BROADCOM®

Data Sheet

BCM84891L

10GBASE-T/5GBASE-T/2.5GBASE-T/1000BASE-T/100BASE-TX Transceiver

General Description

The Broadcom® BCM84891L is a single 10GBASE-T/5GBASE-T/2.5GBASE-T/1000BASE-T/100BASE-TX Ethernet CMOS transceiver. The device performs all physical-layer functions for 10GBASE-T, 5GBASE-T, 2.5GBASE-T, 1000BASE-T, and 100BASE-TX Ethernet on Category 6, 6A, or 7 twisted-pair cable. The 5GBASE-T, 2.5GBASE-T, 1000BASE-T, and 100BASE-TX can operate on standard Category 5e UTP. The BCM84891L supports the XFI/10GBASE-KR, USXGMII, 5000BASE-R, 2500BASE-R, 5000BASE-X, 2500BASE-X, and 1000BASE-X (SGMII) interface for connection to a MAC. The BCM84891L is a highly integrated solution that combines digital adaptive equalizers, ADCs, phase-locked loops, line drivers, encoders, decoders, echo cancelers, crosstalk cancelers, and all the required support circuitry.

The BCM84891L features the Energy Efficient Ethernet (EEE) protocol. EEE enables the BCM84891L to autonegotiate and operate with EEE-compliant link partners to reduce overall system power during low utilization of the link. Broadcom's AutogrEEEn® mode permits legacy systems to enjoy the power saving benefits of EEE.

The BCM84891L automatically negotiates an operating speed with any transceiver on the opposite end of the line. The BCM84891L features Enhanced Cable Diagnostics, which detects common cable problems such as shorts, opens, and cable length.

Applications

- 10-Gigabit Ethernet (GbE) switches and uplinks
- 10GbE server adapter
- 10GbE host control

Features

- Single-chip integrated single Ethernet transceiver-MAC to magnetics:
  - 10GBASE-T IEEE 802.3an
  - 5GBASE-T IEEE 802.3bz
  - 2.5GBASE-T IEEE 802.3bz
  - 1000BASE-T IEEE 802.3ab
  - 100BASE-TX IEEE 802.3u
- IEEE 802.3z
- IEEE 802.3az
- XFI/10GBASE-KR, USXGMII, 5000BASE-X, 2500BASE-X, 5000BASE-R, 2500BASE-R, and 1000BASE-X (SGMII) MAC Interface
- 5G rate over USXGMII/XFI/5000BASE-R/5000BASE-X MAC interface
- 2.5G rate over USXGMII/XFI/2500BASE-R/2500BASE-X MAC interface
- IEEE 1588-2008 version 2-compliant
- IEEE fast retrain and link monitoring
- EEE on 10GBASE-T, 5GBASE-T, 2.5GBASE-T, 1000BASE-T, and 100BASE-TX Auto-sense mode
- Trace matched output impedance
- Line-side loopback
- Low electromagnetic interference (EMI) emissions
- Enhanced cable diagnostics
- Support for jumbo packets up to 18 KB
- Detection and correction of pair swaps (MDI crossover), pair skew, and pair polarity
- Autonegotiation with next page support
- IEEE 1149.1 (JTAG) and IEEE 1149.6 (AC-JTAG) boundary scan support
- Low-power, 0.8V CMOS core
- 1.2V, 1.8V,2.5V, or 3.3V MDIO I/O, 1.8V CMOS I/O
- Device package: 8 mm × 8 mm BGA

1.24 MDIO Command Handler Function

The MDIO Command Handler function is an efficient and robust handshake process to provision and monitor features in the BCM84891L device that guarantees the feature is set properly.

The set of functions that require the MDIO Command Handler are listed in Table 5. The corresponding command CODE value should be written in the Command Register CMD (address: 0x1E.0x4005). During 2-second training, the status returns CMD_SYSTEM_BUSY and the user must wait up to 2 seconds for the command to be executed.

Table 5: Features/Commands Using MDIO Command Handler Process

Feature/Command Name	Description	Command Code
CMD_GET_PAIR_SWAP	—	0x8000
CMD_SET_PAIR_SWAP	—	0x8001
Reserved	—	0x8002
Reserved	—	0x8003
CMD_GET_1588_ENABLE	—	0x8004
CMD_SET_1588_ENABLE	—	0x8005
GET_LIMITED_REACH_MODE_ENABLE	Gets limited reach mode setting.	0x8006
SET_LIMITED_REACH_MODE_ENABLE	Enables or disables limited reach mode setting.	0x8007
CMD_GET_EEE_MODE	—	0x8008
CMD_SET_EEE_MODE	—	0x8009
CMD_GET_EMI_MODE_ENABLE	—	0x800A
CMD_SET_EMI_MODE_ENABLE	—	0x800B
CMD_GET_SUB_LF_RF_STATUS	—	0x800D
GET_KR_MODE_ENABLE	Gets KR mode setting.	0x800E
SET_KR_MODE_ENABLE	Enables or disables KR mode setting.	0x800F
CMD_CLEAR_SUB_LF_RF	—	0x8010
CMD_SET_SUB_LF_RF	—	0x8011
GET_SYNC_E_ENABLE	Get SyncE enable status	0x8012
SET_SYNC_E_ENABLE	Set SyncE enable	0x8013
GET_XFI_2P5G_5G_MODE	—	0X8016
SET_XFI_2P5G_5G_MODE	—	0X8017
SET_EEE_STATISTICS	—	0X801A
GET_EEE_STATISTICS	—	0X801B
GET_JUMBO_PACKET	—	0X801D
SET_JUMBO_PACKET	—	0X801C
GET_PAUSE_FRAME_MODE	—	0x801F
SET_PAUSE_FRAME_MODE	—	0x8020
GET_LED_TYPE	—	0x8021
SET_LED_TYPE	—	0x8022
GET_WAP_MODE_FRAME_TYPE	—	0x8023
SET_WAP_MODE_FRAME_TYPE	—	0x8024
SET_USXGMII	—	0x8026
GET_USXGMII	—	0x8027
GET_XFI_TX_FILTERS	—	0x802B

