.0) entered disabled state
# ifconfig eth0 down
eth0 (Int switch port: 0) (Logical Port: 0) (phyId: 6) Link DOWN.

// step 2: assign subport 9 to eth4
# ethctl eth4 phy-crossbar port 9
Phy endpoint 9 moved from eth0 to eth4

// step 3: bring up eth0 and eth4
# ifconfig eth0 up
IPv6: ADDRCONF(NETDEV_UP): eth0: link is not ready
# ifconfig eth4 up
IPv6: ADDRCONF(NETDEV_UP): eth4: link is not ready
device eth4 entered promiscuous mode
IPv6: ADDRCONF(NETDEV_UP): eth4.0: link is not ready
eth4 (Ext switch port: 3) (Logical Port: 11) (phyId: 6) Link UP at 1000 mbps full duplex
IPv6: ADDRCONF(NETDEV_CHANGE): eth4: link becomes ready
IPv6: ADDRCONF(NETDEV_CHANGE): eth4.0: link becomes ready
br0: port 4(eth4.0) entered forwarding state
br0: port 4(eth4.0) entered forwarding state

// optional showing current PHY port mapping
# ethctl bcmsw phy-crossbar
eth0: on crossbar with phy endpoint: 10
eth4: on crossbar with phy endpoint: 9 13
```

3.3 Limitations

The PHY interface connecting directly to peer MAC (MAC-to-MAC) connection cannot be reassigned at run-time. To support these connections, significant code change is required.

Only boardparams-defined PHY interfaces on the crossbar, and defined ports on the crossbar, can participate in run-time PHY port assignment.

On the BCM63138 device, GPHY3 (subport 13) can only be not used, or assigned to SF2 port 3.

4 Device Crossbar Connections

4.1 BCM63138

Table 1: Direct Connections

Switch - Port	PHY Interface
SF2 (external) –port 0	GPHY0
SF2 (external) –port 1	GPHY1
SF2 (external) –port 2	GPHY2
SF2 (external) –port 5	RGMII1
SF2 (external) –port 7	RGMII2

Table 2: 3x5 Crossbar

Three Internal Endpoints (Ports)	Switch - Port
I0	SF2 (external) – port 3
I1	SF2 (external) – port 4
I2	Runner (internal) – port 0 (WAN)
Five External Endpoints (PHYs)	**Interface**
E0 (subport 9)	Active Ethernet
E1 (subport 10)	GPHY4
E2 (subport 11)	RGMII3
E3 (subport 12)	MII/TMII/RvMII/RGMII
E4 (subport 13)	GPHY3

4.2 BCM63148

Table 3: Direct Connections

Switch - Port	PHY Interface
SF2 (external) –port 0	GPHY0
SF2 (external) –port 1	GPHY1
SF2 (external) –port 2	GPHY2
SF2 (external) –port 3	GPHY3
SF2 (external) –port 5	RGMII1
SF2 (external) –port 7	RGMII2

Table 4: 2x4 Crossbar

Two Internal Endpoints (ports)	Switch - Port
I0	SF2 (external) – port 4
I1	Runner (internal) – port 0 (WAN)
Four External Endpoints (PHYs)	**Interface**
E0 (subport 9)	Active Ethernet
E1 (subport 10)	GPHY4
E2 (subport 11)	RGMII3
E3 (subport 12)	MII/TMII/RvMII/RGMII

4.3 BCM4908

Table 5: Direct Connections

Switch - Port	PHY Interface
SF2 (external) –port 0	GPHY0
SF2 (external) –port 1	GPHY1
SF2 (external) –port 2	GPHY2
SF2 (external) –port 3	GPHY3

Table 6: 2x3 Crossbar

Two Internal Endpoints (ports)	Switch - Port
I0	SF2 (external) – port 7
I1	Runner (internal) – port 3 (WAN)
Three External Endpoints (PHYs)	**Interface**
E0 (subport 9)	SerDes
E1 (subport 10)	GPHY4
E2 (subport 11)	MII/RvMII/RGMII

4.4 BCM63158

Table 7: Direct Connections

Switch - Port	PHY Interface
SF2 (external) –port 0	GPHY0
SF2 (external) –port 1	GPHY1
SF2 (external) –port 2	GPHY2
SF2 (external) –port 3	GPHY3

Table 8: 3x4 Crossbar

Three Internal Endpoints (Ports)	Switch - Port
I0	SF2 (external) – port 4
I1	SF2 (external) – port 6
I2	Runner (internal) – XPORT1 (WAN)
Four External Endpoints (PHYs)	**Interface**
E0 (subport 9)	SerDes
E1 (subport 10)	GPHY4
E2 (subport 11)	RGMII0
E3 (subport 12)	RGMII1

5 Example Initial Configuration

Using BCM4908 device selecting E1 (subport 10) GPHY4 between I1 Runner-port3 (WAN) and I0 SF2-port7 (LAN).

Scenario 1: Initially assigning GPHY4 to WAN (eth0) port

