Application Note

DP83869 Troubleshooting Guide

Alvaro Reyes

ABSTRACT

DP83869HM is a robust, fully-featured gigabit physical layer (PHY) transceiver. *Understanding different modes of operation in DP83869* describes each mode of the DP83869HM in greater detail. This application note was created to help troubleshoot the DP83869HM in a design and to show what to look at in the event that the PHY is not working as intended.

Table of Contents

Trademarks

MSP430F5529™ is a trademark of Texas Instruments.
All trademarks are the property of their respective owners.

1 DP83869 Application Overview

The DP83869HM device is a robust, fully-featured gigabit physical layer (PHY) transceiver with integrated PMD sublayers that supports 10BASE-Te, 100BASE-TX, and 1000BASE-T Ethernet protocols. The DP83869 also supports 1000BASE-X and 100BASE-FX fiber protocols.

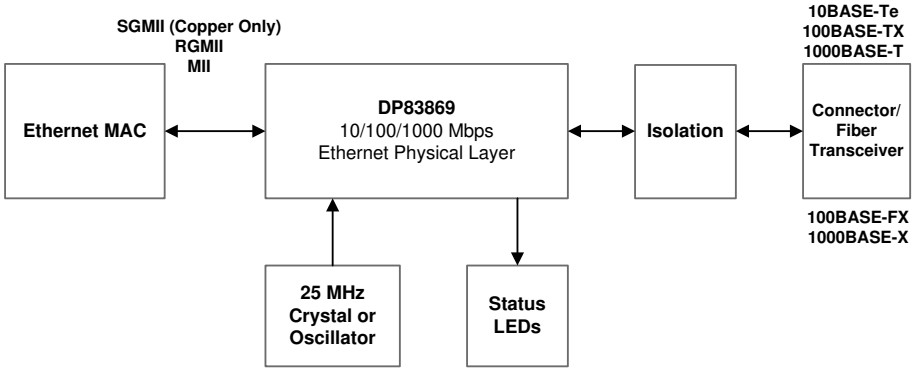

Figure 1-1. Standard Ethernet System Block Diagram

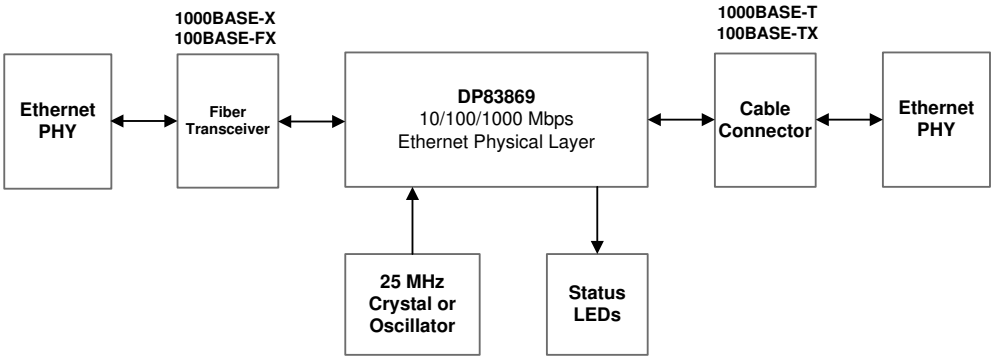

Figure 1-2. Media Convertor System Block Diagram

2 Troubleshooting the Application

The following sections approach the debug from a high level, basic check approach which isolates subsystems of the PHY design to check if the subsystems contribute to an application issue. This document is intended to address common Ethernet issues such as:

- Inability to ping
- Cannot get link OR intermittent linkup
- Linkup but seeing packet errors
- Cannot access registers

The recommendation is to go through the following sections in order unless otherwise specified.

2.1 Schematic and Layout Checklist

The DP83869 Schematic Checklist and DP838XX Industrial Layout Checklist compile the best practices for designing with DP83869 into an easy-to-use document. The recommendation is to go through these documents for a detailed description what connections and components are needed for the PHY to work.

The following sections can present expected behaviors if the PHY is powered and initialized correctly. Any deviations from expected behaviors can point to errors due to incorrect peripheral circuity.

2.2 Device Health Checks

This section dives into device health checks which makes sure the device is powered and initialized correctly. This section can be skipped if DP83869 is:

- Linking up (LED indication or register status) when connected to link partner or showing FLP signals when Ethernet cable is disconnected
- Responding to register access (if applicable)

2.2.1 Voltage Checks

DP83869 needs to have sufficient power as well as:

- One 10nF, and one 10uF decoupling per rail
- One 100nF, and one 1uF decoupling per pin

The DP83869 supports the two configurations for power supplies shown in the Figures below.

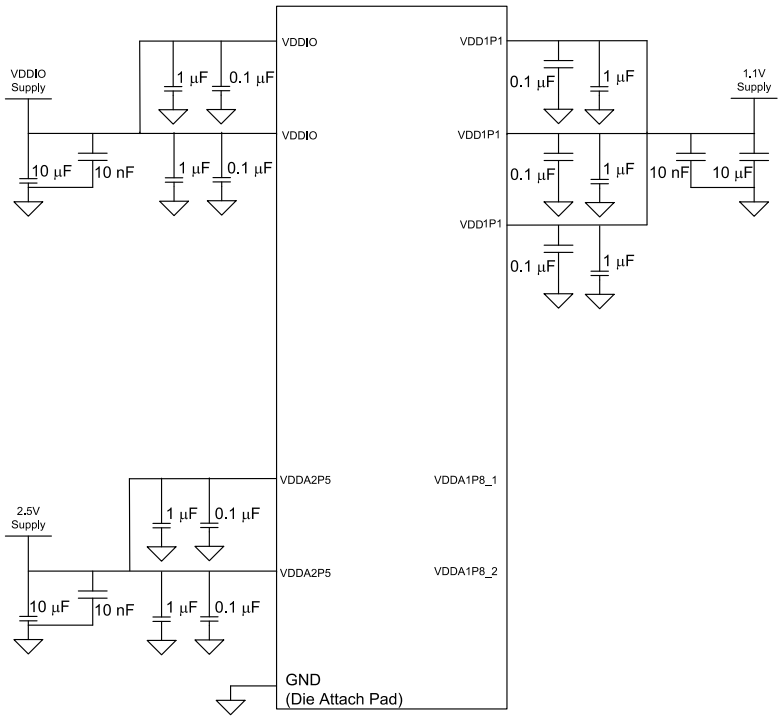

Figure 2-1. Two-Supply Configuration

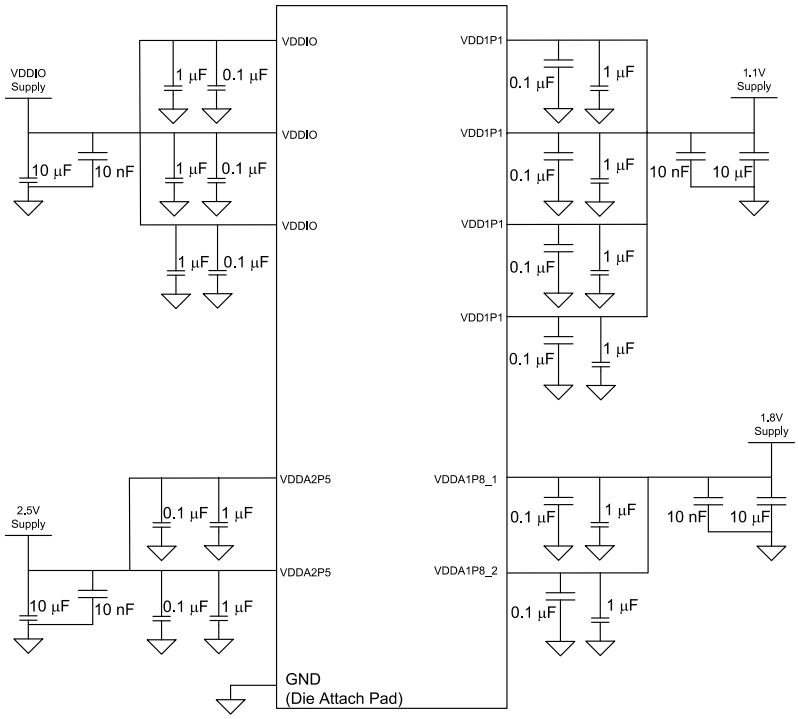

Figure 2-2. Three-Supply Configuration

When operating in the three supply configuration, TI recommends powering all supplies together. If powering all supplies simultaneously is not possible, then power VDD1P1 and VDD2P5 first with VDDIO and VDD1P8 following within 50 ms.

When operating in the two supply configuration, leave both VDDA1P8 pins disconnected and power all supplies together. If powering all supplies simultaneously is not possible, then power VDDA2P5 and VDD1P1 first with VDDIO following within 50 ms.

Power up the device and perform DC measurement of the supplies as close to the pin as possible. Confirm that each measurement is within the limits defined in Table 2-1

Table 2-1. Recommended Operating Conditions

	Min(V)	Typ(V)	Max(V)
VDDIO (1.8V)	1.71	1.8	1.89
VDDIO (2.5V)	2.375	2.5	2.625
VDDIO (3.3V)	3.15	3.3	3.45
VDD1P1	0.99	1.1	1.21
VDDA1P8	1.71	1.8	1.89
VDDA2P5	2.375	2.5	2.625

2.2.2 Probe the RESET_N Signal

The reset functionality on DP83869 is active low. This pin has a weak internal pull-up resistor to provide a default state if left unconnected or not driven externally.

Confirm that the controller is not driving the RESET_N signal low. Otherwise, the device can be held in reset state, and does not respond to register commands nor would link up.

2.2.3 Probe RBIAS

The RBIAS pin is used to set the internal reference current within the DP83869. RBIAS should be a 11kΩ resistor with a 1% tolerance. The preference is to have a single component over multiple in series as the tolerance range can increase.

If properly powered, a 1V signal will appear when probing the RBIAS pin.