Table 5: Features/Commands Using MDIO Command Handler Process (Continued)

Feature/Command Name	Description	Command Code
SET_XFI_TX_FILTERS	—	0x802C
GET_XFI_POLARITY	—	0x802D
SET_XFI_POLARITY	—	0x802E
GET_CURRENT_VOLTAGE	—	0x802F
CMD_GET_SNR	—	0x8030
CMD_GET_CURRENT_TEMP	—	0x8031
CMD_SET_UPPER_TEMP_WARNING_LEVEL	—	0x8032
CMD_GET_UPPER_TEMP_WARNING_LEVEL	—	0x8033
CMD_SET_LOWER_TEMP_WARNING_LEVEL	—	0x8034
CMD_GET_LOWER_TEMP_WARNING_LEVEL	—	0x8035
SET_WOL_ENABLE	Enable/disable wake-on-LAN	0x805A
GET_WOL_ENABLE	Get wake-on-LAN settings	0x805B

To execute a feature, write the associated code (command CODE) in the CMD register.

The MDIO Command Handler uses multiple registers to interface with the user. Table 6 shows the registers potentially used by this function.

Table 6: MDIO Command Handler Register Set

Register Name	Description	Address
CMD	Command	0x1E.0x4005
STATUS	Status of command	0x1E.0x4037
DATA1_REG	Data 1 register	0x1E.0x4038
DATA2_REG	Data 2 register	0x1E.0x4039
DATA3_REG	Data 3 register	0x1E.0x403A
DATA4_REG	Data 4 register	0x1E.0x403B
DATA5_REG	Data 5 register	0x1E.0x403C

From Figure 15, the STATUS register [STATUS (address: 0x1E.0x4037)] can report different types of status messages (codes). The status names and values are shown in Table 7.

The Status register is a special bidirectional register that reports the status and can be written by the user.

Table 7: STATUS Codes

STATUS Code Name	Set by	Description	Code Value
CMD_RECEIVED	FW	A command has been received.	0x0001
CMD_IN_PROGRESS	FW	A command is being serviced.	0x0002
CMD_COMPLETE_PASS/ CMD_OPEN_FOR_CMDS	FW	Last command has been executed successfully.	0x0004
CMD_COMPLETE_ERROR/ CMD_OPEN_FOR_CMDS	FW	Execution of the last command produced an error.	0x0008
CMD_SYSTEM_BUSY	FW	System is not ready to accept commands.	0xBBBB

1.24.1 MDIO Command Handler Theory of Operation

Once the system has booted up, the firmware initializes the STATUS register (in Table 7) to CMD_COMPLETE_PASS/CMD_OPEN_FOR_CMDS and the CMD register to 0x0 as default condition (no operation at 0x0 value). User should poll the status register to check the availability of command handler. If the STATUS register is CMD_IN_PROGRESS or SYSTEM_BUSY, the user must wait for certain period of time and check again. If the STATUS register is not CMD_IN_PROGRESS or SYSTEM_BUSY _CMDS, the user can write the command to CMD register. If the CMD register bit 15 is written as 1, an interrupt is generated and firmware changes the STATUS register to CMD_RECEIVED (see Table 6 and Table 7), and it sends a message to the Service thread to execute the command.

When the Service thread receives the message to execute the user command, it calls the MDIO Command Handler function to process the command.

The MDIO Command Handler reads the command and passes the value of the command into a command array. This value is used as index into the array and returns a function pointer that executes the required command.

Each command function changes the STATUS register to CMD_IN_PROGRESS, and then processes the command. After completion, the command function returns CMD_COMPLETE_PASS/CMD_OPEN_CMDS or CMD_COMPLETE_ERROR/CMD_OPEN_FOR_CMDS. The MDIO Command Handle changes the STATUS register to CMD_COMPLETE_PASS/CMD_OPEN_FOR_CMDS or CMD_COMPLETE_ERROR/CMD_OPEN_FOR_CMDS so that the user can poll the status register and know when the command has completed and reads the register results (see Table 5).

A graphical representation of the process for provisioning and monitoring features is shown in the next section.

