CPE

Software Board Parameters

Software User Manual

Table of Contents

Chapter 1: Overview

The Broadcom BCA CPE reference software provides support for multiple reference design boards for Broadcom devices. The single software image contains board parameters that determine the board configuration—such as Ethernet switch, LED assignment, GPIO control, and GPON/EPON parameters—for each reference board. Customers need only add a new set of board parameters, with no function code change, in order to support a new board based on a mix of features in existing reference designs.

NOTE: Some features may not be available in certain devices. The device data sheet should be referred to for clarity.

1.1 References

The references in this section may be used in conjunction with this document. Broadcom provides customer access to technical documentation and software through its Customer Support Portal (CSP).

For Broadcom documents, replace the “xx” in the document number with the largest number available in the repository to ensure that you have the most current version of the document.

Document (or Item) Name	Number	Source
BCM6316X/BCM6326X, Using GPIO as LED Drivers and Other Alternate Functions	6316X_6326X-AN1xx-R	Broadcom CSP
BCM636X/BCM6328X, Using GPIO as LED and/or NAND Flash Drivers	636X_6328X-AN1xx-R	Broadcom CSP
BCM68138, Operation and Alternate Functions of the GPIO Pins	63148-AN3xx-R	Broadcom CSP
BCM63138 GPIO and LED Assignment Worksheets	—	Broadcom CSP
BCM68148, Operation and Alternate Functions of the GPIO Pins	63148-AN3xx-R	Broadcom CSP
BCM63148 GPIO and LED Assignment Worksheets	—	Broadcom CSP
BCM63158, Operation and Alternate Functions of the GPIO Pins	63158-AN3xx	Broadcom CSP
BCM63158 GPIO and LED Assignment Worksheets	—	Broadcom CSP
BCM6838X Connect and Configure the LEDs	6838X-AN1xx-R	Broadcom CSP
BCM68580 LEDs User Guide	68580X-AN103-R	Broadcom CSP
BCM68460 LEDs User Guide	68460-UG1xx	Broadcom CSP
BCM68360 LEDs User Guide	68360-UG1xx	Broadcom CSP
Configuring AFE_ID for xDSL PHY Firmware	63XX-AN7xx-R	Broadcom CSP
Run-time Selection of Voice Board Parameters	DSLxChange-AN1xx-R	Broadcom CSP
BCM63381, Operation and Alternate Functions of the GPIO Pins	63381-AN1xx-R	Broadcom CSP
Fiber to the Distribution Point Implementation	68380-AN1xx-R	Broadcom CSP

Chapter 2: Working with Board Parameters

2.1 How Board Parameters Work

The board parameters, structure, and constants are defined in:

```
<source tree top level>\shared\opensource\boardparms\bcm963xx\boardparms_<chipset>.c
<source tree top level>\shared\opensource\boardparms\bcm963xx\bp_defs.h
<source tree top level>\shared\opensource\boardparms\bcm963xx\bp_funcs.c
<source tree top level>\shared\opensource\boardparms\bcm963xx\pinmux_table_<chipset>.c
<source tree top level>\shared\opensource\include\bcm963xx\boardparms.h
```

Each board parameter instance defines the board configurations for one particular reference board. It consists of a variable length of board parameter elements defined as:

```
typedef struct bp_elem {
  enum bp_id id;
  union {
    char * cp;
    unsigned char * ucp;
    unsigned char uc;
    unsigned short us;
    unsigned long ul;
  } u;
} bp_elem_t;

static bp_elem_t g_bcm963168xh[] = {
  {bp_cpBoardId,               .u.cp = "963168XH"},
  {bp_usGpioOverlay,           .u.ul =(BP_OVERLAY_SERIAL_LEDS |
                                       BP_OVERLAY_USB_DEVICE|
                                       BP_OVERLAY_PCIE_CLKREQ |
                                       BP_OVERLAY_HS_SPI_SSB5_EXT_CS)},
  {bp_usGpioLedAdsl,           .u.us = BP_GPIO_13_AH},
  {bp_usGpioLedSesWireless,    .u.us = BP_SERIAL_GPIO_7_AL},
  {bp_usGpioLedWanData,        .u.us = BP_GPIO_8_AL},
  {bp_usGpioLedWanError,       .u.us = BP_SERIAL_GPIO_2_AL},
  {bp_usGpioLedBlPowerOn,      .u.us = BP_GPIO_10_AL},
  {bp_usGpioLedBlStop,         .u.us = BP_SERIAL_GPIO_3_AL},
  {bp_usExtIntrResetToDefault, .u.us = BP_EXT_INTR_0},
  {bp_usExtIntrSesBtnWireless, .u.us = BP_EXT_INTR_1},
  {bp_usAntInUseWireless,      .u.us = BP_WLAN_ANT_MAIN},
  {bp_usWirelessFlags,         .u.us = 0},
  {bp_ucPhyType0,              .u.uc = BP_ENET_EXTERNAL_SWITCH},
  {bp_ucPhyAddress,            .u.uc = 0x0},
  {bp_usConfigType,            .u.us = BP_ENET_CONFIG_MMAP},
  {bp_ulPortMap,               .u.ul = 0x58},
  {bp_ulPhyId3,                .u.ul = BP_PHY_ID_4},
  {bp_ulPhyId4,                .u.ul = RGMII_DIRECT | EXTSW_CONNECTED},
  {bp_ulPhyId6,                .u.ul = BP_PHY_ID_25},
  {bp_ucPhyType1,              .u.uc = BP_ENET_EXTERNAL_SWITCH},
  {bp_ucPhyAddress,            .u.uc = 0x0},
  {bp_usConfigType,            .u.us = BP_ENET_CONFIG_HS_SPI_SSB_5},
  {bp_ulPortMap,               .u.ul = 0x0f},
  {bp_ulPhyId0,                .u.ul = BP_PHY_ID_0},
  {bp_ulPhyId1,                .u.ul = BP_PHY_ID_1},
  {bp_ulPhyId2,                .u.ul = BP_PHY_ID_2},
  {bp_ulPhyId3,                .u.ul = BP_PHY_ID_3},
  {bp_ucDspType0,              .u.uc = BP_VOIP_MIPS},
  {bp_ucDspAddress,            .u.uc = 0},
  {bp_usGpioVoip1Led,          .u.us = BP_SERIAL_GPIO_4_AL},
  {bp_usGpioVoip2Led,          .u.us = BP_SERIAL_GPIO_5_AL},
  {bp_usGpioPotsLed,           .u.us = BP_SERIAL_GPIO_6_AL},
  {bp_ulAfeId0,             .u.ul = BP_AFE_CHIP_6306 | BP_AFE_LD_ISIL1556 | BP_AFE_FE_AVMODE_VDSL
| BP_AFE_FE_REV_12_21 | BP_AFE_FE_ANNEXA },
  {bp_usGpioExtAFEReset,       .u.us = BP_GPIO_11_AL},
  {bp_last}
};
```

The BCM963168XH example shown above defines the board parameters for the BCM963168XH reference board. It contains the board parameter elements identified by the board parameter ID, such as board ID (*bp_cpBoardId*) and GPIO overlay (*bp_usGpioOverlay*), that are necessary to define the hardware configuration for the software to run properly.

Multiple board parameter structures are defined for every chip family, providing a distinct set of parameters for each different reference design board used with that chip family. They are grouped into a new array:

```
static bp_elem_t * g_BoardParms[] =
{g_bcm96368vvw, g_bcm96368mvwg, … };
```

Each chip family defines its own *g_BoardParms* array. A compile flag controls which board parameter is compiled into which target profile image.

Both the CFE and Linux images contain a copy of the board parameters and they share the same source of boardparms.c. When changing the board parameters make sure that both the CFE and Linux images are recompiled, to ensure the changes take effect in both places.

When the CFE and Linux kernels start:

1. They read the board ID string *NvramData.szBoardId* from the NVRAM and call the *BpSetBoardId*(*NvramData.szBoardId*) function.
   This function matches the input string to the *bp_cpBoardId* from each entry in the board parameters array *g_BoardParms* and sets the *g_pCurrentBp* to the matching board parameter entry.
2. *g_pCurrentBp* is then pointed to the correct board parameter set and used for all the board parameters accessed from CFE and Linux kernel.

Users can use the **b** command under the CFE prompt to change the board ID string to run the same image on a different reference board.

The boardparms.c module also provides all the helper functions for retrieving the board parameter ID properties. Each board parameter ID has a corresponding helper function. For example, *BpGetGPIOverlays* reads the GPIO overlay (*bp_usGpioOverlay)* property. The CFE and Linux kernels use these helper functions to get the LED, GPIO assignment, and Ethernet switch/PHY connection information etc, so that they can initialize and use the board correctly.

2.2 Adding Support for a New Board

The design architecture makes the process of adding a new board an easy task. A useful approach is to copy the board parameters from a reference board that most closely matches the new design and modify them as necessary. The example below shows the steps to add support for a new board. It uses the BCM963168XH as the reference board and adds a new board called BCM963168EXP.

1. Copy the BCM963168XH board parameter **g_bcm963168xh** and make a new entry **g_bcm963168exp** in the BCM63268 section identified by the _BCM963268_ compiling flag.
2. Change the first board parameter ID, **bp_cpBoardId**, to the new board name.

   ```
   {bp_cpBoardId,              .u.cp = "963168EXP"},
   ```

3. Add **g_bcm963168exp** to the *g_BoardParms* array in the same section.

   ```
   static bp_elem_t * g_BoardParms[] = {g_bcm963168xh, g_bcm963168exp,… };
   ```

4. If the internal WLAN is used, make a new entry of SROM PATCH by copying the SROM patch from a board with similar RF characteristics in the *wlanPaInfo* array. Change the board ID to **963168EXP** in the new entry. Contact Broadcom Support if the change to the SROM parameter value is needed.

   NOTE: Care should be taken when using an external power amplifier. It is important to choose an appropriate SROM patch from devices with an external power AMP; otherwise the device will send its full output power into the input of the external PA.

5. If wireless is used in the new design:
   a. Make a new entry for the wireless LAN PCI ID by copying the BCM963168XH entry in the *wlanPciInfo* array.
   b. In the new entry, change the board ID to **963168EXP**.
   c. If needed, change the wireless PCI ID and sub ID.
6. Check for changes to the LED assignments, such as GPIO pin number changes, a GPIO pin output changing to serial LED output, or any added or removed LEDs. See LED Changes.
   a. If a new LED parameter ID is needed, define a new ID with the name convention of **bp_usGpioLedXXX** in *bp_def.h* and add it to the **bpLedList** and **bpLedList_str** in *bp_func.c*
7. Check for any GPIO assignment changes, such as GPIO pin number changes and any added or removed GPIO pins. In this example, the BCM963168XH board uses GPIO_11 as an external AFE reset. If this is changed to another pin, the pin assignment must be updated accordingly.

   ```
   {bp_usGpioExtAFEReset,       .u.us = BP_GPIO_11_AL}
   ```

   a. If a new GPIO parameter ID is needed, define a new ID with the name convention of **bp_usGpioXXX** in *bp_def.h* and add it to the **bpGpioList** in *bp_func.c*

- Check for a change to the GPIO override function. See GPIO Override.
- Check for changes to the Ethernet switch and PHY connection topology.

  Review the new design and make the respective change for these switch parameters. See Ethernet Switch and PHY Topology for the detailed explanation of switch parameters and their usage.
- Check the AFE connections for DSL chips.
  BCM63268 can support two VDLS2 interfaces. One uses the internal AFE and the other uses an external AFE. Based on the hardware design, the user must specify *bp_ulAfeId0* and *bp_ulAfeId1* if a second AFE interface is used.

Each interface properties must be defined, and which one is internal and which is external.
Refer to *Configuring AFE_ID for xDSL PHY Firmware* (see References) for details about this parameter for the BCM63XX chips.

- Check the GPON/EPON settings for GPON/EPON chips. For GPON or EPON, there are other parameters to check: *bp_usGponOpticsType* and *bp_cpDefaultOpticalParams*. In certain chips, the user must specify the PHY ID to indicate if it is the GPON/EPON WAN interface. See the description of bp_ulPhyIdn for details of this requirement.
- Check external interrupt pin usage.

  Example: The BCM963168XH uses two external interrupt inputs.

```
{bp_usExtIntrResetToDefault, .u.us = BP_EXT_INTR_0},
{bp_usExtIntrSesBtnWireless, .u.us = BP_EXT_INTR_1},
```

  If the new design changes the usage of the external interrupts, these two parameters need to be updated.

  Example: In the BCM6838X, BCM6858X, BCM6836X, BCM6846X, BCM6856X, BCM4908, BCM62118, BCM63158 chips each external interrupt input can be driven from any GPIO.
  The board BCM968380FHGU uses two external interrupt inputs:

```
{bp_usExtIntrResetToDefault, .u.us = BP_EXT_INTR_0},
{bp_usGpio_Intr,             .u.us = BP_GPIO_72_AL},
{bp_usExtIntrSesBtnWireless, .u.us = BP_EXT_INTR_1},
{bp_usGpio_Intr,             .u.us = BP_GPIO_47_AL},
```

  In this example, external interrupt 0 is driven from GPIO_72, and external interrupt 1 is driven from GPIO_47.

- If a voice daughtercard is used, check the voice board parameters. Refer to the *Run-time Selection of Voice Board Parameters* (see References), for details.
- Build the CFE and a system image to test on the new board.

2.3 LED Changes

The LED controller takes two input sources: hardware driven LED and software controlled LED. The hardware driven LEDs use fixed mappings, a certain LED channel is used for a certain hardware LED function. A software controlled LED has the flexibility to assign any LED channel to software defined functions. The LED controller drives the output through a GPIO pin or a serial shift register.

