XAPP809 (v1.2) June 5, 2007

Reference System: PLB Gigabit Ethernet MAC with a SerDes Interface

Author: Norbert Melnikov

Summary

This application note describes a reference system which illustrates how to build an embedded PowerPC® system using the Xilinx 1-Gigabit Ethernet Media Access Controller processor core. This system has the PLB_Gemac configured to use Scatter/Gather Direct Memory Access and the Serializer/Deserializer (SerDes) interface. This application note describes how to set up the specific clocking structure required for the SerDes interface and the constraints to be added to the UCF file. This reference system is complete with a standalone software application to test some of the main features of this core, including access to registers, DMA capabilities, transmit and receive in loopback mode. This reference system is targeted for the ML300 evaluation board.

Included Systems

Included with this application note is the reference system for Xilinx ML300 Evaluation Board. The reference system is available for download at:

- www.xilinx.com/bvdocs/appnotes/xapp809.zip

Introduction

Using Ethernet Media Access Controllers (MACs) in embedded microprocessor systems is becoming increasingly prevalent. Xilinx has a variety of different Ethernet solutions available to be used in embedded applications as peripherals to both the Processor Local Bus (PLB) and On-Chip Peripheral Bus (OPB). The PLB 1-Gigabit Ethernet MAC solution (referred to as PLB_Gemac from here on) supports three different PHY interfaces, Gigabit Media Independent Interface (GMII), Ten Bit Interface (TBI) and the SerDes interface. The transmitter side of the SerDes converts data from serial to parallel and the receiver side converts data from parallel to serial. This allows for a greatly reduced signal count interface and enables PLB_Gemac to take advantage of the RocketIO™ Multi-Gigabit Transceivers (MGTs) available in Virtex™-II Pro and Virtex-4 devices. PLB_Gemac is also DMA capable to provide for lower processor utilization when processing packets. The reference system described in this application note is for a Virtex-II Pro device using the ML300 board with the PLB_Gemac configured to use the SerDes interface and Scatter Gather DMA.

Hardware and Software Requirements

The hardware and software requirements are:

- Xilinx ML300 Evaluation Board
- Xilinx Platform USB cable or Parallel IV programming cable
- RS232 serial cable and serial communication utility (HyperTerminal)
- Xilinx Platform Studio 9.1.01i
- Xilinx Integrated Software Environment (ISE™) 9.1.03i

Reference System Specifics

This reference system contains only the IP cores necessary to provide an example of how to set up PLB Gemac and how to test it. In addition to the PowerPC™ processor and PLB Gemac core, this system includes PLB DDR, PLB BRAM, OPB UARTLite with interrupts, and an OPB GPIO core to control the LED's.

See Figure 1 for the block diagram and Table 1 for the address map of this system.

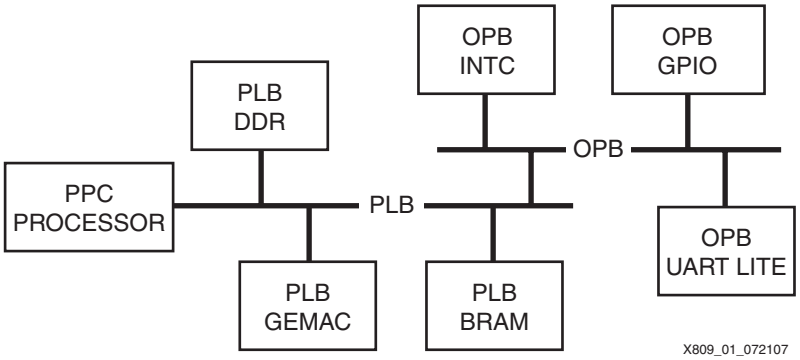

Figure 1: **Reference System Block Diagram**

Address Map

Table 1: **Reference System Address Map**

Peripheral	Instance	Base Address	High Address
PLB_DDR	DDR_SDRAM_32Mx32	0x00000000	0x07FFFFFF
OPB_GPIO	LEDs_13Bit	0x40000000	0x4000FFFF
OPB_UARTLITE	RS232_Uart_1	0x40600000	0x4060FFFF
OPB_INTC	OPB_INTC_0	0x41200000	0x4120FFFF
PLB_GEMAC	PLB_GEMAC__0	0x80400000	0x8040FFFF
PLB_BRAM	PLB_BRAM_IF_CNTLR_1	0xFFFF8000	0xFFFFFFFF

Configuring the PLB_Gemac Core and Setting up its Clocking Structure

The reference system has the PLB Gemac configured to use the SerDes interface and Scatter/Gather DMA.