1.24.2 Command Procedure

To use the MDIO Command Handler either for enabling/disabling a feature or to read the status/information on a feature, use the followings steps:

1. Poll the STATUS register to see whether the previous command is in progress or the system is busy (CMD_IN_PROGRESS or SYSTEM_BUSY). If previous command is in progress or system is busy, check again until the previous command finishes execution and the system is available for taking command.
2. If any parameters are required for the function, write them to the required DATA registers (see Section 1.25, Command Descriptions, for more information)
3. When the firmware is ready for commands, write the Command code/value from Table 5 to the CMD register.
4. Once the command has been written, poll the STATUS register to check whether the command has completed (CMD_COMPLETE_PASS/CMD_FOR_CMDS or CMD_COMPLETE_ERROR/CMD_OPEN_FOR_CMDS).
5. Once the command has completed, read the specified DATA registers (see Table 6) for any saved results for the command, if applicable (see Section 1.25, Command Descriptions, for more information).
6. For provisioning features that require altering port configuration parameters, restart autonegotiation to reestablish link using the new parameters (for example, most provisioning processes require this step).

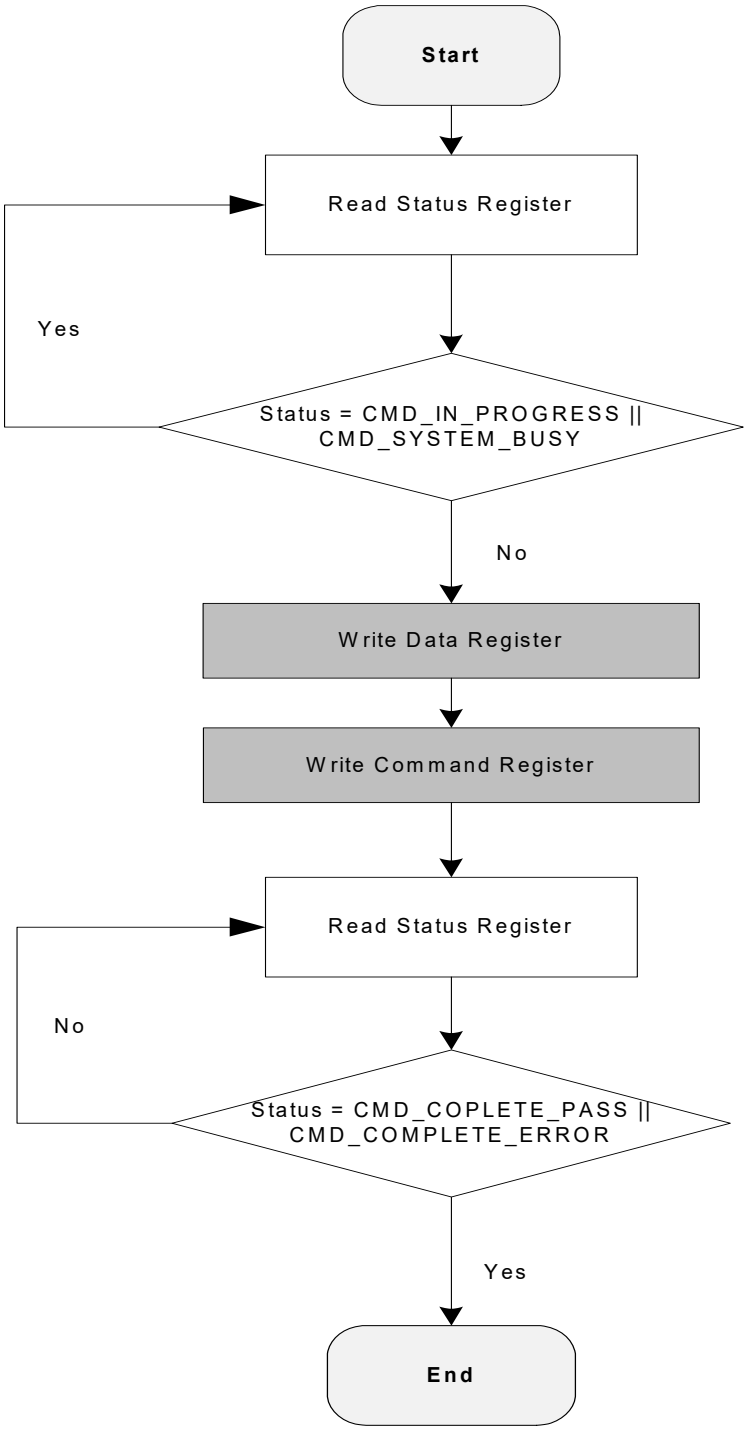
The feature provisioning and monitoring processes are shown in the next section.

NOTE: The recommended polling period for the status register is 100 ms. When a device is in the 10GBASE-T training phase, the command handler status register does not change for up to two seconds. Users should avoid issuing commands during training or start polling the status register after the link status is up.

1.24.3 Provisioning and Monitoring Sequences

Figure 15 provides steps for provisioning the device through the Command Handler.

Figure 15: Provisioning Process #1



1.25 Command Descriptions

1.25.1 Enable/Disable/View Commands

1.25.1.1 CMD_ GET_PAIR_SWAP

Description: Gets pair swap values.	
Input Parameters: None.	**Return Results**: DATA1_REG = Forced/requested current swap pair value. DATA2_REG = Current swap pair value.