There are certain constraints on the LED and GPIO usage and it differs between different chip families:

- For BCM6316X/BCM6326X, refer to *Using GPIO as LED Drivers and Other Alternate Functions* (see References).
- For BCM636X/BCM6328X refer to *Using GPIO as LED and/or NAND Flash Drivers* (see References) for details regarding LED/GPIO usage, mapping, constraints, and limitations.
- For BCM6838X refer to *Connect and Configure the BCM6838X LEDs* (see References).
- For BCM63381 refer to *Operation and Alternate Functions of the GPIO Pins* (see References).
- For BCM63138, refer to *BCM63138 GPIO and LED Assignment Worksheets* (see References)
- For BCM63148, refer to *BCM63148 GPIO and LED Assignment Worksheets* (see References)
- For BCM63158, refer to *BCM63158 GPIO and LED Assignment Worksheets* (see References)
- For BCM68580X, refer to *BCM68580 LEDs User Guide* (see References)
- For BCM68360, refer to *BCM68360 LEDs Configuration User Guide* (see References)

  Example: If a new design changes the WAN error LED to use GPIO_9 direct output, the *bp_usGpioLedWanError* value must be changed:

```
{bp_usGpioLedWanError,       .u.us = BP_GPIO_9_AL},
```

However, if a new design uses EPHY1 from the chip's internal switch, LED bit 9 is hard wired to the EPHY1 LINK/ACT LED function and GPIO_9 cannot be used for WAN errors. This is true even when a shift register is used for the EPHY1 LINK/ACT LED because direct GPIO and shift register outputs share the same source LED register bit (bit 9).

If any new LEDs are added in the new design, a new element must be defined in the board parameters defining the new LED assignment. For example, when secondary DSL is used, the following code must be added in order to map the second DSL line LED with the proper BP_GPIO or BP_SERIAL_GPIO assignment based on the hardware:

```
{bp_usGpioSecLedAdsl,        .u.us = BP_XXX}
```

With the BCM63381, BCM63138, BCM63148, and BCM6838X devices, LEDs that are strictly software controlled and do not need LED controller functions (such as dimming) can be connected as GPIOs, not through the LED controller. For example:

```
 {bp_usGpioSecLedAdsl,        .u.us = BP_XXX | BP_LED_USE_GPIO}
```

The new entry should be appended after the first DSL line LED ID, *bp_usGpioLedAdsl* (even though it can be added anywhere in the *g_bcm963168exp* array, it is not recommended to do so.)

If no second FXS station is used, simply remove the following parameter ID:

```
{bp_usGpioVoip2Led,          .u.us = BP_SERIAL_GPIO_5_AL}
```

For the complete LED board parameter ID list and usage, see "LED Parameter ID Definitions" on page 22.

2.4 GPIO Override

In addition to the usage of the regular GPIO control and LED control, the GPIO pin can be overridden for other functions. The *bp_usGpioOverlay* parameter or the *bp_usSerialLedData/Clk/Mask* functions tell the software what other functions are used on the GPIO pin. For example, the BCM963168XH use GPIO pins 0 and 1 as the serial LED output.

If the board does not use a serial shifted LED output, *BP_OVERLAY_SERIAL_LEDs* should be removed from the overlay bitmap. Each overlay function uses fixed GPIO pin assignments dictated by the chip hardware. The user does not need to specify which GPIO pin is used for that function.

- For the list of overlay functions and a detailed description of their usage, see bp_ulGpioOverlay.
- This function only applies to BCM6362, BCM6328, and BCM63268.

2.5 PinMux and PinMux Check Utility

Devices with PinMux support replace the GPIO Override with a generic PinMux facility. Every chip family has its own unique PinMux option for each pin for different functions such as GPIO, LED, Serial LED, DSL control. Broadcom reference software describes all the PinMux options using a PinMux table. The PinMux table lists board parameter ID usage on all the possible pins supported by the chip. The PinMux initialization code checks each parameter ID for the selected board ID against the table and sets the proper mux. option automatically. The user must not set any PinMux explicitly as the PinMux framework must be in control of the settings.

The PinMux table *g_pinmux_defs_0* is predefined by Broadcom in *bp_func.c* and should not be changed. Each entry in the table has the following field: bp_id, port number, pin number, mux option.

Example:

```
{ bp_usLinkLed, 0, 16, 16|BP_PINMUX_VAL_3 | BP_PINMUX_HWLED },
```

The example means the bp_usLinkLed network link activity LED on port 0 can be used on GPIO 16 with the PinMux setting of 3. It also indicates this is a hardware controlled LED. When the board parameter has this setting: `{bp_usLinkLed, .u.us = BP_GPIO_16_AL}`, the PinMux initialization code finds a match from the PinMux table so it sets this GPIO 16 pin to hardware LED function. With this PinMux framework, the user must only specify the board parameter ID correctly, everything else is done automatically.

Devices with PinMux support have the PinMux test utility, *bptest*, which is built and run during the build process. The utility checks for the most common errors in boardparms, such as requesting that a hardware-controlled LED be configured to a pin other than one of those that are defined in the PinMux table.

If running *bptest* causes a build to abort, carefully check your boardparms settings against the PinMux table and the data sheet.

Devices with the PinMux support:

- BCM63138
- BCM63148
- BCM63381
- BCM6848X
- BCM6858X
- BCM6836X
- BCM62118
- BCM63158
- BCM4908

2.6 Ethernet Switch and PHY Topology

The Broadcom broadband access chips have an internal switch with integrated transceiver (PHY) and external transceiver. They also support an external switch cascaded behind the internal switch. A maximum of two Ethernet switches (one internal and one external) are supported per board. On some chips, such as the BCM6362, the internal switch port cannot be used when an external switch is connected.

On a BCM62368, a maximum of eight ports are supported in the internal switch with three 10/100 Fast Ethernet internal PHY (ports 0–2, PHY ID 1–3), one internal Gigabit PHY (port 3, PHY ID 4), and four RGMII/MII ports (ports 4–7, external PHY ID) for connection with an external PHY or switch.

Port configuration, mapping, and PHY ID information is required for the board parameter design. Refer to the chipset or external switch data sheet for device-specific details. Table 9 through Table 11 list the internal switch configurations for all the supported chips.

The BCM963168XH board supports a BCM53125 external switch. Each switch takes a switch and PHY parameter block, which consists of the following parameter IDs:

- *bp_ucPhyType0*/1
- *bp_ucPhyAddress*
- *bp_usConfigType*
- *bp_ulPortMap*
- *bp_ulPhyId*

The internal switch uses *bp_ucPhyType0* and the external switch uses *bp_ucPhyType1*. The *bp_ucPhyType0* and *bp_ucPhyType1* parameters are always set to BP_ENET_EXTERNAL_SWITCH and *bp_ucPhyAddress* is set to zero. *bp_usConfigType* is set to BP_ENET_CONFIG_MMAP for the internal switch. For the external switch, *bp_usConfigType* can be BP_ENET_CONFIG_HS_SPI_SSB_x, BP_ENET_CONFIG_SPI_SSB_x or BP_ENET_CONFIG_MDIO, depending on the management mechanism for the external switch.

The *bp_ulPortMap* parameter defines the bitmap of the ports used in the switch. BCM963168XH uses GPHY1 (port 3), RGMII 1 (port 4), and RGMII 3 (port 6) in the internal switch, so the mapping is 0x58. For each port that is used, *bp_ulPhyIdx* must be specified. So *bp_ulPhyId3*, *bp_ulPhyId4*, and *bp_ulPhyId6* are set.

Port 3 uses the internal PHY, so it has the internal PHY ID.

Port 4 is connected to an external switch so no PHY ID is needed, but *bp_ulPhyId4* must be set with RGMII_DIRECT | EXTSW_CONNECTED flags so that the software knows that it is connected to an external switch and uses the RGMII interface.

Port 6 is connected to an external PHY, based on the hardware design. The external PHY ID is 25.

2.7 Ethernet Switch LED Configuration

The LEDs of the integrated GPHY ports in the switch are configured through the following LED parameter IDs:

- *bp_usLinkLed*
- *bp_usSpeedLed100*
- *bp_usSpeedLed1000*

These LEDs are controlled by the switch hardware and have fixed mapping to the LED channel.

Example: To configure switch GPHY port 1 to use GPIO 2 and 3 for 100 and 1000 link speed status LED and GPIO 11 for link activity LED. GPHY Port 1 uses fixed LED channel 2, 3 and 11.

```
{bp_ulPhyId1,                .u.ul = (BCM963138_PHY_BASE + 0x1) | (ADVERTISE_ALL_GMII |
PHY_ADV_CFG_VALID)},
{bp_usSpeedLed100,           .u.us = BP_GPIO_2_AL},
{bp_usSpeedLed1000,          .u.us = BP_GPIO_3_AL},
{bp_usLinkLed,               .u.us = BP_GPIO_11_AL},
```

Example: To configure switch GPHY port 1 to use serial LED output channel 2 and 3 for 100 and 1000 link speed status LED and serial LED output 11 for link activity LED. GPHY Port 1 uses fixed LED channel 2, 3 and 11.

```
{bp_ulPhyId1,                .u.ul = (BCM963138_PHY_BASE + 0x1) | (ADVERTISE_ALL_GMII |
PHY_ADV_CFG_VALID)},
{bp_usSpeedLed100,           .u.us = BP_SERIAL_GPIO_2_AL},
{bp_usSpeedLed1000,          .u.us = BP SERIAL_GPIO_3_AL},
{bp_usLinkLed,               .u.us = BP SERIAL_GPIO_11_AL},
```

Advanced LED parameter IDs are introduced in for newer chips with flexible LED configuration support:

- *bp_usNetLed0*
- *bp_usNetLed1*
- *bp_usNetLed2*
- *bp_usNetLed3*
- *bp_ulNetLedLink*
- *bp_ulNetLedActivity*
- *bp_ulNetLedSetting*

Each integrated GPHY port can have up to four hardware-controlled LEDs to indicate its link and/or speed status. This is the generic *bp_usNetLedX* parameter ID which configures the GPIO pin or serial LED output channel for that LED. Each *bp_usNetLedX* must have at least one of the function parameter IDs *bp_ulNetLedLink* or *bp_ulNetLedActivity* to indicate that this LED is used for link speed status or link activity. It is also possible to have both of the functions, combining link activity and status into one LED. This allows flexible LED usage configuration and is no longer limited to link or 100/1000 speed link status fixed functions.

There is also an optional *bp_ulNetLedSetting* parameter ID for additional setting of the each LED *bp_usNetLedX*. For example, it can be used for activity setting such as to configure for TX or RX activity only. This parameter ID can be expanded in the future to support other settings or flags that control the LED behavior. The last LED ID *bp_usNetLed3* is for link activity only for all the link speeds.

Example: Configure GPHY to use serial LED output channel 2 for 10 Mbps and 100 Mbps link status, channel 3 for 100 Mbps and 1 Gbps link status and channel 21 for link activity for all the speed. Here 10 Mbps and 1 Gbps link speed use a single LED to indicate link is on but 100 Mbps link speed uses two LEDs dual color to indicate.

```
{bp_usNetLed0,              .u.us = BP_SERIAL_GPIO_2_AH},
{bp_ulNetLedLink,           .u.ul = BP_NET_LED_SPEED_10|BP_NET_LED_SPEED_100},
{bp_usNetLed1,              .u.us = BP_SERIAL_GPIO_3_AH},
{bp_ulNetLedLink,           .u.ul = BP_NET_LED_SPEED_1G|BP_NET_LED_SPEED_100},
{bp_usNetLed3,              .u.us = BP_SERIAL_GPIO_21_AH},
{bp_ulNetLedActivity,       .u.ul = BP_NET_LED_ACTIVITY_ALL}
```

Example: Configure GPHY to use GPIO 2 for link status LED for all speeds and GPIO 3 for link activity LED for all speeds.

```
{bp_usNetLed0,              .u.us = BP_GPIO_2_AH},
{bp_ulNetLedLink,           .u.ul = BP_NET_LED_SPEED_ALL},
{bp_usNetLed1,              .u.us = BP_GPIO_3_AH},
{bp_ulNetLedActivity,       .u.ul = BP_NET_LED_ACTIVIY_ALL},
```

Example: Configure GPHY to use GPIO 2 for link status and activity LED for Fast Ethernet and Gigabit Ethernet speed and GPIO 3 for link status and activity LED for 2.5 Gbps and 10 Gbps speed

```
{bp_usNetLed0,              .u.us = BP_GPIO_2_AH},
{bp_ulNetLedLink,           .u.ul = BP_NET_LED_SPEED_GBE},
{bp_ulNetLedActivity,       .u.ul = BP_NET_LED_SPEED_GBE},
{bp_usNetLed1,              .u.us = BP_GPIO_3_AH},
{bp_ulNetLedLink,           .u.ul = BP_NET_LED_SPEED_2500|BP_NET_LED_SPEED_10G},
{bp_ulNetLedActivity,       .u.ul = BP_NET_LED_SPEED_2500|BP_NET_LED_SPEED_10G
```

In these examples, a single LED combines both link activity and status functions. The LED is off when the link is down. LED is on when the link is up. LED blinks when there is activity.

2.8 Crossbar Switch Topology

The BCM63138 and BCM63148 chips have a crossbar switch connected to port 0 of the Runner network processor and port 4 of the switch core. PHY configuration corresponds to each port of the crossbar switch rather than to the port to which the switch is connected internally.