```
eth0: <Int sw port: 3> <Logical : 03> MAC : 02:10:18:34:44:01
    Chip Physical Port 10, Cross Bar Port 1, PHY_ID <0x0007f00c:0x0c:GPHY >
eth1: <Ext sw port: 0> <Logical : 08> PHY_ID <0x0007f008:0x08:GPHY> MAC:02:10:18:34:44:01
eth2: <Ext sw port: 1> <Logical : 09> PHY_ID <0x0007f009:0x09:GPHY> MAC:02:10:18:34:44:01
eth3: <Ext sw port: 2> <Logical : 10> PHY_ID <0x0007f00a:0x0a:GPHY> MAC:02:10:18:34:44:01
eth4: <Ext sw port: 3> <Logical : 11> PHY_ID <0x0007f00b:0x0b:GPHY> MAC:02:10:18:34:44:01
eth5: <Ext sw port: 7> <Logical : 15> MAC : 02:10:18:34:44:01
```

In this case, SF2-port7 (eth5) has no PHY endpoint initially. Following is a sample boardparms.c definition.

```
{bp_ucPhyType0,              .u.uc = BP_ENET_NO_PHY}, // Runner
  {bp_usConfigType,            .u.us = BP_ENET_CONFIG_MMAP},
  {bp_ucPhyAddress,            .u.uc = 0x1e},
  {bp_ulPortMap,               .u.ul = 0x9},
  {bp_ulPhyId0,                .u.ul = GMII_DIRECT | EXTSW_CONNECTED},
  {bp_ulPortFlags,             .u.ul = PORT_FLAG_MGMT }, // Managment port is on switch
  {bp_ulPhyId3,                .u.ul = BP_PHY_ID_NOT_SPECIFIED},
  {bp_ulCrossbar,              .u.ul = 10},
  {bp_ulCrossbarPhyId,         .u.ul = (BCM94908_PHY_BASE + 0x04) | (ADVERTISE_ALL_GMII |
PHY_ADV_CFG_VALID)},
  /* use the WAN LED from runner */
  {bp_usSpeedLed100,           .u.us = BP_SERIAL_GPIO_22_AH},
  {bp_usSpeedLed1000,          .u.us = BP_SERIAL_GPIO_23_AH},
  {bp_usLinkLed,               .u.us = BP_SERIAL_GPIO_21_AH},

  {bp_ucPhyType1,              .u.uc = BP_ENET_EXTERNAL_SWITCH},
  {bp_usConfigType,           .u.us = BP_ENET_CONFIG_MMAP}, // Accessing SF2 as MMapped external
switch
  {bp_ulPortMap,               .u.ul = 0x8f},
  {bp_ulPhyId0,                .u.ul = (BCM94908_PHY_BASE + 0x00) | (ADVERTISE_ALL_GMII |
PHY_ADV_CFG_VALID)},
  {bp_usSpeedLed100,           .u.us = BP_SERIAL_GPIO_0_AH},
  {bp_usSpeedLed1000,          .u.us = BP_SERIAL_GPIO_1_AH},
  {bp_usLinkLed,               .u.us = BP_SERIAL_GPIO_26_AH},
  {bp_ulPhyId1,                .u.ul = (BCM94908_PHY_BASE + 0x01) | (ADVERTISE_ALL_GMII |
PHY_ADV_CFG_VALID)},
  {bp_usSpeedLed100,           .u.us = BP_SERIAL_GPIO_2_AH},
  {bp_usSpeedLed1000,          .u.us = BP_SERIAL_GPIO_3_AH},
  {bp_usLinkLed,               .u.us = BP_SERIAL_GPIO_27_AH},
  {bp_ulPhyId2,                .u.ul = (BCM94908_PHY_BASE + 0x02) | (ADVERTISE_ALL_GMII |
PHY_ADV_CFG_VALID)},
  {bp_usSpeedLed100,           .u.us = BP_SERIAL_GPIO_4_AH},
  {bp_usSpeedLed1000,          .u.us = BP_SERIAL_GPIO_5_AH},
  {bp_usLinkLed,               .u.us = BP_SERIAL_GPIO_28_AH},
  {bp_ulPhyId3,                .u.ul = (BCM94908_PHY_BASE + 0x03) | (ADVERTISE_ALL_GMII |
PHY_ADV_CFG_VALID)},
  {bp_usSpeedLed100,           .u.us = BP_SERIAL_GPIO_6_AH},
  {bp_usSpeedLed1000,          .u.us = BP_SERIAL_GPIO_7_AH},
  {bp_usLinkLed,               .u.us = BP_SERIAL_GPIO_29_AH},
  {bp_ulPhyId7,                .u.ul = BP_PHY_ID_NOT_SPECIFIED},  ← no crossbar PHY endpoint
```

During run-time, GPHY4 can be re-assigned to eth5 by:

```
# ifconfig eth5 down; ifconfig eth0 down
# ethctl eth5 phy-crossbar port 10
# ifconfig eth5 up; ifconfig eth0 up
```

Scenario 2: Initially assigning GPHY4 to LAN (eth5) port