1.25.1.2 CMD_SET_PAIR_SWAP

Description: Sets pair swap values.	
Input Parameters: DATA2_REG = Swap pair value to be forced.	**Return Results**: None.
NOTE: CMD_SET_PAIR SWAP requires autonegotiation to take place in order to take effect.	

1.25.1.3 Pair Swap Values

Table 8: Pair Swap Values

Bits	Description
Bits [15:8]	Reserved.
Bits [7:6]	PAIR_D_SEL: 00 = Pair A 01 = Pair B 10 = Pair C 11 = Pair D
Bits [5:4]	PAIR_C_SEL: 00 = Pair A 01 = Pair B 10 = Pair C 11 = Pair D
Bits [3:2]	PAIR_B_SEL: 00 = Pair A 01 = Pair B 10 = Pair C 11 = Pair D
Bits [1:0]	PAIR_A_SEL: 00 = Pair A 01 = Pair B 10 = Pair C 11 = Pair D

1.25.1.4 CMD_ GET_1588_ENABLE

Description:	Returns current value set for internal variable used to enable/disable IEEE 1588 functionality.
Input Parameters: None.	**Return Results**: DATA1: ■ 0 = IEEE 1588 is disabled. ■ 1 = IEEE 1588 is enabled. DATA2 to DATA5: Not used.

1.25.1.5 CMD_SET_1588_ENABLE

Description:	Sets internal variable used to enable/disable IEEE 1588 functionality. Setting to 1 enables IEEE 1588. Setting to 0 disables IEEE 1588. Enable or disable IEEE 1588 functionality prior to link up. If enabling or disabling the IEEE 1588 functionality while the link is already up, bring the link down, and then back up using an AUTONEG restart or similar mechanism.
Input Parameters: DATA1: ■ 0 = Disable IEEE 1588. ■ 1 = Enable IEEE 1588. DATA2 to DATA5: Not used.	**Return Results**: None.
NOTE: CMD_SET_1588_ENABLE requires autonegotiation to take place to take effect.	

1.25.1.6 GET_LIMITED_REACH_MODE_ENABLE

Description:	Returns current value set for internal variable used to enable/disable limited reach mode functionality.
Input Parameters: None.	**Return Results**: DATA1: ■ 0 = Limited reach mode is disabled. ■ 1 = Limited reach mode is enabled. DATA2: Only valid when DATA1 = 1 ■ 0 = Dynamically switch to LR power saving mode level 2 based on cable length. ■ 1 = Always forced to LR power saving mode level 1 (most power saving). ■ 2 = Always forced to LR power saving mode level 2 (less power saving). ■ 3 = Always forced to LR power saving mode level 3 (least power saving). DATA3: Valid only when DATA1 = 1. ■ 0 = Currently is not linked in limited reach mode. ■ 1 = Currently is linked in limited reach mod.e DATA4 to DATA5: Not used.

1.25.1.7 SET_LIMITED_REACH_MODE_ENABLE

Description:	Enables/disables limited reach mode functionality.
Input Parameters: DATA1: ■ 0 = Exit limited reach mode. ■ 1 = Enter limited reach mode. DATA2: Only valid when DATA1 = 1 ■ 0 = Dynamically switch to LR power saving mode level 2 based on cable length. ■ 1 = Always forced to LR power saving mode level 1 (most power saving). ■ 2 = Always forced to LR power saving mode level 2 (less power saving). ■ 3 = Always forced to LR power saving mode level 3 (least power saving). DATA3 to DATA5: Not used.	**Return Results**: None.

1.25.1.8 CMD_GET_EEE_MODE

Description: Gets AutogrEEEn parameters.

Input Parameters: None.

Return Results:

		5G	2.5G	1G/100M	10G
DATA1_REG	Bits	7:6	5:4	3:2	1:0
	EEE Disabled	0	0	0	0
	Native EEE	1	1	1	1
	AutogrEEEn Fixed Latency	2	2	2	2
	AutogrEEEn Variable Latency	3	3	N/A	3
DATA2_REG	AutogrEEEn High Threshold	0	0	N/A	0
DATA3_REG	AutogrEEEn Low Threshold	0x7A12	0x7A12	N/A	0x7A12
DATA4_REG	AutogrEEEn Latency	0x480	0x480	N/A	0x480
DATA5_REG	Reserved	0	0	0	0

NOTE:

- DATA1_REG bit[11:10] defines how bit[3:2] works:
  DATA1_REG Bits: 11:10
  - 0: Bit[3:2] for 1G/100TX EEE
  - 1: Bit[3:2] for 100TX EEE only
  - 2: Bit[3:2] for 1G EEE only
  - 3: Bit[3:2] for 1G/100TX EE
- Although the EEE modes can be different between each speed, the AutogrEEEn parameters for 2.5G/5G/10G in DATA2_REGs to DATA4_REGs are shared. The 1G/100M AutogrEEEn parameters are not programmable and are unaffected by the values in DATA2_REGs to DATA4_REGs.
  DATA1_REG[15:8] = Reserved

Example: When DATA1_REG is 0x00B1, 5G EEE is fixed latency, 2.5G EEE is variable latency, 1G/100M does not advertise EEE, and 10G is native EEE mode. If the data rate is 10G, DATA2_REG through DATA4_REGs are ignored. If the data rate is 5G, DATA4_REG is ignored. If the data rate is 2.5G or 5G, the AutogrEEEn threshold applies.