On crossbar-enabled ports, the *bp_ulPhyIdX* corresponding to the port does not specify the PHY ID but, marks the beginning of one or more Crossbar port descriptions. Each of these port descriptions list their own PHY ID and related parameters.

```
{bp_ulPhyId0, .u.ul = BP_PHY_ID_NOT_SPECIFIED}, // This has one crossbar setting
   {bp_ulCrossbar, .u.ul = 10}, // This is crossbar port 10
       {bp_ulCrossbarPhyId, .u.ul = (BCM963138_PHY_BASE + 0x04) }, // same values as bp_ulPhyIdX
       {bp_usSpeedLed100, .u.us = BP_GPIO_26_AL},
       {bp_usSpeedLed1000, .u.us = BP_GPIO_27_AL},
       {bp_usLinkLed, .u.us = BP_GPIO_14_AL},
{bp_ulPhyId1, .u.ul = GMII_DIRECT | EXTSW_CONNECTED}, // here the next port begins
```

Alternatively:

```
{bp_ulPhyId0, .u.ul = BP_PHY_ID_NOT_SPECIFIED}, // This has more than one crossbar setting
   {bp_ulCrossbar, .u.ul = 10},
       {bp_ulCrossbarPhyId, .u.ul = (BCM963138_PHY_BASE + 0x04) },
       {bp_usSpeedLed100, .u.us = BP_GPIO_26_AL},
       {bp_usSpeedLed1000, .u.us = BP_GPIO_27_AL},
   {bp_ulCrossbar, .u.ul = 11},
       {bp_ulCrossbarPhyId, .u.ul = 0x1 | PHY_INTEGRATED_VALID | MAC_IF_RGMII_1P8V | PHY_EXTERNAL},
       {bp_usLinkLed, .u.us = BP_GPIO_14_AL},
{bp_ulPhyId1, .u.ul = GMII_DIRECT | EXTSW_CONNECTED}, // here the next port begins
```

2.9 Physical Interface Configuration

The existing parameter IDs use an explicit name of the ID to indicate different interfaces such as bp_usGpioLedWanData and bp_usGpioSecLedWanData. This creates a scalability issue when multiple same or similar interfaces must be supported for the same parameter ID.

Physical interface parameter IDs have been introduced for new devices to solve this issue. Board parameter IDs for each interface are grouped together within the interface instance that is identified by the unique interface ID. So the same parameter ID can be used in multiple interfaces. This makes parameter ID generic and expandable and makes the board parameter more organized.

Each interface instance starts with the bp_usIntfId parameter ID which is a unique number among all the interfaces. bp_usIntfEnd signals the end of the interface instance. It also must specify the bp_usIntfType and bp_usPortNum for physical interface type and port number. The rest are interface type specific parameter IDs.

Example: The following code defines two DSL interfaces using the same parameter IDs:

```
     {bp_usIntfId,                .u.us = 0},
     {bp_usIntfType,              .u.us = BP_INTF_TYPE_xDSL},
     {bp_usPortNum,               .u.us = 0},
     {bp_usGpioLedWanLink,        .u.us = BP_GPIO_18_AH},
     {bp_usGpioLedWanAct,         .u.us = BP_GPIO_30_AH},
     {bp_ulAfeId,                 .u.ul = BP_AFE_CHIP_GFAST_CH0 | BP_AFE_LD_6304 |
   BP_AFE_FE_REV_6304_REV_12_4_60 },
     {bp_usGpioAFELDPwr,          .u.us = BP_GPIO_32_AH}, // Line Driver 0 = "Int"
     {bp_usGpioAFELDData,         .u.us = BP_GPIO_33_AH},
     {bp_usGpioAFELDClk,          .u.us = BP_GPIO_34_AH},
     {bp_usGpioAFELDPwrBoost,     .u.us = BP_GPIO_38_AH},
     {bp_usIntfEnd},

     {bp_usIntfId,                .u.us = 1},
     {bp_usIntfType,              .u.us = BP_INTF_TYPE_xDSL},
     {bp_usPortNum,               .u.us = 1},
     {bp_usGpioLedWanLink,        .u.us = BP_SERIAL_GPIO_19_AH},
     {bp_usGpioLedWanAct,         .u.us = BP_GPIO_31_AH},
     {bp_ulAfeId,                 .u.ul = BP_AFE_CHIP_GFAST_CH1 | BP_AFE_LD_6304 |
   BP_AFE_FE_REV_6304_REV_12_4_60 },
     {bp_usGpioAFELDPwr,          .u.us = BP_GPIO_35_AH}, // Line Driver 1 = "Ext"
     {bp_usGpioAFELDData,         .u.us = BP_GPIO_36_AH},
     {bp_usGpioAFELDClk,          .u.us = BP_GPIO_37_AH},
     {bp_usGpioAFELDPwrBoost,     .u.us = BP_GPIO_38_AH},
     {bp_usIntfEnd},
```

Example: The following code defines one SGMII interface and one xPON interface with the same management parameter IDs:

```
    {bp_usIntfId,                .u.us = 7},
    {bp_usIntfType,              .u.us = BP_INTF_TYPE_SGMII},
    {bp_usPortNum,               .u.us = SF2_WAN_PORT_NUM}, /* SGMII on EtherWAN port */
    {bp_usIntfMgmtType,          .u.us = BP_INTF_MGMT_TYPE_I2C},
    {bp_usIntfMgmtBusNum,        .u.us = 1},
    {bp_usExtIntrOpticalModulePresence, .u.us = BP_EXT_INTR_2 | BP_EXT_INTR_TYPE_IRQ_BOTH_EDGE |
BP_EXT_INTR_TYPE_IRQ_SENSE_EDGE},
    {bp_usGpio_Intr,             .u.us = BP_GPIO_20_AL},
    {bp_usGpioSfpModDetect,      .u.us = BP_GPIO_20_AL},
    {bp_usSfpSigDetect,         .u.us = BP_GPIO_19_AH },
    {bp_usIntfEnd},

    {bp_usIntfId,                .u.us = 8},
    {bp_usIntfType,              .u.us = BP_INTF_TYPE_xPON},
    {bp_usPortNum,               .u.us = 0},
    {bp_usIntfMgmtType,          .u.us = BP_INTF_MGMT_TYPE_I2C},
    {bp_usIntfMgmtBusNum,        .u.us = 0},
    {bp_usExtIntrOpticalModulePresence, .u.us = BP_EXT_INTR_3 | BP_EXT_INTR_TYPE_IRQ_BOTH_EDGE |
BP_EXT_INTR_TYPE_IRQ_SENSE_EDGE},
    {bp_usGpio_Intr,             .u.us = BP_GPIO_9_AL},
    {bp_usGpioSfpModDetect,      .u.us = BP_GPIO_9_AL},
    {bp_usSfpSigDetect,          .u.us = BP_GPIO_10_AH },
    {bp_usIntfEnd},
```

As the interface based parameter ID does not need an explicit name, some of the existing board parameter IDs are replaced with a generic parameter ID name. The following tables are a summary of these changes when using the physical interface scheme in the board parameters.

Table 1: Parameter Id Name Update When Using Physical Interface Configuration

Existing Parameter ID Name	New Parameter ID Name
bp_ulAfeId0 bp_ulAfeId1	bp_ulAfeId
bp_usGpioIntAFELDPwr bp_usGpioExtAFELDPwr	bp_usGpioAFELDPwr
bp_usGpioIntAFELDMode bp_usGpioExtAFELDMode	bp_usGpioAFELDMode
bp_usGpioIntAFELDClk bp_usGpioExtAFELDClk	bp_usGpioAFELDClk
bp_usGpioIntAFELDData bp_usGpioExtAFELDData	bp_usGpioAFELDData
bp_usGpioExtAFEReset	bp_usGpioAFEReset
bp_usGpioLedWanData bp_usGpioSecLedWanData	bp_usGpioLedWanAct
bp_usGpioLedWanError bp_usGpioSecLedWanError	bp_usGpioLedWanErr
bp_usGpioLedAdsl bp_usGpioSecLedAdsl	bp_usGpioLedWanLink

Table 1: Parameter Id Name Update When Using Physical Interface Configuration (Continued)

Existing Parameter ID Name	New Parameter ID Name
bp_usGpioLedAdslFail bp_usGpioSecLedAdslFail	bp_usGpioLedWanLinkFail
bp_usGpioLedGpon bp_usGpioLedEpon	bp_usGpioLedWanLink
bp_usGpioLedGponFail bp_usGpioLedEponFail bp_usGpioLedOpticalLinkFail	bp_usGpioLedWanLinkFail
bp_usSFPSerdesMODDEFx bp_usGpioSfpDetect	bp_usGpioSfpModDetect
bp_usSFPSerdesSIGDETx bp_usSgmiiDetect bp_usGpioWanSignalDetected	bp_usSfpSigDetect
bp_usUart1Sdin bp_usUart1Sdout bp_usGpioUart2Sdin bp_usGpioUart2Sdout bp_usGpioUart2Cts bp_usGpioUart2Rts	bp_usUartSdin bp_usUartSdout p_usUartCts bp_usUartRts

Chapter 3: Parameter References

This section lists all the supporting parameters and explains their usage. All chips listed in the chip families below are included. Each of these devices represents the primary chip ID of a family. The BCM63268 chip ID, for example, includes the BCM63168, BCM63169, BCM63268, and BCM63269 devices.

Group A	Group B
BCM6328	BCM6838X
BCM6362	BCM6848X
BCM63268	BCM6858X
BCM63138	BCM6846X
BCM63148	BCM6836X
BCM63158	BCM63158
BCM63381	BCM62119
BCM62118	
BCM4908	

3.1 General Parameter ID Definitions

3.1.1 bp_cpBoardId

Scope

All chips

Type

String

Description

bp_cpBoardId defines the board ID string. The maximum string length is 16.

This parameter must appear first on the list.

3.1.2 bp_cpComment

Scope

All chips

Type

String

Description

bp_cpComment defines additional text information for this board.

3.1.3 bp_elemTemplate

Scope

All chips

Type

pointer to *bp_elem_t*

Description

bp_elemTemplate allows the user to include the parameters from another board. This reduces the size of the code and removes redundant definitions from boardparam.c. The example below defines the board parameters for BCM963168VX_P300, which inherits all parameters from the board BCM963168VX with the exception of *bp_cpBoardId* and *bp_ulAfeId0*, which are different in the new board.

```
static bp_elem_t g_bcm963168vx_p300[] = {
  {bp_cpBoardId,               .u.cp = "963168VX_P300"},
  {bp_ulAfeId0,              .u.ul = BP_AFE_CHIP_INT | BP_AFE_LD_6302 | BP_AFE_FE_ANNEXA
| BP_AFE_FE_REV_6302_REV_7_2_30},
  {bp_elemTemplate,            .u.bp_elemp = g_bcm963168vx},
  {bp_last}
};
```

When the base board parameter has the switch definition block and a new board parameter wants to update one of the parameter IDs, the new parameter must include the entire switch block. For example, if the base board has a switch definition with four ports but the new parameter is an update to port 2, the entire switch block must be kept with the change to port 2 in the new board parameter.

If the base board parameter uses physical interface configuration, the new board parameter can override the interface individually. For example, if the base board parameter defines four interfaces and the new parameter is an update to interface 3, it needs to have only interface 3 in the new parameter, the rest comes from the template.

If the template board parameter has any parameter ID that the new board parameter does not want, then template mechanism cannot be used.

3.1.4 bp_last

Scope

All chips

Type

None

Description

This parameter ID must be the last ID in a board parameter set.

3.2 LED Parameter ID Definitions

An LED can be assigned directly to a GPIO pin or external shift register using the serial LED driver function depending on the hardware design. Certain hardware-controlled LEDs are hard wired to fixed GPIO pins. For example, the WAN activity LED uses GPIO 8 in the BCM63268. These hard wired GPIO pins cannot be used for other LED function. The WLAN and USB device LEDs are also directly controlled by hardware and use other dedicated pins, so they do not need to be specified with a LED parameter ID.

Most of the GPIO pins are multiplexed with alternate functions, if the alternate function is in use, then the GPIO cannot be used to drive an LED. Refer to the following documentation for device specific instructions.

- BCM6316X/BCM6326X: *Using GPIO as LED Drivers and Other Alternate Functions* (see References)
- BCM636X/BCM6328X: *Using GPIO as LED and/or NAND Flash Drivers* (see References)
- BCM63381: *Operation and Alternate Functions of the GPIO Pins* (see References)

The value of the LED parameter ID can be one of the following:

- BP_SERIAL_GPIO_*n*_AL: LED connects to serial shift register and is active low.
- BP_SERIAL_GPIO_*n*_AH: LED connects to serial shift register and is active high.
- BP_GPIO_*n*_AL: LED connects to GPIO pin through LED controller and is active low.
- BP_GPIO_*n*_AH: LED connects to GPIO pin through LED controller and is active high.

  where *n* is the GPIO pin number. The maximum value is device-dependent; check the specific device data sheet for details.
- BP_GPIO_n_AL/BP_LED_USE_GPIO: LED connects to GPIO and is active low.
- BP_GPIO_n_AH/BP_LED_USE_GPIO: LED connects to GPIO and is active high.

If the GPIO pin that drives the LED is also a boot strap pin, the user should always use BP_GPIO_*n*_AL or BP_SERIAL_GPIO_*n*_AL because of the auto-inversion function built into the hardware.

The internal Ethernet PHY link, activity, and speed LEDs are controlled by the GPIO Overlay in "GPIO Parameter ID Definitions" on page 29 and/or Ethernet switch parameter IDs in section "Ethernet Switch Parameter ID Definitions" on page 37.

3.2.1 bp_usGpioLedReserved

Scope

BCM63138, BCM63148, BCM63381, BCM6848X, BCM6858X, BCM4908, BCM62118, BCM63158, BCM6846X, BCM6836X

Type

Unsigned short

Description

Indicates the assignment of a LED function (serial or parallel) without telling the software to do anything with it. This will cause hardware initialization to configure the bit as a LED and initially turn it off. Other software can then control the function by writing to the appropriate bit of the LED register.

3.2.2 bp_usGpioLedAdsl

Scope

BCM6328, BCM6362, BCM63268, BCM63138, BCM63148

Type

Unsigned short

Description

bp_usGpioLedAdsl defines the output pin assignment for ADSL line link up status LED.

3.2.3 bp_usGpioLedAdslFail

Scope

BCM6328, BCM6362, BCM63268, BCM63138, BCM63148

Type

Unsigned short

Description

This parameter ID defines the output pin assignment for ADSL line link down status LED. If it is not defined, software blinks *bp_usGpioLedAdsl* LED to indicate link down.

3.2.4 bp_usGpioSecLedAdsl

Scope

BCM63628, BCM63138, BCM63148

Type

Unsigned short

Description

This parameter ID defines the output pin assignment for second ADSL line link up status LED.

