HB0647
Handbook
CoreMDIO_APB v2.1

Microsemi
Power Matters.™

Microsemi Corporate Headquarters
One Enterprise, Aliso Viejo,
CA 92656 USA
Within the USA: +1 (800) 713-4113
Outside the USA: +1 (949) 380-6100
Fax: +1 (949) 215-4996
Email: sales.support@microsemi.com
www.microsemi.com

Microsemi makes no warranty, representation, or guarantee regarding the information contained herein or the suitability of its products and services for any particular purpose, nor does Microsemi assume any liability whatsoever arising out of the application or use of any product or circuit. The products sold hereunder and any other products sold by Microsemi have been subject to limited testing and should not be used in conjunction with mission-critical equipment or applications. Any performance specifications are believed to be reliable but are not verified, and Buyer must conduct and complete all performance and other testing of the products, alone and together with, or installed in, any end-products. Buyer shall not rely on any data and performance specifications or parameters provided by Microsemi. It is the Buyer's responsibility to independently determine suitability of any products and to test and verify the same. The information provided by Microsemi hereunder is provided “as is, where is” and with all faults, and the entire risk associated with such information is entirely with the Buyer. Microsemi does not grant, explicitly or implicitly, to any party any patent rights, licenses, or any other IP rights, whether with regard to such information itself or anything described by such information. Information provided in this document is proprietary to Microsemi, and Microsemi reserves the right to make any changes to the information in this document or to any products and services at any time without notice.

About Microsemi

Microsemi Corporation (Nasdaq: MSCC) offers a comprehensive portfolio of semiconductor and system solutions for aerospace & defense, communications, data center and industrial markets. Products include high-performance and radiation-hardened analog mixed-signal integrated circuits, FPGAs, SoCs and ASICs; power management products; timing and synchronization devices and precise time solutions, setting the world's standard for time; voice processing devices; RF solutions; discrete components; enterprise storage and communication solutions, security technologies and scalable anti-tamper products; Ethernet solutions; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Microsemi is headquartered in Aliso Viejo, California, and has approximately 4,800 employees globally. Learn more at www.microsemi.com.

Contents

Figures

Tables

1 Revision History

The revision history describes the changes that were implemented in the document. The changes are listed by revision, starting with the most current publication.

1.1 Revision 2.0

Updated changes related to CoreMDIO_APB v2.1.

1.2 Revision 1.0

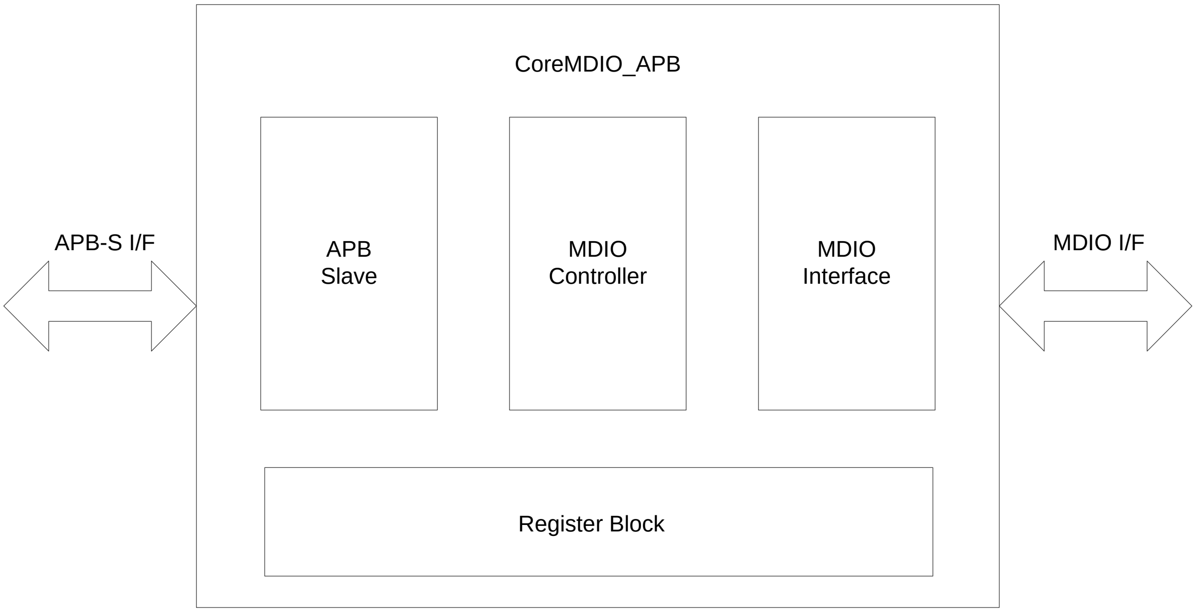
Revision 1.0 was the first publication of this document. Created for CoreMDIO_APB v2.0.