1.25.1.9 CMD_SET_EEE_MODE

Description: Configures IEEE-EEE and AutogrEEEn.

Input Parameters:

		5G	2.5G	1G/100M	10G
DATA1_REG	Bits	7:6	5:4	3:2	1:0
	EEE Disabled	0	0	0	0
	Native EEE	1	1	1	1
	AutogrEEEn Fixed Latency	2	2	2	2
	AutogrEEEn Variable Latency	3	3	N/A	3
DATA2_REG	AutogrEEEn High Threshold	0	0	N/A	0
DATA3_REG	AutogrEEEn Low Threshold	0x7A12	0x7A12	N/A	0x7A12
DATA4_REG	AutogrEEEn Latency	0x480	0x480	N/A	0x480
DATA5_REG	Reserved	0	0	0	0

NOTE:

- DATA1_REG bit[11:10] defines how bit[3:2] works:
  DATA1_REG Bits: 11:10
  - 0: Bit[3:2] for 1G/100TX EEE
  - 1: Bit[3:2] for 100TX EEE only
  - 2: Bit[3:2] for 1G EEE only
  - 3: Bit[3:2] for 1G/100TX EE
- Although the EEE modes can be different between each speed, the AutogrEEEn parameters for 2.5G/5G/10G in DATA2_REGs to DATA4_REGs are shared. The 1G/100M AutogrEEEn parameters are not programmable and are unaffected by the values in DATA2_REGs to DATA4_REGs.
  DATA1_REG[15:8] = Reserved

Example: When DATA1_REG is 0x00B1, 5G EEE is fixed latency, 2.5G EEE is variable latency, 1G/100M does not advertise EEE, and 10G is native EEE mode. If the data rate is 10G, DATA2_REG to DATA4_REGs are ignored. If the data rate is 5G, DATA4_REG is ignored. If the data rate is 2.5G or 5G, the AutogrEEEn threshold applies.

Return Results: None.

NOTE: CMD_SET_EEE_MODE requires autonegotiation to take place to take effect.

1.25.1.10 CMD_ GET_EMI_MODE_ENABLE

Description: Gets SW Fast Retrain current enable status.

Input Parameters: None.

Return Results:

DATA1: 10G fast retrain enable.

- 0 = 10G Fast Retrain is disabled.
- 1 = 10G Fast Retrain is enabled.

DATA2: 5G fast retrain enable.

- 0 = 5G Fast Retrain is disabled.
- 1 = 5G Fast Retrain is enabled.

DATA3: 2.5G fast retrain enable.

- 0 = 2.5G Fast Retrain is disabled.
- 1 = 2.5G Fast Retrain is enabled.

1.25.1.11 CMD_SET_EMI_MODE_ENABLE

Description:	Gets SW Fast Retrain current enable status.
Input Parameters: DATA1: 10G fast retrain enable. ■ 0 = Disable 10G Fast Retrain. ■ 1 = Enable 10G Fast Retrain. DATA2: 5G fast retrain enable. ■ 0 = Disable 5G Fast Retrain. ■ 1 = Enable 5G Fast Retrain. DATA3: 2.5G fast retrain enable. ■ 0 = Disable 2.5G Fast Retrain. ■ 1 = Enable 2.5G Fast Retrain.	**Return Results**: None.
NOTE: CMD_SET_EMI_MODE_ENABLE requires autonegotiation to take place to take effect.	

1.25.1.12 CMD_GET_SUB_LF_RF_STATUS

Description:	Gets current status on the substitution of FAULTS with IDLE going to the line direction.
Input Parameters: None.	**Return Results**: Status. ■ DATA1: ■ 0 = Copper TX direction LF/RF is passing through ■ 1 = Copper TX direction LF/RF is replaced with IDLE DATA2 to DATA5: Not used.

1.25.1.13 CMD_CLEAR_SUB_LF_RF

Description:	Disables substitution of FAULTS with IDLE going to the line direction.
Input Parameters: 1 = None.	**Return Results**: None.

1.25.1.14 CMD_SET_SUB_LF_RF

Description:	Enables substitution of FAULTS with IDLE going to the line direction.
Input Parameters: 1 = None.	**Return Results**: None.

1.25.1.15 GET_KR_MODE_ENABLE

Description:	Gets KR mode status setting (enable/disable).
Input Parameters: None.	**Return Results**: DATA1: ■ 0 = BCM84891L KR mode is disabled ■ 1 = BCM84891L KR mode is enabled. DATA2 to 5: Reserved

1.25.1.16 SET_KR_MODE_ENABLE

Description:	Sets KR mode status setting (enable/disable).
Input Parameters: DATA1: ■ 0 = BCM84891L KR mode is disabled. ■ 1 = BCM84891L KR mode is enabled. DATA2 to 5: Reserved	**Return Results**: None.