3.2.5 bp_usGpioSecLedAdslFail

Scope

BCM63628, BCM63138, BCM63148

Type

Unsigned short

Description

This parameter ID defines the output pin assignment for second ADSL line link down status LED. If it is not defined, software blinks *bp_usGpioSecLedAdsl* LED to indicate link down.

3.2.6 bp_usGpioLedSesWireless

Scope

Any board with wireless LAN support.

Type

Unsigned short

Description

This parameter ID defines the output pin assignment for the Wi-Fi Protected Setup LED.

3.2.7 bp_usGpioLedWanData

Scope

BCM6328, BCM6362, BCM63268, BCM6838X, BCM6848X, BCM6858X, BCM63138, BCM63148

Type

Unsigned short

Description

This parameter ID defines the output pin assignment for internet activity LED. On some devices this is a hardware-controlled LED:

- For the BCM6362 and BCM6328, it always uses GPIO 1 or serial LED bit 1.
- For the BCM63268, it always uses GPIO 8 or serial LED bit 8.

On the above devices, this parameter can only be set to use the hard wired GPIO number.

3.2.8 bp_usGpioLedWanError

Scope

BCM6328, BCM6362, BCM63268, BCM6858X,BCM63138, BCM63148

Type

Unsigned short

Description

This parameter ID defines the output pin assignment for Internet connection error LED. If it is not defined, software blinks *bp_usGpioLedWanDataLED* to indicate a connection error.

3.2.9 bp_usGpioLedBlPowerOn

Scope

BCM6328, BCM6362, BCM63268, BCM63138, BCM63148

Type

Unsigned short

Description

This parameter ID defines the output pin assignment for Power ON LED.

3.2.10 bp_usGpioLedBlStop

Scope

All chips

Type

Unsigned short

Description

This parameter ID defines the output pin assignment for POST Fail LED. This LED will also illuminate if the CFE enters its interactive console mode.

3.2.11 bp_usGpioLedGpon

Scope

BCM6838X, BCM6848X, BCM6858X

Type

Unsigned short

Description

This parameter ID defines the output pin assignment for GPON link up status LED.

3.2.12 bp_usGpioLedMoCAFail

Scope

BCM6829, BCM6816, BCM6819

Type

Unsigned short

Description

This parameter ID defines the output pin assignment for MoCA connection down LED. If it is not defined, software blinks *bp_usGpioLedMoCA* to indicate MoCA connection error.

3.2.13 bp_usGpioLedVoip

Scope

Any board that supports voice card.

Type

Unsigned short

Description

This parameter ID is not used.

3.2.14 bp_usGpioVoip1Led

Scope

Any board that supports voice card.

Type

Unsigned short

Description

This parameter ID defines the output pin assignment for the first VoIP status LED.

3.2.15 bp_usGpioVoip1LedFail

Scope

Any board that supports voice card.

Type

Unsigned short

Description

This parameter ID defines the output pin assignment for the first VoIP status failure LED. If it is not defined, software blinks *bp_usGpioVoip1Led* to indicate the first status failure.

3.2.16 bp_usGpioVoip2Led

Scope

Any board that supports voice card.

Type

Unsigned short

Description

This parameter ID defines the output pin assignment for the second VoIP status LED.

3.2.17 bp_usGpioVoip2LedFail

Scope

Any board that supports voice card.

Type

Unsigned short

Description

This parameter ID defines the output pin assignment for the second status failure LED. If it is not defined, software blinks *bp_usGpioVoip2Led* to indicate the second status failure.

3.2.18 bp_usGpioPotsLed

Scope

Any board that supports voice card.

Type

Unsigned short

Description

This parameter ID defines the output pin assignment for FXO POTS line status LED.

3.2.19 bp_usGpioDectLed

Scope

Any board that supports voice card.

Type

Unsigned short

Description

This parameter ID defines the output pin assignment DECT status LED.

3.2.20 bp_usGpioLedWanAct

Scope

BCM63158

Type

Unsigned short

Description

This parameter ID defines the output pin assignment for WAN activity LED including xDSL and xPON interfaces. This Id replaces the bp_usGpioLedWanData and bp_usGpioSecLedWanData when using physical interface configuration.

3.2.21 bp_usGpioLedWanErr

Scope

BCM63158

Type

Unsigned short

Description

This parameter ID defines the output pin assignment for WAN connection error LED including xDSL and xPON interfaces. This Id replaces the bp_usGpioLedWanError and bp_usGpioSecLedWanError when using physical interface configuration.

3.2.22 bp_usGpioLedWanLink

Scope

BCM63158

Type

Unsigned short

Description

This parameter ID defines the output pin assignment for WAN link up status LED including xDSL and xPON interfaces. This ID replaces the bp_usGpioLedAdsl, bp_usGpioSecLedAdsl, bp_usGpioLedGpon and bp_usGpioLedEpon when using physical interface configuration.

3.2.23 bp_usGpioLedWanLinkFail

Scope

BCM63158

Type

Unsigned short

Description

This parameter ID defines the output pin assignment for WAN link down status LED including xDSL and xPON interfaces. This Id replaces the bp_usGpioLedAdslFail, bp_usGpioSecLedAdslFail, bp_usGpioLedGponFail and bp_usGpioLedEponFail when using physical interface configuration.

3.2.24 bp_usGpioLedUSB

Scope

BCM6838X, BCM6848X, BCM6858X

Type

Unsigned short

Description

This parameter defines the GPIO number connected to USB LED.

3.2.25 bp_usGpioLedUSB2

Scope

BCM6838X, BCM6848X, BCM6858X

Type

Unsigned short

Description

This parameter defines the GPIO number connected to USB 2 LED.

3.3 GPIO Parameter ID Definitions

The value of the GPIO parameter IDs (except for *bp_usGpioOverlay or bp_usSerialLedData/Clk/Mask*) can be one of the following:

- BP_GPIO_*n*_AL, which assigns the function to use GPIO pin *n* and is active low.
- BP_GPIO_*n*_AH, which assigns the function to use GPIO pin *n* and is active high.

  Where *n* is the GPIO pin number and its maximum value is device dependent. Refer to the device-specific data sheet for more details.

3.3.1 bp_ulGpioOverlay

Scope

BCM6318, BCM6328, BCM6362, BCM63268

Type

Unsigned long

Description

This parameter ID defines the bitmap indicating what functions override the regular GPIO function in this board design. The GPIO pins that are overridden with other functions are chip dependent and are not specified in this parameter. For example, when BP_OVERLAY_SERIAL_LEDS bit is set in this parameter, GPIO pins 0 and 1 are overridden with serial LED clock and data pins in the BCM63268, but in the BCM6362 GPIO pins 2 and 3 are overridden. The Broadcom software handles this chip dependency transparently for the user, but the user must ensure that the hardware design connects the right GPIO pins to the shift register's data and clock input.

The value of this parameter is the logic OR of any of the following overlay options. The details of the overlay options are described below.

Overlay Options

BP_OVERLAY_PHY

This overlay option only applies to the BCM63268 devices. It enables the GPIO pins for DSL PHY AFE mode and power control.

When this option is set in the BCM63268, it enables and selects the pair of GPIO 10 and GPIO 11 for internal AFE mode and power control, and the pair of GPIO 12 and GPIO 13 for external AFE mode and power control by default.

User can use this option with the AFE mode and power control parameters, described in "DSL AFE Parameter ID Definitions" on page 47, to manually select the GPIO pins.

BP_OVERLAY_SERIAL_LEDS

This overlay option applies to all chips. It enables two GPIO pins as serial LED clock and data pins. The GPIO pin number is chip dependent. Table 2 shows the pin assignment on each chip.

Table 2: BP_OVERLAY_SERIAL_LEDS GPIO Pin Assignments

Chip	Serial Clock	Serial Data
BCM6362	GPIO_2	GPIO_3
BCM6328	GPIO_7	GPIO_6
BCM63268	GPIO_0	GPIO_1

BP_OVERLAY_EPHY_LED_0

BP_OVERLAY_EPHY_LED_1

BP_OVERLAY_EPHY_LED_2

BP_OVERLAY_EPHY_LED_3

These overlay options apply to the BCM6362, BCM6328, and BCM63268.

The BCM63268 does not have a BP_OVERLAY_EPHY_LED_3 option.

These options override GPIO pins to drive internal EPHY link activity and speed LEDs. The GPIO pin number is chip dependent. Table 3 shows the GPIO pin assignments and corresponding LED register bits (if any) for the LED of EPHYs. See Table 9 through Table 11 for the mapping between these flags and switch port.

For the BCM63268, if a GPIO pin has a corresponding LED register bit, the user does not need to specify the related *BP_OVERLAY_EPHY_LED* x flag if they use the serial shift register to drive the LED diode. The corresponding GPIO pin can then be used for other non-LED purposes.

Table 3: BP_OVERLAY_EPHY_LED_x GPIO Pin Assignments

Chip	LINK ACT LED	SPEED LED
BCM6362	LED_0 to LED_3 Only	GPIO_4 to GPIO_7/LED_4 to LED_7
BCM6328	GPIO_25 to GPIO_28 Only	GPIO_17 to GPIO_20/LED_17 to LED_20
BCM63268	GPIO_9 to GPIO_11/LED_9 to LED_11	GPIO_13 to GPIO_15/LED_13 to LED_15

BP_OVERLAY_GPHY_LED_0

BP_OVERLAY_GPHY_LED_1

These overlay options apply to BCM63268. And BCM63268 does not have BP_OVERLAY_GPHY_LED_1 option. These options override GPIO pins to drive internal GPHY link activity and speed LEDs. The GPIO pin number is chip dependent. Table 2 shows details of GPIO pin assignments and corresponding LED register bits (if any) for the LED of GPHY 0 to GPHY 1. See Table 9 through Table 11 for the mapping between these flags and switch port.

Table 4: BP_OVERLAY_GPHY_LED_x GPIO Pin Assignments

Chip	GPHY 0 LINK ACT	GPHY 1 LINK ACT	GPHY 0 SPEED (SPD0/SPD1)	GPHY 1 SPEED (SPD0/SPD1)
BCM63268	GPIO_12/LED_12	N/A	GPIO_0, 1/LED_0, 1	N/A

BP_OVERLAY_USB_LED

This overlay option applies to BCM6362, BCM6328, and BCM63268. It overrides one GPIO pin as the USB device LED function in some chips. Other chips use hardware dedicate pin. There is no USB host LED. This option should use with BP_OVERLAY_USB_DEVICE. Table 5 shows the GPIO pin assignments.

Table 5: BP_OVERLAY_USB_LED GPIO Pin Assignments

Chip	Device Pin	LED Bit	Notes
BCM6362	GPIO_0	LED 0	Routed through LED 0. Overlay flag required flag required
BCM6328	USB_DEVICE_LED/ USB_PWRON	N/A	Hard wired. Overlay flag NOT required
BCM63268	USB_PWRON2/DEVICE_LED	LED 23	Hard wired but routed through LED 23. Overlay flag required

BP_OVERLAY_USB_DEVICE

All chips support two USB 2.0 ports. By default both ports are configured in USB host mode. This overlay option configure the second port to USB device mode. This option should use with BP_OVERLAY_USB_LED.

This overlay option applies to all chips.

BP_OVERLAY_SPI_SSBn_EXT_CS

BCM6362 Parameters:

- *BP_OVERLAY_SPI_SSB2_EXT_CS*: Enable SPI chip slave select LS_SPI_SSB_2 on GPIO pin 9.
- *BP_OVERLAY_SPI_SSB3_EXT_CS*: Enable SPI chip slave select LS_SPI_SSB_3 on GPIO pin 10.

BCM63268 Parameters:

- *BP_OVERLAY_HS_SPI_SSB4_EXT_CS*: Enable SPI slave chip select SPI_SSB_4 on GPIO pin 16.
- *BP_OVERLAY_HS_SPI_SSB5_EXT_CS*: Enable SPI chip slave select SPI_SSB_5 on GPIO pin 17.
- *BP_OVERLAY_HS_SPI_SSB6_EXT_CS*: Enable SPI chip slave select SPI_SSB_4 on GPIO pin 8.
- *BP_OVERLAY_HS_SPI_SSB7_EXT_CS*: Enable SPI chip slave select SPI_SSB_5 on GPIO pin 9.

BP_OVERLAY_SPI_EXT_CS

These overlay options apply to BCM6328.

For the BCM6328, this overlay option enables the SPI chip slave select function on the SPI_SS_B3 pin. By default this pin is configured as GPIO function.

BP_OVERLAY_PCIE_CLKREQ

This overlay option applies to BCM6328 and BCM63268. It enables the PCIe Clock Request signal PCIE_CLKREQ on GPIO pin. See Table 6 for the detail of the pin assignments.

Table 6: BP_OVERLAY_PCIE_CLKREQ GPIO Pin Assignments

Chip	**GPIO Pin**
BCM6328	GPIO 16
BCM63268	GPIO 23

BP_OVERLAY_UART1

This is option is not used.

3.3.2 bp_usSerialLedData/Clk/Mask

bp_usSerialLedData

bp_usSerialLedClk

bp_usSerialLedMask

Scope

BCM63138, BCM63148, BCM63381, BCM6848, BCM6858X, BCM4908, BCM62118, BCM63158

Description

These three parameter IDs replace the function of enabling BP_OVERLAY_SERIAL_LEDS in the bp_ulGpioOverlay parameter. These three signals identify the pins assigned to the clock, data, and (optional) mask that are sent to an external serial shift register for LEDs.

The bp_usSerialLedData pin can be defined to be active high or active low indicating that the entire shift register should be inverted.

These signals are tied to a fixed GPIO pin. Refer to the PinMux table and data sheet for the supported GPIO pins.