2 Introduction

Management data input/output (MDIO), also known as serial management interface (SMI) is a serial bus defined for the Ethernet family of IEEE 802.3 standards for the media independent interface (MII). The MII connects the media access control (MAC) devices with the Ethernet physical layer (PHY) circuits. The MDIO reads and writes the control and status registers of the PHY, configures each PHY before operation, and monitors link status during operation.

MDIO was originally defined in Clause 22 of IEEE RFC802.3. In the original specification, a single MDIO interface is able to access up to 32 registers in 32 different PHY devices. These registers provide status and control information such as: link status, speed ability and selection, power down for low power consumption, duplex mode (full or half), auto-negotiation, fault signaling, and loopback.

Figure 1 • **Top-Functional Block Diagram**



2.1 Features

CoreMDIO_APB supports the following features:

- IEEE 802.3 Clause 22 standard MDIO interface
- APB Slave interface for register access
- Busy status information for user application
- MDC prescaler

2.2 Core Version

This handbook is for CoreMDIO_APB version 2.1.

2.3 Supported Families

CoreMDIO_APB v2.1 is a generic core and supports all families.

2.4 Device Utilization and Performance

A summary of the utilization data of CoreMDIO_APB is listed in Table 1.

Speed Grade – STD, Core Voltage – 1.2V and Operating Condition- IND.

Table 1 • **Device Utilization and Performance**

		FPGA Resources				
FPGA Family	**Device**	**Combinational**	**Sequential**	**Total**	**%**	**Clock Rate (MHz)**
SmartFusion2	M2S050T	208	128	336	0.5	PCLK = 169
IGLOO2	M2GL050T	207	130	337	0.5	PCLK = 169
RTG4	RT4G150	215	143	358	0.23	PCLK = 169

Note: Data in this table are achieved using synthesis and layout settings optimized for speed.

3 Functional Description

The CoreMDIO_APB is an MDIO controller that provides the set of registers listed in Table 2 to implement the procedures for controlling the MDIO transactions. The CoreMDIO_APB provides the standard APB slave interface to configure the registers required for MDIO read or write transactions. The host or application configures the core registers through the APB slave interface. The CoreMDIO_APB allows management data clock (MDC) pre-scaler and also provides busy status information for the user or host application.

CoreMDIO_APB has the following functional blocks:

- APB Slave Interface
- MDIO Interface
- MDIO Controller Block
- Register Block

3.1 APB Slave Interface

The CoreMDIO_APB provides standard APB slave interface.

3.2 MDIO Interface

The CoreMDIO_APB provides standard MDIO interface.

3.3 MDIO Controller Block

MDIO Controller block drives the MII. The control and status information is exchanged with the attached PHY across MII. It provides an interface between the host processor and one or more PHYs.

3.4 Register Block

Table 2 shows the mapping of the CoreMDIO_APB registers.

Table 2 • **CoreMDIO_APB Registers**

Name	Address	Type	Width	Reset Value	Description
ADDRESSREG	Base + 0×00	R/W	32	0×0	Address register
PHYADDRREG	Base + 0×04	R/W	32	0×0	PHY ID address register
CONTROLREG	Base + 0×08	W/O	32	0×0	Control register
STATUSREG	Base + 0×0C	R/O	32	0×0	Status register
DATAINREG	Base + 0×10	W/O	32	0×0	Data input register
DATAOUTREG	Base + 0×14	R/O	32	0×0	Data output register
CLKPRESCALERREG	Base + 0×18	R/W	32	0×0	Clock prescaler register

Table 3 shows the description of the CoreMDIO_APB registers.

Table 3 • **CoreMDIO_APB Registers Description**

Bit (s)	Field	Type	Function
ADDRESSREG			
Bit [4:0]	ADDR	R/W	MDIO register address
Bit [31:5]	-	-	Reserved
PHYADDRREG			
Bit [4:0]	PHYADDR	R/W	MDIO PHY address
Bit [31:5]	-	-	Reserved
CONTROLREG			
Bit [0]	RD	W/O	To perform single MDIO read cycle. This bit is cleared when the read cycle is performed.
Bit [1]	WR	W/O	To perform single MDIO write cycle. This bit is cleared when the write cycle is performed.
Bit [31:2]	-	-	Reserved
STATUSREG			
Bit [0]	BUSY	R/O	MDIO interface status information. When 1 is returned, it indicates that the MDIO read or write cycle is currently being performed.
Bit [31:1]	-	-	Reserved
DATAINREG			
Bit [15:0]	DATAIN	W/O	When written, a write cycle is performed using the 16-bit data and the preconfigured PHY and register address. The CONTROLREG register bit 1 must be set to perform write cycle.
Bit [31:16]	-	-	Reserved
DATAOUTREG			
Bit [15:0]	DATAOUT	R/O	Following a MDIO read cycle, the 16-bit data is read from this register.
Bit [31:16]	-	-	Reserved
CLKPRESCALERREG			
Bit [2:0]	PRESCALER	R/W	Prescaler value. This determines the MDC frequency. 000 - MGTCLK/4 001 - MGTCLK/4 010 - MGTCLK/6 011 - MGTCLK/8 100 - MGTCLK/10 101 - MGTCLK/14 110 - MGTCLK/20 111 - MGTCLK/28
Bit [31:3]	-	-	Reserved

4 Operation

The CoreMDIO_APB supports the Clause 22 of IEEE 802.3u. The MDIO bus has MDC and MDIO signals.

MDIO has a specific terminology to define the various devices on the bus. The device driving the MDIO bus is placed as the station management entity (STA). The target devices that are handled by the MDC are referred to as MDIO Manageable Devices (MMD).

The STA initiates all communication in MDIO and drives the clock on MDC. MDC has a frequency of 2.5 MHz or higher.

Article 22 defines the basic format of the MDIO communication as shown in Figure 2.

Figure 2 • **Basic MDIO Frame Format**

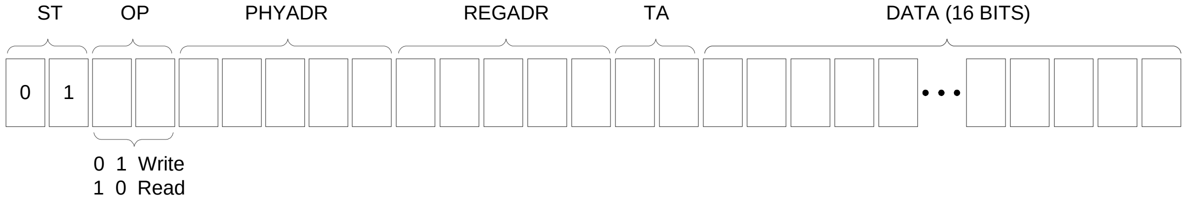


Table 4 shows the description of the basic MDIO frame format.

Table 4 • **I/O Basic MDIO Frame Format Description**

Field	Bits	Description
ST	2 bits	Start of frame (01 for Clause 22)
OP	2 bits	OP code
PHYADR	5 bits	PHY address
REGADR	5 bits	Register address
TA	2 bits	Turnaround time
DATA	16 bits	Data Driven by STA during write Driven by MMD during read

The frame format allows only a 5-bit number for both the PHY address and the register address, which limits the number of MMDs that the STA can interface.

4.1 Write Cycle

When requested by the host, the MDIO controller performs a write cycle using the preconfigured PHY register addresses and the supplied 16-bit data.

4.2 Read Cycle

When requested by the host, the MDIO controller performs a read cycle using the preconfigured PHY register addresses. 16-bit read data is loaded into the MII management status register for use by the host.

5 Interface

5.1 Ports

The port signals for CoreMDIO_APB are described in Table 5 and as shown in Figure 3.

Figure 3 • **CoreMDIO_APB I/O Signals**

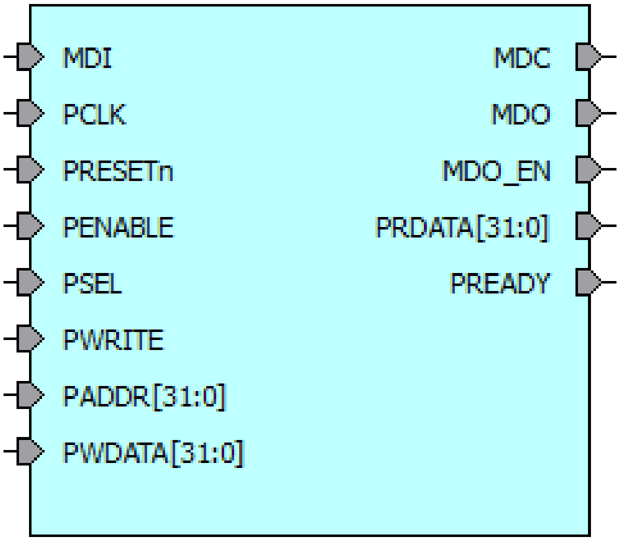


Table 5 • **I / O Signal Description**

Port Name	**Type**	**Description**
Clocks and Reset		
PCLK	Input	APB clock.
PRESETn	Input	Asynchronous reset. Active low.
APB Slave Interface		
PSEL	Input	APB Select.
PWRITE	Input	APB read/write enable.
PADDR[31: 0]	Input	APB address.
PENABLE	Input	APB enable.
PWDATA[31: 0]	Input	APB write data.
PRDATA[31: 0]	Output	APB read data.
PREADY	Output	APB ready.
MDIO Interface		
MDO	Output	MDIO data output.
MDO_EN	Output	MDIO data output enable.
MDI	Input	MDIO data input.
MDC	Output	MDIO data clock. This is an internally generated clock using PCLK.

5.2 Core Parameters

The CoreMDIO_APB does not have any parameter or generics.

6 Timing Diagrams

Refer to *http://standards.ieee.org/getieee802/download/802.3-2012_section2.pdf*

- Figure 22–15: Behavior of MDIO during TA field of a read transaction
- Figure 22–18: MDIO sourced by STA
- Figure 22–19: MDIO sourced by PHY

7 Tool Flow

7.1 License

CoreMDIO_APB is available with the RTL source code.

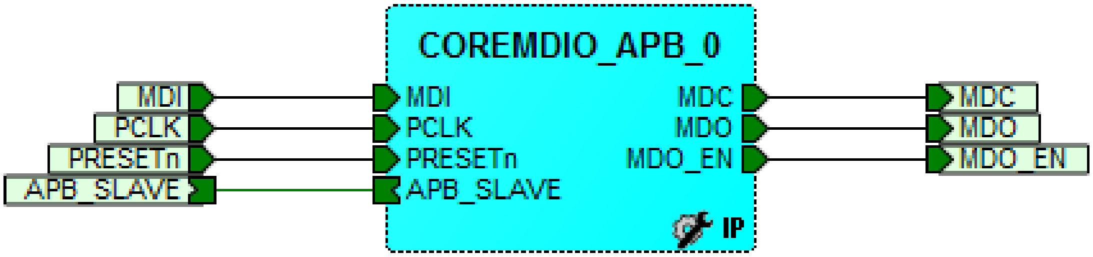
7.2 SmartDesign

CoreMDIO_APB is available for download in the Libero IP catalog through the web repository. Once it is listed in the catalog, the core can be instantiated using the SmartDesign flow. For information on using SmartDesign to configure, connect, and generate cores, refer to the Libero online help. An example instantiated view is shown in Figure 4.

After configuring and generating the core instance, the basic functionality can be simulated using the test-bench supplied with the CoreMDIO_APB. The testbench parameters automatically adjust to the CoreMDIO_APB configuration. The CoreMDIO_APB can be instantiated as a component of a larger design.

CoreMDIO_APB is compatible with Libero SoC.

Figure 4 • **SmartDesign CoreMDIO_APB Instance View**



7.3 Simulation Flows

The user testbench for CoreMDIO_APB is included in all releases. To run simulations, select the user testbench flow within SmartDesign and click Generate Design under the SmartDesign menu. The user testbench is selected through the Core Testbench Configuration GUI. When SmartDesign generates the Libero project, it will install the user testbench files. To run the user testbench, set the design root to the CoreMDIO_APB instantiation in the Libero Design Hierarchy pane and click the Simulation icon in the Libero Design Flow window. This invokes ModelSim® and automatically runs the simulation. After generating CoreMDIO_APB, the presynthesis testbench hardware description language (HDL) files are installed in Libero.

7.4 Synthesis in Libero

To run synthesis on the CoreMDIO_APB, set the design root to the IP component instance and run the synthesis tool from the Libero design flow pane.

7.5 Place-and-Route in Libero

After the design is synthesized, run the compilation and then place-and-route the tools. CoreMDIO_APB does not require no special place-and-route settings.

8 Testbench

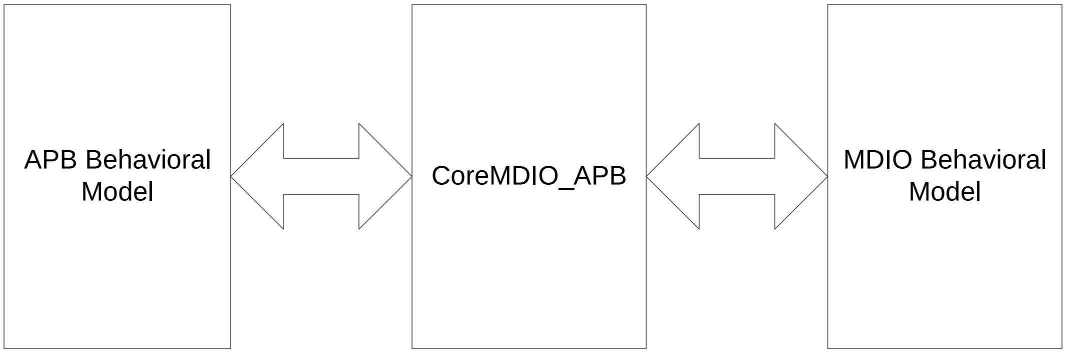
A unified test-bench is used to verify and test CoreMDIO_APB called as user test-bench.

8.1 User Testbench

Basic unit-level verification environment is created as part of the CPZ delivery.

Figure 5 shows the block diagram of the user testbench. The user testbench instantiates the CoreMDIO_APB and the behavioral code that generates APB transactions required to generate an MDIO transaction.

Figure 5 • **User Testbench**



The user testbench is included with the CoreMDIO_APB releases that verify the operations of MDIO.

8.1.1 Test Cases:

The following are the MDIO write and read test sequences:

1. **Configure the CoreMDIO_APB registers for an MDIO write transaction:**
   - Write to REG ADDR register with MDIO register address
   - Write to PHY ADDR register with MDIO PHY address
   - Write to DATA IN register with MDIO write data
   - Write to CONTROL register with initiate write bit asserted.
   - Wait for Busy status to be zero in Read status register.
2. **Configure the CoreMDIO_APB registers for an MDIO read transaction:**
   - Write to REG ADDR register with MDIO register address
   - Write to PHY ADDR register with MDIO PHY address
   - Write to CONTROL register with initiate read bit asserted.
   - Poll for Busy status in Read status register.
   - Read to DATA OUT register for MDIO read data
3. **Compare read data with write data**