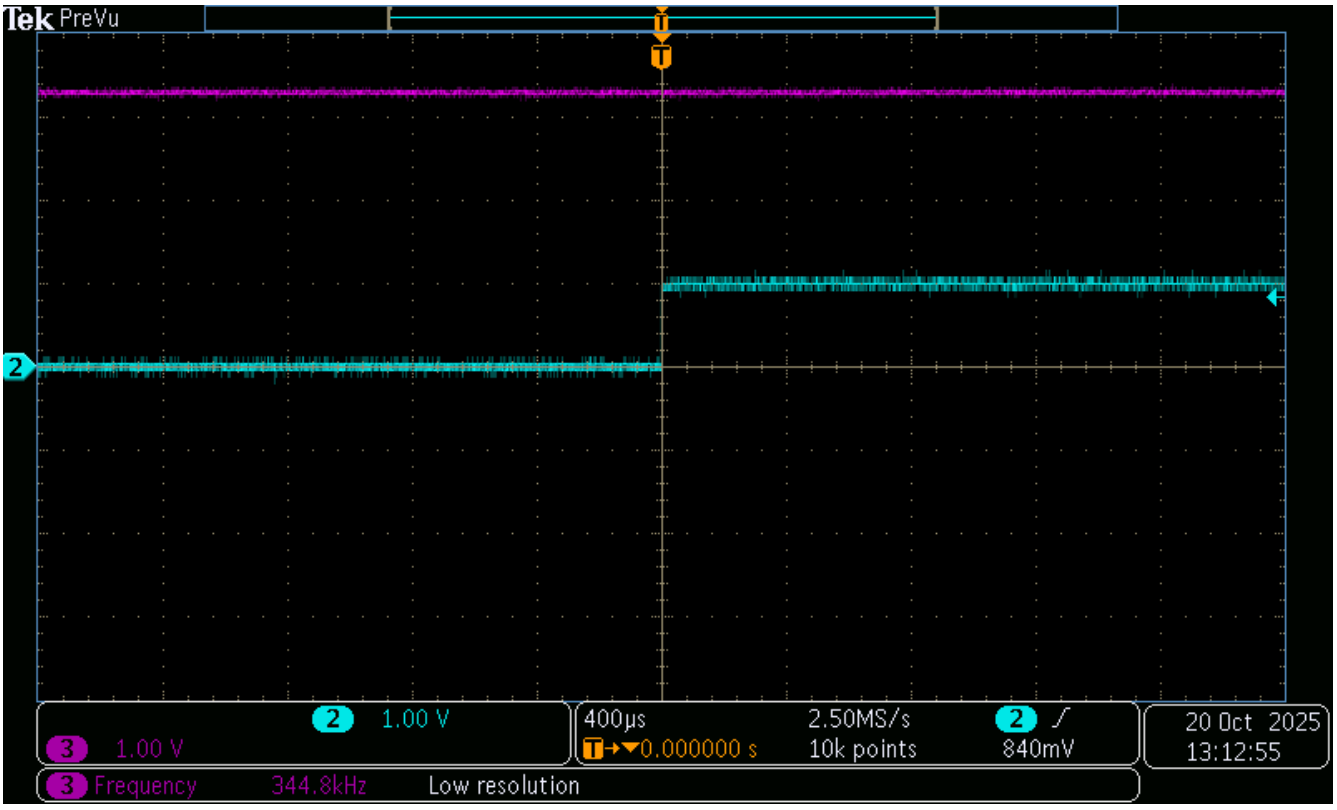

Figure 2-3. RBIAS Voltage (Blue) and VDDIO (Purple)

2.2.4 Probe the XI Clock

The following guidelines are the main specifications to reference for compatible input clocks:

Table 2-2. 25MHz Crystal Specifications

Parameter	Min	Typ	Max	Unit
Frequency		25		MHz
Frequency Tolerance	-100		100	ppm
Load Capacitance	15		40	pF
ESR			50	Ω

Probing on the crystal nodes can change the capacitive loading and therefore change the operational frequency. If using a crystal as the clock source, probe the CLK_OUT signal. The default signal on CLK_OUT is a buffered version of the XI reference and provides a representative measurement.

Table 2-3. 25MHz Oscillator Specifications

Parameter	Min	Typ	Max	Unit
Frequency		25		MHz
Frequency Tolerance	-100		100	ppm
Rise or Fall Time			5	ns
Duty Cycle	40		60	%
Jitter RMS			11	ps

Note

For more information on designing with a crystal network, please refer to the Selection and specification of crystals for Texas Instruments Ethernet physical layer transceivers, application note.

2.2.5 Probe the Strap Pins During Initialization

The DP83869 has strap pins for configuring the device in a predetermined mode. The voltage at these strap pins determines which mode the DP83869 can operate in.

On initialization, the external strap network along with the internal resistor creates a voltage divider that the PHY samples. No other component on the line should affect the DC bias set by this network.

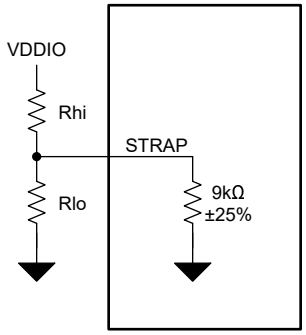

Figure 2-4. DP83869 Strap Circuit

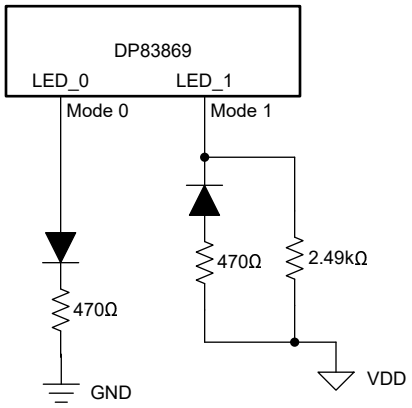

Figure 2-5. DP83869 LED Strap Circuit

In some cases, other devices on the board (for example, the MAC) will drive the strap pins unexpectedly. The strap values can be read from register 0x006E (STRAP_STS). If there is power cycle dependency to an issue, the strapping may be marginal and can be observed cycle to cycle against this register to determine if the PHY is strapped in an unintended state.

Measurements can be made during power up and after power up when the RESET_N signal is asserted.

Note

Register 0x6E is an extended register and cannot be accessed directly. Please reference Section 4.1.

2.2.5.1 Configuring Correct Operational Mode

The operational mode of the DP83869 is configured through the OPMODE[0], OPMODE[1], and OPMODE[2] straps. A brief summary of each OPMODE configuration is provided in Table 2-4. More information can be found in the *Programming* section of the data sheet.

To verify DP83869's operational mode, register 0x6E can be read to confirm. Register 0x6E is read-only, meaning the Operational Mode cannot be changed by writing to this register. Software configuration of the DP83869 is possible through register 0x1DF which allows writes to configure the OPMODE. Some operational modes require more register writes than just register 0x1DF, this information is provided in the *Register Configuration for Operational Modes* Section in the data sheet.

Note

Registers 0x6E and 0x1DF are extended registers and cannot be accessed directly. Please reference Section 4.1.

Table 2-4. Functional Mode Strap Table

PIN NAME	STRAP NAME	PIN #	DEFAULT	OPMODE[2]	OPMODE[1]	OPMODE[0]	FUNCTIONAL MODES
JTAG_TDO/ GPIO_1	OPMODE[0]	22	0	0	0	0	RGMII to Copper (1000Base-T/ 100Base-TX/10Base-Te)
				0	0	1	RGMII to 1000Base-X
RX_D3	OPMODE[1]	36	0	0	1	0	RGMII to 100Base-FX
				0	1	1	RGMII-SGMII Bridge Mode
RX_D2	OPMODE[2]	35	0	1	0	0	1000Base-T to 1000Base-X
				1	0	1	100Base-Tx to 100Base-FX
				1	1	0	SGMII to Copper (1000Base-T/ 100Base-TX/10Base-Te)
				1	1	1	JTAG for boundary scan

2.2.6 Probe the Serial Management Interface (MDC, MDIO)

The Serial Management Interface (SMI) can be useful in providing status fields during a debug. Make sure the MDIO line has a pull up resistor to VDDIO as this pin is an open-drain to the PHY. When idle, the voltage needs to be VDDIO. Make sure the SMI access uses the following sequence:

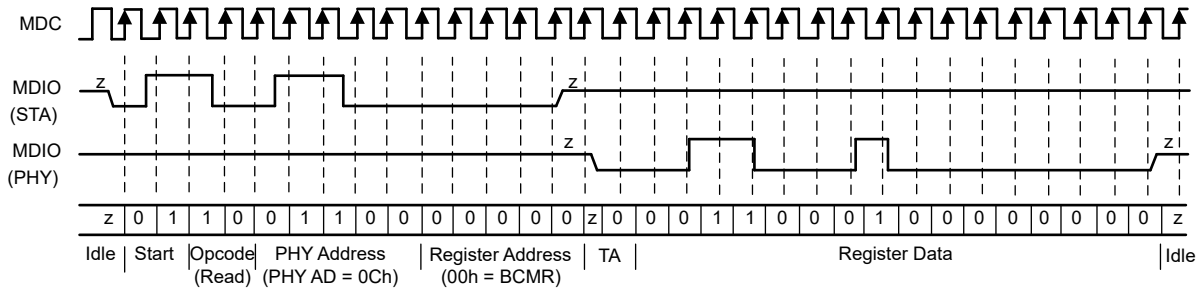

Figure 2-6. SMI Read Operation

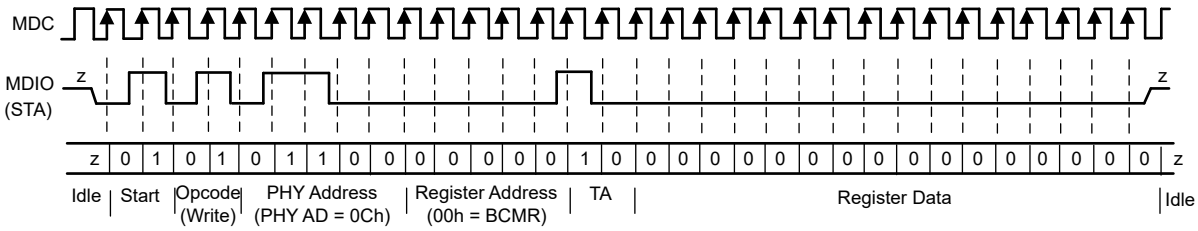

Figure 2-7. SMI Write Operation

2.2.6.1 Read and Check Register Values

Read the registers and verify the default values shown in the data sheet. Note that the initial values of some registers can vary based on strap options. An example of expected register values for PHY operation and link in 100/1000Mbps with auto-negotiation enabled are shown in Table 2-5.