3.3.3 bp_usGpioFpgaReset

Scope

Special test boards only

Type

Unsigned short

Description

This parameter ID defines the output pin assignment for FGPA reset pin

3.3.4 bp_usGpioWirelessPowerDown

Scope

Any board that supports wireless module power down

Type

Unsigned short

Description

This parameter ID defines the output pin assignment for wireless module power down

3.3.5 bp_usGpioPassDyingGasp

Scope

All chips

Type

Unsigned short

Description

This parameter ID defines the output pin that passes the dying gasp signal to other module on the board.

3.3.6 bp_usUart1Sdin

Scope

BCM63268, BCM63381, BCM63138, BCM63148, BCM6838X, BCM6848X, BCM6858X, BCM62118, BCM4908, BCM63158, BCM6846X, BCM6836X

Type

Unsigned short

Description

This parameter defines the GPIO number connected to UART 1 data input.

3.3.7 bp_usUart1Sdout

Scope

BCM63268, BCM63381, BCM63138, BCM63148, BCM6838X, BCM6848X, BCM6858X, BCM62118, BCM4908, BCM63158, BCM6846X, BCM6836X

Type

Unsigned short

Description

This parameter defines the GPIO number connected to UART1 data output.

3.3.8 bp_usGpioUart2Cts

Scope

BCM6848X, BCM6858X, BCM6836X, BCM6846X, BCM6856X, BCM63138, BCM63148, BCM63158, BCM4908, BCM62118

Type

Unsigned short

Description

This parameter defines the GPIO number connected to UART 2 CTS.

3.3.9 bp_usGpioUart2Rts

Scope

BCM6848X, BCM6858X, BCM6836X, BCM6846X, BCM6856X, BCM63138, BCM63148, BCM63158, BCM4908, BCM62118

Type

Unsigned short

Description

This parameter defines the GPIO number connected to UART 2 RTS:w.

3.3.10 bp_usGpioUart2Sdin

Scope

BCM63268, BCM63381, BCM63138, BCM63148, BCM6838X, BCM6848X, BCM6858X, BCM62118, BCM4908, BCM63158, BCM6846X, BCM6836X

Type

Unsigned short

Description

This parameter ID defines the input pin assignment for UART 2 as a standard input. The GPIO number is chip dependent and is shown in Table 7.

Table 7: bp_usGpioUart2Sdin GPIO Pin Assignments

Chip	GPIO Pin
BCM63268	GPIO_12 or GPIO_26 (active high)
BCM6838X	GPIO_14 (active high)
All new chips	Refer to PinMux table

3.3.11 bp_usGpioUart2Sdout

Scope

BCM63268, BCM63381, BCM63138, BCM63148, BCM6838X, BCM6848X, BCM6858X, BCM62118, BCM4908, BCM63158, BCM6846X, BCM6836X

Type

Unsigned short

Description

This parameter ID defines the output pin assignment for UART 2 as standard output. The GPIO number is chip dependent and shown in Table 8.

Table 8: bp_usGpioUart2Sdout GPIO Pin Assignments

Chip	GPIO Pin
BCM63268	GPIO_13 or GPIO_27 (active high)
BCM6838X	GPIO_15 (active high)
All new chips	Refer to PinMux table

3.3.12 bp_usGpioFemtoReset

Scope

All boards with Femto Chip

Type

Unsigned short

Description

This parameter ID defines the output pin assignment for FEMTO chip reset signal.

3.3.13 bp_usGpioSfpDetect

Scope

BCM63138, BCM63148, BCM62118

Type

Unsigned short

Description

This parameter defines the GPIO pin connecting to the MOD_ABS pin of the SFP optical module. Its status indicates if SPF module is plugged in or not.

3.3.14 bp_usGpioSfpModDetect

Scope

BCM63158

Type

Unsigned short

Description

This parameter defines the GPIO pin connecting to the MOD_ABS pin of the SFP optical module. Its status indicates if the SPF module is plugged in or not. This ID replaces the bp_usGpioSfpDetect when using physical interface configuration.

3.3.15 bp_usGpio10GTxDis

Scope

BCM6858X

Type

Unsigned short

Description

This parameter defines the GPIO number connected to the 10G TX transceiver.

It is used to reset the TX in order to handle a remote fault event.

3.3.16 bp_usGpioOpticalModuleFixup

Scope

BCM6858X

Type

Unsigned short

Description

This parameter defines the optical module fixup GPIO number.

The fixup is a power down that is done by a pull up on pin 9 of the Source Photonics XGPON SFP, when it is detected.

3.3.17 bp_usSFPSerdesSIGDET0/1/2/3

Scope

BCM6858X

Type

Unsigned short

Description

This parameter defines the GPIO number connected to the signal detect pin in an SFP transceiver.

3.3.18 bp_usSFPSerdesMODEDEF0/1/2/3

Scope

BCM6858X

Type

Unsigned short

Description

This parameter defines the GPIO number connected to the pin in the SFP transceiver, allowing for optical module presence detection.

3.3.19 bp_usGpioWanSignalDetected

Scope

BCM6838X, BCM6848X, BCM6858X

Type

Unsigned short

Description

This parameter defines the pin assignment and polarity for PON optical TRX signal detect signal.

3.4 Ethernet Switch Parameter ID Definitions

The following parameter IDs define the parameters for internal and external switch configuration on the board. Each switch must define the following parameter IDs:

- *bp_ucPhyType0/1*
- *bp_ucPhyAddress*
- *bp_usConfigType, bp_ulPortMap*
- *bp_ulPhyId*

3.4.1 bp_ucPhyType0

Scope

All chips

Type

Unsigned char

Description

This parameter ID defines the internal port connected to the internal switch. For historical reasons, it is always set to *BP_ENET_EXTERNAL_SWITCH*.

3.4.2 bp_ucPhyType1

Scope

All chips

Type

Unsigned char

Description

This parameter ID defines the external switch type. It is always set to *BP_ENET_EXTERNAL_SWITCH* if present.

3.4.3 bp_ucPhyAddress

Scope

All chips

Type

Unsigned char

Description

This parameter is always set to zero.

3.4.4 bp_usConfigType

Scope

All chips

Type

Unsigned short

Description

This parameter ID defines the switch configuration type. For internal switch it is always set to *BP_ENET_CONFIG_MMAP*. For external switch, it can be set to the following value the management mechanism for the external switch:

BP_ENET_CONFIG_MDIO

Use MDIO interface to configure external switch

BP_ENET_CONFIG_GPIO_MDIO

Use GPIO simulated MDIO interface to configure external switch

BP_ENET_CONFIG_MDIO_PSEUDO_PHY

Use MDIO PSEDUO PHY to configure external switch

BP_ENET_CONFIG_SPI_SSB_x

Use SPI interface to configure external switch.

- BP_ENET_CONFIG_SPI_SSB_0
- BP_ENET_CONFIG_SPI_SSB_1
- BP_ENET_CONFIG_SPI_SSB_2
- BP_ENET_CONFIG_SPI_SSB_3

Use SPI interface to configure external switch

BP_ENET_CONFIG_HS_SPI_SSB_x

Use High Speed SPI interface to configure external switch:

- BP_ENET_CONFIG_HS_SPI_SSB_0
- BP_ENET_CONFIG_HS_SPI_SSB_1
- BP_ENET_CONFIG_HS_SPI_SSB_2
- BP_ENET_CONFIG_HS_SPI_SSB_3
- BP_ENET_CONFIG_HS_SPI_SSB_4
- BP_ENET_CONFIG_HS_SPI_SSB_5
- BP_ENET_CONFIG_HS_SPI_SSB_6
- BP_ENET_CONFIG_HS_SPI_SSB_7

3.4.5 bp_ulPortMap

Scope

All chips

Type

Unsigned long

Description

This parameter ID defines the bitmap of the ports used in the switch. When a port is populated with RJ45 connector, either a port with integrated PHY or a port using external PHY, or connected to external switch, or connected to WAN interface, the corresponding bit in *bp_ulPortMap* should be set to one. For example, if port 0 and 1 are populated and port 4 is connected to external switch, the *bp_ulPortMap* for the internal switch should be set to 0x13.

Table 9 through Table 11 show the internal switch's port type, mapping, PHY ID, and other properties.

Table 9: BCM63268 Switch Configuration

Port	Name	Type	PHY ID	LED Overlay Flag
0	EPHY1	GEMAC + EPHY	1+EPHY Base Addr	BP_OVERLAY_EPHY_LED_0
1	EPHY2	GEMAC + EPHY	2+EPHY Base Addr	BP_OVERLAY_EPHY_LED_1
2	EPHY3	GEMAC + EPHY	3+EPHY Base Addr	BP_OVERLAY_EPHY_LED_2
3	GPHY1	GEMAC + GPHY	4	BP_OVERLAY_GPHY_LED_0
4	RGMII_1	RGMII/MII/RvMII GEMAC	External	N/A
5	RGMII_2	RGMII GEMAC	External	N/A
6	RGMII_3	RGMII/MII/RvMII GEMAC	External	N/A
7	RGMII_4	RGMII GEMAC	External	N/A

Table 10: BCM6328 Switch Configuration

Port	Name	Type	PHY ID	LED Overlay Flag
0	EPHY1	FEMAC + EPHY	1	BP_OVERLAY_EPHY_LED_0
1	EPHY2	FEMAC + EPHY	2	BP_OVERLAY_EPHY_LED_1
2	EPHY3	FEMAC + EPHY	3	BP_OVERLAY_EPHY_LED_2
3	EPHY4	FEMAC + EPHY	4	BP_OVERLAY_EPHY_LED_3
4	GMII_1	RGMII/MII/RvMII GEMAC	External	N/A

Table 11: BCM6362 Switch Configuration

Port	Name	Type	PHY ID	LED Overlay Flag
0	EPHY1	FEMAC + EPHY	1	BP_OVERLAY_EPHY_LED_0
1	EPHY2	FEMAC + EPHY	2	BP_OVERLAY_EPHY_LED_1
2	EPHY3	FEMAC + EPHY	3	BP_OVERLAY_EPHY_LED_2
3	EPHY4	FEMAC + EPHY	4	BP_OVERLAY_EPHY_LED_3
4	GMII_1	RGMII/MII/RvMII GEMAC	External	N/A
5	GMII_2	RGMII GEMAC	External	N/A

Table 12: BCM63138/BCM63148/BCM62118/BCM4908/BCM63158 Switch Configuration

Port	Name	Type	PHY ID
0	GPHY0	GEMAC + GPHY	GPHY_BASE_ADDRESS+0
1	GPHY1	GEMAC + GPHY	GPHY_BASE_ADDRESS+1
2	GPHY2	GEMAC + GPHY	GPHY_BASE_ADDRESS+2
3	GPHY3	GEMAC + GPHY	GPHY_BASE_ADDRESS+3
Cross bar	GPHY4	GEMAC + GPHY	GPHY_BASE_ADDRESS+4

Each can have bp_usLinkLed, bp_usSpeedLed100, or bp_usSpeedLed1000 parameters.

3.4.6 bp_ulPhyId*n*

Scope

All chips

Type

Unsigned long

Description

These parameter IDs define the PHY ID for corresponding port. Only the eight LSBs are used to represent the actual PHY ID. The upper bits are used for special flags. Refer to *boardparam.h* for all the flag definitions. Specify *bp_ulPhyIdx* in the board parameter set when the port is used.

For internal switch ports with integrated PHY, the port PHY ID is fixed for each port. See the switch configurations in Table 9 through Table 11 for integrated PHY ID on each chip.
The PHY ID is defined as:

```
BP_PHY_ID_x | PHY_INTERNAL | PHY_INTEGRATED_VALID
```

where BP_PHY_ID_x is the internal PHY ID, as shown below.

```
{bp_ulPhyId2,  .u.ul = BP_PHY_ID_3 | PHY_INTERNAL | PHY_INTEGRATED_VALID }
```

The PHY ID for the port on the internal switch with external PHY is defined as:

```
BP_PHY_ID_y | MAC_IFACE_VALID | MAC_IF_type | PHY_EXTERNAL| PHY_INTEGRATED_VALID
```

where *BP_PHY_ID_y* is external PHY ID and *MAC_IF_type* can be MAC_IF_RGMII, MAC_IF_GMII, MAC_IF_MII, or MAC_IF_RvMII, as in this example:

```
{bp_ulPhyId4,          .u.ul = BP_PHY_ID_24 | MAC_IFACE_VALID | MAC_IF_RGMII |
PHY_INTEGRATED_VALID | PHY_EXTERNAL},
```

The PHY ID for the port on the external switch is defined same as internal port with integrated PHY.

The software has some backward-compatibility logic that permits existing less explicit definitions to work. For example, the user can just define the PHY ID without any flags if the PHY ID is less than 0x10 for the port with integrated PHY. But new boards should explicitly define the port PHY ID based on rule set forth above.

If an internal switch port is connected to an external switch, the PHY ID should be defined as: Interface_Type|EXTSW_CONNECTED for BCM63268 chips and Interface_Type for all other chips, where Interface_Type can be RGMII_DIRECT, GMII_DIRECT and MII_DIRECT.

Example:

```
{bp_ulPhyId4,             .u.ul = RGMII_DIRECT | EXTSW_CONNECTED},
```

Note that for the non BCM63268 chips, the internal switch ports are not available when external switch is used.

When connecting to other external MII entity such as Femto cell, specify the MII interface type and set the bit for that port in the bp_ulPortMap.

.

```
{bp_ulPortMap,               .u.ul = 0x1f},
{bp_ulPhyId4,                .u.ul = MII_DIRECT},
```

3.4.7 bp_usEphyBaseAddress

Scope

BCM63268

Type

Unsigned short

Description

These parameter IDs define the integrated EPHY base address. The EHPY base address is a 5-bit number. This parameter allows the user to specify the two MSB bits of the base address, so the three LSB bits must be zero.

3.4.8 bp_usGphyBaseAddress

Scope

BCM63138, BCM63148, BCM63268, BCM62118, BCM4908, BCM63158

Type

Unsigned short

Description

These parameter IDs define the integrated GPHY base address. The GHPY base address is a 5-bit number. This parameter allows user to specify the two MSB bits of the base address, so the three LSB bits must be zero.