It is important to note that the when the PLB GEMAC core is configured to use the SerDes interface, only the SERDES interface must be turned ON while the TBI and GMII interfaces must be turned OFF. This is done by setting the parameters in the GEMAC features tab in the PLB GEMAC core. The parameter *Enable SERDES* is set to 1and the parameters *Enable GMII Interface* and *Enable TBI Interface* are turned off by setting them to 0.

Note: **If two or more interfaces are turned on at the same time, the core will not function**.

Figure 2 shows the GeMAC features settings for the PLB GEMAC core to use the SERDES interface.

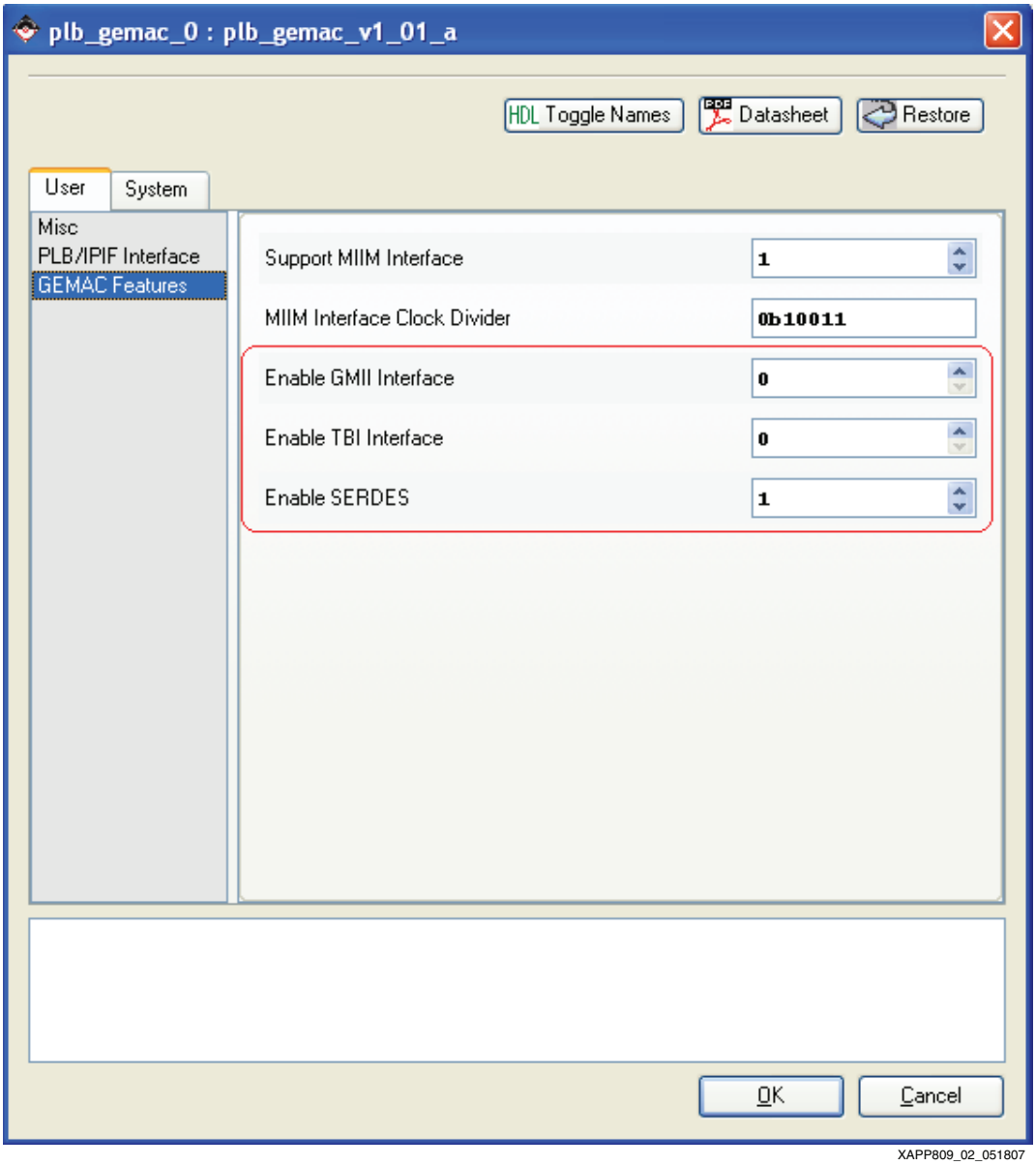

Figure 2: **GeMAC Features for PLB GEMAC**

The PLB GEMAC is set to include the Scatter/Gather DMA capabilities by setting the parameter *DMA Mode* to 3 in the PLB /IPIF Interface tab in the PLB GEMAC core.

Figure 3 shows the DMA parameter settings for the PLB GEMAC core.