Table 2-5. Example Register Values on Link up

Register Address	Register Value		Comments
	100Mbps	1000Mbps	
0x0000	1140	1140	Auto-negotiation enabled
0x0001	796D	796D	Link established, Auto-negotiation status
0x0004	01E1	01E1	10/100Mbps advertisement
0x0009	0000	0300	1000Mbps advertisement
0x0011	4F02	BC02	Link properties. With the PHY linked in a given speed, use these values as a reference to identify any variance from the expected operation. Note that not all registers need to be the same as what is shown in this table.

Example: After powering and linking the PHY in 1000Mbps, Reg 0x11 contains the value 0xBC02. This confirms:

- 1000Mbps Mode
- Full-Duplex
- Auto-Negotiation is complete
- Link established

If register access is not readily available in the application, USB-2-MDIO GUI is available from TI and can be used with an MSP430F5529™ Launchpad, purchasable through the TI eStore. The GUI supports reading and writing registers, running script files, and can be used with the DP83869HM and the other devices in TI's Ethernet portfolio. USB-2-MDIO User's Guide and GUI are available for download.

2.3 MDI Health Checks

This section dives into device health checks which makes sure that the device's MDI section is operating properly. This section can be skipped if DP83869 is linked up and reporting no errors on Reg 0x15 when sending traffic through the device.

2.3.1 Magnetics

The following guidelines are the main specifications to reference for compatible magnetics:

Table 2-6. Magnetic Isolation Specifications

PARAMETER	TEST CONDITIONS	TYP	UNIT
Turns Ratio	±2% Tolerance	1:1	-
Insertion Loss	1-100MHz	-1	dB
Return Loss	1-30MHz	-16	dB
	30-60MHz	-12	dB
	60-80MHz	-10	dB
Differential to Common Mode Rejection	1-50MHz	-30	dB
	60-150MHz	-20	dB
Crosstalk	30MHz	-35	dB
	60MHz	-30	dB
Open Circuit Inductance	8mA DC Bias	350	µH
Isolation	HPOT	1500	Vrms

If these exact requirements cannot be met, the following allowances can be made:

- Turns ratio: 3% is tolerable
- Insertion loss: -1dB or closer to 0dB
- Return loss: Meets or exceeds values in table above.

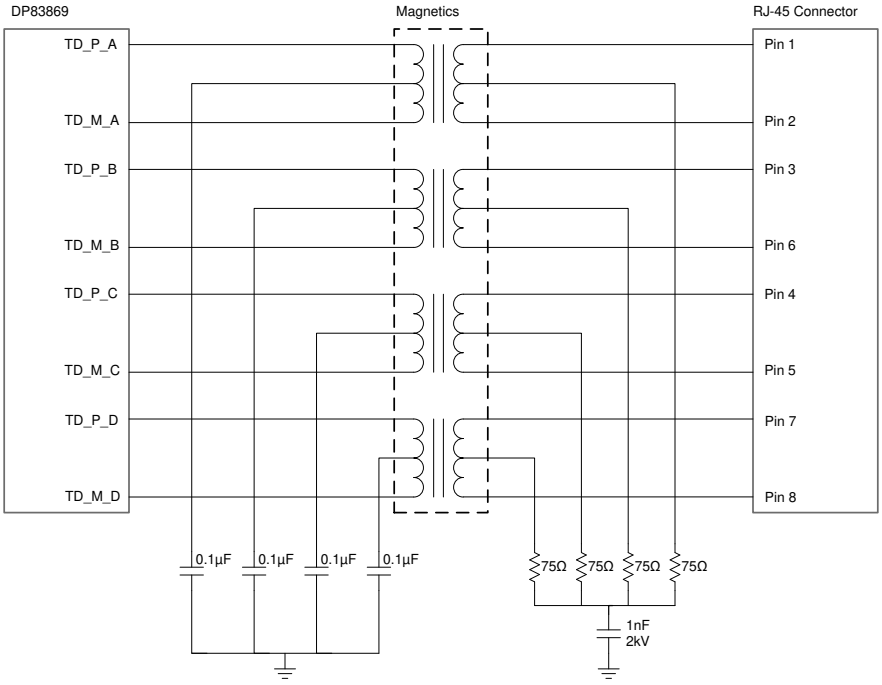

Figure 2-8. PHY to RJ45 and Magnetics

- Each center tap on the side connected to the PHY must be isolated from one another and connected to ground by a decoupling capacitor (0.1µF recommended).

2.3.2 Probe the MDI Signals

In the default configuration, Auto-negotiation and Auto-MDIX can be enabled. A link pulse needs to be visible on the channel transmit (TD_P, TD_M) and will occasionally toggle to the receive pair (RD_P, RD_M). If set to MDI, this pulse is only available on the transmit pair while if set in MDI-X, this will only be available on the receive pair. A short Ethernet cable terminated with 100 Ohm differential should be used for measuring the MDI signals.

A terminated cable is shown in Figure 2-7. A connection diagram for making measurements with the terminated cable is shown in Figure 2-8.

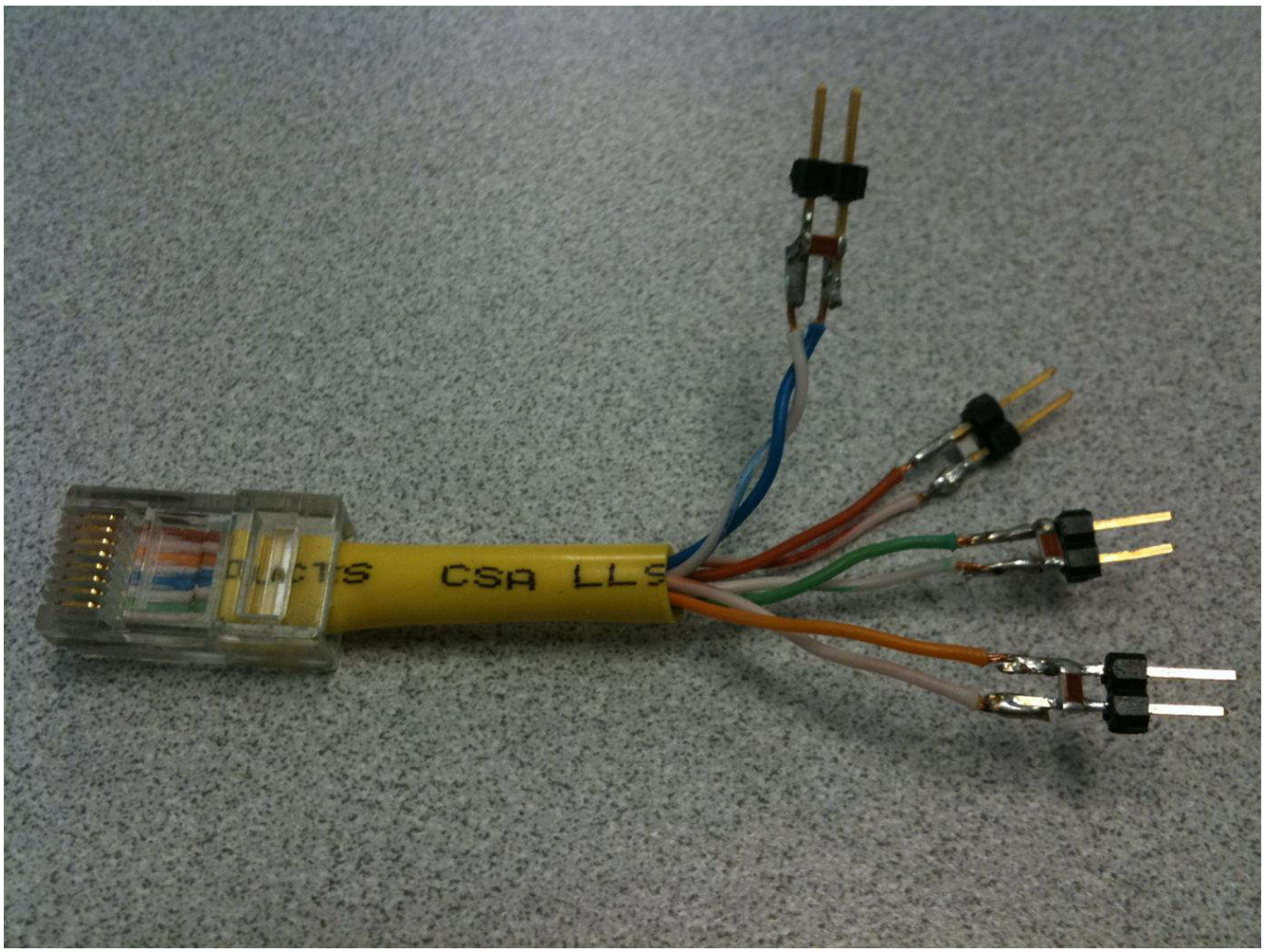

Figure 2-9. 100Ω Terminated Cable

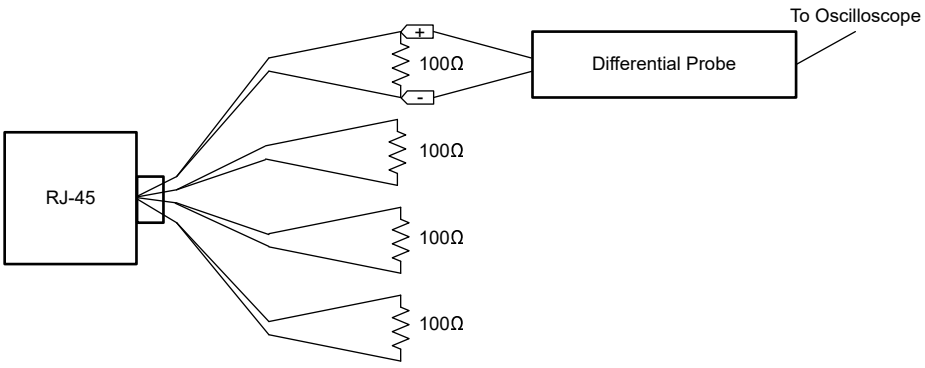

Figure 2-10. 100Ω Terminated Cable Connection Diagram

Auto-negotiation link pulses are nominally 100ns wide. Pulses are spaced by 62µs or 125µs and are transmitted in bursts. The bursts are nominally 2ms in duration and occur every 16ms. Figure 2-9 shows a link pulse.