3.4.9 bp_usSpeedLed100/bp_usSpeedLed1000

Scope

BCM63381, BCM63138, BCM63148, BCM62118, BCM4908, BCM63158

Type

Unsigned short

Description

These parameter IDs define the GPIO output pin or serial LED channel assignment for the integrated GPHY’s link speed LEDs. These are switch, hardware-controlled LEDs and can only be used with certain GPIO pins. Refer to the PinMux table and data sheet for the supported GPIO pins.

3.4.10 bp_usLinkLed

Scope

BCM63381, BCM63138, BCM63148, BCM62118, BCM4908, BCM6848X, BCM63158

Type

Unsigned short

Description

This parameter defines the GPIO output pin or serial LED channel assignment for the integrated GPHY's link activity LED. This is a switch hardware-controlled LED and can only be used with certain GPIO pins. Refer to the PinMux table and data sheet for the supported GPIO pins.

3.4.11 bp_usNetLed0/bp_usNetLed1/bp_usNetLed2/bp_usNetLed3

Scope

BCM63158, BCM6858X, BCM6846X, BCM6836X

Type

Unsigned short

Description

These parameter IDs define GPIO output pin or serial LED channel assignment for the switch port LEDs. These are switch hardware-controlled LEDs and can only be used with certain GPIO pins. Refer to the PinMux table and data sheet for the supported GPIO pins.

The LED must be configured as either a link speed LED or a link activity LED or both, based on the LED function parameter ID following them.

bp_usNetLed3 can only be used for link activity for all speeds.

3.4.12 bp_ulNetLedLink

Scope

BCM63158, BCM6858X, BCM6846X, BCM6836X

Type

Unsigned long

Description

This LED function parameter ID indicates the preceding bp_usNetLedX is used for link speed status LED purpose and it defines the link speed that the LED should turn ON or OFF. This is switch hardware-controlled LED and can only be used with certain GPIO pins. Refer to the PinMux table and data sheet for the supported GPIO pins.

The supported link speed definitions are:

- BP_NET_LED_SPEED_10
- BP_NET_LED_SPEED_100
- BP_NET_LED_SPEED_1G
- BP_NET_LED_SPEED_2500
- BP_NET_LED_SPEED_10G

Some common speed combination is also supported:

- BP_NET_LED_SPEED_FAE 10 Mbps and 100 Mbps fast Ethernet speed
- BP_NET_LED_SPEED_GBE 1 Gbps and fast Ethernet speed

And this is for all supported speed:

- BP_NET_LED_SPEED_ALL

3.4.13 bp_ulNetLedActivity

Scope

BCM63158, BCM6858X, BCM6846X, BCM6836X

Type

Unsigned long

Description

This LED function parameter ID indicates the preceding bp_usNetLedX is used for link activity LED purpose and it defines the corresponding link speed that the LED should blink.

It can use all the speed definitions for bp_ulNetLedLink above or use the following definition for all supported speed:

- BP_NET_LED_ACTIVITY_ALL

3.4.14 bp_usGpioLedAggregateAct

Scope

BCM63138, BCM63148, BCM62118, BCM4908, BCM63158

Type

Unsigned short

Description

This LED function parameter ID defines the output pin assignment for the switch aggregated link activity LED. It is a hardware controlled LED. Refer to the PinMux table for supported GPIO pins.

3.4.15 bp_usGpioLedAggregateLnk

Scope

BCM63138, BCM63148, BCM62118, BCM4908, BCM63158

Type

Unsigned short

Description

This LED function parameter ID defines the output pin assignment for the switch aggregated link status LED. It is a hardware controlled LED. Refer to the PinMux table for supported GPIO pins.

3.4.16 bp_usPhyConnType

Scope

All

Type

Unsigned short

Description

This parameter is used to identify the phy connection type for a given PHY ID. This field is required for MoCA and PLC. The following values can be used:

PHY_CONN_TYPE_INT_PHY
PHY_CONN_TYPE_EXT_PHY
PHY_CONN_TYPE_EXT_SW
PHY_CONN_TYPE_EPON
PHY_CONN_TYPE_GPON
PHY_CONN_TYPE_MOCA
PHY_CONN_TYPE_PLC
PHY_CONN_TYPE_FEMTO
PHY_CONN_TYPE_MOCA_ETH

3.4.17 bp_ucPhyDevName

Scope

All

Type

Unsigned char

Description

This parameter is used to specify an alternative name for a given device. The default name for a port is ETHX. This parameter is required for MoCA and PLC interfaces.

3.4.18 bp_usOamIndex

Scope

All

Type

Unsigned short

Description

This parameter is used to specify an alternative index for the management layer.

The default index is the index of the port : PhyIdX.

3.4.19 bp_ulPortMaxRate

Scope

BCM6362 and BCM63268

Type

long

Description

This parameter is used to specify a maximum bit rate for a given port. It is intended for use with RGMII connections where the connected device does not support the full rate.

3.4.20 bp_ulPortFlags

Scope

All chips

Type

Unsigned long

Description

This parameter defines extra information for a PhyId, possible flags can be:

PORT_FLAG_MGMT (FTTDP Only)

PORT_FLAG_ATTACHED (FTTDP Only)

PORT_FLAG_WAN_ONLY – Port is capable to be used only for WAN role

PORT_FLAG_RX_INTERNAL_DELAY – RGMII RX delay needed

PORT_FLAG_TX_INTERNAL_DELAY – RGMII TX delay needed

PORT_FLAG_SWAP_PAIR – Swap the Ethernet pairs

3.5 DSL AFE Parameter ID Definitions

3.5.1 bp_ulAfeId0

Scope

BCM6328, BCM6362, BCM63268, BCM63138, BCM63148, BCM63381

Type

Unsigned Long

Description

This parameter ID defines internal AFE ID for the board. Refer to *Configuring AFE_ID for xDSL PHY Firmware* for more details (see References).

3.5.2 bp_ulAfeId1

Scope

BCM63268, BCM63138, BCM63148

Type

Unsigned short

Description

This parameter ID defines external AFE ID for the board. Refer to *Configuring AFE_ID for xDSL PHY Firmware* for more details (see References).

3.5.3 bp_usGpioExtAFEReset

Scope

BCM63268, BCM63138, BCM63148

Type

Unsigned short

Description

This parameter ID defines output pin assignment for the external AFE reset signal.

3.5.4 bp_usGpioExtAFELDPwr, bp_usGpioExtAFELDMode, bp_usGpioIntAFELDPwr, and bp_usGpioIntAFELDMode

Scope

BCM63268, BCM63138, BCM63148, BCM63381

These parameters must be used in conjunction with the BP_OVERLAY_PHY overlay option for BCM63268.

Type

Unsigned short

Description

These parameter IDs define output pin assignment for the external and internal AFE line driver power and mode control signals for BCM63268. The DSL PHY hardware block has the choices to enable i_gpio_vdsl_ctrl[1:0] and i_gpio_vdsl_ctrl[3:2] output to certain pairs of GPIO pin for line driver control based on these parameters' setting. The following table shows all the possible combinations of the GPIO pin usage:

Table 13: LD Control Pin Combinations

LD Control	GPIO Pin Assignment			
vdsl_ctrl[1:0]	GPIO[11:10]	GPIO[11:10]	GPIO[25:24]	GPIO[25:24]
vdsl_ctrl[3:2]	GPIO[13:12]	GPIO[27:26]	GPIO[27:26]	GPIO[13:12]

The user can assign the internal and/or external pair of LD Power and Mode parameter to any pair of GPIO pin pairs: [BP_GPIO_10_AH, BP_GPIO_11_AH], [BP_GPIO_12_AH, BP_GPIO_13_AH], [BP_GPIO_24_AH, BP_GPIO_25_AH], [BP_GPIO_26_AH, BP_GPIO_27_AH] based on the values in Table 13. GPIO pins must be assigned in a pair to the line driver mode and power control. If one GPIO pin in the pair is used for one line driver control function, the other GPIO pin in the pair must be used for the other function of line driver control and cannot be used for any other GPIO purpose.

For newer DSL chips, refer to the PinMux table for the supported GPIO pins.

3.5.5 bp_usGpioAFELDRelay

Scope

BCM6328, BCM6362, BCM63268, BCM63138, BCM63148, BCM63158

Type

Unsigned short

Description

This parameter ID defines output pin assignment for the AFE LD Relay signal, that switches AFE from xDSL to Gfast mode.

3.5.6 bp_usGpioAFELDPwrBoost

Scope

BCM63138, BCM63148, BCM63158

Type

Unsigned short

Description

This parameter ID defines the GPIO output pin assignment for enabling more transmit power for Gfast 106b mode.

3.5.7 bp_usGpioExtAFELDClk

Scope

BCM6328, BCM6362, BCM63268, BCM63138, BCM63148

Type

Unsigned short

Description

This parameter ID defines output pin assignment for the external AFE LD clock signal.

3.5.8 bp_usGpioExtAFELDData

Scope

BCM6328, BCM6362, BCM63268, BCM63138, BCM63148

Type

Unsigned short

Description

This parameter ID defines output pin assignment for the external AFE LD data signal.

3.5.9 bp_usGpioIntAFELDClk

Scope

BCM63138, BCM63148, BCM63268, BCM63381

Type

Unsigned short

Description

This parameter ID defines output pin assignment for the internal AFE LD clock signal.

3.5.10 bp_usGpioIntAFELDData

Scope

BCM63138, BCM63148, BCM63268, BCM63381

Type

Unsigned short

Description

This parameter ID defines output pin assignment for the internal AFE LD data signal.

3.5.11 bp_ulAfeId

Scope

BCM63158

Type

Unsigned long

Description

This parameter ID defines AFE ID for the board. Refer to *Configuring AFE_ID for xDSL PHY Firmware* (see References) for more details. This ID replaces the `bp_ulAfeId0` and `bp_ulAfeId1` when using physical interface configuration.

3.5.12 bp_usGpioAFELDPwr, bp_usGpioAFELDMode, bp_usGpioAFELDClk, bp_usGpioAFELDData

Scope

BCM63158

Type

Unsigned short

Description

These parameter IDs define output pin assignment for the AFE LD power, mode, clock and data signal. These IDs replace the Int and Ext version of the IDs when using physical interface configuration.

3.5.13 bp_usGpioAFEReset

Scope

BCM63158

Type

Unsigned short

Description

This parameter ID defines output pin assignment for the AFE reset signal.This Id replaces the bp_usGpioExtAFEReset when using physical interface configuration.

3.5.14 bp_usGpioAFEVR5P3PwrEn

Scope

BCM63138, BCM63148, BCM63158

Type

Unsigned short

Description

This parameter ID defines output pin assignment for the AFE 5.3V voltage regular enabling signal.

3.6 GPON/EPON Parameter ID Definitions

3.6.1 bp_usGponOpticsType

Scope

BCM6838X, BCM6848X, BCM63158, BCM6858X, BCM6846X, BCM6836X

Type

Unsigned short

Description

This parameter ID defines GPON optics type. It supports the following types:

BP_GPON_OPTICS_TYPE_LEGACY

BP_GPON_OPTICS_TYPE_PMD

3.6.2 bp_usAePolarity

Scope

BCM6838X, BCM6848X, BCM63158, BCM6858X, BCM6846X, BCM6836X

Type

Unsigned short

Description

This parameter defines the polarity of the AE TRX

3.6.3 bp_usRogueOnuEn

Scope

BCM6838X, BCM6848X, BCM63158, BCM6858X, BCM6846X, BCM6836X

Type

Unsigned short

Description

This parameter defines if the Rogue ONU feature is supported

3.6.4 bp_usGpioPonTxEn

Scope

BCM6838X, BCM6848X, BCM63158, BCM6858X, BCM6846X, BCM6836X

Type

Unsigned short

Description

This parameter ID defines the pin assignment and polarity for PON TX.

3.6.5 bp_usGpioPonRxEn

Scope

BCM6838X, BCM6848X, BCM63158, BCM6858X, BCM6846X, BCM6836X

Type

Unsigned short

Description

This parameter ID defines the pin assignment and polarity for PON RX.

3.6.6 bp_InvSerdesTxPol

Scope

BCM6838X, BCM6848X, BCM6858X, BCM6836X, BCM6846X, BCM6856X, BCM63158

Type

Unsigned short

Description

This parameter defines the polarity of the WAN SerDes in the TX direction.

It can be set to either pmd_use_def_polarity or pmd_polarity_invert

3.6.7 bp_InvSerdesRxPol

Scope

BCM6838X, BCM6848X, BCM6858X, BCM6836X, BCM6846X, BCM6856X, BCM63158

Type

Unsigned short

Description

This parameter defines the polarity of the WAN SerDes in the RX direction.

It can be set to either pmd_use_def_polarity or pmd_polarity_invert.

3.6.8 bp_usPonLbe

Scope

BCM6858X, BCM6836X, BCM6846X, BCM6856X

Type

Unsigned short

Description

This parameter defines the GPIO number for the laser burst enable pin in the PON module.

The pin controls the time in which the PON can send data.

3.6.9 bp_usExtIntrPmdAlarm

Scope

BCM6838X, BCM6848X, BCM63158, BCM6858X, BCM6846X, BCM6836X

Type

Unsigned short

Description

This parameter ID represents the GPIO from which the interrupt from the PMD is received (used as external interrupt).

3.6.10 bp_pmdFunc

Scope

BCM6838X, BCM6848X, BCM6858X, BCM6836X, BCM6846X, BCM6856X, BCM63158

Type

Unsigned short

Description

This parameter defines the functionality of the PMD and BOSA.

Available flags are:

- BP_PMD_APD_REG_DISABLED/BP_PMD_APD_REG_ENABLED
- BP_PMD_APD_TYPE_FLYBACK/BP_PMD_APD_TYPE_BOOST

3.6.11 bp_usGpioPmdReset

Scope

BCM6838X, BCM6848X, BCM63158, BCM6858X, BCM6846X, BCM6836X

Type

Unsigned short

Description

This parameter ID represents the GPIO used to reset the PMD chip.