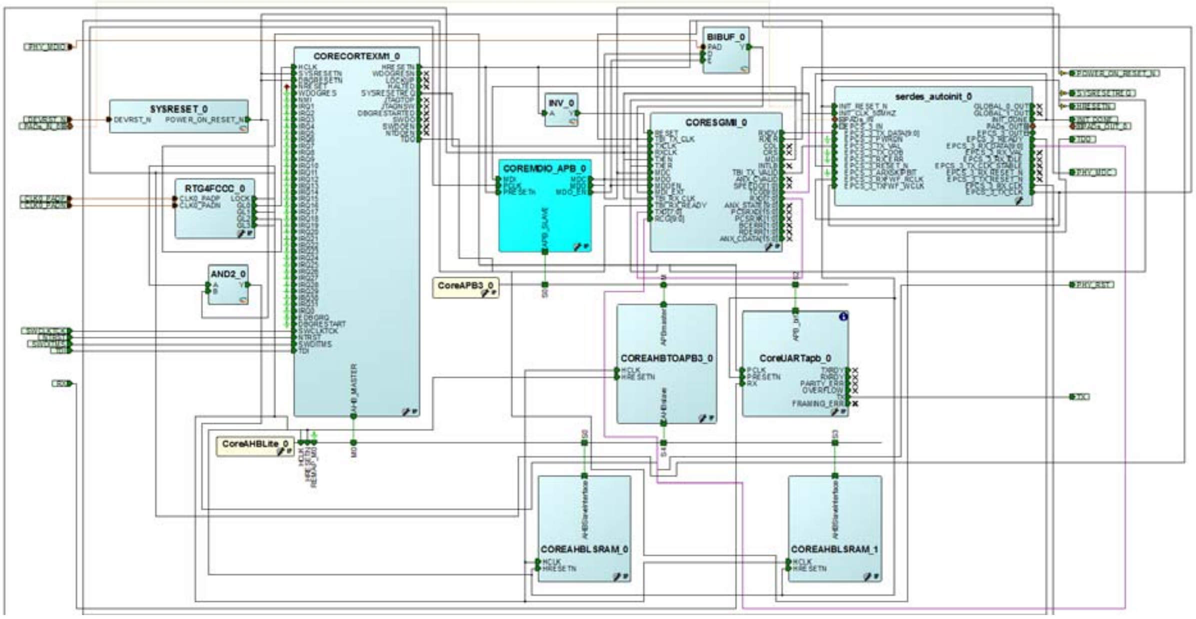
9 System Integration

Figure 6 shows an example design with the CoreMDIO_APB features and the loopback implementation at CoreSGMII’s GMII interface on RTG4 Development Kit.

This example design requires the following:

- CoreCortexM1 connected to CoreMDIO_APB through CoreAHBLite, CoreAHBtoAPB3, and CoreAPB3.
- CoreSGMII connected to an external serial gigabit media independent interface (SGMII) PHY through serdes_autoinit.
- Firmware application for CoreCortexM1 to configure CoreSGMII and external PHY through CoreMDIO_APB.

Figure 6 • **CoreMDIO_APB Example Design**



- POWER_ON_RESET_N from SYSRESET_0 is the reset used for CoreMDIO_APB_0.
- 50 MHz PCLK is generated from RTG4FCCC_0.
- The COREMDIO_APB_0 generates the 2.5 MHz MDC clock from PCLK.
- The CORESGMII_0 has TXCLK, RXCLK, TBI_TXCLK, TBI_RXCLK and MDC clocks.
- TBI_TX_CLK and TXCLK are connected to 125MHz EPCS_3_TX_CLK of serdes_autoinit _0.
- TBI_RX_CLK and RXCLK are connected to 125MHz EPCS_3_RX_CLK of serdes_autoinit _0.
- CORECORTEXM1_0 configures the CORESGMII_0 and external PHY through COREMDIO_APB_0.

Run the Libero flow with enabling the Timing Driven and High Effort Place and Route options enabled. The example design can be obtained from the Microsemi technical support team.

10 Ordering Information

10.1 Ordering Codes

No license is required for CoreMDIO_APB.