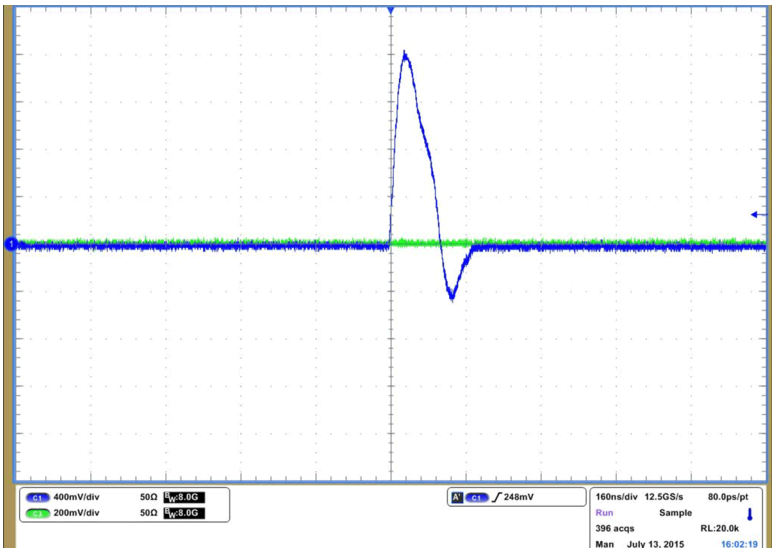

Figure 2-11. Link Pulse Example

2.3.3 Check the Link Quality

After establishing a valid link, confirming the key status register values and visually verifying that the link LED is lit, the next data transfer debug step is to check the MDI Interface. There are several possible sources of link problems:

1. Link partner transmit problem
2. Cable length and quality
3. 25MHz reference clock quality
4. MDI signal quality

With the PHY powered and connected to a link partner, the registers in Table 2-6 can be used to determine the Mean Square Error (MSE). For 100Mbps communication please refer to channel A only. With the MSE value, refer to Table 2-8 to determine link quality:

Table 2-7. Link Quality MSE Registers for 1000Mbps

Channel	Register Address
A	0x0225
B	0x0265
C	0x02A5
D	0x02E5

Table 2-8. MSE - Link Quality Conversion

Link Quality	MSE Range
Excellent	0x020A > MSE
Good	0x033B > MSE > 0x020A
Poor	MSE > 0x033B

2.3.3.1 Improving Short Cable Link Margin

If DP83869 encounters link quality issues with short cables 1m or less in length, consider the section below.

The PHY's digital signal processing (DSP) block may converge to suboptimal filter values at shorter lengths which can result in poor Signal to Noise Ratio (SNR). The register configuration below can improve the SNR by adjusting timing bandwidths to help the DSP converge correctly:

```
begin
// Hard Reset
001F 8000
// Threshold for consecutive amount of Idle symbols for Viterbi Idle detector to assert Idle Mode
set to 5
0053 2054
// CAGC DC Compensation Disable
00EF 3840
// Leader Training Timers - increasing time in different training states
0102 7477
0103 7777
0104 4577
// Timing Loop Bandwidth
010C 7777
01C2 7FDE
// Follower Timers - increasing time in different training states
0115 5555
0118 0771
// Timing Loop Bandwidth
011D 6DB2
011E 3FFB
01C3 FFC6
01C4 0FC2
01C5 0FF0
// FFE Fix
012C 0E81
// Soft Reset
001F 4000
end
```

2.3.3.2 Improving Inter-channel Link Margin

The DP83869 uses an AGC gain convergence circuit (automatic gain control of MDI receiver) to provide faster linkup. There is a tradeoff between the linkup time and gain mismatch between pairs. In applications where packet errors are observed, gain matching can be improved by increasing the gain convergence time with the following register writes:

```
begin
// Hard reset
001F 8000
// Increase time for AGC
0102 7477
// No AGC Re-train
00E4 0080
// Soft reset
001F 4000
end
```

2.3.4 Compliance

IEEE compliance measurements can be made to verify the signaling characteristics. For details on these measurements and how to properly configure the PHY, please refer to How to Configure DP8386x for Ethernet Compliance Testing, application note.

2.4 MII Health Checks

This section dives into device health checks which makes sure that the device's MAC interface is operating properly.

2.4.1 MII Check

The Media Independent Interface (MII) is a synchronous 4-bit wide nibble data interface that connects the PHY to the MAC. The MII is fully compliant with IEEE 802.3-2002 clause 22.

MII mode cannot be enabled via straps alone and register configuration will be necessary. Enable MII mode using the following process:

1. Write register 0x01DF = 0x0060 for copper MDI, or 0x0062 for fiber MDI
2. Write register 0x0018 = 0x000E

When using auto-negotiation to resolve MDI speed in MII mode, TI recommends disabling gigabit speed advertisement through register 0x0009.

The MII signals are summarized below:

Table 2-9. MII Signals

Function	Pins
Data Signals	TX_D[3:0]
	RX_D[3:0]
Transmit and Receive Signals	TX_EN
	RX_DV
Error Signal	RX_ER

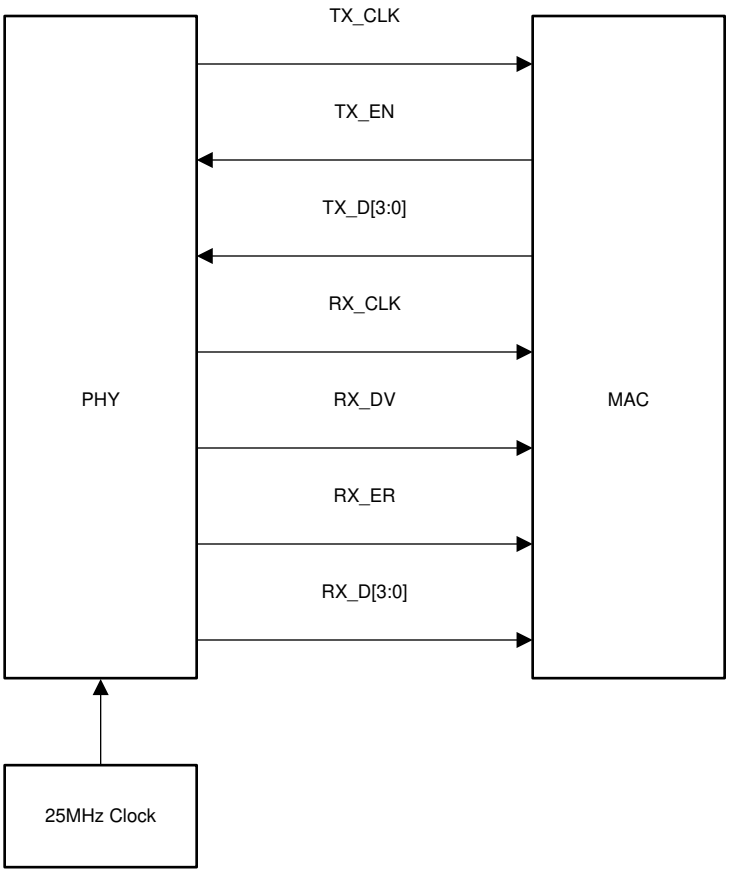

Figure 2-12. MII Signaling

Data on TX_D[3:0] is latched at the PHY with reference to TX_CLK. Data on RX_D[3:0] is provided with reference to RX_CLK. If a MAC TX or RX bus is suspected to be problematic, probe the lines at the receiver side of the trace to make sure that the receiver's setup and hold times are met.

Table 2-10. 100M MII Timings

Spec	Min	Typ	Max	Units
TX_CLK High/Low Time	16	20	24	ns
TX_D[3:0], TX_EN Setup to TX_CLK	10			ns
TX_D[3:0], TX_EN Hold from TX_CLK	0			ns
RX_CLK High/Low Time	16	20	24	ns
RX_D[3:0], RX_ER, RX_DV delay from RX_CLK rising	10		30	ns