3.6.12 bp_usPmdMACEwakeEn

Scope

BCM6838X, BCM6848X, BCM63158, BCM6858X, BCM6846X, BCM6836X

Type

Unsigned short

Description

This parameter ID represents the GPIO used for the early wake up signal to the PMD.

3.7 External Interrupt Parameter ID Definitions

3.7.1 bp_usGpio_Intr

Scope

BCM6838X, BCM6858X, BCM6836X, BCM6846X, BCM6856X, BCM4908, BCM62118, BCM4908, BCM63158

Type

Unsigned short

Description

This parameter may be used following any interrupt parameter ID, including bp_usButtonExtIntr, bp_usExtIntrSesBtnWireless, bp_usExtIntrResetToDefault, bp_usExtIntrMocaHostIntr, bp_usExtIntrMocaSBIntr0, bp_usExtIntrMocaSBIntr1, bp_usExtIntrLTE, and bp_usExtIntrTrplxrTxFail.

If specified, the system will map the previously specified interrupt against a state change on the specified GPIO. This parameter takes values in the range BP_GPIO_0_AL to BP_GPIO_141_AL and GPIO_0_AH to GPIO_141_AH. The AL suffix implies active-low, and the AH suffix implies active-high. The interrupt will be mapped when the GPIO transitions into its active state.

Example:

```
{bp_usButtonExtIntr,          .u.us = BP_EXT_INTR_1},
{bp_usGpio_Intr,              .u.us = BP_GPIO_72_AL},
```

Will cause Ext Intr 1 to trigger when the GPIO signal 72 transitions to the low state.

3.7.2 bp_usExtIntrResetToDefault

Scope

All chips

Type

Unsigned short

Description

This parameter ID defines the external interrupt pin signal that connects to the reset factory default button.

Available values for the BCM63268, BCM6328, and BCM6362 are BP_EXT_INTR_0 to BP_EXT_INTR_3.

This parameter may be immediately followed by a bp_usGpio_Intr parameter ID. If so, the system will automatically configure the specified GPIO to trigger the interrupt specified in this parameter.

NOTE: While still supported, use of the new button parameter IDs is recommended. Refer to the ButtonConfiguration.pdf document included in the release tarball for more details.

3.7.3 bp_usExtIntrSesBtnWireless

Scope

All boards with Wi-Fi support.

Type

Unsigned short

Description

This parameter ID defines the external interrupt pin signal that connects to the reset Wi-Fi protected setup push button.

Available values for the BCM63268, BCM6328, and BCM6362 are BP_EXT_INTR_0 to BP_EXT_INTR_3.

This parameter should be immediately followed by a bp_usGpio_Intr parameter ID. If so, the system will automatically configure the specified GPIO to trigger the interrupt specified in this parameter.

NOTE: While still supported, use of the new button parameter IDs is recommended. Refer to the *ButtonConfiguration.pdf* document included in the release tarball for more details.

3.7.4 bp_usExtIntrOpticalModulePresence

Scope

BCM63138, BCM63148, BCM62118, BCM63158, BCM6858X, BCM6846X, BCM6836X

Type

Unsigned short

Description

This parameter ID defines the external interrupt pin signal that connects to the SFP transceiver MOD_ABS pin.

It allows detection of when an optical module (SFP) is inserted.

3.7.5 bp_usButtonIdx

Scope

All chips

Type

Unsigned short

Description

This parameter specifies a new button. Its value will be a 0-based index to the button number, this value should be less than 3. This parameter ID must be followed by bp_usButtonExtIntr and bp_usGpioIntr parameters, to indicate which GPIO and interrupt the button is associated with. Following these, may be one or more bp_usButtonAction and bp_ulButtonActionParm parameter IDs, which associate actions with button events.

Example:

```
  {bp_usButtonIdx,           .u.us = 1},
  {bp_usButtonExtIntr,       .u.us = BP_EXT_INTR_2 | BP_EXT_INTR_TYPE_IRQ_HIGH_LEVEL },
  {bp_usGpio_Intr,           .u.us = BP_GPIO_11_AH },
  {bp_usButtonAction,        .u.us = BP_BTN_ACTION_PRINT | BP_BTN_TRIG_PRESS },
  {bp_ulButtonActionParm,    .u.ul = (unsigned long)"Hold for 5s to restore to default" },
  {bp_usButtonAction,        .u.us = BP_BTN_ACTION_RESTORE_DEFAULTS | BP_BTN_TRIG_HOLD |
BP_BTN_TRIG_5S },
```

This creates a button associated with external interrupt 2, on GPIO 11, which is considered pressed when the GPIO signal 11 is high. This further associates two actions with the button. When the button is first pressed, it will trigger the 'print' action, printing the string, and when held for five seconds it will trigger the restore to default action.

In addition, other actions may be associated with the specified button index at runtime using the registerPushButtonPressNotifyHook, registerPushButtonHoldNotifyHook, and registerPushButtonReleaseNotifyHook APIs.

3.7.6 bp_usButtonExtIntr

Scope

All chips

Type

Unsigned short

Description

This parameter must be used following bp_usButtonIdx. It specifies which interrupt the given button is mapped to. This parameter may be immediately followed by a bp_usGpioIntr parameter.

3.7.7 bp_usButtonAction

Scope

All chips

Type

Unsigned short

Description

This parameter registers a button action against a button. It must be specified after a bp_usButtonIdx parameter. It may optionally be followed by a bp_ulButtonActionParm parameter, which will specify a parameter to be passed to the action handler.

The value is made up of two parts, an action and a trigger. The action is one of the parameters listed in Table 14.

Table 14: Button Action

Parameter	Description
BP_BTN_ACTION_NONE	Does nothing. This may be used to invalidate a previous release action after a period of time. If you registered a release action, with time 0, and then registered a NONE action with time 3, then the original release action would only get invoked if the button was held for less than three seconds.
BP_BTN_ACTION_SES	Initiates a wireless SES key exchange. If 1905 is compiled in, this will initiate key exchanges on all 1905 interfaces instead.
BP_BTN_ACTION_PLC_UKE	Initiates a PLC key exchange.
BP_BTN_ACTION_RANDOMIZE_PLC	Causes the PLC to select a new random key.
BP_BTN_ACTION_RESTORE_DEFAULTS	Restores the device to factory default settings, and resets the board.
BP_BTN_ACTION_RESET	Causes the board to reset.
BP_BTN_ACTION_PRINT	Causes a message to be printed to the CLI. Takes a parameter, which will be a pointer to a string (cast to unsigned long).

The trigger may be one of the parameters listed in Table 15.

Table 15: Button Trigger

Parameter	**Description**
BP_BTN_TRIG_PRESS	The action is invoked when the button is first pressed. If multiple actions are registered against this trigger, all actions will occur.
BP_BTN_TRIG_HOLD	The action is invoked when the button is held for a period of time. A button release is not required to cause the event to occur.
BP_BTN_TRIG_RELEASE	The action is invoked after the button is released after a specified time (if no time is specified 0 seconds is assumed). Only the action(s) with the largest timeout less than the release time are invoked. If multiple actions are registered with the same timeout all actions with that timeout will be invoked.

NOTE: The trigger may be OR'ed with a timeout trigger of BP_BTN_TRIG_0s to BP_BTN_TRIG_10s.

3.7.8 bp_usCfeResetToDefaultBtnIdx

Scope

All chips

Type

Unsigned short

Description

This parameter specifies the button index to be used to perform a reset to default when the board is in CFE mode. This is used in conjunction with the bp_usButtonIdx parameter, however, it is not considered a button action; it should not be placed between the bp_usButtonIdx parameter and any bp_usButtonAction parameters which apply to that button.

It is not possible to specify a custom trigger for the event – the restore to default will take place immediately when the button is pressed, when the modem is in CFE mode, or if held down over the course of a power cycle.

3.7.9 bp_usExtIntrWanSignalDetected

Scope

BCM6838X

Type

Unsigned short

Description

This parameter defines the interrupt number to trigger for the WAN signal detect event.

3.8 Physical Interface Parameter ID Definitions

3.8.1 bp_usIntfId

Scope

BCM63158

Type

Unsigned short

Description

This parameter specifies the interface Id for the interface and also marks the beginning of the interface instance. This ID is unique number for each interface and usually starts from zero.

3.8.2 bp_usIntfEnd

Scope

BCM63158

Type

Unsigned short

Description

This parameter marks the end of the current interface instance. It does not need to be assigned to any value.

3.8.3 bp_usIntfType

Scope

BCM63158

Type

Unsigned short

Description

This parameter defines the type of the interface. The following types are supported:

BP_INTF_TYPE_xDSL: ADSL, VDSL, GFAST

BP_INTF_TYPE_xPON: GPON, EPON, Active Ethernet

BP_INTF_TYPE_xMII

BP_INTF_TYPE_GPHY

BP_INTF_TYPE_SGMII

BP_INTF_TYPE_I2C

BP_INTF_TYPE_UART

3.8.4 bp_usPortNum

Scope

BCM63158

Type

Unsigned short

Description

This parameter defines the physical port number for the interface within the same group or block such as the port number within the SF2 and DSL block.

3.8.5 bp_usIntfMgmtType

Scope

BCM63158

Type

Unsigned short

Description

If the interface requires management bus to operate, this parameter specifies the management bus type. For example, an xPON interface may require I2C bus to manage its optical module. The supported types are:

BP_INTF_MGMT_TYPE_I2C

BP_INTF_MGMT_TYPE_MDIO

BP_INTF_MGMT_TYPE_SPI

3.8.6 bp_usIntfMgmtBusNum

Scope

BCM63158

Type

Unsigned short

Description

If the interface requires management bus to operate, this parameter specifies the management bus number, such as I2C bus number.

3.9 Voice Parameter ID Definitions

3.9.1 bp_ucDspType0 and bp_ucDspType1

Scope

All chips

Type

Unsigned char

Description

These parameters ID define the type of first and second DSP if available. Available values are:

BP_VOIP_MIPS

BP_VOIP_DSP

BP_VOIP_NO_DSP

3.9.2 bp_ucDspAddress

Scope

All chips

Type

Unsigned char

Description

This parameter ID is always zero.

3.10 SPI Slave Parameter ID Definitions

3.10.1 bp_usGpioSpiSlaveReset

Scope

All chips

Type

Unsigned short

Description

This parameter ID defines the output pin assignment for SPI slave reset pin.

3.10.2 bp_usGpioSpiSlaveBootMode

Scope

All chips

Type

Unsigned short

Description

This parameter ID defines an output pin assignment used to control the boot mode of the SPI slave device.

3.10.3 bp_usSpiSlaveBusNum

Scope

All chips

Type

Unsigned short

Description

This parameter ID defines SPI slave bus number. Set to LEG_SPI_BUS_NUM for low-speed SPI slave and HS_SPI_BUS_NUM for high-speed SPI slave.

3.10.4 bp_usSpiSlaveSelectNum

Scope

All chips

Type

Unsigned short

Description

This parameter ID defines SPI slave chip select number. If the specified chip select signal is multiplexed with a GPIO pin, the user must set one of the following overlay options in *bp_usGpioOverlay* depending on target chip:

BP_OVERLAY_SPI_SSBx_EXT_CS

BP_OVERLAY_HS_SPI_SSBx_EXT_CS

BP_OVERLAY_SPI_EXT_CS

See "bp_ulGpioOverlay" on page 29 for more details.

3.10.5 bp_usSpiSlaveSelectGpioNum

Scope

BCM63138, BCM63148, BCM63158, BCM63381, BCM6848, BCM6858X, BCM62118

Type

Unsigned short

Description

In BCM63138, BCM63148, BCM63381 chips, there may be multiple choices of the GPIO pin selection for a particular SPI slave selection signal through the PinMux setting. This parameter allow user to select which GPIO pin to be used. The board parameter framework automatically set the PinMux properly based on the board parameter configuration.

Also for these chips, it is required to explicitly specify the bp_usSpiSlaveSelectNum and bp_usSpiSlaveSelectGpioNum setting in the board parameter for the SPI interface usage on the voice daughter card.

3.10.6 bp_usSpiSlaveMode

Scope

All chips

Type

Unsigned short

Description

This parameter ID defines SPI slave mode. It can be one of the following values:

SPI_MODE_0 (0)

SPI_MODE_1 (SPI_CPHA)

SPI_MODE_2 (SPI_CPOL)

SPI_MODE_3 (SPI_CPOL|SPI_CPHA).

3.10.7 bp_ulSpiSlaveCtrlState

Scope

All chips

Type

Unsigned short

Description

This parameter ID defines SPI controller state. It can be one of the following values:

SPI_CONTROLLER_STATE_SET

SPI_CONTROLLER_STATE_CPHA_EXT

SPI_CONTROLLER_STATE_GATE_CLK_SSOFF

SPI_CONTROLLER_STATE_ASYNC_CLOCK

3.10.8 bp_ulSpiSlaveMaxFreq

Scope

All chips

Type

Unsigned short

Description

This parameter ID defines SPI slave maximum clock rate in Hertz (Hz).

3.10.9 bp_usSpiSlaveProtoRev

Scope

All chips

Type

Unsigned short

Description

This parameter ID defines SPI slave protocol revision.

3.11 Wireless Parameter ID Definitions

3.11.1 bp_usAntInUseWireless

Scope

All boards with wireless support.

Type

Unsigned short

Description

This parameter ID defines wireless antenna setting. The following values are supported:

BP_WLAN_ANT_MAIN

BP_WLAN_ANT_AUX

BP_WLAN_ANT_BOTH

3.11.2 bp_usWirelessFlags

Scope

All boards with wireless support.

Type

Unsigned short

Description

This parameter ID defines wireless flag. The following values are supported:

BP_WLAN_MAC_ADDR_OVERRIDE

BP_WLAN_EXCLUDE_ONBOARD

BP_WLAN_EXCLUDE_ONBOARD_FORCE

BP_WLAN_USE_OTP

3.12 FTTdp

For Fiber To The Distribution Point parameters refer to the *Fiber to the Distribution Point Implementation* Application Note, see References.