1.25.1.17 GET_SYNC_E_ENABLE

Description:	Get SyncE enable setting.
Input Parameters: None.	**Return Results**: DATA1: SyncE enable. ■ 0 = SyncE is disabled. ■ 1 = SyncE is enabled. DATA2: M0 mux select. ■ 0 = Select P0 10G clock/lock for output on M0. ■ 1 = Select P1 10G/1G clock/lock for output on M0. ■ 2 = Select P2 10G/1G clock/lock for output on M0. ■ 3 = Select P3 10G/1G clock/lock for output on M0. ■ 8 = Select P0 1G clock/lock for output on M0. DATA3: M1 mux select. ■ 0 = Select P0 10G clock/lock for output on M1. ■ 1 = Select P1 10G/1G clock/lock for output on M1. ■ 2 = Select P2 10G/1G clock/lock for output on M1. ■ 3 = Select P3 10G/1G clock/lock for output on M1. ■ 8 = Select P0 1G clock/lock for output on M1. DATA4: FW Control SyncE Configuration. ■ 0 = User Control Sync E configuration. ■ 1 = FW Control Sync E configuration. DATA5[0]: M0 mux SYNCE_OEN. [1]: M1 mux SYNCE_OEN. ■ 0 = Output enabled. ■ 1 = Output disable. DATA5[15:2]: Reserved.

1.25.1.18 SET_SYNC_E_ENABLE

Description:	Set SyncE enable setting.
Input Parameters: DATA1: SyncE enable. ■ 0 = SyncE is disabled. ■ 1 = SyncE is enabled. DATA2: M0 mux select. ■ 0 = Select P0 10G clock/lock for output on M0 (default). ■ 1 = Select P1 10G/1G clock/lock for output on M0. ■ 2 = Select P2 10G/1G clock/lock for output on M0. ■ 3 = Select P3 10G/1G clock/lock for output on M0. ■ 8 = Select P0 1G clock/lock for output on M0. DATA3: M1 mux select. ■ 0 = Select P0 10G clock/lock for output on M1 (default). ■ 1 = Select P1 10G/1G clock/lock for output on M1. ■ 2 = Select P2 10G/1G clock/lock for output on M1. ■ 3 = Select P3 10G/1G clock/lock for output on M1. ■ 8 = Select P0 1G clock/lock for output on M1. DATA4: FW Control SyncE Configuration. ■ 0 = User Control SyncE configuration (default). ■ 1 = Firmware Control SyncE configuration. DATA5[0]: M0 mux SYNCE_OEN. [1]: M1 mux SYNCE_OEN. ■ 0 = Output enabled. ■ 1 = Output disable. DATA5[15:2]: Reserved.	**Return Results**: None.

1.25.1.19 GET_XFI_TX_FILTERS

Description:	Gets current XFI TX filter taps.
Input Parameters: None.	**Return Results**: DATA2: ■ Bit [0:7] = Pre tap. ■ Bit [8:15] = Main tap. DATA3: ■ Bit [0:7] = Post1 tap. ■ Bit [8:15] = Post2 tap. DATA4: ■ Bit [8:15] = TX HPF DATA5: Reserved **NOTE:** The current link speed is the filter for that speed. If a link is down, the filter tap is invalid.

1.25.1.20 SET_XFI_TX_FILTERS

Description:	Sets XFI TX filter taps.
Input Parameters: DATA1: ■ Bit[0:7] = Reserved. ■ Bit [8:15] = Speed. – 0 = Backward-compatible 10G – 1 = 100BT – 2 = 1G – 3 = 10G – 4 = 2.5G – 5 = 5G DATA2: Bit [0:7] = Pre tap. Bit [8:15] = Main tap. DATA3: Bit [0:7] = Post1 tap. Bit [8:15] = Post2 tap. DATA4: Bit [8:15] = TX HPF DATA5: Reserved **NOTE:** Speed = 0 means 10G; this is for backward-compatibility.	**Return Results**: None.

1.25.1.21 GET_XFI_POLARITY

Description:	Gets XFI polarity.
Input Parameters: None.	**Return Results**: ■ DATA2: TX polarity. ■ DATA3: RX polarity.

1.25.1.22 SET_XFI_POLARITY

Description:	Sets XFI polarity.
Input Parameters: None. DATA2: ■ TX polarity. DATA3: ■ RX polarity.	**Return Results**: None.

1.25.1.23 GET_CURRENT_VOLTAGE

Description:	Gets current voltage.
Input Parameters: None.	**Return Results**: DATA1 = For 0.8V measurement. DATA2 = For 0.8V measurement. DATA3 = For 1.88V measurement. DATA4 = Reserved.

1.25.1.24 CMD_GET_SNR

Description:	Gets SNR value.
Input Parameters: DATA1: 0 = Do not display SNR in IEEE registers 1 = Display SNR in IEEE registers	**Return Results**: DATA2/10 = SNR for Channel A in dB DATA3/10 = SNR for Channel B in dB. DATA4/10 = SNR for Channel C in dB DATA5/10 = SNR for Channel D in dB

1.25.1.25 CMD_GET_CURRENT_TEMP

Description:	Returns the internally derived temperature of the die. DATA1_REG returns the temperature, when supported.
Input Parameters: None.	**Return Results**: DATA1_REG = Temperature value.

1.25.1.26 CMD_SET_UPPER_TEMP_WARNING_LEVEL

Description:	Sets the upper-bound of the temperature warning to the value in DATA1_REG.
Input Parameters: DATA1: Upper temperature limit in °C (2s compliment). DATA2: Upper temperature limit margin in °C (positive number). DATA3: Enable upper temperature warning. ■ 0 = Disable. ■ 1 = Enable.	**Return Results**: None.