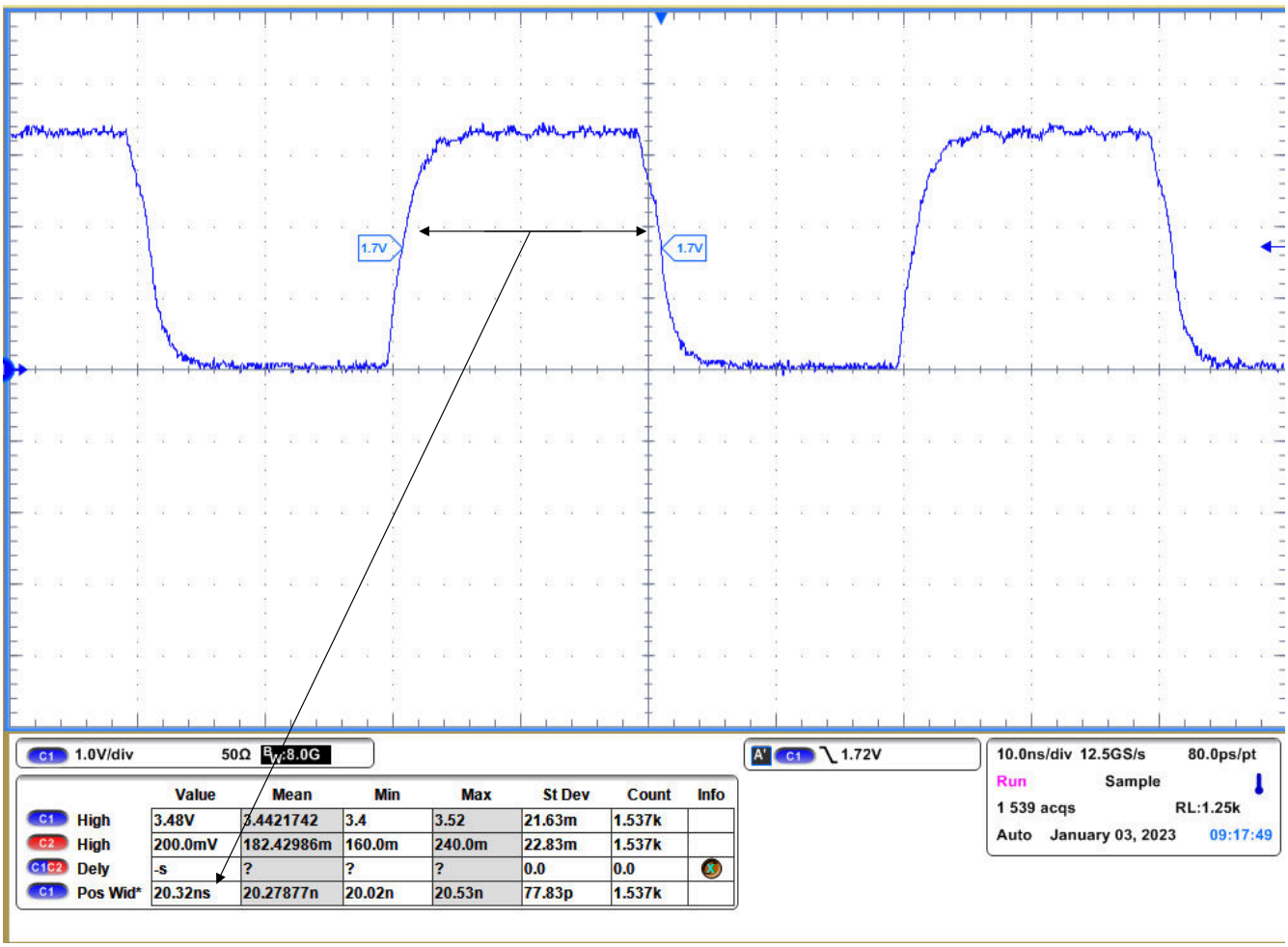

Figure 2-13. 100M RX_CLK High Time

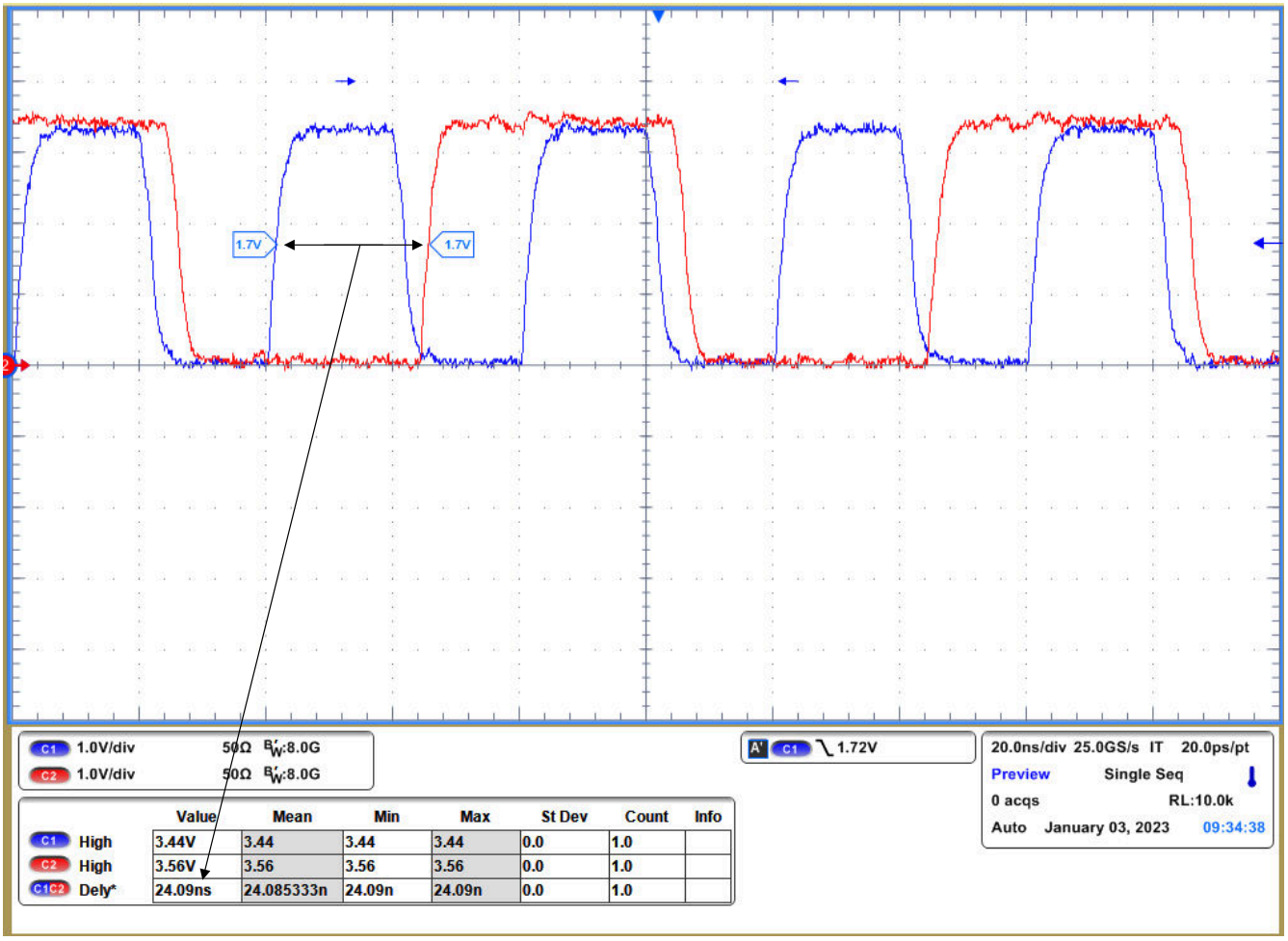

Figure 2-14. 100M RX_D1 Delay From RX_CLK

2.4.2 RGMII Check

Reduced Gigabit Media Independent Interface (RGMII) is a 4-bit wide data interface that supports up to 1000Mbps communication between a PHY and MAC. RGMII mode can be enabled via strapping the OPMODE[2:0] pins low, or by writing '000' to register 0x01DF[2:0].

The RGMII signals are summarized below:

Table 2-11. RGMII Signals

Function	Pins
Data Signals	TX_D[3:0]
	RX_D[3:0]
Transmit and Receive Signals	TX_CTRL
	RX_CTRL

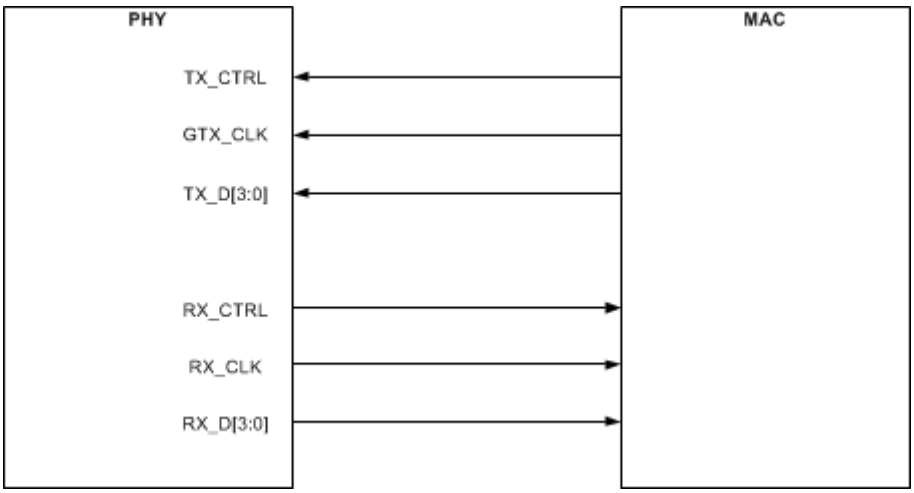

Figure 2-15. RGMII Signaling

Reference the waveforms in this section to verify the expected MAC data and clock signals for RGMII in shift and align modes. To capture data and clock signals, measure close to the receiver end. Note the following requirements for selecting the correct delay mode:

Table 2-12. Selecting the Correct RGMII Delay Mode

If MAC's Configuration is:	Required PHY Configuration
RGMII Align Mode on TX side	RGMII Shift Mode on TX side
RGMII Align Mode on RX side	RGMII Shift Mode on RX side
RGMII Shift Mode on TX side	RGMII Align Mode on TX side
RGMII Shift Mode on RX side	RGMII Align Mode on RX side

RX_D[3:0] Data Aligned with RX_CLK

For the PHY set in RX align mode in 10/100Mbps, probe the clock and data signals on the MAC end and compare to the reference waveforms shown below.

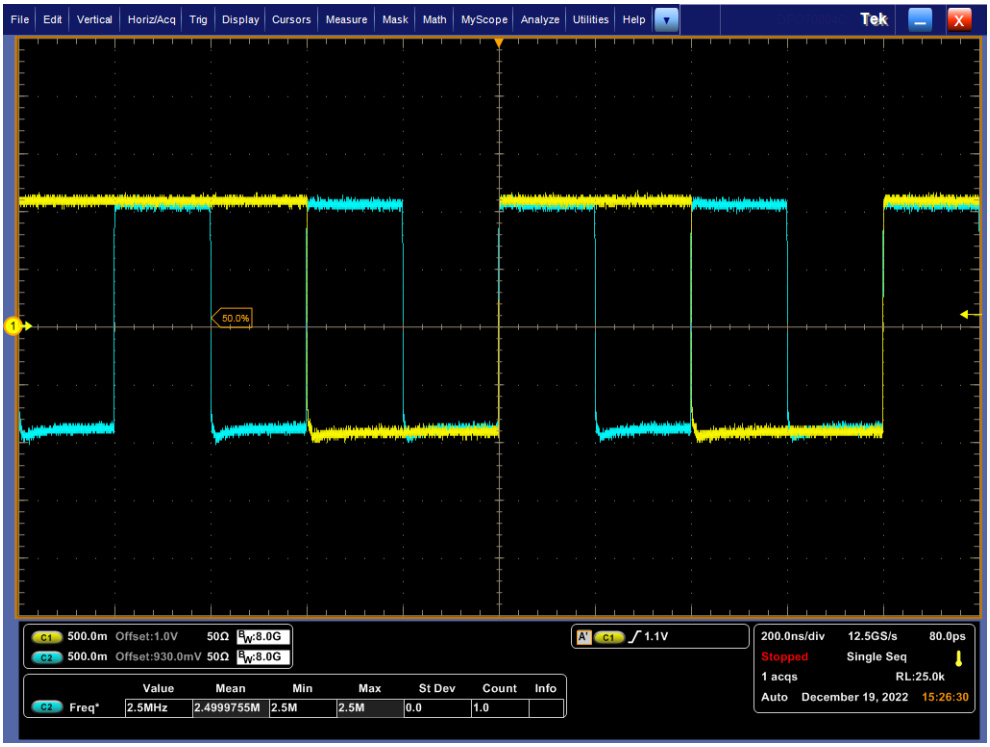

Figure 2-16. 10Mbps Data Aligned With RX_CLK

Verify the frequency of the clock (C2) as 2.5 MHz and the data (C1) being sampled at the rising edge of the clock.