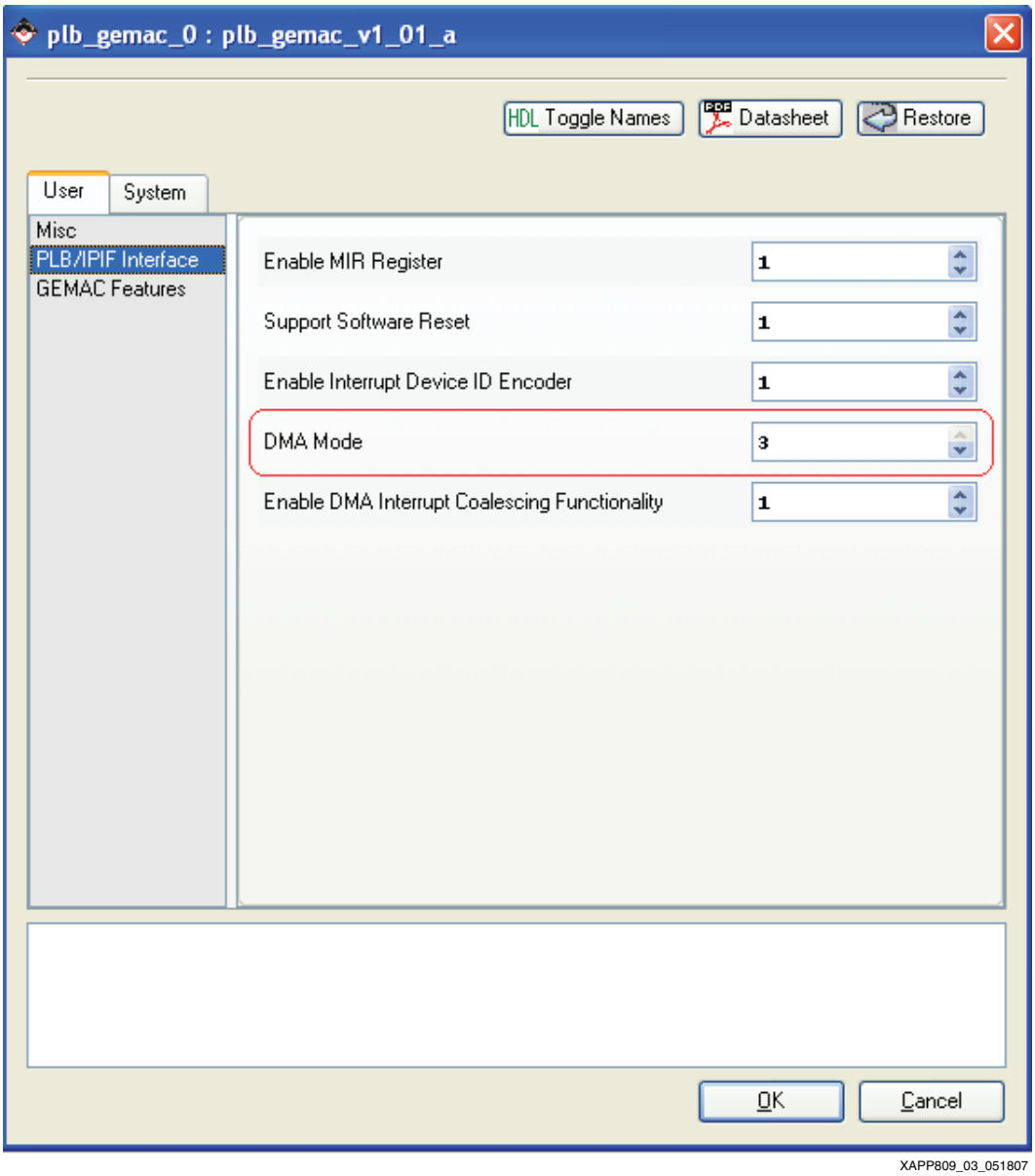

Figure 3: **DMA parameter setting for PLB GEMAC**

The SerDes interface requires a special clocking structure, shown in Figure 4, is set up with a Digital Clock Manager (DCM). The only requirement is the availability of a 62.5 MHz high quality clock. This 62.5 MHz clock is fed into the DCM to additionally create a clock that has twice the frequency and 180° out-of-phase (CLK2X180).

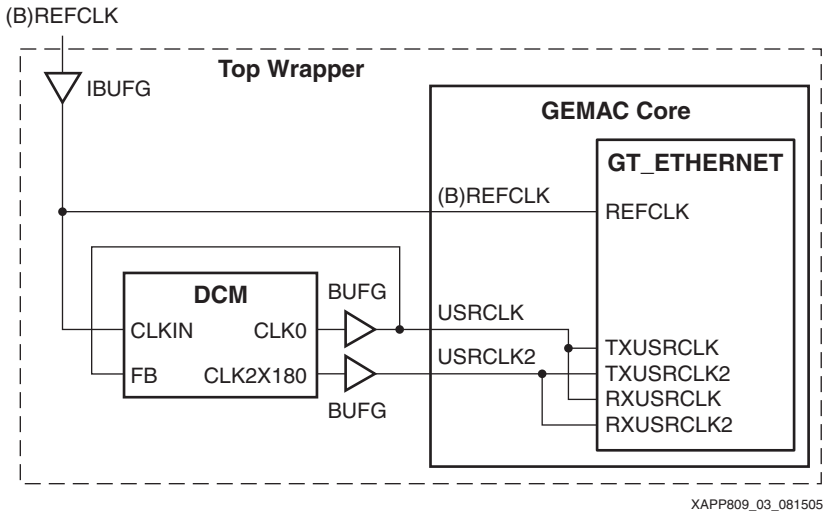

Figure 4: **Clock Management for the SerDes Interface**

The ML300 evaluation board provides the necessary 62.5 MHz clock that is fed into a custom hardware module included in the pcores area of this project. This hardware module, dcm_ip, functions as illustrated in Figure 4. The period constraints for the three different SerDes clocks have also been added to the UCF file. Additionally, the UCF file contains some RocketIO MGT specific constraints and a constraint to choose REFCLK from the four available high-quality reference clocks: REFCLK, REFCLK2, BREFCLK, and BREFCLK2.

In addition to the DCM used for the PLB_Gemac clocks, this reference systems also uses two other DCMs — one for generating the processor clock and phase shifted clocks for the DDR and the other for phase shifting the DDR feedback clock. The locked signals of these three DCMs have been connected to some LEDs on the ML300 board to ensure that the DCMs lock as expected. The locations of these LEDs are M1, K5, P6 as can be seen in the UCF file.

Software Application

The software application for this reference system runs the self-test application for the PLB_Gemac core. These tests are also available in the software area of the EDK installation directory as standalone tests, however this reference system has combined all of them. Specifically, the tests exercise the PLB_Gemac core in polled, FIFO driven interrupt, and SGDMA driven interrupt modes. All of these tests are run in internal loopback mode that is available when the SerDes interface is enabled.

In addition to the PLB_Gemac tests, also a simple GPIO test is included. This test is run before any of the other tests and it walks through a series of 14 LEDs on the ML300 board numerous times.

All of the tests described above are run out of the on-board DDR memory.