1.25.1.27 CMD_GET_UPPER_TEMP_WARNING_LEVEL

Description:	Gets the upper-bound of the temperature warning and saves the value to DATA1_REG.
Input Parameters: None.	**Return Results**: DATA1: Upper temperature limit in °C (2s compliment). DATA2: Upper temperature limit margin in °C (positive number). DATA3: Enable upper temperature warning. ■ 0 = Disable. ■ 1 = Enable.

1.25.1.28 CMD_SET_LOWER_TEMP_WARNING_LEVEL

Description:	Sets the lower-bound of the temperature warning and saves the value to DATA1_REG.
Input Parameters: DATA1: Lower temperature limit in °C (2s compliment). DATA2: Lower temperature limit margin in °C (positive number). DATA3: Enable upper temperature warning. ■ 0 = Disable. ■ 1 = Enable.	**Return Results**: None.

1.25.1.29 CMD_GET_LOWER_TEMP_WARNING_LEVEL

Description:	Gets the lower bound of the temperature warning to the value in DATA1_REG.
Input Parameters: None.	**Return Results**: DATA1: Lower temperature limit in °C (2s compliment). DATA2: Lower temperature limit margin in °C (positive number). DATA3: Enable upper temperature warning. ■ 0 = Disable. ■ 1 = Enable.

1.25.1.30 GET_WAP_MODE_FRAME_TYPE

Description:	—
Input Parameters: None.	**Return Results**: DATA1: 0 = Reserved DATA1: 1 = Priority protocol is to IEEE 802.3bz. DATA2: Current link frame type. ■ 0 = Reserved ■ 1 = IEEE 802.3bz ■ 2 = NBASE-T ■ 3 = Reserved DATA3 to DATA5: Reserved.

1.25.1.31 SET_WAP_MODE_FRAME_TYPE

Description:	—
Input Parameters: DATA1: Frame type priority. ■ 0: Reserved. ■ 1: IEEE 802.3bz. DATA2 to DATA5: Reserved.	**Return Results**: None.

1.25.1.32 GET_XFI_2P5G_5G_MODE

Description:	—
Input Parameters: None.	**Return Results**: DATA1: 0 = 10G Idle Stuffing mode for 2.5G. 1 = 2500BASE-X_5000BASE-X mode for 2.5G. 2 = 2500BASE-R_5000BASE-R mode for 2.5G. DATA2: 0 = 10G Idle Stuffing mode for 5G. 1 = 25000BASE-X_5000BASE-X mode for 5G. 2 = 2500BASE-R_5000BASE-R mode for 5G. DATA3 to DATA5: Not used.

1.25.1.33 SET_XFI_2P5G_5G_MODE

Description: —	
Input Parameters: DATA1: 0 = 10G Idle Stuffing mode for 2.5G. 1 = 2500BASE-X_5000BASE-X mode for 2.5G. 2 = 2500BASE-R_5000BASE-R mode for 2.5G. DATA2: 0 = 10G Idle Stuffing mode for 5G. 1 = 25000BASE-X_5000BASE-X mode for 5G. 2 = 2500BASE-R_5000BASE-R mode for 5G. DATA3 to DATA5: Not used.	**Return Results**: None.
NOTE: KR mode is not supported when 2500X or 5000X mode is enabled.	

1.25.1.34 SET_EEE_STATISTICS

Description: Enable EEE statistics.	
Input Parameters: None.	**Return Results**: None.
NOTE: This command must be issued after the link is up.	

1.25.1.35 GET_EEE_STATISTICS

Description: —	
Input Parameters: Counter selection: 0 = Select event counters. 1 = Select time duration.	**Return Results**: DATA2: TX (LOC) counters/time lower 16 bits. DATA3: TX (LOC) counters/time upper 16 bits. DATA4: RX (REM) counters/time lower 16 bits. DATA5: RX (REM) counters/time upper 16 bits.
NOTE: PHY selects 10G or 1G based on current link speed.	

1.25.1.36 GET_JUMBO_PACKET

Description: Gets jumbo packet status setting (enable/disable).	
Input Parameters: None.	**Return Results**: DATA1: Current jumbo packet enable status. ■ 0 = Jumbo packet is disabled. ■ 1 = Jumbo packet is enabled. DATA2: Jumbo packet size option ■ 0 = 10K packet ■ 1 = 18k packet ■ 2 = 9k packet DATA3 to DATA5: Not used.

1.25.1.37 SET_JUMBO_PACKET

Description:	Sets jumbo packet status setting (enable/disable).
Input Parameters: DATA1: Enable or disable jumbo packet. 0 = Disable jumbo packet. 1 = Enable jumbo packet. DATA2: Jumbo packet size option ■ 0 = 10K packet ■ 1 = 18k packet ■ 2 = 9k packet DATA3 to DATA5: Not used.	**Return Results**: None.