```
eth0: <Int sw port: 3> <Logical : 03> MAC : 02:10:18:34:44:01
eth1: <Ext sw port: 0> <Logical : 08> PHY_ID <0x0007f008:0x08:GPHY> MAC:02:10:18:34:44:01
eth2: <Ext sw port: 1> <Logical : 09> PHY_ID <0x0007f009:0x09:GPHY> MAC:02:10:18:34:44:01
eth3: <Ext sw port: 2> <Logical : 10> PHY_ID <0x0007f00a:0x0a:GPHY> MAC:02:10:18:34:44:01
eth4: <Ext sw port: 3> <Logical : 11> PHY_ID <0x0007f00b:0x0b:GPHY> MAC:02:10:18:34:44:01
eth5: <Ext sw port: 7> <Logical : 15> MAC : 02:10:18:34:44:01
    Chip Physical Port 10, Cross Bar Port 1, PHY_ID <0x0007f00c:0x0c:GPHY >
```

In this case, runner-port3 (eth0) has no PHY endpoint initially. Following is a sample boardparms.c definition.

```
  {bp_ucPhyType0,              .u.uc = BP_ENET_NO_PHY}, // Runner
  {bp_usConfigType,            .u.us = BP_ENET_CONFIG_MMAP},
  {bp_ucPhyAddress,            .u.uc = 0x1e},
  {bp_ulPortMap,               .u.ul = 0x9},
  {bp_ulPhyId0,                .u.ul = GMII_DIRECT | EXTSW_CONNECTED},
  {bp_ulPortFlags,             .u.ul = PORT_FLAG_MGMT }, // Managment port is on switch
  {bp_ulPhyId3,                .u.ul = BP_PHY_ID_NOT_SPECIFIED},  ← no crossbar PHY endpoint
  /* use the WAN LED from runner */
  {bp_usSpeedLed100,           .u.us = BP_SERIAL_GPIO_22_AH},
  {bp_usSpeedLed1000,          .u.us = BP_SERIAL_GPIO_23_AH},
  {bp_usLinkLed,               .u.us = BP_SERIAL_GPIO_21_AH},

  {bp_ucPhyType1,              .u.uc = BP_ENET_EXTERNAL_SWITCH},
  {bp_usConfigType,           .u.us = BP_ENET_CONFIG_MMAP}, // Accessing SF2 as MMapped external
switch
  {bp_ulPortMap,               .u.ul = 0x8f},
  {bp_ulPhyId0,                .u.ul = (BCM94908_PHY_BASE + 0x00) | (ADVERTISE_ALL_GMII |
PHY_ADV_CFG_VALID)},
  {bp_usSpeedLed100,           .u.us = BP_SERIAL_GPIO_0_AH},
  {bp_usSpeedLed1000,          .u.us = BP_SERIAL_GPIO_1_AH},
  {bp_usLinkLed,               .u.us = BP_SERIAL_GPIO_26_AH},
  {bp_ulPhyId1,                .u.ul = (BCM94908_PHY_BASE + 0x01) | (ADVERTISE_ALL_GMII |
PHY_ADV_CFG_VALID)},
  {bp_usSpeedLed100,           .u.us = BP_SERIAL_GPIO_2_AH},
  {bp_usSpeedLed1000,          .u.us = BP_SERIAL_GPIO_3_AH},
  {bp_usLinkLed,               .u.us = BP_SERIAL_GPIO_27_AH},
  {bp_ulPhyId2,                .u.ul = (BCM94908_PHY_BASE + 0x02) | (ADVERTISE_ALL_GMII |
PHY_ADV_CFG_VALID)},
  {bp_usSpeedLed100,           .u.us = BP_SERIAL_GPIO_4_AH},
  {bp_usSpeedLed1000,          .u.us = BP_SERIAL_GPIO_5_AH},
  {bp_usLinkLed,               .u.us = BP_SERIAL_GPIO_28_AH},
  {bp_ulPhyId3,                .u.ul = (BCM94908_PHY_BASE + 0x03) | (ADVERTISE_ALL_GMII |
PHY_ADV_CFG_VALID)},
  {bp_usSpeedLed100,           .u.us = BP_SERIAL_GPIO_6_AH},
  {bp_usSpeedLed1000,          .u.us = BP_SERIAL_GPIO_7_AH},
  {bp_usLinkLed,               .u.us = BP_SERIAL_GPIO_29_AH},
  {bp_ulPhyId7,                .u.ul = BP_PHY_ID_NOT_SPECIFIED},
  {bp_ulCrossbar,              .u.ul = 10},
  {bp_ulCrossbarPhyId,         .u.ul = (BCM94908_PHY_BASE + 0x04) | (ADVERTISE_ALL_GMII |
PHY_ADV_CFG_VALID)},
```

During run-time, GPHY4 can be re-assigned to eth0 by:

```
# ifconfig eth5 down; ifconfig eth0 down
# ethctl eth0 phy-crossbar port 10
# ifconfig eth5 up; ifconfig eth0 up
```

Revision History

April 16, 2018

- Initial release