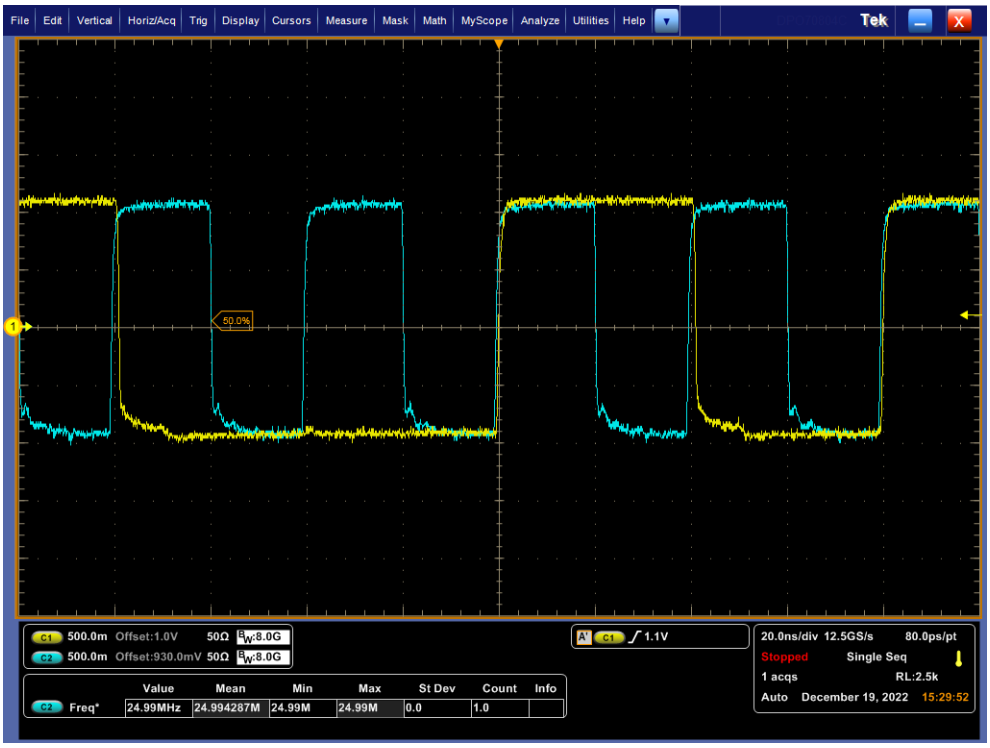

Figure 2-17. 100Mbps Data Aligned With RX_CLK

Verify the frequency of the clock (C2) as 25 MHz and the data (C1) being sampled at the rising edge of the clock.

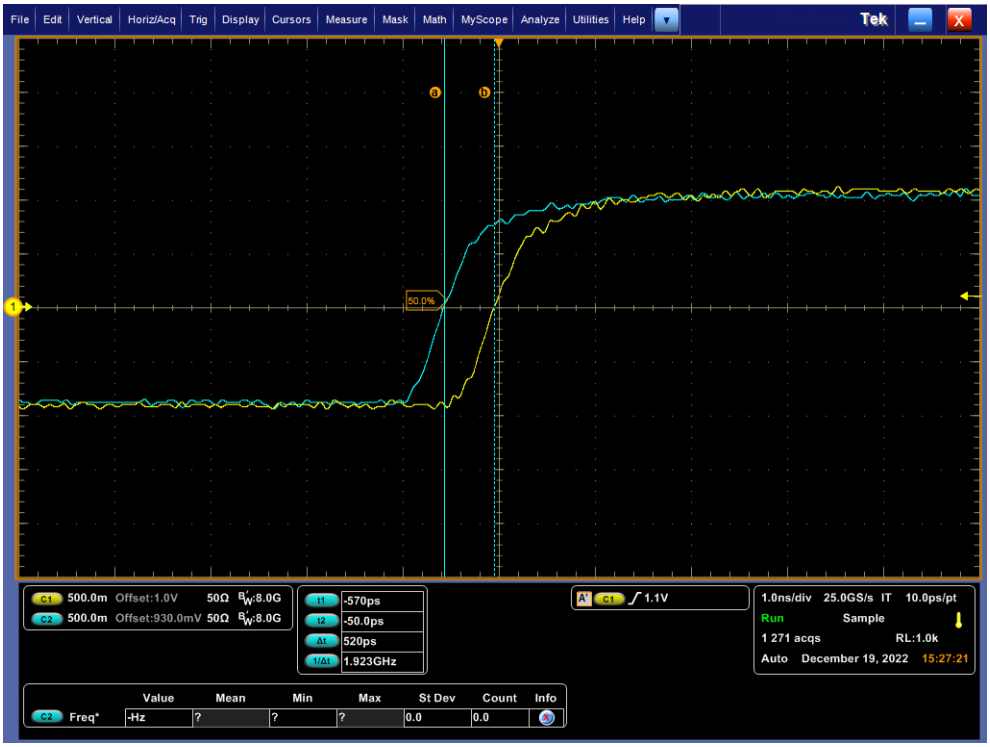

Figure 2-18. 10Mbps Data and Clock Delay in Align Mode

Verify the delay between clock and data is <500ps in align mode.

RX_D[3:0] Data and RX_CLK in Shift Mode

For the PHY set in RX shift mode (0x32) in 10/100Mbps, probe the clock and data signals on the MAC end and compare to the following reference waveforms.

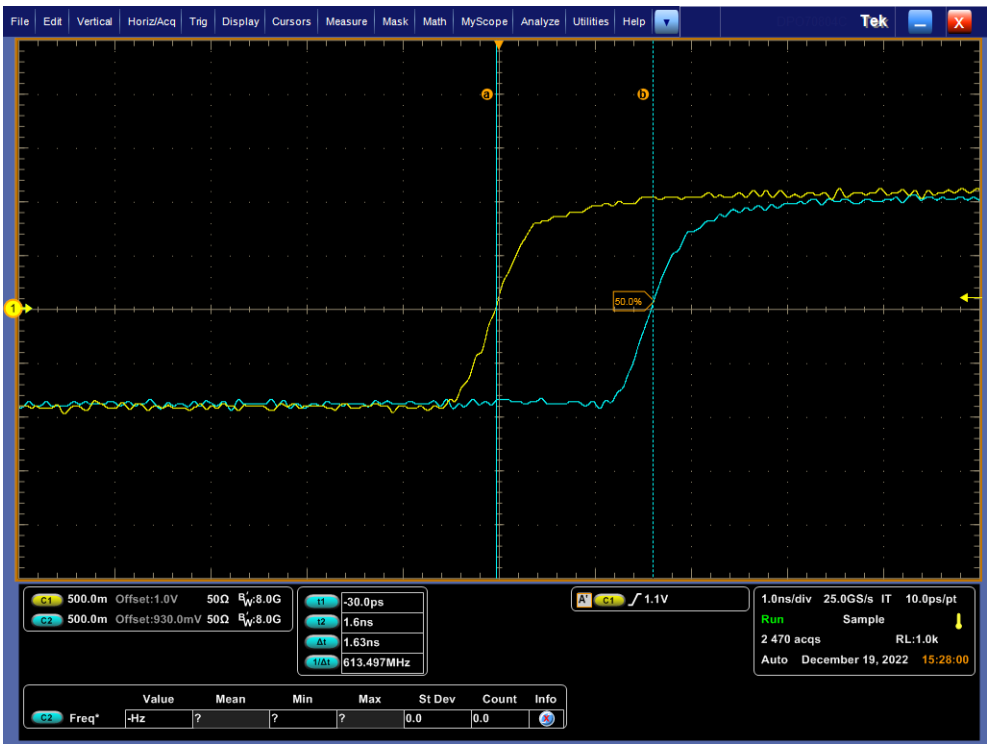

Figure 2-19. 10Mbps Data and RX_CLK in Shift Mode (4ns Programmed Delay)

Verify the delay between clock and data is >1ns in shift mode. The programmed delay is relative to the clock's initial position in aligned mode. Measuring the difference in the clock's position before and after setting shift mode yields a value closer to the programmed delay.

TX_D[3:0] and TX_CLK in Shift and Align Mode

With the PHY set in TX shift or align mode, probe the data and clock signals on the PHY end and verify the timing requirements below are met:

Table 2-13. RGMII Timings

PARAMETER		MIN	NOM	MAX	UNIT
T_{skewT}	Data to Clock output Skew (at Transmitter)	–500	0	500	ps
T_{skewR}	Data to Clock input Skew (at Receiver)	1	1.8	2.6	ns
T_{setupT}	Data to Clock output Setup (at Transmitter – internal delay)	1.2	2		ns
T_{holdT}	Clock to Data output Hold (at Transmitter – internal delay)	1.2	2		ns
T_{setupR}	Data to Clock input Setup (at Receiver – internal delay)	1	2		ns
T_{holdR}	Clock to Data input Hold (at Receiver – internal delay)	1	2		ns
T_{cyc}	Clock Cycle Duration	7.2	8	8.8	ns
Duty_G	Duty Cycle for Gigabit	45	50	55%	
Duty_T	Duty Cycle for 10/100T	40	50	60%	
T_R	Rise Time (20% to 80%)			0.75	ns
T_F	Fall Time (20% to 80%)			0.75	ns

2.4.3 SGMII Check

The Serial Gigabit Media Independent Interface (SGMII) provides a means of conveying network data and port speed between a 100M/1000M PHY and a MAC with significantly less signal pins (four or six pins) than required for GMII (24 pins) or RGMII (12 pins). The SGMII interface uses 1.25Gbps LVDS differential signaling which has the added benefit of reducing EMI emissions relative to GMII or RGMII.

The SGMII interface includes Auto-Negotiation capability. Auto-Negotiation provides a mechanism for control information to be exchanged between the PHY and the MAC. This allows the interface to be automatically configured based on the media speed mode resolution on the MDI side. SGMII Auto-Negotiation is the default mode of the operation but can be disabled by writing register 0x14[7] = 0.

The SGMII Output spec included in the data sheet states an Output Differential Voltage which refers to the Peak-to-Peak, SO_P - SO_N. This means that the SO_P and SO_N signals are roughly ±0.55V each.

Table 2-14. SGMII Output Spec

SGMII Output		Min	Typ	Max	Unit
Output Differential Voltage	SO_P and SO_N, AC Coupled	0.95	1.00	1.05	Vpp

All SGMII connections must be AC-coupled through a 0.1µF capacitor.

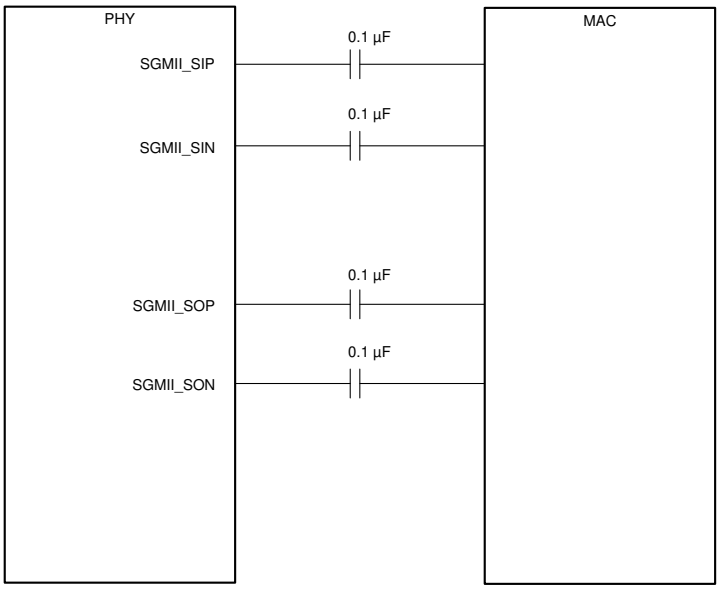

Figure 2-20. SGMII 4-Wire Connections

2.5 Loopback and PRBS

2.5.1 Loopback Modes

There are several options for loopback that test and verify various functional blocks within the PHY. Enabling loopback mode allows in-circuit testing of the MII and MDI data paths. DP83869 can be configured to one of the near-end (MII) loopback modes or to the reverse (MDI) loopback mode.

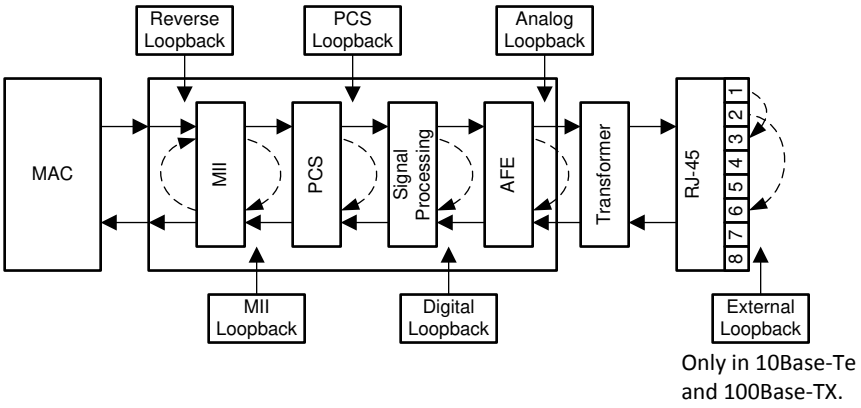

Figure 2-21. Loopbacks

The availability of Loopback depends on the operational mode of the PHY. The Link Status in these loopback modes is also effected by the operational mode. Table 2-15 lists out the exceptions where loopbacks are not available.

Table 2-15. Loopback Availability Exception

OP MODE	LOOPBACK	EXCEPTION
Copper	PCS	10M
Fiber	MII	100M
	PCS	100M
	Analog	100M, 1000M
SGMII to RGMII	PCS	10M, 100M, 1000M
	Digital	10M, 100M, 1000M
	Analog	10M, 100M, 1000M
	External	10M, 100M, 1000M
RGMII to SGMII	PCS	10M, 100M, 1000M
	External	10M, 100M, 1000M
Media Convertor	MII	100M, 1000M
	Analog	100M on Fiber Interface
	External	100M on Fiber Interface
		100M, 1000M on Copper Interface

MII loopback can be used to verify the MAC interface, while reverse loopback is used with a link partner to verify the data path along the MDI.

- MII loopback is enabled by setting register 0x0000[14]
  - In 100Base-TX mode, set register 0x0016[3] as well
- Reverse loopback is enabled by setting register 0x0016[5]

2.5.2 Transmitting and Receiving Packets with the MAC

If generating and checking packets with the MAC is possible, and the PHY has a working link partner with reverse loopback capability, verify the full data path as follows:

1. Connect the PHY to the MAC and a working link partner.
2. Enable reverse loopback on the link partner.

3. Transmit test packets from the MAC to the PHY.
4. Verify the MAC receives the same test packets.

If the MAC receives the same test packets transmitted without issue, the full MAC → PHY → MDI data path is valid. If this test does not pass, perform MII loopback to isolate the issue along the data path:

1. Power and connect the PHY to the MAC.
2. Enable MII loopback on the PHY.
3. Transmit test packets from the MAC to the PHY.
4. Verify the MAC receives the same test packets.

If the MAC receives the same test packets, the MAC → PHY data path is valid, and the issue has been isolated to the MDI data path. If this test does not pass, the issue is likely on the MAC interface. To verify the MAC interface, refer to the Section 2.4 of this application note. To verify the PHY internal data path, perform the above procedure using analog loopback mode.

2.5.3 Transmitting and Receiving Packets with BIST

The device incorporates an internal PRBS Built-in Self-Test (BIST) circuit to accommodate in-circuit testing or diagnostics. The BIST circuit can be used to test the integrity of the transmit and receive data paths. BIST can be performed using various loopback modes to isolate any issues to specific parts of the data path. The BIST generates packetized data with variable content and IPG.

If generating and checking packets with the MAC is not possible, use PRBS packet generation and checking functionality to verify the data path. Perform reverse loopback with PRBS and a working link partner as follows:

1. Power and connect the PHY to a link partner.
2. Enable PRBS packet generation on the PHY (write Reg 0x0016 = 0xF000).
3. Enable reverse loopback on the link partner (If Link Partner is DP83869 write Reg 0x0016 = 0x0020).
4. Wait at least one second, then check PRBS lock status on the PHY by reading register 0x0017[11].

If register 0x0017[11] is high, the data path through PHY → MDI is valid. If this test does not pass, the issue can be on the PHY's internal data path or the MDI. To verify the internal data path, perform PRBS with analog loopback using the following procedure:

1. Write register 0x001F = 0x8000 //PHY reset
2. Write register 0x0000 = 0x0140 //Disable Auto-neg, force 1000Mbps
3. Write register 0x0016 = 0x0008 //Enable Analog loopback, use 100Ω MDI terminations
4. Write register 0x0016 = 0xF008 //Enable PRBS generator
5. Read register 0x0017[11] //Should be high for PRBS locked
6. Read register 0x0072[7:0] //Should be 0x00 for no errors

If the internal data path is valid the issue is isolated to the MDI or the link partner.

3 Operational Mode Clarification

While the data sheet offers sufficient information, this section is a condensed version to help understand how to properly use and configure these modes.

3.1 Bridge Modes

The DP83869HM supports two types of Bridge Modes to translate data between two MAC interface types. The two types are:

- RGMII-to-SGMII mode
- SGMII-to-RGMII mode

The naming convention implies MAC-to-PHY and the functionality of the DP83869HM changes depending on which mode is selected. More information can be found in the *Bridge Modes* section of the data sheet.

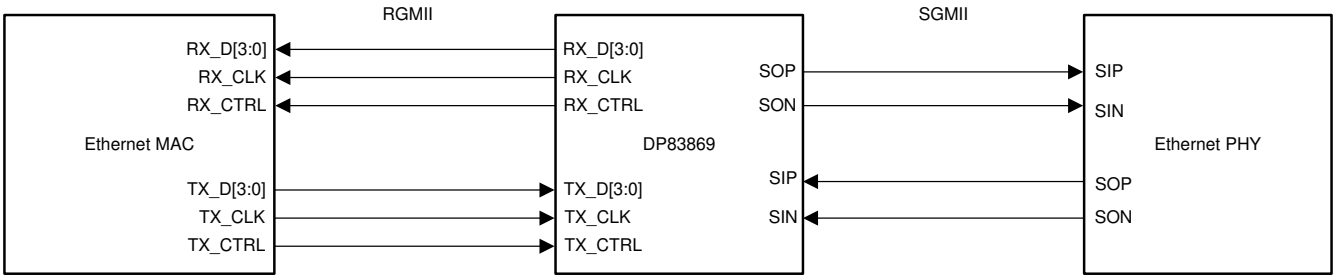

Figure 3-1. DP83869HM RGMII-to-SGMII Bridge

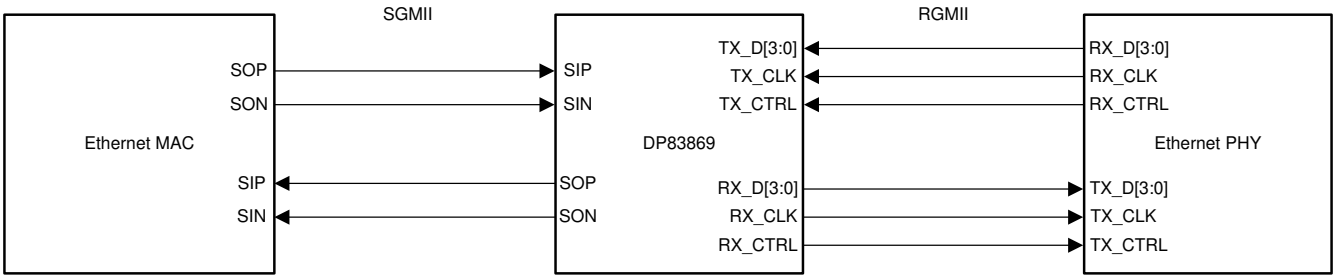

Figure 3-2. DP83869HM SGMII-to-RGMII Bridge

3.2 Fiber Configuration

DP83869HM is capable of 100Base-FX and 1000Base-X fiber communication. When using fiber, speed is not determined by auto-negotiation. Both sides of the link must be configured to the same operating speed. The pins used for fiber are the same pins used for SGMII: SIP/SIN and SOP/SON. The DP83869HM must be set to either *SGMII to Copper, RGMII/MII to fiber, or Copper to fiber*. A SGMII to fiber connection is not possible. For more information on debugging a fiber interface, please see our Ethernet PHY Fiber Debug Guide.