1.25.1.38 GET_PAUSE_FRAME_MODE

Description:	Gets the pause frame mode status (enable/disable).
Input Parameters: None.	**Return Results**: DATA1: Pause frame enable status. ■ 0 = No pause frame. ■ 1 = Pause frame is enabled. DATA2 to 5: Reserved.
NOTE: This is only valid for 2.5/5G Idle Stuffing mode in 10G XFI WAP mode.	

1.25.1.39 SET_PAUSE_FRAME_MODE

Description:	Sets the pause frame mode status (enable/disable).
Input Parameters: DATA1: Pause frame enable status. ■ 0 = No pause frame. ■ 1 = Pause frame is enabled.	**Return Results**: None.
NOTE: This is only valid for 2.5/5G Idle Stuffing mode in 10G XFI WAP mode.	

1.25.1.40 GET_LED_TYPE

Description:	Gets the LED control mode status (enable/disable).
Input Parameters: None.	**Return Results**: DATA1: Current LED type ■ 0 = Firmware control ■ 1 = User control DATA2 to DATA5: Reserved

1.25.1.41 SET_LED_TYPE

Description:	Sets the LED control mode status (enable/disable).
Input Parameters: DATA1: Current LED type ■ 0 = FW control ■ 1 = User control DATA2 to DATA5: Reserved	**Return Results**: None.

1.25.1.42 GET_USXGMII

Description:	Gets the USXGMII mode and AUTONEG function (enable/disable).
Input Parameters: None.	**Return Results**: DATA1: USXGMII enable status. ■ 0 = USXGMII is disabled. ■ 1 = USXGMII is enabled. DATA2: USXGMII autonegotiation enable status (only valid if USXGMII is enabled). ■ 0 = Autonegotiation is disabled. ■ 1 = Autonegotiation is enabled. DATA3: ■ [0:2] = single_max_speed – 1 = 2.5G – 2 = 5G – 4 = 10G DATA4: Reserved DATA5: ■ 1 = MAC/PHY frequency unlocked ■ 0 = MAC/PHY frequency is locked

1.25.1.43 SET_USXGMII

Description:	Sets the USXGMII mode and AUTONEG function (enable/disable).
Input Parameters: DATA1: USXGMII enable. ■ 0 = Disable USXGMII ■ 1 = Enable USXGMII. DATA2: USXGMII autonegotiation enable status (only valid if USXGMII is enabled). ■ 0 = Disable autonegotiation. ■ 1 = Enable autonegotiation. DATA3: ■ [0:2] = single_max_speed – 1 = 2.5G – 2 = 5G – 4 = 10G DATA4: Reserved DATA5: ■ 1 = MAC/PHY frequency unlocked ■ 0 = MAC/PHY frequency is locked	**Return Results**: None.

1.25.1.44 SET_WOL_ENABLE

Description:	Enable/disable wake-on-LAN (WOL) mode.
Input Parameters:DATA1: WOL mode enable ■ 0 = Disable WOL mode ■ 1 = Enable WOL mode DATA2 to DATA5: Reserved	**Return Results**: None.
NOTE: This command takes effect immediately without need of link re-establishment. After WOL mode is set, do a soft reset to the PHY prior to its normal operation. WOL mode is only necessary for 2.5G speed and above.	

1.25.1.45 GET_WOL_ENABLE

Description:	Get wake-on-LAN (WOL) mode settings.
Input Parameters: None.	**Return Results**: DATA1: WOL mode enable ■ 0 = Disable WOL mode ■ 1 = Enable WOL mode DATA2 to DATA5: Reserved

1.26 General-Purpose LED Programmability

The BCM84891L has five LED pins. These LED pins can be functionally programmed, and are referred to in this document as LED1 through LED5. control register words are provided to enable the selection of LED functions for each of the pins. These control words are located in register LED Control (DEVAD = 1, Address = 0xA83B). For each of the five programmable LEDs, their corresponding register bits, and default settings.

The LED sources are masked in the Control LEDx Mask register. LED mask bit should be set to 1 to allow the source to driver the LED pin. When multiple bits in the Control LEDx Mask register are set, the LED sources are OR'd together.

The LEDs are controlled by the LED Control (DEVAD = 1, Address = 0xA83B). The LED Control Source (DEVAD = 1, Address = 0xA83C) registers drives the LED signals on the pins when the mask associated with the LED is set to enable the bit.

In 10G mode, each LED is controlled separately by the individual bits of the PHYC_CTL_LED_CTL. The LEDx_OE_N enables the LEDx output. LEDx_CTL controls the mode of operation of LEDx.

- When set to 00, the LED is off.
- When set to 01, the LED blinks based on the selected source. The blink cycle is programmed by 10GBASE-T Control LEDx bit [3:0].
- When set to 10, the LED is ON based on the selected source registers.
- 11 is reserved.

Each of the five LEDs has a maskable source registers bit (see Table 9).

Activity LED can be enabled by unmasking bit 1 (for receiver activity) or/and bit 2 (for transmitter activity) in the LED associated Control LEDx Mask registers. The LED Control Source (DEVAD = 1, Address = 0xA83C) bit 1 provides receiver activity status, and bit 2 provides transmitter activity status. When both bits are set, the activity LEDs indicated activity of transmitter and receiver. To make the activity pulse visible to eye, the Stretch_En bit associate for the selected LED pin to use as activity LED signal should be set to 1.

NOTE: