XAPP1041 (v2.0) September 24, 2008

Reference System: XPS LL Tri-Mode Ethernet MAC Embedded Systems for MicroBlaze and PowerPC Processors

Author: Ed Hallett

Abstract

This application note describes three reference systems and outlines how to use the XPS Local Link Tri-Mode Ethernet Media Access Controller (XPS_LL_TEMAC).

1. This reference system uses the Virtex®-5 FPGA with the built-in PowerPC® 440 processor. This system uses the embedded PowerPC 440 as the microprocessor and is built for the Xilinx ML507 board.
2. This reference system uses the Virtex-4 FPGA with the built-in PowerPC 405 processor. This system uses the embedded PowerPC 405 as the microprocessor and is built for the Xilinx ML405 board.
3. This reference system uses a Virtex-5 FPGA and the MicroBlaze™ processor. This system uses the soft MicroBlaze (v7.10.a) embedded processor and is built for the Xilinx ML505 board.

The reference systems configure the XPS_LL_TEMAC to use the Scatter/Gather DMA engine integrated into the Multi-Ported Memory Controller (MPMC) and to support a Gigabit Media Independent Interface (GMII) PHY.

The reference systems include a software test suite that provides examples of how to verify the functionality of the XPS_LL_TEMAC IP core and how to measure raw Ethernet performance. This application is a simple stand-alone program capable of transmitting and receiving raw Ethernet packets for measuring performance and looping back received data.

Included Systems

Included with this application note are three reference systems:

1. PowerPC 440 Processor Reference System

The project name used in `xapp1041_ppc440.zip` is `ml507_ppc_xps_ll_temac`.

2. PowerPC 405 Processor Reference System

The project name used in `xapp1041_ppc405.zip` is `ml405_ppc_xps_ll_temac`.

3. MicroBlaze Processor Reference System

The project name used in `xapp1041_mb.zip` is `ml505_mb_xps_ll_temac`.

Introduction

Using Ethernet Media Access Controllers (EMACs) in embedded microprocessor systems is becoming increasingly prevalent. In fact, the usage is so high that Xilinx has integrated an Ethernet MAC into the fabric of the latest members of the Virtex family of FPGAs. Both the Virtex-4 FX and Virtex-5 LXT/FXT architectures include hard Tri-mode Ethernet controllers. This hardened Ethernet MAC is capable of transmitting and receiving data at 10, 100, and 1000 Mbps and supports interfacing to MII, GMII, Reduced GMII (RGMII), Serial GMII (SGMII), and 1000BASE-X, all while consuming no FPGA resources because the Ethernet MAC is embedded in the Virtex devices mentioned. Additionally, for customers who need to use an Ethernet controller in an architecture that does not provide a hard Ethernet MAC, the

XPS_LL_TEMAC can be configured to be implemented completely using the programmable logic cells as a soft core. Xilinx also provides a parameterizable bus interface, PLBv46, to the hard and soft Tri-mode Ethernet controllers so that they can be easily connected in embedded processor systems. The XPS_LL_TEMAC core is complete with variable size FIFOs and ports to the Scatter/Gather DMA engine, via a LocalLink interface. This is made available through either MPMC with the SDMA PIM or through the PPC440MC DDR2 controller using the HDMA built into the processor block to make building embedded processor systems much easier.

The reference systems described in this application note were originally created using Base System Builder (BSB) and have been further enhanced to optimize raw Ethernet performance. The XPS_LL_TEMAC core is configured to use Scatter/Gather DMA, hardware Checksum Offloading (CSO), and interfaces to the on-board PHY chip through a GMII interface from a Virtex-5 LXT, Virtex-4 FX, or Virtex-5 LXT FPGA. The MPMC Soft Direct Memory Access (SDMA) interface includes a receive (Rx) and transmit (Tx) hardware Data Realignment Engine (DRE), which is always enabled. The SDMA controller port on the MPMC provides direct connection between the Ethernet MAC and the memory controller, thereby reducing congestion on the system bus.

A brief overview of the XPS_LL_TEMAC core features will be presented as a basis for discussing enhancements. A section describing the clocking structure at the system level and the XPS_LL_TEMAC and MPMC core levels for each system is also included.

This application note demonstrates how to optimize each system to obtain the highest raw Ethernet performance. Although BSB sets most of the system and core parameters optimal for Ethernet performance, there are some considerations depending on how the system is utilized.

A standalone performance application, PerfApp, provided with each reference system is described in detail. A basic overview of the test suite menu and test flow is provided as well as details of how to run each of the tests. Raw Ethernet performance data will be presented and discussed for each of the reference systems.

TCP/IP performance is beyond the scope of this application note. Future application notes are in process to provide this information.

Hardware and Software Requirements

The hardware and software requirements are:

- Xilinx ML507 Development Board for the PowerPC 440 processor reference system
- Xilinx ML405 Development Board for the PowerPC 405 processor reference system
- Xilinx ML505 Development Board for the MicroBlaze processor reference system
- Xilinx Platform USB Download Cable or Parallel IV Download Cable
- RS232 Serial Cable
- Ethernet Cable
- Serial Communications Utility Program (e.g. HyperTerminal)
- Xilinx Platform Studio 10.1.02
- ISE® 10.1.02 design tools

Reference System Specifics

PowerPC 440 Processor Reference System

The `ml507_ppc_xps_ll_temac` reference system is composed of an embedded PowerPC 440 processor EDK project. The included PerfApp software application is based upon the Xilinx standalone Board Support Package (BSP).

This reference system contains the IP cores necessary to provide an example of how to set up XPS_LL_TEMAC, how to verify that the core is operational, and how to measure raw Ethernet performance. In addition to the PowerPC 440 processor and the XPS_LL_TEMAC, this system includes the PowerPC 440 MC memory controller for DDR2, the Block RAM memory controller,

a UART core, two GPIO cores, and an INTC interrupt controller. The XPS_LL_TEMAC PHY interface signals are connected to the tri-speed Marvell Alaska 88E1111 PHY on the ML507 board. For this reference system the GMII PHY interface type is used.

See Figure 1 for the block diagram and Table 1 for the address map of the PowerPC 440 processor system.

Block Diagram for PowerPC 440 Processor System

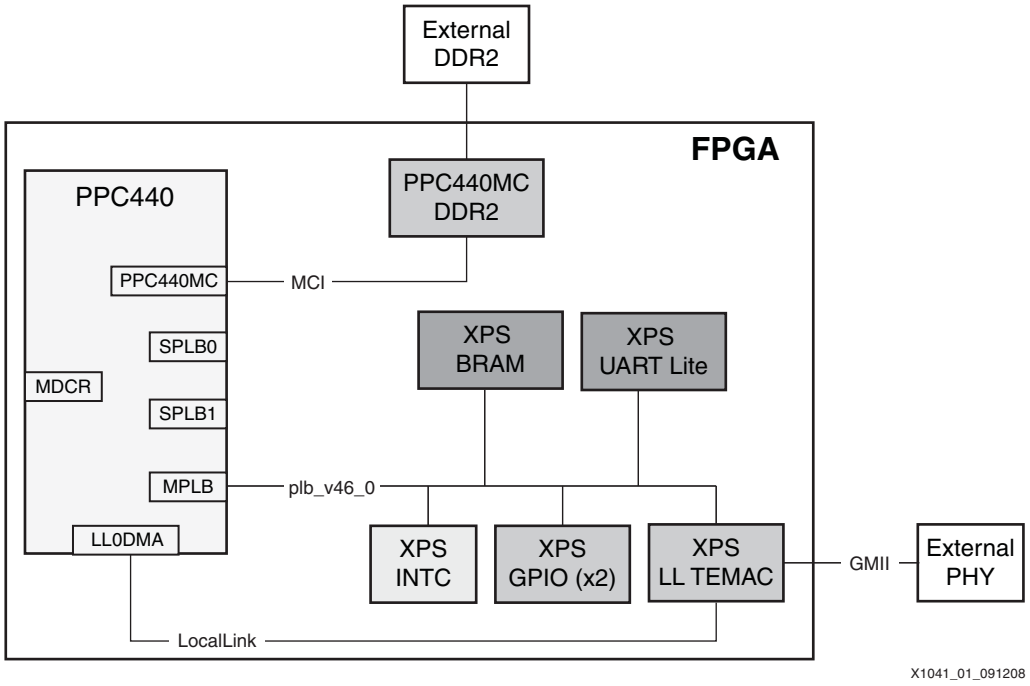

Figure 1: **PowerPC 440 Processor Reference System Block Diagram**

Address Map

Table 1: **PowerPC 440 Processor Reference System Address Map**

Instance	Peripheral	Base Address	High Address
xps_bram_if_cntrl_1	xps_bram_if_cntrl	0xFFFFC000	0xFFFFFFFF
DDR2_SDRAM	ppc440mc_ddr2	0x00000000	0x0FFFFFFF
RS232_Uart	xps_uartlite	0x84000000	0x8400FFFF
xps_intc_0	xps_intc	0x81800000	0x8180FFFF
Hard_Ethernet_MAC	xps_ll_temac	0x81C00000	0x81C0FFFF
LEDs_8Bit	xps_gpio	0x81400000	0x8140FFFF
Push_Buttons_Position	xps_gpio	0x81420000	0x8142FFFF

PowerPC 405 Processor Reference System

The `ml405_ppc_xps_ll_temac` reference system is composed of an embedded PowerPC 405 processor EDK project. The included PerfApp software application is based upon the Xilinx standalone Board Support Package (BSP).

This reference system contains the IP cores necessary to provide an example of how to set up XPS_LL_TEMAC, how to verify that the core is operational, and how to measure raw Ethernet performance. In addition to the PowerPC405 processor (with the PLBv46 Wrapper) and

XPS_LL_TEMAC, this system includes controllers for DDR and Block RAM memory, a UART core, two GPIO cores, and an interrupt controller. The XPS_LL_TEMAC PHY interface signals are connected to the tri-speed Marvell Alaska 88E1111 PHY on the ML405 board. For this reference system the GMII PHY interface type is used.

See Figure 2 for the block diagram and Table 2 for the address map of the PowerPC 405 processor system.

Block Diagram for PowerPC 405 Processor System

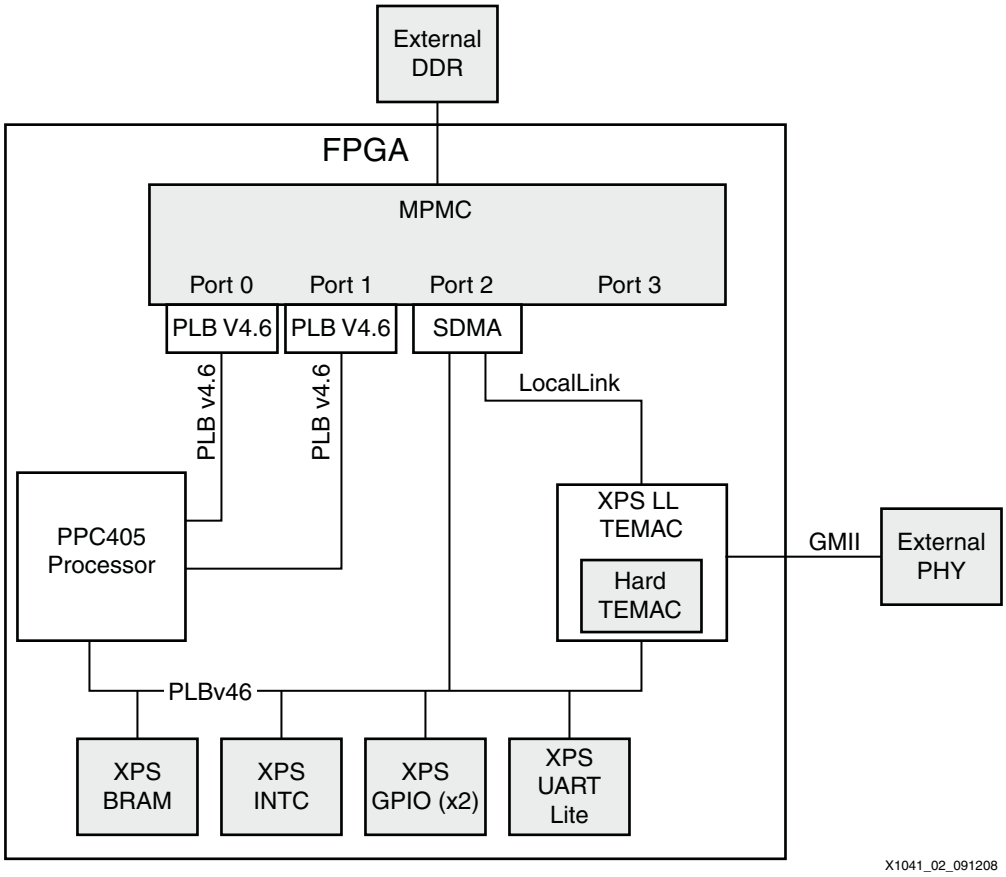

Figure 2: **PowerPC 405 Processor Reference System Block Diagram**

Address Map

Table 2: **PowerPC 405 Processor Reference System Address Map**

Instance	Peripheral	Base Address	High Address
xps_bram_if_cntrl_1	xps_bram_if_cntrl	0xFFFFE000	0xFFFFFFFF
DDR_SDRAM	mpmc (v4.00.a)	0x00000000	0x07FFFFFF
DDR_SDRAM SDMA	mpmc (v4.00.a)	0x84600000	0x8460FFFF
RS232_Uart	xps_uartlite	0x84000000	0x8400FFFF
xps_intc_0	xps_intc	0x81800000	0x8180FFFF
TriMode_MAC_GMII	xps_ll_temac	0x81C00000	0x81C0FFFF
LEDs_4Bit	xps_gpio	0x81420000	0x8142FFFF
Push_Buttons_Position	xps_gpio	0x81400000	0x8140FFFF

MicroBlaze Processor Reference System

The `ml505_mb_xps_ll_temac` reference system is composed of an embedded MicroBlaze EDK project. The included PerfApp software application is based upon the Xilinx standalone Board Support Package (BSP).

This reference system contains the IP cores necessary to provide an example of how to set up XPS_LL_TEMAC, how to verify that the core is operational and how to measure raw Ethernet performance. In addition to the MicroBlaze processor and XPS_LL_TEMAC, this system includes controllers for Block RAM memory and DDR2, a UART core, two GPIO cores, and an interrupt controller. The MicroBlaze processor does not have an integral timer, as does the PowerPC 405 processor, so this system also includes an XPS Timer IP core to handle the timer functions needed in the software application. The XPS_LL_TEMAC PHY interface signals are connected to the tri-speed Marvell Alaska 88E1111 PHY on the ML505 board. For this reference system the GMII PHY interface type is used. See Figure 3 for the block diagram and Table 3 for the address map of the MicroBlaze processor system.

Block Diagram for MicroBlaze Processor System

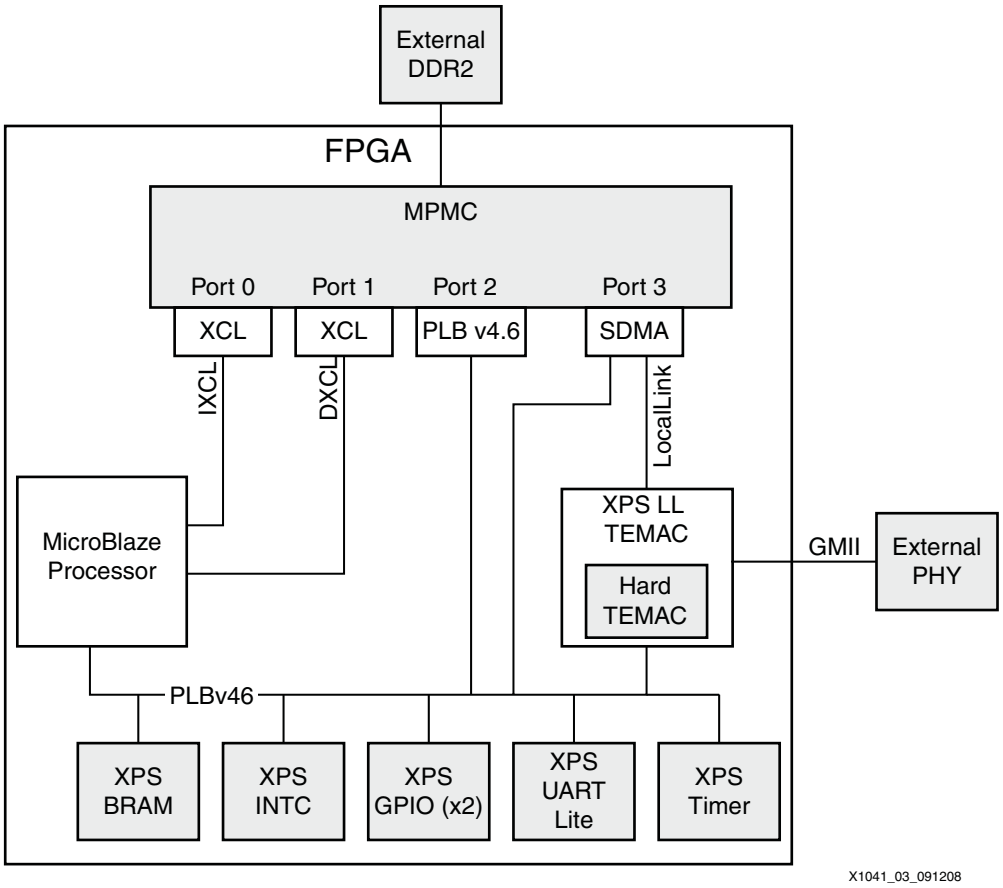

Figure 3: **MicroBlaze Processor Reference System Block Diagram**

Address Map

Table 3: **MicroBlaze Processor Reference System Address Map**

Instance	Peripheral	Base Address	High Address
xps_bram_if_cntrl_1	xps_bram_if_cntrl	0x8A308000	0x8A309FFF
DDR2_SDRAM	mpmc (v4.00.a)	0x90000000	0x9FFFFFFF
DDR2_SDRAM SDMA	mpmc (v4.00.a)	0x84600000	0x8460FFFF
RS232_Uart_1	xps_uartlite	0x84000000	0x8400FFFF
xps_intc_0	xps_intc	0x81800000	0x8180FFFF
Hard_Ethernet_MAC	xps_ll_temac	0x81C00000	0x81C0FFFF
LEDs_8Bit	xps_gpio	0x81420000	0x8142FFFF
Push_Buttons_5Bit	xps_gpio	0x81400000	0x8140FFFF
xps_timer_1	xps_timer	0x83C00000	0x83C0FFFF

Clock Management

There are slight differences in clocking between the Virtex-4 and Virtex-5 hard TEMAC cores and the soft core implementation. There are also differences in the clocking for the PowerPC 440 processor and PowerPC 440MC memory controller compared to the PowerPC 405 and MicroBlaze processors using the MPMC. The following sections describe the clocking for the ML507 PowerPC 440 processor reference system using the Virtex-5 hard Temac core, the ML405 PowerPC 405 processor reference system using the Virtex-4 hard Temac core, and the ML505 MicroBlaze processor reference system using the Virtex-5 hard Temac core. A section is also included to describe a PowerPC 405 processor system that will give higher Ethernet performance when using a faster speed grade part for Virtex-4. Refer to the XPS_LL_TEMAC data sheet for more details on clocking.

Clocking Structure for the Included ML507 PowerPC 440 Processor System

System Level Clocking for the Included PowerPC 440 Processor Reference System

The ML507 board has a slow speed grade Virtex-5 device, so the fastest that the processor clock can run is 400MHz. PPC440MC DDR2 memory controller clock needs to be an integral multiple of the PLBv46 bus clock and can run at a maximum of 266.66 MHz for this speed grade. With these limitations, the system clock frequencies shown in Table 4 were chosen to obtain the highest possible raw Ethernet performance for this system.

Table 4: **PowerPC 440 Processor System Clock Frequencies**

Clock	Enhanced System Frequency	BSB System Frequency
PLBv46 Bus Clock	133.33 MHz	160 MHz
PPC440MC DDR2 Clock	266.66 Mhz	160 Mhz
PowerPC 440 Processor Clock	400 MHz	400 MHz

The PowerPC 440 processor reference system included with this application note uses the Clock Generator core (v2.01.a) to provide the system clocking. The BSB-created clocking scheme was modified to obtain the highest possible raw Ethernet performance, as shown in Table 4 and Figure 4. The Clock Generator provides automatic instantiation of the DCMs and connections. Automatic BUFG insertion and reset sequence determination are also provided by the Clock Generator. The Clock Generator provides a grouping function to force specified clocks to be generated using the same DCM to minimize clock skew. This grouping can be seen

by viewing the parameters in the Clock Generator instance in the MHS file or the Clock Generator Wizard in XPS. Refer to DS614 for further details regarding the Clock Generator.

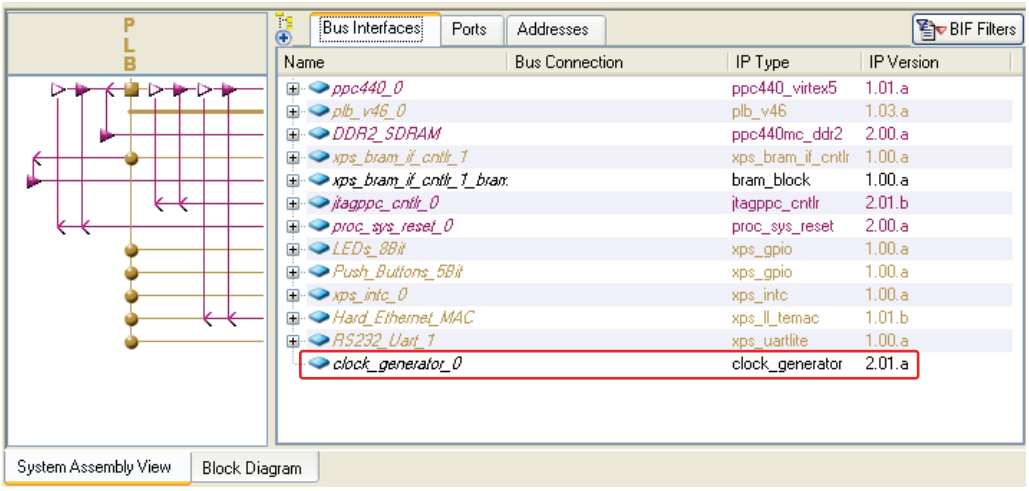

Figure 4: **EDK System Assembly View Showing Clock Generator**

A block diagram of the PowerPC 440 processor system clocking scheme, using the Clock Generator, is shown in Figure 5.

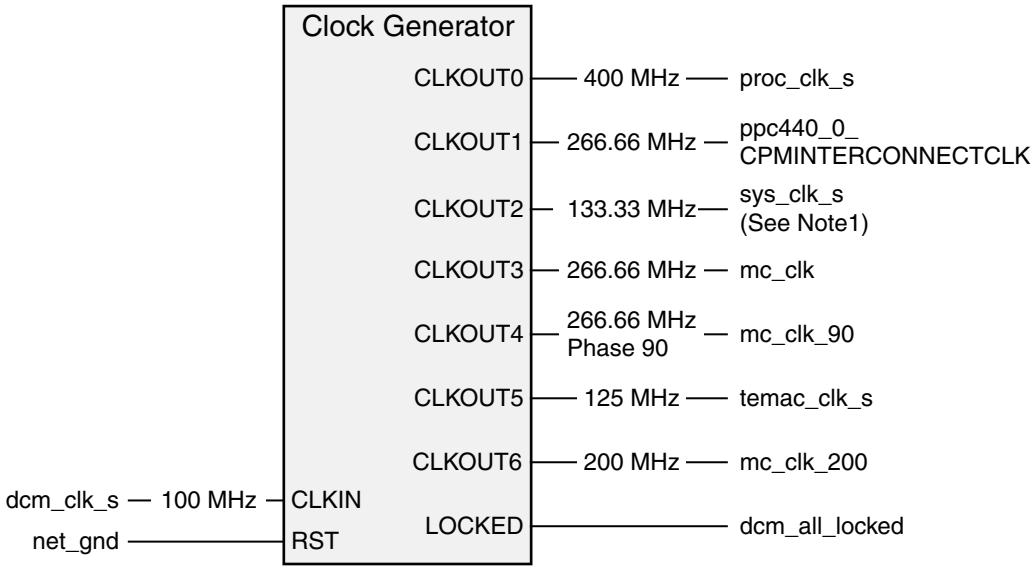

Figure 5: **PowerPC 440 Processor System Level Clocking Scheme**

PowerPC 440 System Clocking for the Included PowerPC 440 Processor Reference System

In order to obtain the highest data throughput for the PowerPC 440 processor reference system, the clock frequencies shown in Table 6 are defined. The PowerPC 440 processor can run at 400 MHz for the -1 device on the ML507 board. Refer to the Virtex-5 FPGA Data Sheet, DS202, for further details concerning the PowerPC 440 processor clocking.

Table 5: **PowerPC 440 Processor Clocking**

Clock Signal	Type	Frequency	Description
CPMC440CLK	I	400 MHz	CPU clock
CPMINTERCONNECTCLK	I	266.66 Mhz	Xbar clock
CPMPPCMPLBCLK	I	133.33 MHz	Master PLBv46 clock (connected to slave devices)
CPMPPCMPLBCLK	I	133.33 MHz	Slave PLBv46 clock (no Master PLB devices in this design)
CPMMCCLK	I	266.66 MHz	Memory interface clock
CPMDMA0LLCLK	I	133.33 MHz	DMA0 LocaLink clock

PPC440MC DDR2 Clocking for the Included PowerPC 440 Processor Reference System

In order to obtain the highest data throughput for the PowerPC 440 processor reference system through the PPC440MC DDR2 memory controller, the clock frequencies shown in Table 6 are defined. The PPC440MC DDR2 can run at an integer multiple of the PLBv46 bus frequency and it can run at 266.66 MHz for the -1 device on the ML507 board, so 266.66 MHz was chosen to optimize the throughput to the DDR2. Refer to DS567 for maximum clock speed of designs using the DDR2 Memory Controller for PowerPC 440 Processors.

Table 6: **PPC440MC DDR2 Clocking**

Clock Signal	Type	Frequency	Description
MI_MCCLK (mc_mibclk)	I	266.66 MHz	System input clock
MI_MCCLK90	I	266.66 Mhz	System input clock phase shifted by 90 degrees
MI_MCCLKDIV2	I	133.33 MHz	System input clock divided by 2
MI_MCCLK_200	I	200 MHz	IDELAY reference clock
SPLB_Clk	I	133.33 MHz	PLBv46 clock

XPS_LL_TEMAC Clocking for the Included PowerPC 440 Processor Reference System

This section shows the GMII clock management scheme for Virtex-5. Only the clocks relevant to the GMII interface will be discussed here, as shown in Table 7. Details for clocking schemes for interfaces other than GMII can be found in the XPS_LL_TEMAC data sheet.

Table 7: **XPS_LL_TEMAC Clocking**

Clock Signal	Type	Frequency	Description
GTX_CLK	I	125 MHz	Gigabit TX Clock. Input clock on global clock routing used to derive all other clocks for GMII PHY mode
LlinkTemac0_CLK	I	133.33 MHz	TEMAC LocalLink clock. Connect to the SDMA2_Clk on MPMC.
SPLB_Clk	I	133.33 MHz	PLBv46 clock
REFCLK	I	200 MHz	For signal delay primitives (IDELAY Controllers) for GMII PHY mode
GMII_TX_CLK	O	125 MHz or 1000BASE-T	TEMAC to PHY transmit clock, used for Gigabit speed (GMII) *Supplied by TEMAC core
GMII_RX_CLK	I	125 MHz	PHY to TEMAC receive clock *Supplied by PHY
MII_TX_CLK	I	25 MHz for 100BASE-T and 2.5 MHz for 10BASE-T	TEMAC to PHY transmit clock, used for 10/100 speeds (MII) *Supplied by PHY
MII_RX_CLK	I	n/a	Not used with the Marvell PHY, GMII_RX_CLK supplies all receive clocks

Clocking constraints for the PowerPC 440 processor system can be found in the system constraints file, `system.ucf`, for this reference system.

When the GMII interface is selected with parameters for the XPS_LL_TEMAC, a GMII/MII PHY interface is used which is capable of all three Ethernet speeds (10 Mbps/100 Mbps/1000 Mbps).

An example clock management diagram for GMII for the Virtex-5 hard TEMAC core implementation, taken from the XPS_LL_TEMAC data sheet, can be seen in Figure 6. The PowerPC 440 processor reference system uses C_INCLUDE_IO = 1. This makes the XPS_LL_TEMAC easier to use by allowing the core to include BUFG, IBUFG, IBUF, OBUF, and other FPGA resources to correctly connect the external interface signals to the FPGA I/O.

Note: These resources can be turned off for custom applications by setting the C_INCLUDE_IO parameter to "0". Refer to the XPS_LL_TEMAC data sheet for details about which resources are removed.

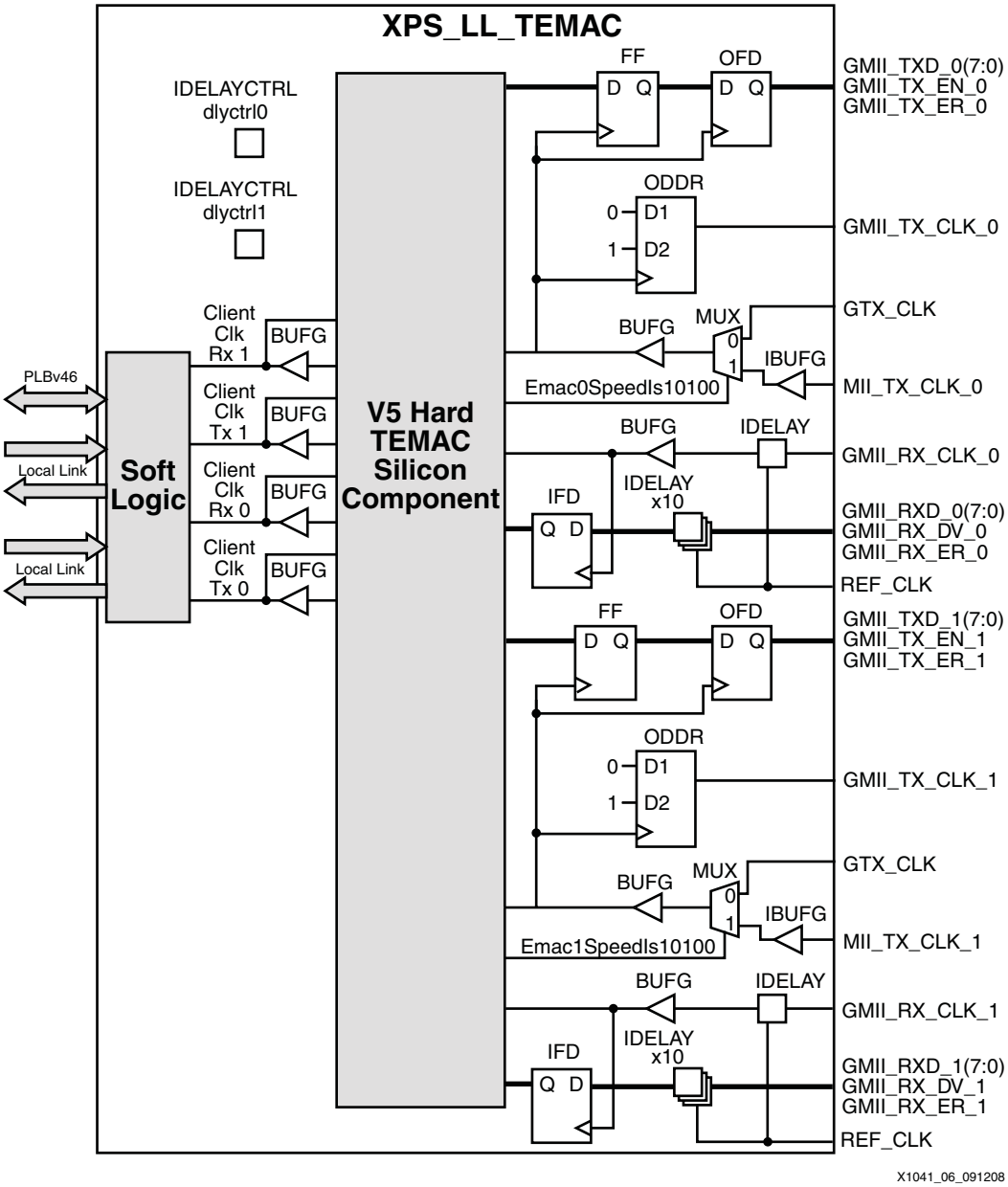

Figure 6: **Virtex-5 GMII Clock Management for C_INCLUDE_IO = "1"**

Clocking Structure for the Included ML405 PowerPC 405 Processor System

System Level Clocking for the Included PowerPC 405 Processor Reference System

The ML405 board has a slow speed grade Virtex-4 device (-10), so the fastest that the processor clock can run is 350 MHz. The supported F_{MAX} for the MIG PHY for Virtex-4 in this speed grade part is 175 MHz. The MPMC clock and the PowerPC 405 processor clock need to be an integral multiple of the PLBv46 clock. With these limitations, for the slow speed grade part (-10), the system clock frequencies shown in Table 8 were chosen to obtain the highest possible raw Ethernet performance for this system. These frequencies are available for the PowerPC 405 processor system on the ML405 board.

Table 8: **PowerPC 405 Processor System Clock Frequencies**

Clock	Enhanced System Frequency	BSB System Frequency
PLBv46 Bus Clock	100 MHz	100 MHz
MPMC/DDR Clock	100 Mhz	100 Mhz
PowerPC 4405 Processor Clock	300 MHz	300 MHz

The PowerPC 405 processor reference system included with this application note uses the Clock Generator core (v2.01.a) to provide the system clocking. BSB automatically created this clocking scheme, as shown in Table 8 and Figure 7. The Clock Generator provides automatic instantiation of the DCMs and connections. The DCMs no longer need to be instantiated directly into the system MHS file, as they previously were, except in special circumstances, such as in the higher performance PowerPC 405 processor reference system that will be described in the next section. Automatic BUFG insertion and reset sequence determination are also provided by the Clock Generator. The Clock Generator provides a grouping function to force specified clocks to be generated using the same DCM to minimize clock skew. This grouping can be seen by viewing the parameters in the Clock Generator instance in the MHS file or the Clock Generator Wizard in XPS. See DS614 for details regarding the Clock Generator.

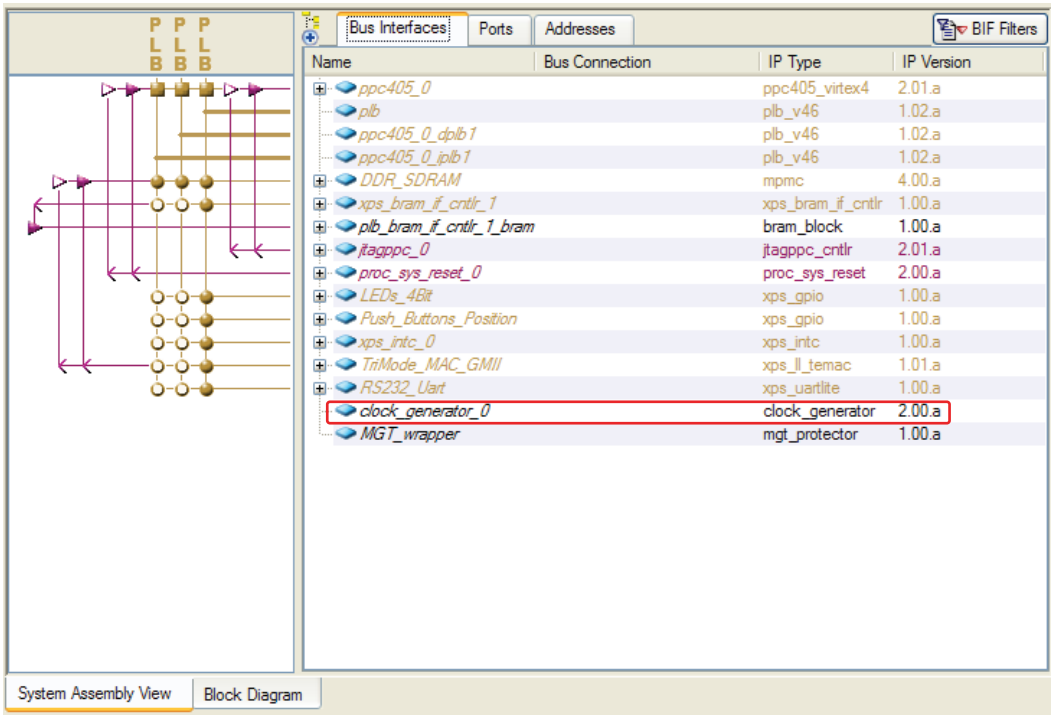

Figure 7: **EDK System Assembly View Showing Clock Generator**

A block diagram of the PowerPC 405 processor system clocking scheme, using the Clock Generator, is shown in Figure 8.

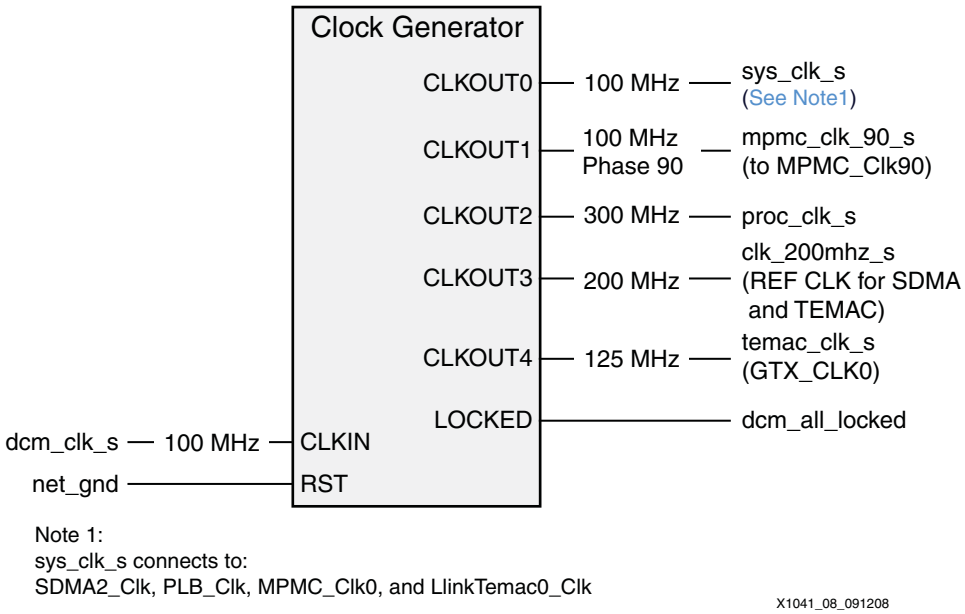

Figure 8: **PowerPC 405 Processor System Level Clocking Scheme**

MPMC Clocking for the Included PowerPC 405 Processor Reference System

In order to obtain the highest data throughput for the PowerPC 405 processor reference system through the MPMC, the clock frequencies shown in Table 9 are defined. The MPMC can run at an integer multiple of the PLBv46 bus frequency but 200 MHz is beyond the data sheet specification for this clock input so the bus clock frequency of 100 MHz was chosen.

Table 9: **MPMC Clocking**

Clock Signal	Type	Frequency	Description
MPMC_Clk0	I	100 MHz	System input clock
MPMC_Clk90	I	100 Mhz	System input clock phase shifted by 90 degrees
SDMA2_Clk	I	100 MHz	LocalLink clock. Connect to the LlinkTemac0_CLK on xps_ll_temac.
SPLB_Clk	I	100 MHz	PLBv46 clock
MPMC_Clk_200MHz	I	200 MHz	For IDELAY control logic only (MIG-based Virtex-4 and Virtex-5 PHY only)

XPS_LL_TEMAC Clocking for the Included PowerPC 405 Processor Reference System

This section shows the GMII clock management scheme for Virtex-4. Only the clocks relevant to the GMII interface will be discussed here, as shown in Table 10. Details for clocking schemes for interfaces other than GMII can be found in the XPS_LL_TEMAC data sheet.

Table 10: **XPS_LL_TEMAC Clocking**

Clock Signal	Type	Frequency	Description
GTX_CLK	I	125 MHz	Gigabit TX Clock. Input clock on global clock routing used to derive all other clocks for GMII PHY mode
LlinkTemac0_CLK	I	100 MHz	TEMAC LocalLink clock. Connect to the SDMA2_Clk on MPMC.
SPLB_Clk	I	100 MHz	PLBv46 clock
REFCLK	I	200 MHz	For signal delay primitives (IDELAY Controllers) for GMII PHY mode
GMII_TX_CLK	O	125 MHz or 1000BASE-T	TEMAC to PHY transmit clock, used for Gigabit speed (GMII) *Supplied by TEMAC core
GMII_RX_CLK	I	125 MHz	PHY to TEMAC receive clock *Supplied by PHY
MII_TX_CLK	I	25 MHz for 100BASE-T and 2.5 MHz for 10BASE-T	TEMAC to PHY transmit clock, used for 10/100 speeds (MII) *Supplied by PHY
MII_RX_CLK	I	n/a	Not used with the Marvell PHY, GMII_RX_CLK supplies all receive clocks

Clocking constraints for the PowerPC 405 processor system can be found in the system constraints file, `system.ucf`, for this reference system.

When the GMII interface is selected with parameters for the XPS_LL_TEMAC, a GMII/MII PHY interface is used which is capable of all three Ethernet speeds (10 Mbps/100 Mbps/1000 Mbps).

An example clock management diagram for GMII for the Virtex-4 hard TEMAC core implementation, taken from the XPS_LL_TEMAC data sheet, can be seen in Figure 9. The PowerPC 405 processor reference system uses C_INCLUDE_IO = 1. This makes the XPS_LL_TEMAC easier to use by allowing the core to include BUFG, IBUFG, IBUF, OBUF, and other FPGA resources to correctly connect the external interface signals to the FPGA I/O.

Note: These resources can be turned off for custom applications by setting the C_INCLUDE_IO parameter to "0". Refer to the XPS_LL_TEMAC data sheet for details about which resources are removed.

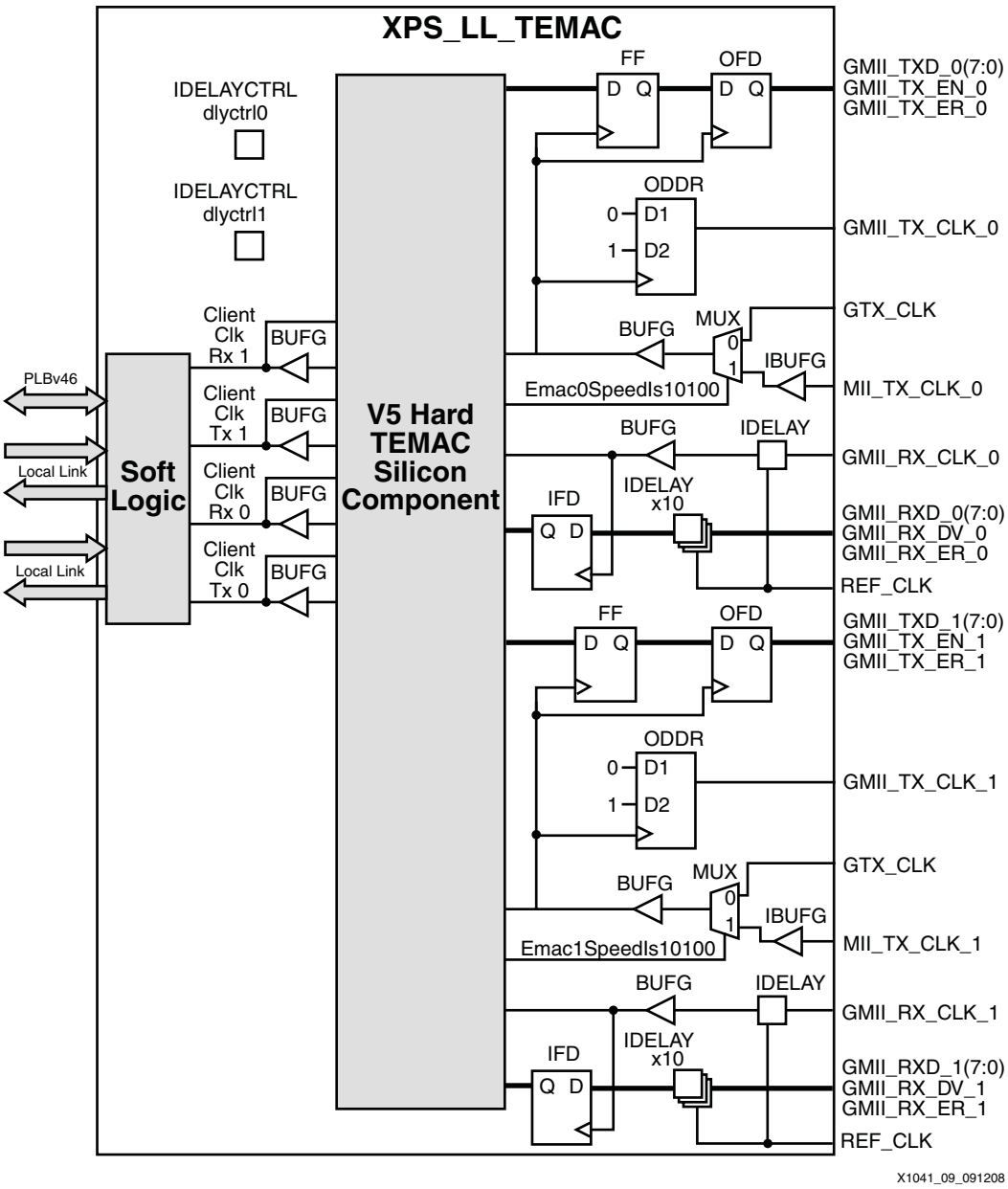

Figure 9: **Virtex-4 GMII Clock Management for C_INCLUDE_IO = "1"**

Clocking Structure for a Higher Performance PowerPC 405 Processor System

This section describes a PowerPC 405 processor system that uses a faster Virtex-4 -11 or -12 device to obtain the best possible raw Ethernet performance for a PowerPC 405 processor reference system.

Note: This higher performance PowerPC 405 processor reference system is not included with this application note because the Xilinx ML405 board only comes with the Virtex-4 -10 device which does not support the 375 MHz processor clock frequency needed.

System Level Clocking for the Higher Performance PowerPC 405 Processor System

The system clock frequencies shown in Table 11 are recommended.

Table 11: **PowerPC 405 Processor Higher Performance System Clock Frequencies**

Clock	Enhanced System Frequency	BSB System Frequency
PLBv46 Bus Clock	125 MHz	100 MHz
MPMC/DDR Clock	125 Mhz	100 Mhz
PowerPC 4405 Processor Clock	375 MHz	300 MHz

This PowerPC 405 processor reference system has manually instantiated Digital Clock Management (DCM) circuits to provide the system clocking needed. Although BSB will build most embedded systems using the Clock Generator core (v2.01.a) for system clocking, it needs to be modified to obtain the clock frequencies shown in Table 11.

Even though the Clock Generator can produce the exact PowerPC 405 processor clock frequency of 375 MHz required for this PowerPC 405 processor system, it was decided to manually instantiate the DCMs through the GUI in EDK to show the user how that is accomplished, as shown in Figure 10.

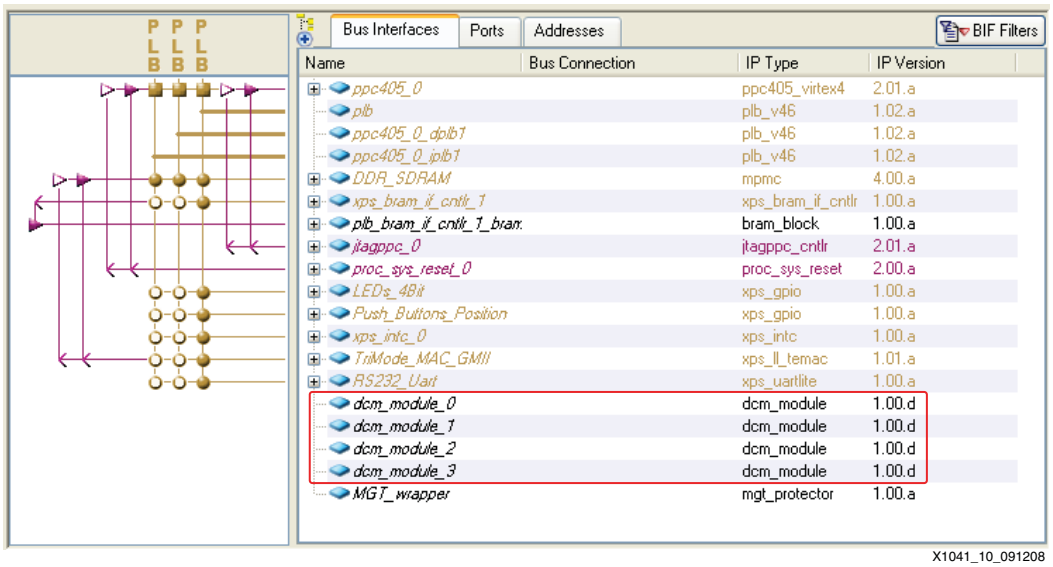

Figure 10: **EDK System Assembly View Showing DCMs**

A block diagram of the PowerPC 405 system clocking scheme, using DCMs, is shown in Figure 11.

Note: The dcm1 parameter, **C_DCM1_DLL_RFREQUENCY_MODE,** needs to be set to **HIGH** due to the frequency of the CLKIN input and the frequency of the CLK2X output.

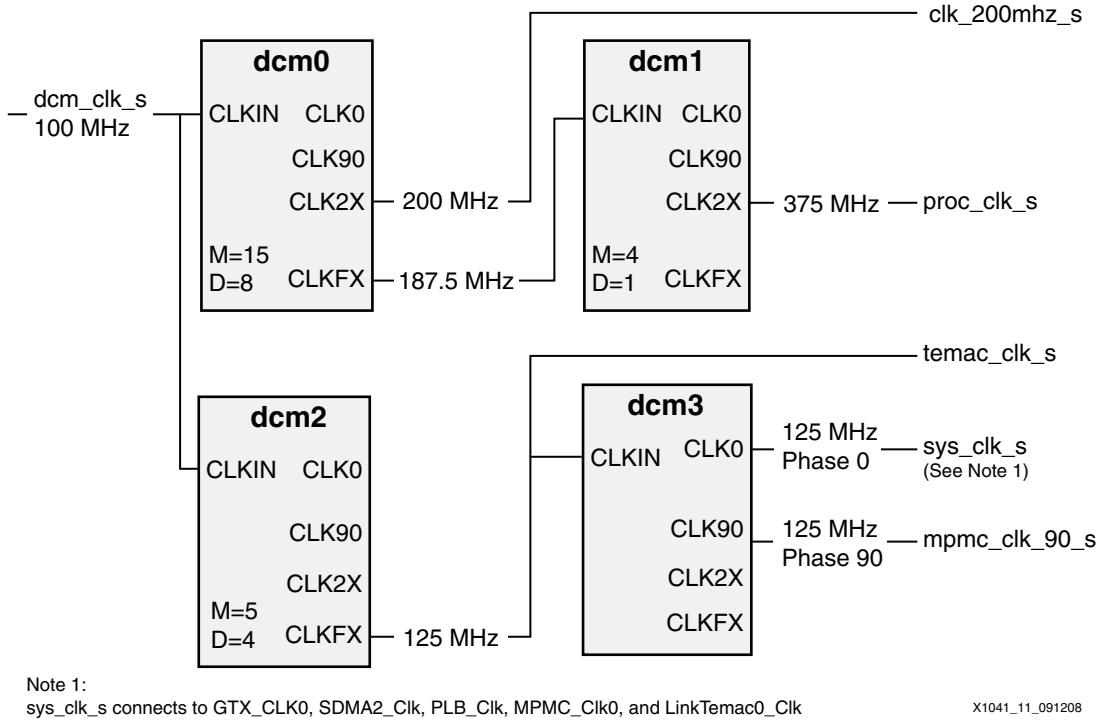

Figure 11: **PowerPC 405 Processor System Level Clocking Scheme**

MPMC Clocking for the Higher Performance PowerPC 405 Processor System

In order to obtain the highest data throughput for the PowerPC 405 processor reference system through the MPMC, the clock frequencies shown in Table 12 are defined. The MPMC can run at an integer multiple of the PLBv46 bus frequency but 250 MHz is beyond the data sheet specification for this clock input so the bus clock frequency of 125 MHz was chosen.

Table 12: **MPMC Clocking**

Clock Signal	Type	Frequency	Description
MPMC_Clk0	I	125 MHz	System input clock
MPMC_Clk90	I	125 Mhz	System input clock phase shifted by 90 degrees
SDMA2_Clk	I	125 MHz	LocalLink clock. Connect to the LlinkTemac0_CLK on xps_ll_temac.
SPLB_Clk	I	125 MHz	PLBv46 clock
MPMC_Clk_200MHz	I	200 MHz	For IDELAY control logic only (MIG-based Virtex-4 and Virtex-5 PHY only)

XPS_LL_TEMAC Clocking for the Higher Performance PowerPC 405 Processor System

This section shows the GMII clock management scheme for Virtex-4. Only the clocks relevant to the GMII interface will be discussed here, as shown in Table 13. Details for clocking schemes for interfaces other than GMII can be found in the XPS_LL_TEMAC data sheet.

Table 13: **XPS_LL_TEMAC Clocking**

Clock Signal	Type	Frequency	Description
GTX_CLK	I	125 MHz	Gigabit TX Clock. Input clock on global clock routing used to derive all other clocks for GMII PHY mode
LlinkTemac0_CLK	I	125 MHz	TEMAC LocalLink clock. Connect to the SDMA2_Clk on MPMC.
SPLB_Clk	I	125 MHz	PLBv46 clock
REFCLK	I	200 MHz	For signal delay primitives (IDELAY Controllers) for GMII PHY mode
GMII_TX_CLK	O	125 MHz or 1000BASE-T	TEMAC to PHY transmit clock, used for Gigabit speed (GMII) *Supplied by TEMAC core
GMII_RX_CLK	I	125 MHz	PHY to TEMAC receive clock *Supplied by PHY
MII_TX_CLK	I	25 MHz for 100BASE-T and 2.5 MHz for 10BASE-T	TEMAC to PHY transmit clock, used for 10/100 speeds (MII) *Supplied by PHY
MII_RX_CLK	I	n/a	Not used with the Marvell PHY, GMII_RX_CLK supplies all receive clocks

Clocking Structure for the Included ML505 MicroBlaze Processor System

System Level Clocking

In order to obtain the best possible raw Ethernet performance for the MicroBlaze processor reference system using the ML505 board, it was necessary to use the system clock frequencies shown in Table 14. These frequencies are available for the MicroBlaze processor system on the ML505 board.

Table 14: **MicroBlaze Processor System Clock Frequencies**

Clock	Enhanced System Frequency	BSB System Frequency
PLBv46 Bus Clock	125 MHz	125 MHz
MPMC/DDR2 Clock	125 Mhz	125 Mhz
MicroBlaze Processor Clock	125 MHz	125 MHz

The MicroBlaze processor reference system uses the Clock Generator core (v2.01.a) to provide the system clocking. BSB automatically created this clocking scheme, as shown in Table 14 and Figure 12. The Clock Generator provides automatic instantiation of the DCMs and connections. The DCMs no longer need to be instantiated directly into the system MHS file. Automatic BUFG insertion and reset sequence determination are also provided by the Clock Generator. The Clock Generator provides a grouping function to force specified clocks to be generated using the same DCM to minimize clock skew. This grouping can be seen by viewing the parameters in the Clock Generator instance in the MHS file or the Clock Generator Wizard in XPS. Refer to DS614 for further details regarding the Clock Generator.

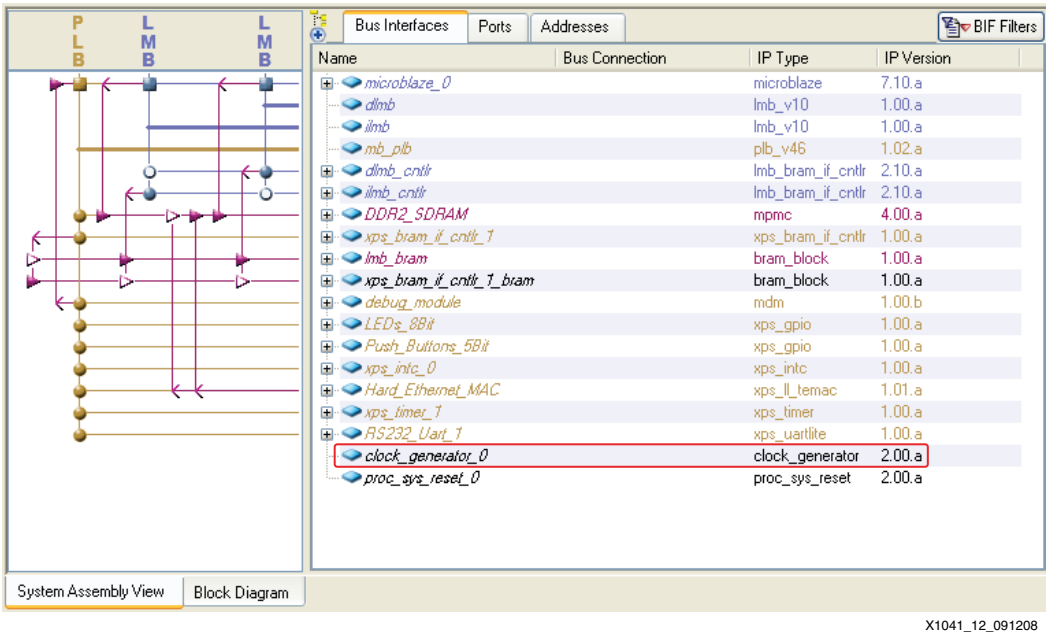

Figure 12: **EDK System Assembly View Showing Clock Generator**

A block diagram of the MicroBlaze processor system clocking scheme, using the Clock Generator, is shown in Figure 13.

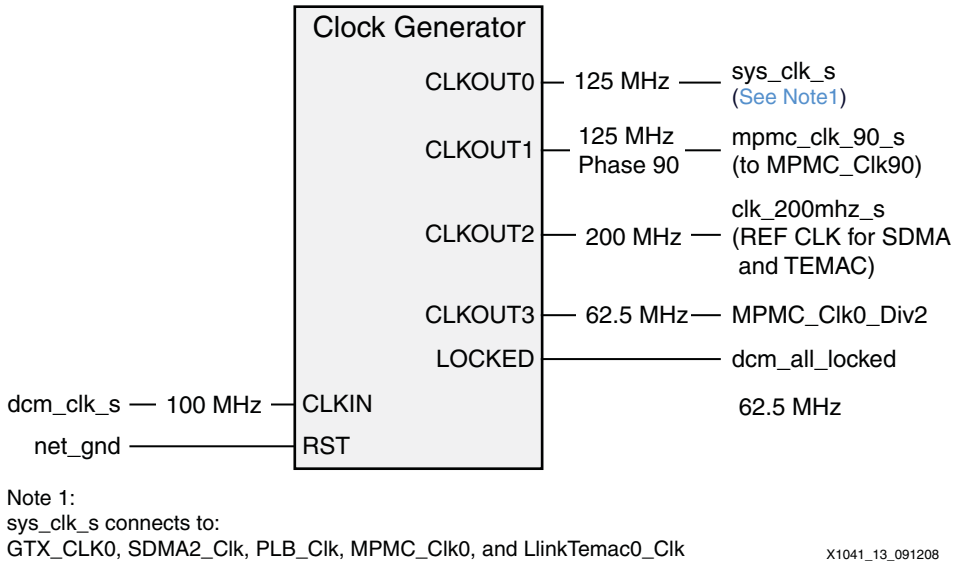

Figure 13: **MicroBlaze Processor System Level Clocking Scheme**

MPMC Clocking for MicroBlaze Processor System

In order to obtain the highest data throughput for the MicroBlaze processor reference system through the MPMC, the clock frequencies shown in Table 15 are defined. The MPMC can run at an integer multiple of the PLBv46 bus frequency but 250 MHz is beyond the data sheet specification for this clock input so the bus clock frequency of 125 MHz was chosen.

Table 15: **MPMC Clocking**

Clock Signal	Type	Frequency	Description
MPMC_Clk0	I	125 MHz	System input clock
MPMC_Clk90	I	125 Mhz	System input clock phase shifted by 90 degrees
SDMA2_Clk	I	125 MHz	This is the LocalLink clock. Connect to the LlinkTemac0_CLK on xps_ll_temac.
SPLB_Clk	I	125 MHz	PLBv46 clock
MPMC_Clk_200MHz	I	200 MHz	For IDELAY control logic only (MIG-based Virtex-4 and Virtex-5 PHY only)
MPMC_Clk0_DIV2	I	62.5 MHz	MPMC_clk0 divided by 2; used in MIG-based Virtex-5 DDR2 PHY only; new for MPMC v4.00.a

XPS_LL_TEMAC Clocking

This section shows the GMII clock management scheme for Virtex-5. Only the clocks relevant to the GMII interface will be discussed here, as shown in Table 16. Details for clocking schemes for interfaces other than GMII can be found in the XPS_LL_TEMAC data sheet.

Table 16: **XPS_LL_TEMAC Clocking**

Clock Signal	Type	Frequency	Description
GTX_CLK	I	125 MHz	Gigabit TX Clock. Input clock on global clock routing used to derive all other clocks for GMII PHY mode
LlinkTemac0_CLK	I	125 MHz	TEMAC LocalLink clock. Needs to be the same clock as connected to the SDMA2_Clk on MPMC.
SPLB_Clk	I	125 MHz	PLBv46 clock
REFCLK	I	200 MHz	For signal delay primitives (IDELAY Controllers) for GMII/MII PHY mode
GMII_TX_CLK	O	125 MHz or 1000BASE-T	TEMAC to PHY transmit clock, used for Gigabit speed (GMII) *Supplied by TEMAC core
GMII_RX_CLK	I	125 MHz for 1000BASE-T, 25 MHz for 100BASE-T and 2.5 MHz for 10BASE-T	PHY to TEMAC receive clock, used for GMII *Supplied by PHY
MII_TX_CLK	I	25 MHz for 100BASE-T and 2.5 MHz for 10BASE-T	TEMAC to PHY transmit clock, used for 10/100 speeds (MII) *Supplied by PHY
MII_RX_CLK	I	n/a	Not used with the Marvell PHY, GMII_RX_CLK supplies all receive clocks

Clocking constraints for the MicroBlaze processor system can be found in the system constraints file, `system.ucf`, for this reference system.

When the GMII interface is selected with parameters for the XPS_LL_TEMAC, a GMII / MII interface is used which is capable of all three Ethernet speeds (10/100/1000). An example clock management diagram for GMII for the Virtex-5 hard TEMAC core implementation, taken from the XPS_LL_TEMAC data sheet, can be seen in Figure 14. The MicroBlaze processor reference system uses C_INCLUDE_IO = 1. This makes the XPS_LL_TEMAC easy to use by allowing the core to include BUFG, IBUFG, IBUF, OBUF, and other FPGA resources to correctly connect the external interface signals to the FPGA I/O.

Note: These resources can be turned off for custom applications by setting the C_INCLUDE_IO parameter to "0". Refer to the XPS_LL_TEMAC data sheet for details about which resources are removed.

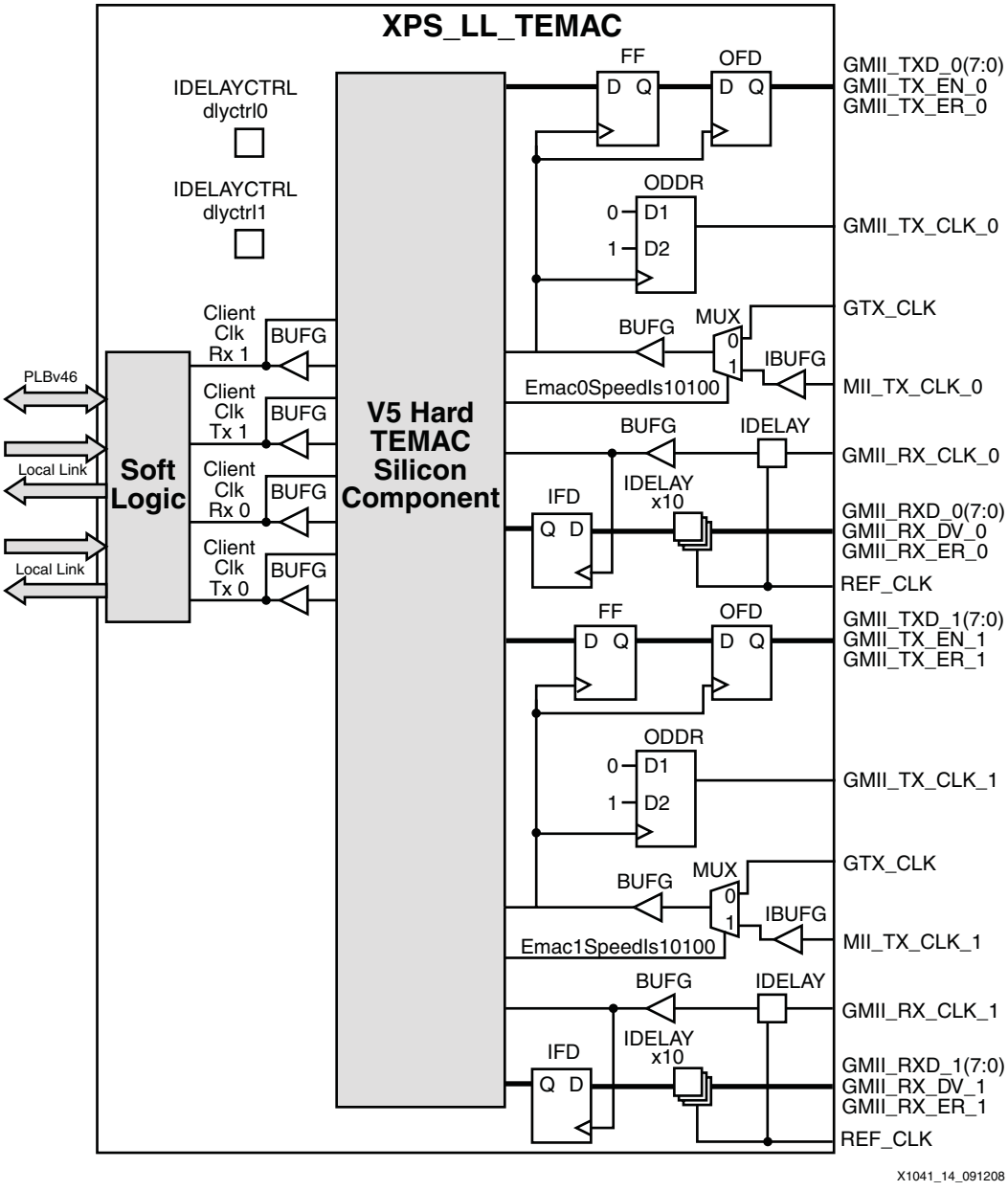

Figure 14: **Virtex-5 GMII Clock Management for C_INCLUDE_IO = "1"**

Overview of the XPS_LL_TEMAC Core

The XPS_LL_TEMAC provides a standard PLBv46 bus interface for simple connection to PowerPC or MicroBlaze processor cores to allow control and status access to internal registers. The 32-bit PLBv46 slave interface supports single beat read and write data transfers (no burst transfers).

Two Xilinx LocalLink 32-bit buses are provided for moving transmit and receive Ethernet data independently to and from the XPS_LL_TEMAC and external memory. The LocalLink buses are designed to provide support for TCP/UDP partial checksum offload in hardware, if that function is required.

Independent configurable TX and RX data FIFOs for queueing frames are incorporated into the core. FIFOs can be configured to be as large as 32K bytes to hold multiple frames at once as well as jumbo frames. There is optional support in the core for jumbo frames up to 9000 bytes.

The core is based on the Xilinx hard silicon Ethernet MAC in the Virtex-5 FXT, Virtex-5 LXT, and Virtex-4 FX devices. The core also provides a soft Ethernet option for these and other supported devices, such as Spartan® FPGAs. The parameter C_TEMAC_TYPE must be set appropriately for each option.

The core can support one or two full duplex Ethernet bus interfaces with a shared control interface and independent data and interrupt interfaces. Only full duplex operation is supported in the XPS_LL_TEMAC.

The XPS_LL_TEMAC provides one or two Ethernet interfaces. If two Ethernet interfaces are selected they are completely independent except that they must use the same type of PHY interface.

Each processor block with hard Ethernet controllers actually includes two Ethernet MACs and they share a unified host interface. The host interface for the XPS_LL_TEMAC in this reference system is controlled through the PLBv4.6 rather than the DCR. The implementation of the Host Interface varies depending on the core implementation (V4 hard, V5 hard, or soft core). The user of this core only needs to understand that all of the registers are accessed via the PLBv46. The registers are either directly accessed or, in the case of some of the Hard core registers and the PHY registers, they are indirectly accessed via the PLBv46. Access to the external PHY registers is provided via the MII Management bus. Refer to the XPS_LL_TEMAC data sheet for a complete description of how to access the registers.

The core optionally includes logic to help calculate TCP/UDP checksums for transmit and verify TCP/UDP checksums for receive. Using this logic can significantly increase the maximum Ethernet bus data rate because processor utilization is reduced. The checksum information is included with each Ethernet frame passing over the LocalLink bus interface. On transmit, the location where the checksum should be inserted is identified. On receive, the checksum calculated by the XPS_LL_TEMAC, which includes the “entire packet data”, is sent. These are just overhead bytes for the LocalLink interface.

This core provides FIFO buffering of transmit and receive Ethernet frames allowing more optimal transfer to and from the core with DMA. The number of frames that can be buffered in each direction is based on the size of each frame and the size of the FIFOs.

Support for many PHY interfaces is included but it varies on the TEMAC type selected. Refer to the XPS_LL_TEMAC data sheet for more details on which PHY types are supported for each TEMAC type. More PHY types will be supported in future upgrades to the core.

There is a Media Independent Interface Management (MIIM) port to indirectly access PHY registers for each Ethernet interface.

For reference, the XPS_LL_TEMAC block diagram is shown in Figure 15 highlighting each functional section of the core. The empty boxes shown in the block diagram contain control logic and packet FIFOs for both the Tx and Rx for the Ethernet Interfaces.

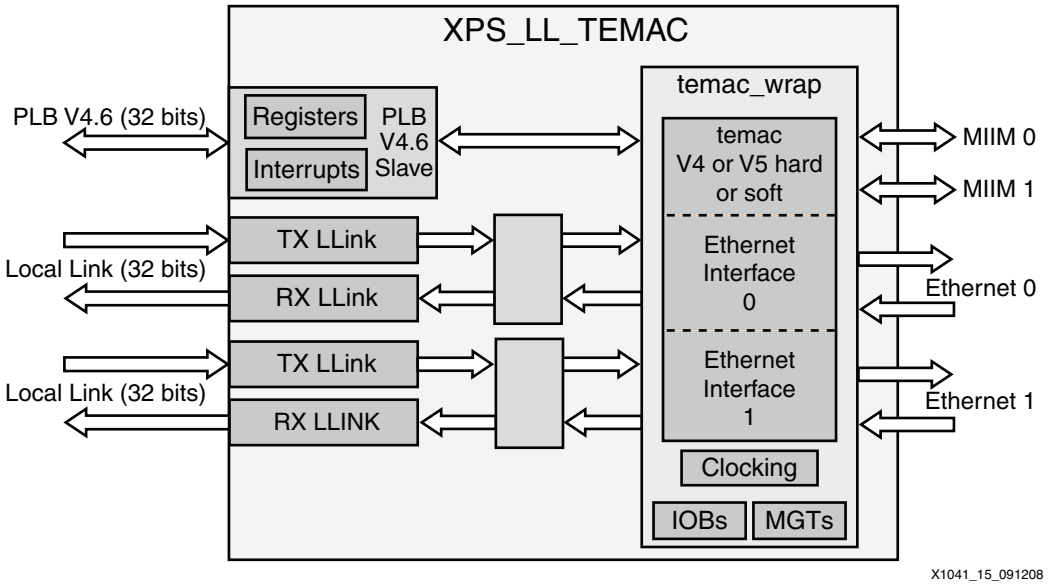

Figure 15: **XPS_LL_TEMAC Block Diagram**

Configuring the XPS_LL_TEMAC Core

All three reference systems configure the XPS_LL_TEMAC core to use Scatter/Gather DMA mode and enable Checksum Offloading on both the receive and transmit channels.

These reference systems use only one XPS_LL_TEMAC instance, even though the core allows for connecting up two instances of the XPS_LL_TEMAC.

The PowerPC 440, PowerPC 405, and MicroBlaze processor reference systems have receive and transmit FIFOs set to 16 KB in order to handle Jumbo frames (up to 9000 bytes).

The included reference systems utilize C_PHY_TYPE = 1, which is GMII/MII.

As an example, Figure 16 shows the settings of the XPS_LL_TEMAC parameters for the `ml405_ppc_xps_ll_temac` reference system included.

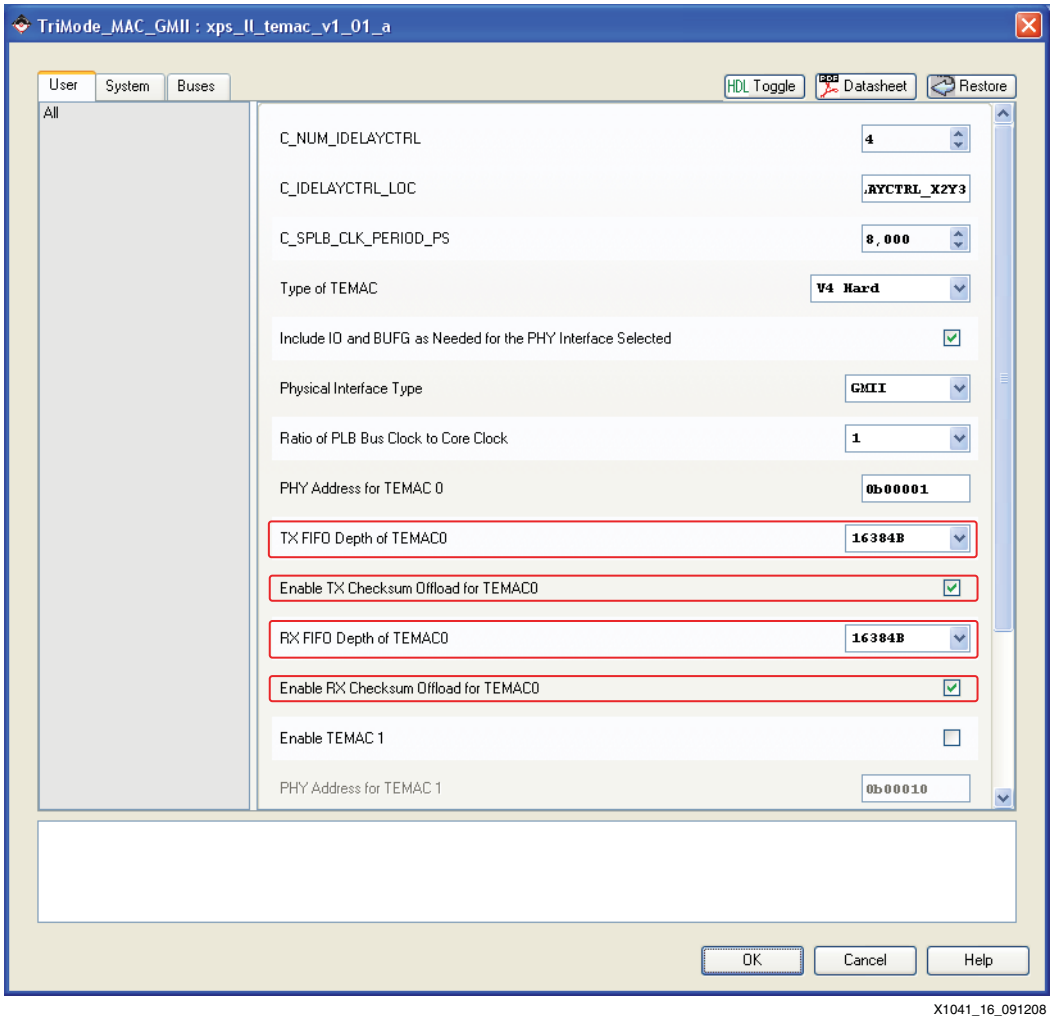

Figure 16: **XPS_LL_TEMAC Parameters for PowerPC 405 Processor Reference System**

Standalone PerfApp Application

Overview of the Standalone PerfApp Application

The intent of this software application is to enable users to run a raw ethernet performance test suite on an embedded PowerPC 440, PowerPC 405, or MicroBlaze processor system. The PerfApp test suite provides a menu interface to select which test to execute, gathers metrics, and displays the results. There are several tests that can be run by the user to obtain raw performance results for receive or transmit.

Each reference system comes with the standalone PerfApp software application, modified for that system. This application can measure raw Ethernet throughput (receive or transmit) and also transmit out whatever data was received by swapping the source and destination MAC addresses in the packet data and retransmitting the data (echo server option).

The application runs a suite of tests for measuring raw Ethernet performance on a system with the XPS_LL_TEMAC IP core. The version of the software application included with each reference system has been updated to work with the XPS_LL_TEMAC driver functions. It does not work with other Xilinx Ethernet cores without modification.

This application does not use an operating system. It runs under Standalone BSP.

The receive and transmit channels use 128 DMA buffer descriptors each. The DMA buffer descriptors and the data are contained in external memory (DDR or DDR2, depending on the reference system).

Due to the large size of the PerfApp application, it does not fit in BRAM, so it needs to be executed out of external memory.

Note: For XPS_LL_TEMAC the DMA buffer descriptors cannot be put into BRAM.

The application included with each system is currently set up to run with the Marvell Alaska 88E1111 PHY, which is the PHY used on the ML405, ML505, and ML507 boards. The application could work with other PHY devices if the source files were modified to accommodate a particular PHY.

Note: The FIFO depth needs to be large enough to hold at least one full frame. The software does not check Rx and Tx FIFO depths to see if Jumbo frames can be handled, so the user needs to be aware that 16K depths are needed to support Jumbo frames (9000 bytes). The FIFO depths are parametric settings for the XPS_LL_TEMAC and can only be changed at build time.

Even though the Rx and Tx Checksum Offload are enabled for these reference systems, this functionality does not affect the raw Ethernet performance. The checksum calculation and verification are only used for TCP/UDP packets.

Most of the tests in the test suite have the capability to change the DMA interrupt coalescing parameters. With DMA interrupt coalescing, multiple packets can be queued up before sending an interrupt, in order to reduce processor overhead. Interrupt coalescing allows for increased performance by decreasing CPU overhead, but does affect the TCP/IP performance slightly. Instead of servicing every DMA packet-complete interrupt, interrupt coalescing collects a set number of packets before sending a single interrupt. The interrupt handler then walks the completed packet DMA Ring moving the buffers and descriptors from complete to available for future use. This test parameter is called the **`Rx/Tx Packet Threshold Count`** in the PerfApp application.

The packet waitbound timer is used to interrupt the CPU if after a programmable amount of time after processing the last packet, no new packets were processed. At which point the completed buffers are moved to the available state for future use. This processing is similar to the packet count processing except that the interrupt source was the time-out of the waitbound timer. The time unit is equal to *1024 times the LocalLink clock period*. This test parameter is called the **`Rx/Tx Waitbound Timer`**. The waitbound timer is especially useful on the Rx side because of the asynchronous nature of receiving packets.

Without Interrupt Coalescing, the XPS_LL_Temac would send a packet-complete interrupt for each packet sent or received, which significantly slows down Ethernet performance due to the processor spending many of its cycles on servicing interrupts.

Performance Definitions

CPU Utilization

CPU Utilization is calculated as the percentage of time the processor is not ideal. Under heavy network loads, we expect CPU Utilization to increase.

Link Utilization (Network Utilization)

Link Utilization Percentage is calculated as a percentage of the maximum effective rate over a period of time.

Util % = A / MER

In the above equation, *A* = Actual observed rate in Mbps and *MER* = Maximum Effective Rate in Mbps.

For example, on a 1000 Mbit link observed transferring 64 byte packets at a rate of 620 Mbps, the link utilization would be calculated as: 620 / 761.9 = 81.4%.

The goal for maximum performance is a link utilization of 100%.

Maximum Effective Rate

The maximum effective rate describes the maximum transfer rate of non-overhead data for a particular packet size. Although it is natural to expect a 1 Gbps link to have a 1 Gbps effective data rate, the overhead required by Ethernet reduces the maximum possible effective rate. Every packet on the link has an additional 8 bytes of overhead for the preamble and start-of-stream delimiter (SSD), plus an inter-frame gap (IFG), which is typically 96 bit times. For small packets, this is significant overhead and lowers the maximum effective rate. For large packets this overhead is less significant resulting in higher maximum effect rates.

For example, the number of bits required to send a 512 bit (64-byte) packet is:

*(512) + (8 bytes overhead * 8) + (96 bit IFG) = 672 bits.*

The percentage of time spent transferring overhead is:

(672 - 512) / 672 = 23.809%.

The remainder of the time, 76.19%, is used to transfer actual packet data. This percentage forms the basis of the maximum effective rate. Table 17 lists various maximum effective rates for different link speeds.

Table 17: **Maximum Effective Rates**

Packet Size (bytes)	10 Mbit Link (Mbps)	100 Mbit Link (Mbps)	1000 Mbit Link (Mbps)
64	7.619	76.19	761.9
128	8.611	86.11	861.1
512	9.624	96.24	962.4
1518	9.870	98.70	987.0
9000	9.978	99.78	997.8

PerfApp Test Flow

The following describes the basic standalone PerfApp application flow. The application performs these steps automatically and these are not steps that the user needs to take.

- Initialization Section
  - Set up the UART console
  - Set the MAC address to ‘00 0A 35 01 8D 36’
  - Set up and enable the caches
  - Set up the Interrupt Controller and enable interrupts
  - If a MicroBlaze processor system, set up the XPS Timer (uses 2 timers)

Note: The MAC address is defined in the source code and is a random address chosen for testing. It is advised not to put the board under test on a live corporate network.

  - MAC set up
    - PHY type capabilities (GMII for these reference systems)
    - Link speed and Duplex mode (default is 1000 Mbps/Full Duplex)
    - Set up the Rx and Tx Buffer Descriptor spaces and Frame spaces (memory requirements for packet data storage)

- Main Menu Section
- User has option to change Link speed (cannot change Duplex for XPS_LL_TEMAC, always Full Duplex)
- User selects Test to be run
- Application is capable of running in one of four different modes for XPS_LL_TEMAC:
  - Polled FIFO mode (requires XPS_LL_FIFO IP core, not in these systems)
  - FIFO mode- Interrupt driven (requires XPS_LL_FIFO IP core, not in these systems)
  - Simple DMA (not supported in XPS_LL_TEMAC IP core)
  - Scatter/Gather (SG)DMA mode (this is the only supported setup for the included reference systems)

Note: Only those Tests appear in the menu that apply to the mode of the system (for instance, DMA tests only appear if DMA is present in the system)

- The basic application flow for the Tx SG DMA test is as follows.
  - Read the parameter values in from the User selections
    - Tx frame size, interrupt coalescing settings, test run time
  - Initialize the MAC and PHY based on User settings
    - Set MAC address and PHY and MAC link speed
    - Set interrupt coalescing settings for SG DMA
  - Set frame header and frame data in memory (Tx Frame Space)
  - Create the Buffer Descriptor (BD) Ring
  - Start the test timer and then the selected test
    - Allocate all the TxBDs
    - Point all TxBDs to the frame in memory
    - Send TxBDs to hardware for processing (get the frame header and data memory and send to LocalLink)
  - At this point it runs in interrupt context: reading interrupts, acknowledging them and servicing them
    - Interrupts are serviced when waitbound timer times out or when packet threshold count is reached
    - When interrupt occurs, go to interrupt handler routine to service
  - Extract processed BDs from hardware
  - Free processed BDs (used BDs to be reused)
  - Allocate and prepare new BDs and send to hardware
  - Stop test when test run time expires

Note: Rx SG DMA and Tx Canned tests use similar flows

Standalone PerfApp Memory Mapping

Figure 17 shows an example DDR2 memory map for the PowerPC 440 processor system after PerfApp has set up the Buffer Descriptors and initialized the constants for the Scatter Gather DMA tests. The actual DDR memory address locations will vary depending on the application program, stack, and heap sizes for the system.

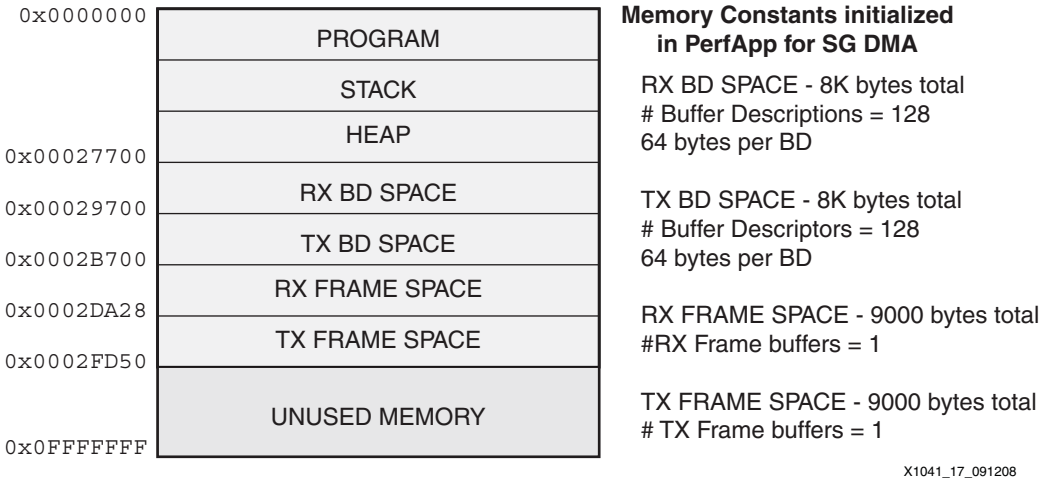

Figure 17: **PowerPC 440 Processor DDR Memory Map Initialization**

Figure 18 shows an example DDR memory map for the PowerPC 405 processor system after PerfApp has set up the Buffer Descriptors and initialized the constants for the Scatter Gather DMA tests. The actual DDR memory address locations will vary depending on the application program, stack, and heap sizes for the system.

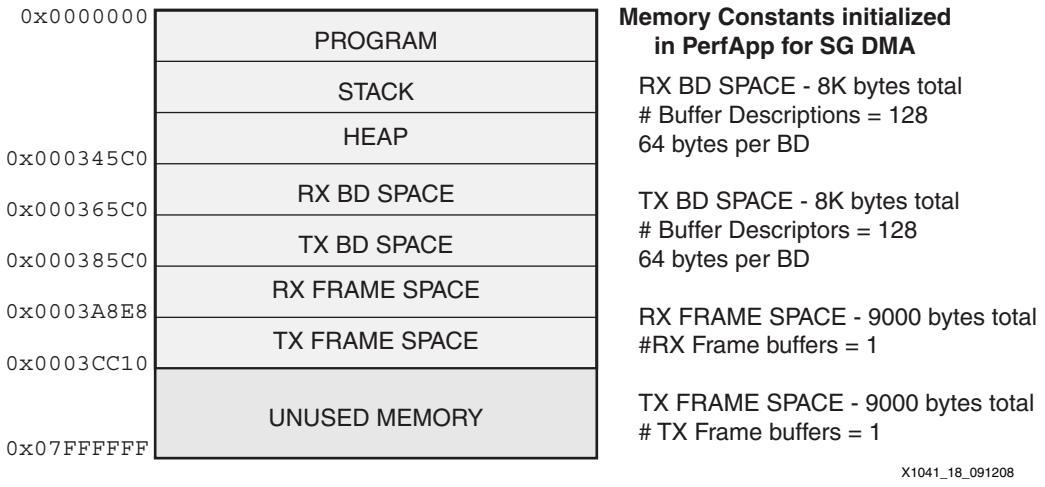

Figure 18: **PowerPC 405 Processor DDR Memory Map Initialization**

Figure 19 shows an example DDR2 memory map for the MicroBlaze processor system after PerfApp has set up the Buffer Descriptors and initialized the constants for the Scatter Gather DMA tests. The actual DDR2 memory address locations will vary depending on the application program, stack, and heap sizes for the system.

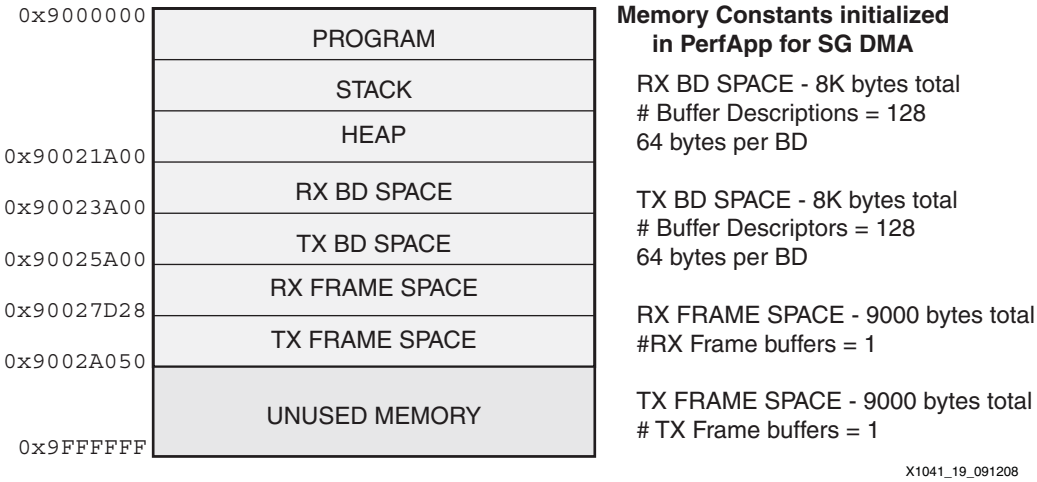

Figure 19: **MicroBlaze Processor DDR2 Memory Map Initialization**

In a real world application, each BD will have it's own unique buffer instead of sharing one like is done in this test application.

For more details on how the buffer descriptors are set up and used in a XPS_LL_TEMAC Scatter/Gather DMA system, refer to the MPMC v4.00.a data sheet, the PPC440MC DDR2 HDMA documentation, and the XPS_LL_TEMAC data sheet.

Standalone PerfApp Menu for PowerPC 440 Processor System

A typical Main Menu for the PowerPC 440 processor system is shown in Figure 20. The CPU core and PLB bus clock speeds are highligthed. These frequencies are used by the application in calculating performance data for the test selected. The Rx and Tx Buffer Descriptor spaces and Frame spaces are highlighted.

Note: The only tests available in the menu are Scatter/Gather mode tests.

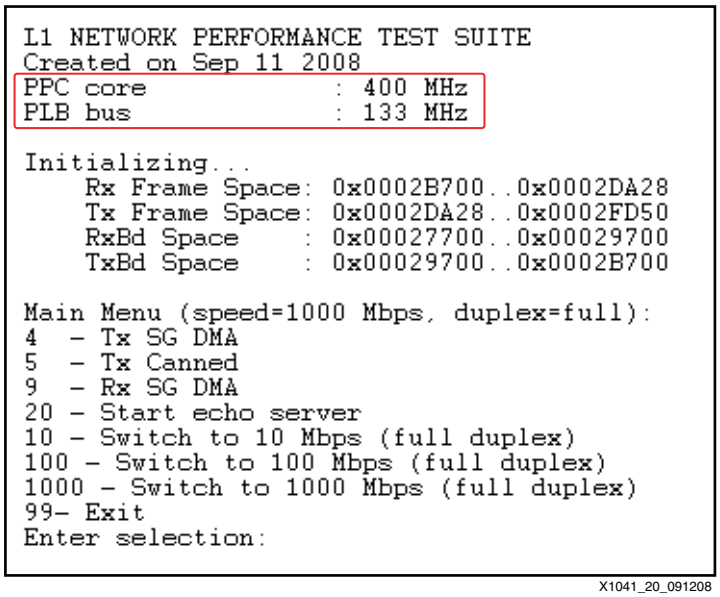

Figure 20: **PerfApp Main Menu for PowerPC 440 Processor System**

Standalone PerfApp Menu for PowerPC 405 Processor System

A typical Main Menu for the PowerPC 405 processor system is shown in Figure 21. The CPU core and PLB bus clock speeds are highligthed. These frequencies are used by the application in

calculating performance data for the test selected. The Rx and Tx Buffer Descriptor spaces and Frame spaces are highlighted.

Note: The only tests available in the menu are Scatter/Gather mode tests.

Standalone PerfApp Menu for MicroBlaze Processor System

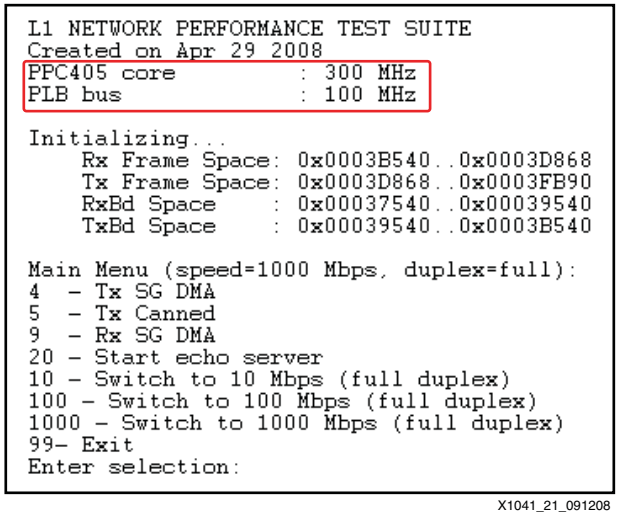

Figure 21: **PerfApp Main Menu for PowerPC 405 Processor System**

A typical Main Menu for the MicroBlaze processor system is shown in Figure 22. The CPU core and PLB bus clock speeds are highlighted. These frequencies are used by the application in calculating performance data for the test selected. The Rx and Tx Buffer Descriptor spaces and Frame spaces are highlighted.

Note: The only tests available in the menu are Scatter/Gather mode tests.

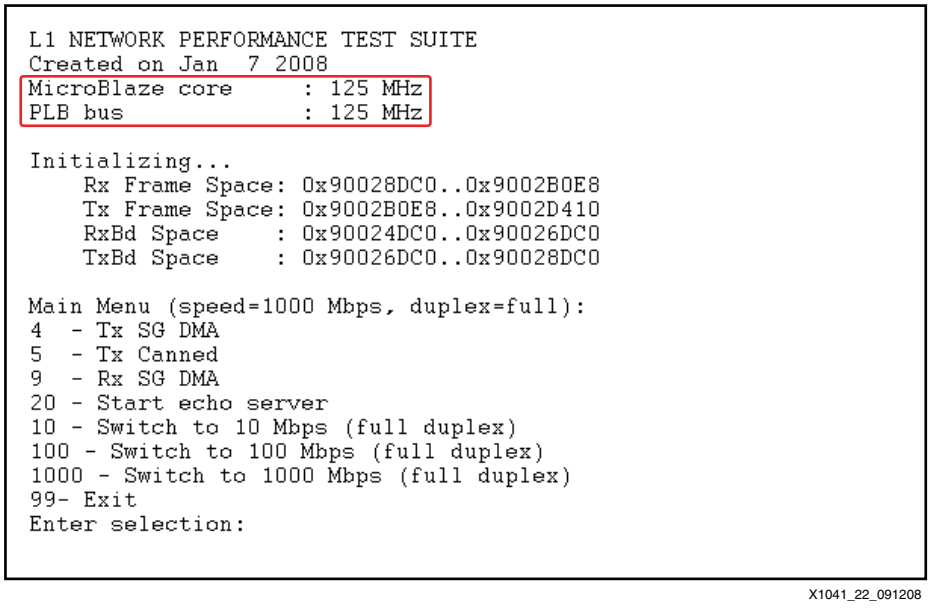

Figure 22: **PerfApp Main Menu for MicroBlaze Processor System**

Standalone PerfApp Test Suite

Transmit Scatter/Gather DMA Test

Menu selection **`Tx SG DMA`** (option #4) will transmit frames of the size specified and for the number of seconds selected. Through menu selections, configuration of the interrupt coalescing parameters are selectable via the TX/RX packet threshold count and the TX/RX waitbound timer. The following raw Ethernet performance data is output to the terminal:

- # of Frames Sent
- Throughput (in Mbps)
- Network Utilization (in percentage)
- CPU Utilization (in percentage)

See Figure 23 for an example of the test parameters, default selections, and the output of the **`Tx SG DMA`** test run for the PowerPC 440 processor reference system.

Note: The theoretical maximum line rate (Maximum Effective Rate) for a 1000 Mbps link, with a minimum packet size of 64 bytes, is 761.9 Mbps. This is due to the overhead required by Ethernet. Every packet on the link has an additional 8 bytes of overhead for the preamble and start-of-stream delimiter (SSD), plus an inter-frame gap (IFG) of 96 bits. As can be seen from the data in Figure 23, the PowerPC 440 processor system is capable of reaching the Maximum Effective Rate of 761.9 Mbps.

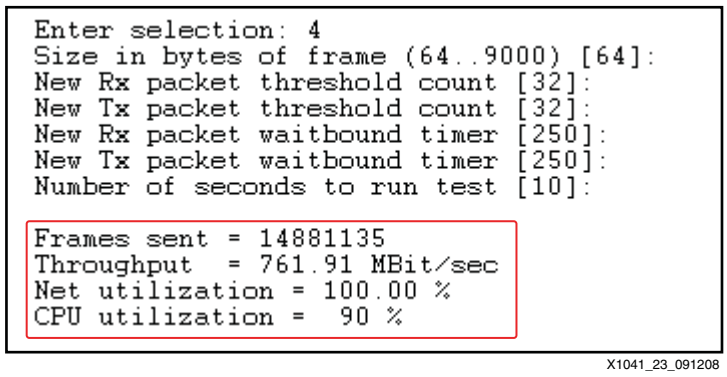

Figure 23: **PowerPC 440 Processor Performance Data for Tx SG DMA Test**

See Figure 24 for an example of the test parameters, default selections, and the output of the **`Tx SG DMA`** test run for the PowerPC 405 processor reference system.

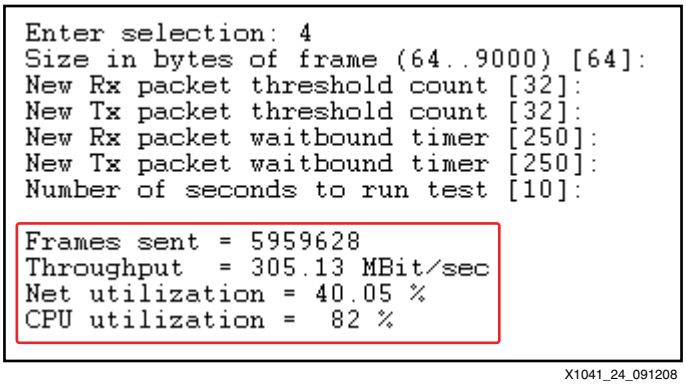

Figure 24: **PowerPC 405 Processor Performance Data for Tx SG DMA Test**

Refer to Figure 25 for an example of the test parameters, default selections, and the output of the **`Tx SG DMA`** test run for the MicroBlaze processor reference system.

```
Enter selection: 4
Size in bytes of frame (64..9000) [64]:
New Rx packet threshold count [32]:
New Tx packet threshold count [32]:
New Rx packet waitbound timer [250]:
New Tx packet waitbound timer [250]:
Number of seconds to run test [10]:

Frames sent = 366336
Throughput  = 18.75 MBit/sec
Net utilization =  2.46 %
CPU utilization = 100 %
```

X1041_25_091208

Figure 25: **MicroBlaze Processor Performance Data for Tx SG DMA Test**

Transmit Canned SGDMA Test

Menu selection **`Tx Canned`** (option #5) uses a set of pre-selected test parameters for collecting transmit raw Ethernet performance data for using Scatter/Gather DMA mode. The selected parameters cause the test to transmit various canned frame sizes, from 64 bytes up to the maximum jumbo frame size of 9000 bytes. Test data is also collected for packet threshold counts of 1, 2, 8, and 64 packets. This is a comprehensive test for collecting raw transmit Ethernet performance data over a wide range of frame sizes and packet threshold counts.

From the data, in Figure 26, it can be seen that the performance increases significantly as the frame size is increased. It is also significant to note that the packet threshold count has a dramatic effect on the CPU utilization, which is expected due to the fact that the CPU does not need to process an interrupt for each packet transmitted.

Figure 26 shows a typical **`Tx Canned`** test output for the PowerPC 440 processor reference system. This test offers a good snapshot of the transmit performance of the PowerPC 440 processor reference system.

```
                Packet Threshold settings
          +------------------+------------------+------------------+------------------+
          |        1         |        2         |        8         |       64         |
          +------------------+------------------+------------------+------------------+
Frame           Net   CPU          Net   CPU          Net   CPU          Net   CPU
 Size    Mbps Util  Util    Mbps Util  Util    Mbps Util  Util    Mbps Util  Util
 ----   ----- ----- ----   ----- ----- ----   ----- ----- ----   ----- ----- ----
   64   761.9 100.0  100   761.9 100.0  100   761.9 100.0  100   761.9 100.0   84
  128   864.9 100.0  100   864.9 100.0  100   864.9 100.0   71   864.9 100.0   47
  512   962.4 100.0   74   962.4 100.0   43   962.4 100.0   20   962.4 100.0   13
 1518   987.0 100.0   26   987.0 100.0   15   987.0 100.0    7   986.9 100.0    4
 9000   997.8 100.0    4   997.8 100.0    3   997.7 100.0    1   997.2  99.9    1
```

X1041_26_091208

Figure 26: **PowerPC 440 Processor Performance Data for Tx Canned Test**

Figure 27 shows a typical **Tx Canned** test output for the PowerPC 405 processor reference system. This test offers a good snapshot of the transmit performance of the PowerPC 405 processor reference system.

```
**************************************************************************
Interrupt Driven SG DMA Performance
**************************************************************************
                    Packet Threshold settings
                +-----------------+-----------------+-----------------+-----------------+
                |        1        |        2        |        8        |       64        |
                +-----------------+-----------------+-----------------+-----------------+
Frame           Net   CPU         Net   CPU         Net   CPU         Net   CPU
 Size    Mbps  Util  Util    Mbps  Util  Util    Mbps  Util  Util    Mbps  Util  Util
 ----    ----- ----- ----    ----- ----- ----    ----- ----- ----    ----- ----- ----
   64   304.2  39.9  100   304.0  39.9  100   304.0  39.9  100   305.1  40.0   80
  128   546.4  63.2  100   546.6  63.2  100   548.4  63.4   95   547.2  63.3   75
  512   959.2  99.7  100   959.2  99.7   70   959.2  99.7   40   959.2  99.7   32
 1518   985.7  99.9   39   985.7  99.9   24   985.7  99.9   14   985.7  99.9   11
 9000   997.5 100.0    7   997.5 100.0    4   997.5 100.0    2   997.2  99.9    2
```

X1041_27_091208

Figure 27: **PowerPC 405 Processor Performance Data for Tx Canned Test**

Figure 28 shows a typical **Tx Canned** test output for the MicroBlaze processor reference system. This test offers a good snapshot of the transmit performance of the MicroBlaze processor reference system.

```
**************************************************************************
Interrupt Driven SG DMA Performance
**************************************************************************
                    Packet Threshold settings
                +-----------------+-----------------+-----------------+-----------------+
                |        1        |        2        |        8        |       64        |
                +-----------------+-----------------+-----------------+-----------------+
Frame           Net   CPU         Net   CPU         Net   CPU         Net   CPU
 Size    Mbps  Util  Util    Mbps  Util  Util    Mbps  Util  Util    Mbps  Util  Util
 ----    ----- ----- ----    ----- ----- ----    ----- ----- ----    ----- ----- ----
   64   18.7   2.5   100   18.7   2.5   100   18.7   2.5   100   18.7   2.5   100
  128   37.4   4.3   100   37.4   4.3   100   37.4   4.3   100   37.4   4.3   100
  512   149.7  15.6  100   149.6  15.5  100   149.6  15.5  100   149.6  15.5  100
 1518   441.5  44.7  100   441.9  44.8  100   441.5  44.7  100   441.4  44.7  100
 9000   996.4  99.9  100   996.4  99.9  100   995.2  99.7   61   997.4 100.0   41
```

X1041_28_091208

Figure 28: **MicroBlaze Processor Performance Data for Tx Canned Test**

Note: These raw performance data for the Tx Canned Tests imply that more coalescing equals better performance and this is not always the case. This application does nothing with the packet data which would take time to process in a real application. For instance, if 64 packets suddenly arrived to be handled at once, it might take longer to handle them all then it would take for the next 64 packets to arrive, hence causing a bottleneck. In this case, the larger coalescing values may not be the optimum choice for throughput. For each system/application there is more likely an optimum “magic number”.

Receive Scatter/Gather DMA Test

Menu selection **Rx SG DMA** (option #9) will receive frames and print the performance results to the terminal window at 1 second time intervals until the test is aborted by typing any key on the console terminal. Configuration of the interrupt coalescing parameters are selectable for this test, as they were for the **Tx SG DMA** test.

This test reports any frames dropped or errors. The number of frames received may differ from the number of frames transmitted due to interrupt coalescing. Frames may be dropped if the received frames count is extremely large and the frame sizes are small. This scenario could lead to running out of buffer descriptors.

See Figure 29, Figure 30 and Figure 31 for examples of the test parameters, default selections, and the output of the test run for the PowerPC 440, PowerPC 405, and MicroBlaze processor system, respectively.

Figure 29 shows a typical **`Rx SG DMA`** test output for the PowerPC 440 processor reference system.

```
Main Menu (speed=1000 Mbps, duplex=full):
4   - Tx SG DMA
5   - Tx Canned
9   - Rx SG DMA
20  - Start echo server
10  - Switch to 10 Mbps (full duplex)
100 - Switch to 100 Mbps (full duplex)
1000 - Switch to 1000 Mbps (full duplex)
99- Exit
Enter selection: 9
New Rx packet threshold count [32]:
New Tx packet threshold count [32]:
New Rx packet waitbound timer [250]:
New Tx packet waitbound timer [250]:
Test runs in 1 sec intervals. Press any key to stop...

Total    Run     Frames Rx Throughput                Frames
 sec     sec      per/sec   MBit/Sec  Util%  CPU% Dropped Errors
----- -------- ---------- ---------- ------ ---- ------- ------
    0  1.0000       0.00      0.000   0.00    0       0      0
    1  1.0000       0.00      0.000   0.00    0       0      0
    2  1.0000  962307.44    492.701  64.67   65       0      0
    3  1.0000 1488040.75    761.877 100.00  100       1      0
    4  1.0000 1488042.38    761.878 100.00  100       1      0
    5  1.0000 1488041.38    761.877 100.00  100       1      0
    6  1.0000 1488040.75    761.877 100.00  100       1      0
    7  1.0000 1488040.75    761.877 100.00  100       1      0
    8  1.0000 1488040.75    761.877 100.00  100       1      0
    9  1.0000 1488041.50    761.877 100.00  100       1      0
   10  1.0000 1488320.12    762.020 100.02  100       0      0
   11  1.0000 1488040.38    761.877 100.00  100       1      0
   12  1.0000  460019.00    235.530  30.91   31       1      0
   13  1.0000       0.00      0.000   0.00    0       0      0
   14  1.0000       0.00      0.000   0.00    0       0      0
```

X1041_29_091208

Figure 29: **Rx SG DMA Test Output for PowerPC 440 Processor System**

Figure 30 shows a typical **`Rx SG DMA`** test output for the PowerPC 405 processor reference system.

```
Main Menu (speed=1000 Mbps, duplex=full):
4  - Tx SG DMA
5  - Tx Canned
9  - Rx SG DMA
20 - Start echo server
10 - Switch to 10 Mbps (full duplex)
100 - Switch to 100 Mbps (full duplex)
1000 - Switch to 1000 Mbps (full duplex)
99- Exit
Enter selection: 9
New Rx packet threshold count [32]:
New Tx packet threshold count [32]:
New Rx packet waitbound timer [250]:
New Tx packet waitbound timer [250]:
Test runs in 1 sec intervals. Press any key to stop...

Total    Run    Frames Rx Throughput              Frames
 sec     sec     per/sec   MBit/Sec  Util%  CPU% Dropped Errors
-----  -------  --------- ---------- ------ ---- ------- ------
    0  1.0000       0.00      0.000   0.00    0       0      0
    1  1.0000       0.00      0.000   0.00    0       0      0
    2  1.0000       0.00      0.000   0.00    0       0      0
    3  1.0002  593854.31    304.053  39.91   72       0      0
    4  1.0002  829528.19    424.718  55.74  100       2      0
    5  1.0001  829568.38    424.739  55.75  100       2      0
    6  1.0002  829494.69    424.701  55.74  100       2      0
    7  1.0002  829459.62    424.683  55.74  100       2      0
    8  1.0001  829541.50    424.725  55.75  100       2      0
    9  1.0001  829472.88    424.690  55.74  100       2      0
   10  1.0002  829451.75    424.679  55.74  100       2      0
   11  1.0001  829519.69    424.714  55.74  100       2      0
   12  1.0002  829419.50    424.663  55.74  100       2      0
   13  1.0000  193304.00     98.972  12.99   23       2      0
   14  1.0000       0.00      0.000   0.00    0       0      0
   15  1.0000       0.00      0.000   0.00    0       0      0
```

X1041_30_091208

Figure 30: **Rx SG DMA Test Output for PowerPC 405 Processor System**

Figure 31 shows a typical **Rx SG DMA** test output for the MicroBlaze processor reference system.

```
Main Menu (speed=1000 Mbps, duplex=full):
4  - Tx SG DMA
5  - Tx Canned
9  - Rx SG DMA
20 - Start echo server
10 - Switch to 10 Mbps (full duplex)
100 - Switch to 100 Mbps (full duplex)
1000 - Switch to 1000 Mbps (full duplex)
99- Exit
Enter selection: 9
New Rx packet threshold count [32]:
New Tx packet threshold count [32]:
New Rx packet waitbound timer [250]:
New Tx packet waitbound timer [250]:
Test runs in 1 sec intervals. Press any key to stop...

Total   Run    Frames Rx Throughput              Frames
 sec    sec     per/sec   MBit/Sec  Util%  CPU% Dropped Errors
----- ------- --------- ---------- ------ ---- ------- ------
    0  1.0095  32586.56     16.684   2.19  100       1      0
    1  1.0094  32590.30     16.686   2.19  100       0      0
    2  1.0094  32589.56     16.686   2.19  100       0      0
    3  1.0094  32589.66     16.686   2.19  100       0      0
    4  1.0094  32589.88     16.686   2.19  100       0      0
    5  1.0094  32590.01     16.686   2.19  100       0      0
    6  1.0074  32653.42     16.719   2.19  100       0      0
    7  1.0000  13184.47      6.750   0.89   40       2      0
    8  1.0000      0.00      0.000   0.00    0       0      0
    9  1.0000      0.00      0.000   0.00    0       0      0
```

X1041_31_091208

Figure 31: **Rx SG DMA Test Output for MicroBlaze Processor System**

Echo Server SGDMA Test

Menu selection **Start Echo Server** (#20) will receive packets and immediately transmit them back to the sending MAC address as fast as possible.

The application keeps track of CPU utilization and the number of frames received, transmitted, and dropped. These statistics are displayed every 5 seconds. At the time of outputting the counts, no new frames are received. The maximum length for any received frame is 9000 bytes of data.

The number of frames received may differ from the number of frames transmitted due to interrupt coalescing. Frames may be dropped if the received frames count is extremely large and the frame sizes are small. This scenario could lead to running out of buffer descriptors.

This performance measurement application does not verify the correctness of data.

Figure 32 shows a typical **Echo Server** test output for the PowerPC 440 processor reference system. The output shown is an example of the test parameters, default selections, and the output of the this test.

```
Main Menu (speed=1000 Mbps, duplex=full):
4   - Tx SG DMA
5   - Tx Canned
9   - Rx SG DMA
20 - Start echo server
10 - Switch to 10 Mbps (full duplex)
100 - Switch to 100 Mbps (full duplex)
1000 - Switch to 1000 Mbps (full duplex)
99- Exit
Enter selection: 20
New Rx packet threshold count [32]:
New Tx packet threshold count [32]:
New Rx packet waitbound timer [250]:
New Tx packet waitbound timer [250]:

Frame storage:
   Base address : 0x0FDCC000
   Frame count  : 256
   Frame length : 9024 bytes
Press any key to stop, output every 5 sec.

CPU         Rx         Tx   Dropped
Load    Frames     Frames    Frames  Errors
----  -------  -------  -------  ------
   0        0        0        0       0
  72  2290516  2290516        0       0
 100  3185797  3185797        0       0
 100  3184986  3184986        0       0
 100  3185399  3185399        0       0
 100  3186035  3186035        0       0
 100  3184868  3184868        0       0
  28   887621   887621        0       0
   0        0        0        0       0
```

X1041_32_091208

Figure 32: **Echo Server Test Output for PowerPC 440 Processor System**

Figure 33 shows a typical **`Echo Server`** test output for the PowerPC 405 processor reference system. The output shown is an example of the test parameters, default selections, and the output of the this test.

```
Main Menu (speed=1000 Mbps, duplex=full):
4  - Tx SG DMA
5  - Tx Canned
9  - Rx SG DMA
20 - Start echo server
10 - Switch to 10 Mbps (full duplex)
100 - Switch to 100 Mbps (full duplex)
1000 - Switch to 1000 Mbps (full duplex)
99- Exit
Enter selection: 20
New Rx packet threshold count [32]:
New Tx packet threshold count [32]:
New Rx packet waitbound timer [250]:
New Tx packet waitbound timer [250]:

Frame storage:
   Base address : 0x07DCC000
   Frame count  : 256
   Frame length : 9024 bytes
Press any key to stop, output every 5 sec.

CPU         Rx         Tx   Dropped
Load    Frames     Frames    Frames  Errors
----  -------  -------  --------  ------
 100  1365787  1365787         0       0
 100  1365888  1365888         0       0
 100  1365888  1365888         0       0
 100  1365888  1365888         0       0
 100  1365888  1365888         0       0
  55   751745   751745         0       0
   0        0        0         0       0
   0        0        0         0       0
```

X1041_33_091208

Figure 33: **Echo Server Test Output for PowerPC 405 Processor System**

Figure 34 shows a typical **Echo Server** test output for the MicroBlaze processor reference system.

```
Main Menu (speed=1000 Mbps, duplex=full):
4  - Tx SG DMA
5  - Tx Canned
9  - Rx SG DMA
20 - Start echo server
10 - Switch to 10 Mbps (full duplex)
100 - Switch to 100 Mbps (full duplex)
1000 - Switch to 1000 Mbps (full duplex)
99- Exit
Enter selection: 20
New Rx packet threshold count [32]:
New Tx packet threshold count [32]:
New Rx packet waitbound timer [250]:
New Tx packet waitbound timer [250]:

Frame storage:
   Base address : 0x9FDCC000
   Frame count  : 256
   Frame length : 9024 bytes
Press any key to stop, output every 5 sec.

CPU         Rx         Tx   Dropped
Load    Frames     Frames    Frames  Errors
----   -------   -------   -------  ------
 100    67840     67840         0       0
 100    67840     67840         0       0
 100    67840     67840         0       0
 100    67840     67840         0       0
 100    67840     67840         0       0
  50    33537     33537         0       0
   0        0         0         0       0
   0        0         0         0       0
```

X1041_34_091208

Figure 34: **Echo Server Test Output for MicroBlaze Processor System**

Memory Controller Comparison in a PowerPC 440 Processor System

This section shows a comparison between using the PPC440MC DDR2 versus the MPMC memory controller in a PowerPC 440 processor systems. This system is not described in detail or provided as a reference system because the only difference between the two systems is the memory controller. It is recommended that the PPC440MC DDR2 controller be used whenever possible due to the increase in performance that can be obtained. It has been optimized for a PowerPC 440 processor system using DDR2 memory.

PowerPC 440 Memory Controller Comparison

The major differences in using the PPC440MC DDR2 memory controller compared to the MPMC memory controller in the PowerPC 440 processor system are shown in Table 18.

Table 18: Major Differences in Memory Controllers for PowerPC 440 Processor Systems

Parameter	PPC440MC DDR2	MPMC
Memory Interface / Speed	DDR2 Bus / 266.66 MHz	DDR2 Bus / 100 MHz
TEMAC to Memory Interface / Speed	LocalLink / 133.33 MHz	LocalLink / 100 MHz
PLBV46 Bus Speed	133.33 MHz	100 MHz
PowerPC 440 Processor Core Speed	400 MHz	400 MHz
DDR2 Bus Width	64 Bits	32 Bits

The raw Ethernet performance data shown in Table 19 highlights the increase in performance that can be achieved using the PPC440MC DDR2 over the MPMC. All raw Ethernet performance test data is taken with the following parameters set in the PerfApp application:

- Frame size in bytes as noted in Table
- Rx packet threshold count = 32
- Tx packet threshold count = 32
- Rx packet waitbound timer = 250
- Tx packet waitbound timer = 250

These data show the increased performance obtained by using the PPC440MC DDR2 memory controller over the MPMC, specifically with the minimum frame size of 64 bytes.

Table 19: Raw Performance Differences in Memory Controllers for PowerPC 440 Processor Systems

Raw Ethernet Performance	PPC440MC DDR2 System	MPMC System
Transmit SG DMA - 64 bytes	761 Mbps	589 Mbps
Transmit SG DMA - 1518 bytes	986 Mbps	986 Mbps
Transmit SG DMA - 9000 bytes	997 Mbps	997 Mbps
Receive SG DMA - 64 bytes	761 Mbps	595 Mbps
Receive SG DMA - 1518 bytes	986 Mbps	986 Mbps
Receive SG DMA - 9000 bytes	997 Mbps	997 Mbps

Raw Performance Comparisons for the PowerPC Processor Systems

The raw Ethernet performance is dependent on many things in the system. Because CPU utilization is high on smaller packet sizes due to increased interrupt overhead, there is a strong dependence on the system bus (PLBv46) speed, the Memory (DDR/DDR2) speed, and the processor speed.

Comparison of PowerPC 405 Processor Systems

The data shown in Table 20 through Table 24 was taken using the PerfApp application on PowerPC 405 processor systems with different bus, MPMC, and CPU frequencies to show the variability in performance.

There are restrictions on the frequencies that can be set. For instance, the PowerPC 405 processor frequency must be an integer multiple of the PLBv46 frequency and the MPMC to SDMA Clock Ratio can be either one or two times the PLBv46 frequency. Also, there are limits based on the speed grade of the part for how fast the PowerPC 405 processor and the MPMC can operate. For the ML405 board using the -10 speed grade, the PowerPC 405 processor is specified to run at a maximum of 350 MHz and the MPMC is specified to run at a maximum clock frequency of 180 MHz.

For each experimental system, the TX SGDMA test in PerfApp was used to collect raw Ethernet performance data using a set packet threshold count of 64 and a waitbound timer of 250. Frame sizes were varied from the minimum allowable frame size of 64 bytes to the maximum jumbo

frame size of 9000 bytes. The data reported from PerfApp is `Throughput` in Mbps, `Network Utilization` in percentage, and `CPU Utilization` in percentage.

The first set of raw Ethernet performance results are obtained from the ML405 PowerPC 405 processor system included with this application note.

The data for this system has the following clock speeds and is shown in Table 20:

- PLBv46 bus = 100 MHz
- DDR memory = 100 MHz
- PowerPC 405 processor = 300 MHz

Table 20: PerfApp Results: Bus = 100 MHz, DDR = 100 MHz, CPU = 300 MHz

Frame Size (bytes)	Throughput (Mbps)	Net Utilization (%)	CPU Utilization (%)
64	305.1	40.0	82
128	547.3	63.3	77
512	957.0	99.4	33
1518	985.7	99.9	11
9000	997.4	99.9	2

The data for a similar system for the ML405 board with the following clock speeds is shown in Table 21:

- PLBv46 bus = 100 MHz
- DDR memory = 200 MHz
- PowerPC 405 processor = 300 MHz

Table 21: PerfApp Results: Bus = 100 MHz, DDR = 200 MHz, CPU = 300 MHz

Frame Size (bytes)	Throughput (Mbps)	Net Utilization (%)	CPU Utilization (%)
64	361.6	47.5	78
128	644.6	74.5	70
512	959.2	99.7	26
1518	985.7	99.9	9
9000	997.2	99.9	2

The difference between these two systems is only due to an increase in the MPMC clock frequency. Because the CPU clock frequency has not changed there is only a small change in CPU utilization for even the smaller packet sizes. All of the increase in throughput is due to the increase in the MPMC clock frequency.

The data for a similar system for the ML405 board with the following clock speeds is shown in Table 22:

- PLBv46 bus = 90 MHz
- DDR memory = 180 MHz
- PowerPC 405 processor = 270 MHz

Table 22: PerfApp Results: Bus = 90 MHz, DDR = 180 MHz, CPU = 270 MHz

Frame Size (bytes)	Throughput (Mbps)	Net Utilization (%)	CPU Utilization (%)
64	325.6	42.7	79
128	579.4	67.0	71
512	955.7	99.3	29
1518	984.6	99.8	10
9000	997.2	99.9	2

Comparing the 100 - 200 - 300 system to the 90 - 180 - 270 system, there is approximately a 10% decrease in all three clock frequencies but CPU utilization did not change significantly. The slight performance decrease is due to both the system bus and the MPMC running slower.

The data for a similar system for the ML405 board with the following clock speeds is shown in Table 23:

- PLBv46 bus = 90 MHz
- DDR memory = 180 MHz
- PowerPC 405 processor = 360 MHz

Table 23: PerfApp Results: Bus = 90 MHz, DDR = 180 MHz, CPU = 360 MHz

Frame Size (bytes)	Throughput (Mbps)	Net Utilization (%)	CPU Utilization (%)
64	324.8	42.6	69
128	579.4	67.0	60
512	955.7	99.3	25
1518	984.6	99.8	9
9000	997.2	99.9	1

Comparing the 90 - 180 - 270 system to the 90 - 180 - 360 system, there is a 33% increase in CPU clock frequency. As expected, CPU utilization for the smaller frame sizes did change significantly but the throughput or network utilization did not change. This points to the throughput bottleneck being either in the system bus or the MPMC.

The data for a similar system for the ML405 board with the following clock speeds is shown in Table 24:

- PLBv46 bus = 125 MHz
- DDR memory = 125 MHz
- PowerPC 405 processor = 375 MHz

Table 24: PerfApp Results: Bus = 125MHz, DDR = 125 MHz, CPU = 375 MHz

Frame Size (bytes)	Throughput (Mbps)	Net Utilization (%)	CPU Utilization (%)
64	381.5	50.0	83
128	685.2	79.2	77
512	962.4	100.0	26
1518	986.9	100.0	9
9000	997.6	100.0	2

The best raw Ethernet performance was obtained from the 125 - 125 - 375 system, which is also the Higher Performance PowerPC 405 processor system described in an earlier section. Based on this data it appears that to obtain the highest throughput, the system bus needs to run as fast as possible to be able to get the small packets to the CPU. Next in importance is to keep the CPU utilization as low as possible so the CPU clock needs to run as fast as possible. The clock speed for the MPMC/DDR can only run at integer multiples of the PLBv46 bus speed. If the system bus is running above 100 MHz, the speed of the DDR has to run at the same speed as the bus to pass the data sheet specification for the MPMC.

Note: To exhibit the highest raw Ethernet performance capability of the Virtex-4 FX part, the PowerPC 405 processor is running at 375 MHz. To duplicate these results the user would be required to use either a -11 or -12 production part. Refer to previous sections of this document for details of how to build this system.

PowerPC 405 Processor System Comparison Conclusion

In general, most of the performance variation is evident in the smaller packet sizes where the CPU has to do more packet and interrupt processing. This is significant because it directly translates to TCP/UDP performance in that any decreases in CPU Utilization shown for raw packets requiring a lot of CPU processing will mean decreases in CPU processing of TCP/UDP packet headers also.

Performance of the PowerPC 440 Processor System

The data for the ML507 board with the PPC440 reference system with the following clock speeds is shown again in this section for comparison. Refer to Table 25:

- PLBv46 bus = 133.33 MHz
- DDR2 memory = 266.67 MHz
- PowerPC 440 processor = 400 MHz

Table 25: PerfApp Results: Bus = 133 MHz, DDR2 = 267 MHz, CPU = 400 MHz

Frame Size (bytes)	Throughput (Mbps)	Net Utilization (%)	CPU Utilization (%)
64	761.9	100.0	100
128	864.9	100.0	100
512	962.4	100.0	73

Table 25: PerfApp Results: Bus = 133 MHz, DDR2 = 267 MHz, CPU = 400 MHz

Frame Size (bytes)	Throughput (Mbps)	Net Utilization (%)	CPU Utilization (%)
1518	987.0	100.0	25
9000	997.8	100.0	4

The raw Ethernet performance for the PowerPC 440 processor system is by far superior to the other PowerPC 405 processor systems as can be seen by the data. This is due to several factors such as the processor running at 400 MHz, the PLBv46 bus running at 133 MHz, but the biggest effect is due to the DDR2 memory running at 266.66 MHz.

Note: The throughput data in the table shows that the PowerPC 440 processor system is running at Maximum Effective Rate for all frame sizes, which is also evident by the 100% network utilization data. This is a significant improvement in performance over the other reference systems.

Executing the Reference System

To execute these reference systems, the ML405, ML505, or ML507 board must be set up correctly and the bitstream must have been updated and programmed into the Virtex-4 or Virtex-5 FPGA device. The standalone PerfApp software is then loaded into external memory. Programming the bitstream can be done by either downloading the pre-built bitstream from the `ready_for_download/` directory, under the project root directory, or by generating and downloading it from XPS. Similarly, the PerfApp application executable can be downloaded from the `ready_for_download/` directory or built and downloaded through XPS.

PerfApp Hardware Setup

A hardware setup similar to what it shown in Figure 35 was used to obtain the PerfApp data shown in the above sections for the PerfApp test suite menu option results.

A live network connection must be made to the board under test to obtain a Link connection to the PHY.

For the tests that require receiving packets, **`Rx SG DMA`** and **`Echo Server,`** an Ethernet source is needed to supply packets as fast as possible. The ML507 board was used in all cases to transmit Ethernet frames to the other boards because the PowerPC 440 processor reference system is able to reach theoretical maximum line rate for transmitting all frame sizes including smaller 64 byte frames. With this set-up, the maximum possible receive rate can be obtained for each board / reference system. Figure 35 shows the board connections for a basic testing set-up using two ML507 development boards.

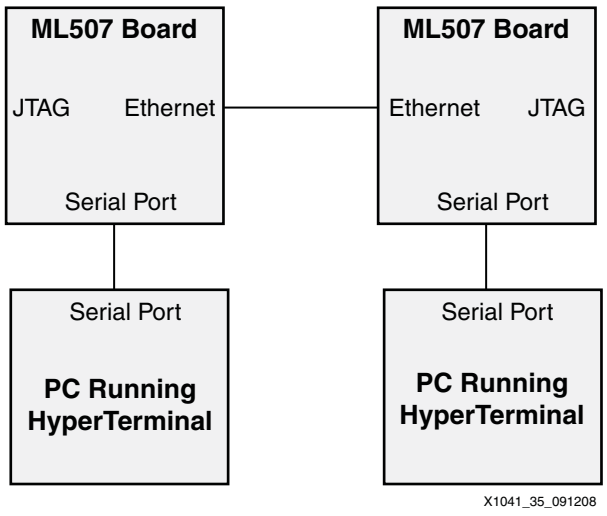

Figure 35: **Hardware Setup for Running PowerPC 440 Processor PerfApp Tests**

To monitor and control the software applications provided with this application note, set the HyperTerminal to the Bits per second of **115200**, Data Bits to **8**, Parity to **None**, and Flow Control to **None**. Refer to Figure 36 for the proper settings.

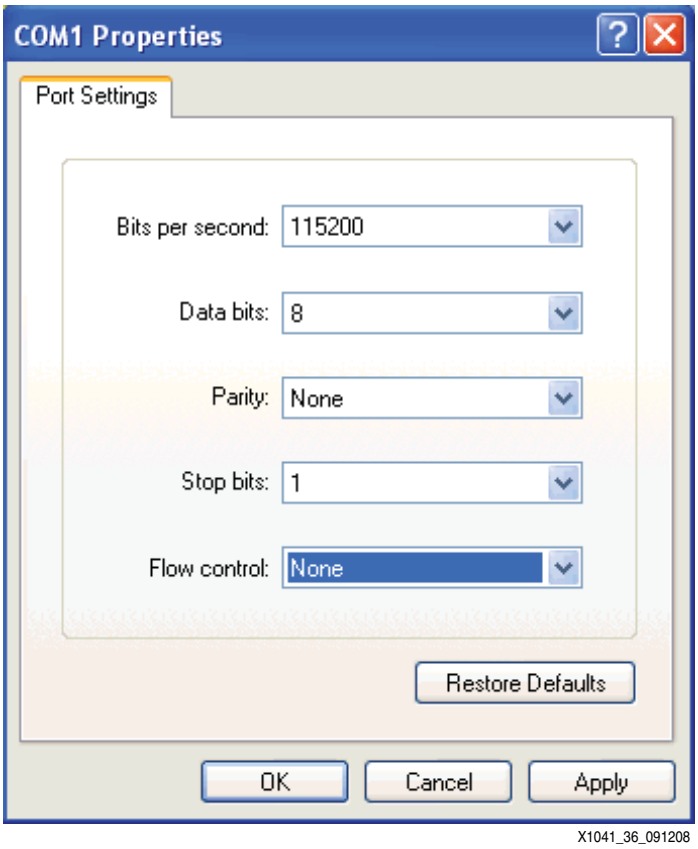

Figure 36: **HyperTerminal Settings**

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

Use the following steps to execute the system using files inside the `ready_for_download/` directory under the project root directory.

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

   ```
   impact -batch xapp1041.cmd
   ```

3. Invoke XMD and connect to the MicroBlaze processor by using the following command:

   ```
   xmd -opt xapp1041.opt
   ```

4. Download the standalone PerfApp executable by using the following command:

   ```
   dow lltemac_perfapp.elf
   ```

Executing the Reference System from EDK

Use the following steps to execute the system using EDK.

1. Open `system.xmp` inside EDK.
2. Use **Hardware→Generate Bitstream** to generate a bitstream for the system.
3. Download the bitstream to the board with **Device Configuration→Download Bitstream**.
4. Launch XMD with **Debug→Launch XMD...**

5. Download the standalone PerfApp executable by using the following command:

```
dow lltemac_perfapp.elf
```

Running the Standalone PerfApp Application

Use the **run** command, inside XMD, to run the PerfApp software application.

Refer back to the standalone PerfApp software application for the details of the menu options and each of the tests.

Before running any of the tests in the PerfApp test suite, the user needs to set the Link speed from the menu. The Link speed does not actually change until the desired test is selected and runs. This is important to note if monitoring the Link speed on the board LEDs or if debugging the Link speed with ChipScope.

Note: A live Ethernet connection must be made to the board otherwise the following link error will occur when a test is run:

```
PhySetup_Marvell_88e111: Link is down
```

Running Miscellaneous Software Applications

Supplied with the reference systems are a few other software applications that can be run to check out the hardware. These are very basic tests that are included as a reference and starting place for checking functionality of the different parts of the reference system.

TestApp_Memory Software Application

This basic BSB memory test application, `TestApp_Memory`, does a cursory check on a section of the DDR/DDR2. It can be useful for checking out basic memory functionality of the embedded reference system.

TestApp_Peripheral Software Application

This basic BSB peripheral test application, `TestApp_Peripheral`, does a basic functionality check on some of the peripheral cores in the systems. It does a test on the GPIO input and output, a test on the interrupt controller, a test on the timer (for MicroBlaze processor), a test on the UART, and some basic functional tests on the XPS_LL_TEMAC core.

Hello_TEMAC Software Application

The `hello_temac` test application does slightly more in depth functional testing on the xps_ll_temac core. The `hello_temac` application is not provided with the PowerPC 440 processor reference system.

This application is made up of example software that is supplied with the XPS_LL_TEMAC drivers in the EDK tools. This application does the following tests:

- Single Frame Example
- Coalescing Example
- Checksum Offload Example (if Checksum Offloading is turned on)

Note: This test can take several minutes to complete on the MicroBlaze processor reference system if the I-cache is not enabled.

Conclusion

This application note and associated reference systems demonstrate how to use the XPS_LL_TEMAC IP core in a PowerPC 440, PowerPC 405, and MicroBlaze processor embedded system. A software application providing a menu driven test suite is also supplied that provides examples of how to measure raw Ethernet performance on all three reference systems. All of the raw Ethernet performance data presented in this application note was verified independently using an IXIA 400T Ethernet Traffic Generator / Performance Analyzer test system.

References

1. DS537 *XPS Local Link TEMAC Product Specification*
2. DS643 *Multi-Port Memory Controller Product Specification*
3. DS614 *Clock Generator Product Specification*

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
5/29/08	1.0	Initial Xilinx release.
9/24/08	2.0	Added PowerPC 440 Processor Performance section and reference system.

Notice of Disclaimer