3.13 I2C Parameter ID Definitions

3.13.1 bp_usGpioI2cSda

Scope

All chips with I2C support

Type

Unsigned short

Description

This parameter ID defines the pin assignment for I2C SDA signal.   This is hardware controlled pin. Refer to the PinMux table and data sheet for the supported GPIO pins.

3.13.2 bp_usGpioI2cScl

Scope

All chips with I2C support

Type

Unsigned short

Description

This parameter ID defines the pin assignment for I2C SCL signal.   This is hardware controlled pin. Refer to the PinMux table and data sheet for the supported GPIO pins.

3.13.3 bp_usGpioI2c2Sda

Scope

BCM6858X

Type

Unsigned short

Description

This parameter defines the GPIO number connected to the I2C data.

3.13.4 bp_usGpioI2c2Scl

Scope

BCM6858X

Type

Unsigned short

Description

This parameter defines the GPIO number connected to the I2C clock.

3.13.5 bp_usI2cmuxAddr

Scope

BCM6858X

Type

Unsigned short

Description

This parameter defines the I2C address of the mux for the WAN type.

This address must also be configured in the i2cmux_i2c.c probe list.

Current supported addresses are 0x70 and 0x72.

3.14 I2S Parameter ID Definitions

3.14.1 bp_usI2sSdata

Scope

All chips with I2S support

Type

Unsigned short

Description

This parameter ID defines the pin assignment for I2S SDATA signal. This is a hardware controlled pin. Refer to the PinMux table and data sheet for the supported GPIO pins.

3.14.2 bp_usI2sSclk

Scope

All chips with I2S support

Type

Unsigned short

Description

This parameter ID defines the pin assignment for the I2S SCLK signal. This is a hardware controlled pin. SRefer to the PinMux table and data sheet for the supported GPIO pins.

3.14.3 bp_usI2sLrck

Scope

All chips with I2S support

Type

Unsigned short

Description

This parameter ID defines the pin assignment for I2S LRCK signal. This is a hardware controlled pin. Refer to the PinMux table and data sheet for the supported GPIO pins.

3.15 PCM Parameter ID Definitions

3.15.1 bp_usPcmSdin

Scope

All chips with PCM support

Type

Unsigned short

Description

This parameter ID defines the pin assignment for PCM SDIN signal. This is a hardware controlled pin. Refer to the PinMux table and data sheet for the supported GPIO pins.

3.15.2 bp_usPcmSdout

Scope

All chips with PCM support

Type

Unsigned short

Description

This parameter ID defines the pin assignment for PCM SDOUT signal. This is a hardware controlled pin. Refer to the PinMux table and data sheet for the supported GPIO pins.

3.15.3 bp_usPcmClk

Scope

All chips with PCM support

Type

Unsigned short

Description

This parameter ID defines the pin assignment for PCM CLK signal. This is a hardware controlled pin. Refer to the PinMux table and data sheet for the supported GPIO pins.

3.15.4 bp_usPcmFs

Scope

All chips with PCM support

Type

Unsigned short

Description

This parameter ID defines the pin assignment for PCM FS signal. This is a hardware controlled pin. Refer to the PinMux table and data sheet for the supported GPIO pins.

3.16 Miscellaneous Parameter ID Definitions

3.16.1 bp_usVregSel1P2

Scope

Femtocell test board

Type

Unsigned short

Description

This parameter ID is not supported.

3.16.2 bp_ulInterfaceEnable

Scope

BCM63138, BCM63148, BCM63381, BCM6848X, BCM6858X, BCM62118, BCM4908, BCM63158, BCM6846X, BCM6836X

Type

Unsigned long

Description

This parameter configures the hardware to explicitly enable a particular interface that would not otherwise be enabled. For example, by setting this to BP_PINMUX_FNTYPE_NAND, the pins used for NAND will be connected to the NAND controller even if the device was strapped to boot from SPI.

Available interface enables:

BP_PINMUX_FNTYPE_HS_SPI | chip_select_number

BP_PINMUX_FNTYPE_IRQ | irq_number

BP_PINMUX_FNTYPE_NAND

BP_PINMUX_FNTYPE_SATA

BP_PINMUX_FNTYPE_DECT

BP_PINMUX_FNTYPE_LPORT

3.16.3 bp_usSgmiiDetect

Scope

BCM963148, BCM63138

Type

Unsigned short

Description

This parameter ID defines the input pin for the SerDes optical signal detection status. It is connected to the RX_LOS pin of the optic module if an SPF is used. This is a hardware controlled pin. Refer to the PinMux table and data sheet for the supported GPIO pins.

3.16.4 bp_usSfpSigDetect

Scope

BCM963158

Type

Unsigned short

Description

This parameter ID defines the input pin for the SerDes optical signal detection status. It is connected to the RX_LOS pin of the optic module if an SPF is used. This is a hardware controlled pin. Refer to the PinMux table and data sheet for the supported GPIO pins. This ID replaces bp_usSgmiiDetect when using the physical interface configuration.

3.16.5 bp_ulSimInterfaces

Scope

BCM6838X, BCM6848X

Type

Unsigned long

Description

This parameter defines the type of the smart card (SIM) interface supported.

Available options are:

BP_SIMCARD_GROUPA

BP_SIMCARD_GROUPA_OD

BP_SIMCARD_GROUPB

3.16.6 bp_ulSlicInterfaces

Scope

BCM6838X, BCM6848X

Type

Unsigned long

Description

This parameter defines the type of the SLIC interface supported.

Available options are:

BP_SLIC_GROUPC

BP_SLIC_GROUPD

3.16.7 bp_usMiiMdc

Scope

BCM6848X, BCM6836X, BCM6856X, BCM4908, BCM62118, BCM4908,

Type

Unsigned short

Description

This parameter defines the GPIO number connected to an MDC line of an MDC/MDIO bus.

3.16.8 bp_usMiiMdio

Scope

BCM6848X, BCM6836X, BCM6856X, BCM4908, BCM62118

Type

Unsigned short

Description

This parameter defines the GPIO number connected to an MDIO line of an MDC/MDIO bus.

3.16.9 bp_usUsbPwrOn0/1

Scope

BCM6858X, BCM6836X, BCM6846X, BCM6856X, BCM63148, BCM63158

Type

Unsigned short

Description

This parameter defines the GPIO number connected to USB power.

3.16.10 bp_usUsbPwrFlt0/1

Scope

BCM6858X, BCM6836X, BCM6846X, BCM6856X, BCM63148, BCM63158

Type

Unsigned short

Description

This parameter defines the GPIO number connected to USB power fault.

3.16.11 bp_ulMemoryConfig

Scope

BCM63138, BCM63148, BCM4908, BCM62118, BCM6836X, BCM6856X, BCM63158

Type

Unsigned long

Description

This parameter defines the DDR memory configuration. Details are in the DDR application notes.

3.16.12 bp_ulLedChannelId

Scope

BCM963178

Type

Unsigned long

Description

LED Channel ID that is used when selecting the correct PinMux for a specific GPIO.

On chipsets such as BCM963178, more than one PinMux is associated with different LED channels for a particular GPIO.

Example:

In the BCM963178, GPIO 25 in PinMux 6 sets up LED channel 24, and in PinMux 3 sets up LED channel 16.

This parameter should follow the software LED definition to specify the intended LED channel that is used by the software LED.

Revision History

963XX-SWUM405, November 11, 2018

- Updated: “How Board Parameters Work” on page 9
- Added: “bp_ulLedChannelId” on page 73

963XX-SWUM404, January 15, 2018

- Updated: Chapter 3, Parameter References
- Updated: “bp_elemTemplate” on page 21
- Updated: “LED Changes” on page 12
- Added: “Ethernet Switch LED Configuration” on page 15
- Added: “Physical Interface Configuration” on page 17
- Added: “bp_usGpioLedWanAct” on page 27
- Added: “bp_usGpioLedWanErr” on page 27
- Added: “bp_usGpioLedWanLink” on page 28
- Added: “bp_usGpioLedWanLinkFail” on page 28
- Added: “bp_usGpioLedUSB” on page 28
- Added: “bp_usGpioLedUSB2” on page 28
- Added: “bp_usUart1Sdin” on page 33
- Added: “bp_usUart1Sdout” on page 33
- Added: “bp_usGpioUart2Cts” on page 33
- Added: “bp_usGpioUart2Rts” on page 34
- Added: “bp_usGpioSfpDetect” on page 35
- Added: “bp_usGpioSfpModDetect” on page 35
- Added: “bp_usGpio10GTxDis” on page 35
- Added: “bp_usGpioOpticalModuleFixup” on page 36
- Added: “bp_usLinkLed” on page 42
- Added: “bp_usNetLed0/bp_usNetLed1/bp_usNetLed2/bp_usNetLed3” on page 42
- Added: “bp_ulNetLedLink” on page 43
- Added: “bp_ulNetLedActivity” on page 43
- Added: “bp_usGpioLedAggregateAct” on page 44
- Added: “bp_usGpioLedAggregateLnk” on page 44
- Added: “bp_usOamIndex” on page 45
- Added: “bp_ulPortFlags” on page 46
- Added: “bp_usGpioAFELDPwrBoost” on page 49
- Added: “bp_usGpioIntAFELDClk” on page 49
- Added: “bp_usGpioAFELDPwrBoost” on page 49
- Added: “bp_ulAfeId” on page 50
- Added: “bp_usGpioAFELDPwr, bp_usGpioAFELDMode, bp_usGpioAFELDClk, bp_usGpioAFELDData” on page 50
- Added: “bp_usGpioAFEReset” on page 50
- Added: “bp_usGpioAFEVR5P3PwrEn” on page 51
- Added: “bp_InvSerdesTxPol” on page 52
- Added: “bp_InvSerdesRxPol” on page 52

- Added: "bp_usPonLbe" on page 53
- Added: "bp_usExtIntrOpticalModulePresence" on page 56
- Added: "Physical Interface Parameter ID Definitions" on page 59
- Added: "bp_usExtIntrWanSignalDetected" on page 58
- Added: "I2C Parameter ID Definitions" on page 66
- Added: "I2S Parameter ID Definitions" on page 67
- Added: "bp_usSfpSigDetect" on page 70
- Added: "bp_usMiiMdc" on page 71
- Added: "bp_usMiiMdc" on page 71
- Added: "bp_usSFPSerdesMODEDEF0/1/2/3" on page 36
- Added: "bp_usMiiMdc" on page 71
- Added: "bp_usMiiMdio" on page 72
- Added: "bp_usUsbPwrOn0/1" on page 72
- Added: "bp_usUsbPwrOn0/1" on page 72
- Added: "bp_usUsbPwrFlt0/1" on page 72

963XX-SWUM403-R, May 12, 2015

- Updated: "LED Parameter ID Definitions"
- Updated: "bp_usSerialLedData/Clk/Mask"
- Updated: "bp_usGponOpticsType"
- Added: Support for the BCM6848X devices.
- Added: "bp_usAePolarity"
- Added: "bp_usRogueOnuEn"
- Added: "bp_usGpioPonTxEn"
- Added: "bp_usGpioPonRxEn"
- Added: "bp_ulSimInterfaces"
- Added: "bp_ulSlicInterfaces"
- The BCM6828, BCM6368, BCM6816, and BCM6818 devices are no longer supported in software version 4.16L.XX and have been removed from this document.
- Removed: bp_usGpioLedGponFail
- Removed: bp_usGpioLedMoCA
- Removed: bp_usGpioLedMoCAFail
- Removed: bp_usDuplexLed
- Removed: bp_cpDefaultOpticalParams

963XX-SWUM402-R, July 28, 2014

- Updated: "How Board Parameters Work"
- Updated: "Parameter References"
- Updated: "bp_ulGpioOverlay"
- Updated: Table 1: "BP_OVERLAY_SERIAL_LEDS GPIO Pin Assignments,"
- Updated: Table 2: "BP_OVERLAY_EPHY_LED_x GPIO Pin Assignments,"
- Updated: Table 3: "BP_OVERLAY_GPHY_LED_x GPIO Pin Assignments,"
- Updated: Table 4: "BP_OVERLAY_USB_LED GPIO Pin Assignments,"
- Updated: "bp_usGpioUart2Sdin"
- Updated: "bp_ulInterfaceEnable"

- Added: Table 10: “BCM6328 Switch Configuration,”
- Added: “bp_usPhyConnType”
- Added: “bp_usPhyDevName”
- Added: “bp_ulPortMaxRate”
- Added: “bp_usGpio_Intr”
- Added: “bp_usButtonIdx”
- Added: “bp_usButtonExtIntr”
- Added: “bp_usButtonAction”
- Added: “bp_usCfeResetToDefaultBtnIdx”
- Added: “bp_usGpioSpiSlaveReset”
- Added: “bp_usSpiSlaveSelectGpioNum”
- Added: “bp_usSgmiiDetect”
- Removed: bp_usGpioLaserDis
- Removed: bp_usGpioLaserTxPwrEn
- Removed: Table 11: “BCM6368 Switch Configuration”
- Removed: Table 13: “BCM6818G Switch Configuration”

963XX-SWUM401-R, December 18, 2013

- Updated: Scope of “bp_ulGpioOverlay”
- Updated: “bp_usGpioUart2Sdin”
- Updated: “Adding Support for a New Board”
- Added: “PinMux Check Utility”
- Added: “Crossbar Switch Topology”
- Added: New Parameters:
  “bp_usGpioLedReserved”
  “bp_usSerialLedData/Clk/Mask”
  “bp_usGphyBaseAddress”
  “bp_ulInterfaceEnable”
- Added: Table 14: “BCM63138 Switch Configuration,”
- Added: BCM6838X chip family
- Removed: BCM6828 Switch Configuration Table
- Removed: BCM6816 Switch Configuration Table
- Removed: bp_ulNonPeriphGpioCtrlMap

963XX-SWUM400-R, March 16, 2012

Initial release