Executing the Reference System

To execute this reference system, the ML300 board must be set up correctly and the bitstreams must have been updated and be ready for download. Pre-built bitstreams that have been verified to work properly are also available in the `ready_for_download/` directory under the project root directory. A HyperTerminal (or any other terminal) must be connected to the COM port that is connected to the innermost UART terminal on the board. The terminal settings must

have the baud rate set to 9600 and data bits to 8. See Figure 5 for the HyperTerminal settings. The UART terminal is used to capture the results of the tests.

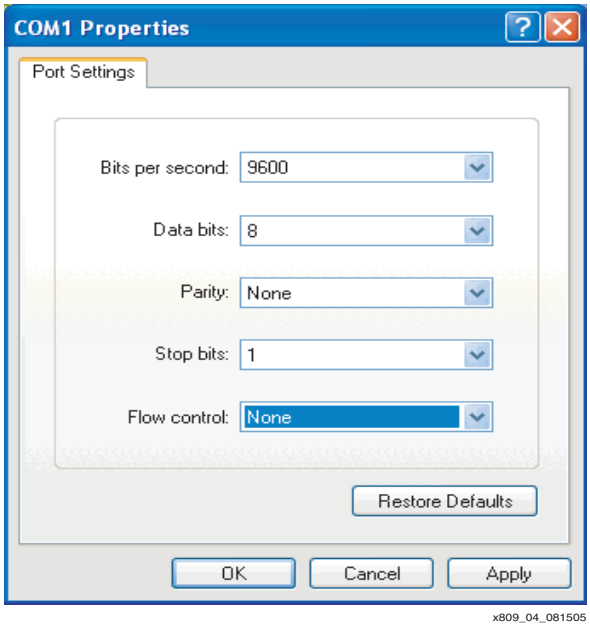

Figure 5: **HyperTerminal Settings**

Executing the Reference System using the Pre-Built Bitstream and the Compiled Software Applications

To execute the system using files inside the `ready_for_download/` in the project root directory, follow these steps:

1. Change directories to the `ready_for_download` directory.
2. Use iMPACT to download the bitstream by using the following:

   ```
   impact -batch xapp809.cmd
   ```

3. Invoke XMD and connect to the PowerPC 405 processor by the following command:

   ```
   xmd -opt xapp809.opt
   ```

4. Download the executables by the following command:

   ```
   dow executable.elf
   ```

Executing the Reference System from EDK

To execute the system using EDK, follow these steps:

1. Open `system.xmp` inside EDK.
2. Use **Hardware** → **Generate Bitstream** to generate a bitstream for the system.
3. Use **Software** → **Build All User Applications** to build the software applications.
4. Download the bitstream to the board with **Device Configuration** → **Download Bitstream**.
5. Launch XMD with **Debug** → **Launch XMD...**
6. Download the executables by the following command:

   ```
   dow executable.elf
   ```

Running the Software Applications

To run the either of software applications, use the ***run*** command inside XMD. The status of the software application is displayed in the HyperTerminal data screen.

Running the OPB Central DMA Software Application

After downloading the hardware bitstream, an XMD connection must be established. Once connected, the compiled software application, TestApp_Peripheral/executable.elf, must be downloaded and executed. Since the software application starts running a GPIO test, the LEDs on board will start toggling. The blinking of the LEDs alone is a good indication of whether the hardware is configured correctly and whether the software application is running as expected.

The PLB_Gemac tests produce an output on the HyperTerminal and run over the course of several minutes. If all the tests pass, the output should read as follows:

```
-- Entering main() --

Starting XGemacFifoIntrExample
Successfully finished XGemacFifoIntrExample

Starting XGemacPolledExample
Successfully finished XGemacPolledExample

Starting XGemacSgDmaIntrExample
Successfully finished XGemacSgDmaIntrExample

-- Exiting main() --
```

If the tests produce no output, but the GPIO test runs properly, the problem is most likely related to the HyperTerminal settings or communication cable connection. If not even the GPIO test ran, the project must have not built correctly or there was an error in downloading the hardware or software bitstreams.

References

PLB 1-Gigabit Ethernet Media Access Controller (MAC) with DMA Product Specification, Xilinx DS460

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
09/22/05	1.0	Initial Xilinx release.
9/20/06	1.1	Modified for Xilinx Iron version of tools.
6/5/07	1.2	Updated for EDK 9.1.01i.