3.2.1 Fiber Registers

DP83869HM has several fiber related registers, this section is intended to summarize a few of their functionalities and show how to debug. Link status can always be checked in register 0x01, but register 0xC01 checks Fiber link.

Table 3-1. Register 0xC01 Behavior

Mode	Register 0xC01 behavior
RGMII to Fiber	Behaves the same as register 0x01. There is no added benefit of reading register 0xC01
Media Convertor Mode	**Case 1:** Copper link is down but Fiber link is up Register 0x01 indicates link is down (copper side), but register 0xC01 shows link is up (fiber side). This is helpful for being able to debug which cable is at fault. **Case 2:** Copper link is up but Fiber link is down Register 0x01 indicates link is down. If register 0xC01 says link is down, register 0x01 is also down.

Case 2 in Table 3-1 does not give full confidence in which link is down. The Fiber Interrupt Status Register (0xC19) can be used to isolate where the fault is. In Case 2, 0xC19[9] (Fiber Far End Fault) and 0xC19[4] (Link Status Change) can flag. In Case 1, only 0xC19[4] can flag.

3.2.2 Media Converter LED Behavior

The DP83869 LEDs can be configured through register 0x0018. The LEDs have 15 modes of operation, denoted from 0 through E. With DP83869 configured as a media converter, the following LED modes can be used to determine the link status:

- Mode 0: Link Ok
- Mode 1: RX/TX Activity
- Mode 2: TX Activity
- Mode 3: RX Activity
- Mode 5: 1000Base-T/1000Base-X
- Mode 7: 10BT Link
- Mode 8: 10/100BT Link
- Mode 9: 100/1000BT Link
- Mode B: Copper Link + Blink on TX/RX Activity

For copper link status, use LED mode 0. Modes 7, 8, and 9 can differentiate between the copper link speeds. For fiber link status, use LED mode 5. LED mode 5 tracks both 1000Base-X and 100Base-X link status. It is not recommended to use LED mode 6 to track the fiber link status in media converter operation. For TX/RX activity use LED mode 1. Modes 2 and 3 can differentiate between TX or RX activity, while mode B allows the LED to blink on activity.

4 Tools and References

The following chapter contains additional tools and references relevant for debugs.

4.1 Extended Register Access

The DP83869HM's Serial Management Interface (SMI) function supports read or write access to the extended register set using registers REGCR (0x0D) and ADDAR (0x0E) and the MDIO Manageable Device (MMD) indirect method defined in IEEE 802.3ah Draft for clause 22 for accessing the clause 45 extended register set.

The standard register set, MDIO registers 0 to 31, is accessed using the normal direct-MDIO access or the indirect method, except for register REGCR (0x0D) and ADDAR (0x0E), which is accessed only using the normal MDIO transaction. The SMI function ignores indirect accesses to these registers.

REGCR (0x0D) is the MDIO Manageable MMD access control. In general, register REGCR(4:0) is the device address DEVAD that directs any accesses of ADDAR (0x0E) register to the appropriate MMD.

The PHY'S supports one MMD device address. The vendor-specific device address DEVAD[4:0] = 11111 is used for general MMD register accesses.

All accesses through registers REGCR and ADDAR must use the correct DEVAD. Transactions with other DEVAD are ignored. REGCR[15:14] holds the access function: address (00), data with no post increment (01), data with post increment on read and writes (10), and data with post increment on writes only (11).

Table 4-1. REGCR DEVAD Functions

REGCR[15:14]	Function
00	Accesses to register ADDAR modify the extended register 'set address' register. This address register must always be initialized to access any of the registers within the extended register set.
01	Accesses to register ADDAR access the register within the extended register set selected by the value in the address register.
10	Access to register ADDAR access the register within the extended register set selected by the value in the address register. After that access is complete, for both reads and writes, the value in the address register is incremented.
11	Access to register ADDAR access the register within the extended register set selected by the value in the address register. After that access is complete, for write accesses only, the value in the address register is incremented. For read accesses, the value of the address register remains unchanged.

The following sections describe how to perform operations on the extended register set using register REGCR and ADDAR. The descriptions use the device address for general MMD register accesses (DEVAD[4:0] = 11111).

4.1.1 Read (No Post Increment) Operation

To read a register in the extended register set:

Instruction	Example: Read 0x0170
1. Write the value 0x001F (address function field = 00, DEVAD = 31) to register REGCR.	Write register 0x0D to value 0x001F
2. Write the desired register address to register ADDAR.	Write register 0x0E to value 0x0170
3. Write the value 0x401F (data, no post increment function field = 01, DEVAD = 31) to register REGCR.	Write register 0x0D to value 0x401F
4. Read the content of the desired extended register set register to register ADDAR.	Read register 0x0E

Subsequent reads from register ADDAR (step 4) continue reading the register selected by the value in the address register.

Note

Steps (1) and (2) can be skipped if the address register was previously configured.

4.1.2 Write (No Post Increment) Operation

To write a register in the extended register set:

Instruction	Example: Set reg 0x0170 = 0C50
1. Write the value 0x001F (address function field = 00, DEVAD = 31) to register REGCR (0x0D).	Write register 0x0D to value 0x001F
2. Write the desired register address to register ADDAR (0x0E).	Write register 0x0E to value 0x0170
3. Write the value 0x401F (data, no post increment function field = 01, DEVAD = 31) to register REGCR.	Write register 0x0D to value 0x401F
4. Write the content of the desired extended register set register to register ADDAR.	Write register 0x0E to value 0x0C50

Subsequent writes to register ADDAR (step 4) continue to rewrite the register selected by the value in the address register.

Note

Steps (1) and (2) can be skipped if the address register was previously configured.

4.2 Software and Driver Debug on Linux

The two essential components required for the PHY to function on a Linux system are the device tree and driver file, for which the DP83869HM drivers can be found here. Below is a sample format of what a device tree looks like. This can be found on any open source kernal under the following path: root/Documentation/devicetree/bindings/net/ti.

```
#include <dt-bindings/net/ti-dp83869.h>
   mdio0 {
     #address-cells = <1>;
     #size-cells = <0>;
     ethphy0: ethernet-phy@0 {
       reg = <0>;
       tx-fifo-depth = <DP83869_PHYCR_FIFO_DEPTH_4_B_NIB>;
       rx-fifo-depth = <DP83869_PHYCR_FIFO_DEPTH_4_B_NIB>;
       ti,op-mode = <DP83869_RGMII_COPPER_ETHERNET>;
       ti,max-output-impedance;
       ti,clk-output-sel = <DP83869_CLK_O_SEL_CHN_A_RCLK>;
       rx-internal-delay-ps = <2000>;
       tx-internal-delay-ps = <2000>;
     };
   };
```

4.2.1 Common Terminal Outputs

The following section is intended to provide common terminal commands that can be used to debug driver related issues.

```
$ dmesg | grep "mdio"
```

One of the possible outputs is as follows:

```
$ mdio_bus xxx.ethernet-x: MDIO device at address 8 is missing
```

This message indicates that the PHY is not found on the MDIO bus, which can be caused by several issues. The most common one being a missing or incorrect device tree, but can also be due to a non-functional PHY or a bad SMI connection.

Once the PHY can be detected on the MDIO bus, another common error message is as follows:

```
$ Generic PHY xxx.ethernet-x: attached PHY driver [Generic PHY]
```

This message indicates that the driver file for the corresponding PHY is not loaded correctly or not present at all, and Linux loaded in a generic driver that most likely won't work with the PHY. In that case, verify that the driver successfully compiled, was added when building Linux, and that the driver matches with the model of PHY used.

Finally, a message like this can display:

```
am65-cpsw-nuss 8000000.ethernet eth1: PHY [mdio_mux-0.1:03] driver [TI DP83869] (irq=POLL)
```

This message shows that the PHY has the correct driver loaded and is detected successfully. Run *ifconfig* to verify the network interface is present. Example *ifconfig* output when the PHYs are successfully recognized as network adapters:

```
root@am64xx-evm:~# ifconfig
eth0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether 34:08:e1:80:b5:f8  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth1: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether 70:ff:76:1e:9e:a6  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 90  bytes 6824 (6.6 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 90  bytes 6824 (6.6 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

5 Summary

This application note provides a suggested flow for evaluating a new application and confirming the expected functionality. The step-by-step recommendations helps ease board bring-up and initial evaluation of DP83869HM designs.

6 References

- Texas Instruments, DP83869HM, product page.
- Texas Instruments, *DP83869HM High Immunity 10/100/1000 Ethernet Physical Layer Transceiver With Copper and Fiber Interface*, datasheet.
- Texas Instruments, Understanding the different modes of operation in DP83869HM, application note.

7 Revision History

Changes from Revision * (December 2023) to Revision A (October 2025)	Page
• Updated the numbering format for tables, figures, and cross-references throughout the document	1
• Reformatted content into PHY and interface health check sections	3
• Added schematic and layout checklist references	3
• Added LED strap Circuit image	7
• Moved USB2MDIO instructional section to 'Read and Check register values' section	9
• Added MSE registers	12
• Added short cable script	12
• Added AGC gain script	13
• Added LED behavior in media converter mode	27

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you fully indemnify TI and its representatives against any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale, TI’s General Quality Guidelines, or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products. Unless TI explicitly designates a product as custom or customer-specified, TI products are standard, catalog, general purpose devices.

TI objects to and rejects any additional or different terms you may propose.

Last updated 10/2025