

Creating an Ethernet Subsystem with Micrel Ethernet PHY and Xilinx Spartan-3A DSP FPGAs: A Step-By-Step Guide

APPLICATION NOTE

September 2009 (Version 1.0)

Introduction

Many modern applications use FPGAs to implement complex system level building blocks. These building blocks can contain processors, DSPs, bus interface and even video and audio functions. Getting high-speed data on and off the FPGA can be accomplished by using a standard Ethernet interface and this leverages a wide range of established protocols, software platforms and IP. FPGA's usually requires an Ethernet Interface IP Core inside the FPGA and some external devices, typically an Ethernet Physical (PHY) layer device and a connector.

Creating an Ethernet subsystem involves a variety of design considerations. This application note provides a step-by-step guide for a designer who needs to quickly and efficiently create an Ethernet subsystem using a Xilinx Spartan-3A DSP FPGA, and a Micrel Ethernet PHY.

Topics in this application note are organized by design task (Device Selection, FPGA Design, and Board Design) and each topic is a stand-alone section, with a short introduction or overview, followed by the step-by-step design guidelines. The reader may skip over sections and go directly to the topic of interest, or start at the beginning and follow each topic in order to quickly go through the entire design process, step-by-step. The topics covered in this application note include the following:

- 1) Device Selection
 - a. Micrel Ethernet PHY
 - i. Ethernet PHY Functional Overview
 - ii. Pin Descriptions
 - iii. MII Connectivity
 - b. Xilinx Spartan-3A DSP FPGAs
 - i. Spartan-3A DSP Functional Overview
 - ii. Family Overview
 - iii. Device Selection
- 2) FPGA Design
 - a. Using the Xilinx Ethernet IP Core
 - i. IP Core Overview
 - ii. Functional Description
 - iii. Pin Descriptions
 - iv. Example Design- Step-by-Step
 - b. Integration of the IP Core
 - i. Integration Overview
 - ii. Functional Description
 - iii. Pin Descriptions
- 3) Board Design Considerations
 - a. Pin Assignment Considerations
 - b. Simultaneous Switching Output Considerations
 - c. Power Supply and Decoupling Considerations
 - d. Board Layout and Signal Routing Considerations

A significant amount of supporting documentation is referenced in this application note and the reference section at the end of this application note provides links to on-line versions of each reference, making it easy to locate additional information.

Device Selection

The first phase of the design project is device selection, where the designer decides on the key components to be used in the design. As a target application example we will use a video inspection system used in an assembly line application. This type of embedded video processor needs to fit into a very small form factor and will monitor and control inspection of product produced by an assembly. Ethernet is being used to communicate to the controller so that commands and data received by the controller can be relayed to the central processing location. A large number of controllers will be distributed throughout the factory at various stages of the manufacturing process. Because of the data transmission speed required high-speed Ethernet is needed, but backwards compatibility to standard Ethernet is an advantage. Significant DSP functionality is desired so that video images may be processed quickly and any product quality issues spotted quickly. Let's assume that an initial review of possible suppliers has been conducted and that Micrel has been selected as the supplier for the Ethernet PHY and Xilinx has been selected as the FPGA supplier.

An FPGA was selected for the flexibility, small form factor and the ability to provide a variety of I/O options required by the equipment. The Spartan-3A DSP Family was selected because of its low cost, high-I/O count, DSP functionality and the availability of a robust Ethernet MAC. The Micrel Ethernet PHY was selected because of its small footprint, flexible interface, power management capability and advanced features.

In this section of the application note we will review the Micrel Ethernet PHY, and the Xilinx Spartan-3A DSP Family. This combination of devices is excellent for a wide range of applications in the consumer, industrial, communications and networking industries- and in particular for our target application where a small form factor, DSP processing power and future extensibility are important. The following sections provide a quick overview of each device family, identifies key features that make them good selections for our target application and determines the best device within the family for our specific application.

Micrel Ethernet Devices

Micrel offers a wide range of Ethernet physical layer devices and recently announced a single-port 10/100/1000 Base-T Gigabit Transceiver, the KSZ9021. All Micrel PHY offer robust performance and low power consumption, in the industry's smallest footprint. A summary list of the main Micrel PHY devices is given in Table 1 below. A selector guide with all the specific part numbers is available at http://www.micrel.com/page.do?page=product-info/fastether_trans.jsp

| Product | Voltages | Interface | Industrial | Crossover |
|---------|-----------|-------------------|------------|-----------|
| KSZ9021 | 3.3V | GMII/MMII RMII | Available | Available |
| KS8041 | 3.3V | MII, RMII, SMII | Available | Available |
| KS8721 | 2.5V/3.3V | MII, RMII | Available | Available |

Table 1: Micrel Ethernet Controller Products Summary

Micrel KS8041NL Overview

With the core at 1.8 volts to meet low voltage and low - power requirements, the KSZ8041NL is a 10BASE-T/100BASE-TX Physical Layer Transceiver with Media Independent Interface (MII) or the Reduced Media Independent Interface (RMII) to transmit and receive data. A unique mixed signal design extends signaling distance while reducing power consumption.

HP Auto MDI/MDI-X provides the most robust solution for eliminating the need to differentiate between crossover and straight-through cables. The KSZ8041NL represents a new level of features and performance and is an ideal choice of physical layer transceiver for 100BASE-TX/10BASE-T applications.

The signal list for the KSZ8041NL is shown in Table 2 below. The interface to the MAC can use either the MII or RMII standard. The signals listed in Figure 2 follow most closely the MII standard since that is the one we will use in our target application. These 2 bus types are described in more detail following Table 2.

| Name | Type | Description |
|-----------------------------|-------|--|
| TXD0 | I | Transmit Data 0: Bit 0 of the 4 data bits that are accepted by the PHY for transmission. |
| TXD1 | I | Transmit Data 1: Bit 0 of the 4 data bits that are accepted by the PHY for transmission. |
| TXD2 | I | Transmit Data 2: Bit 0 of the 4 data bits that are accepted by the PHY for transmission. |
| TXD3 | I | Transmit Data 3: Bit 0 of the 4 data bits that are accepted by the PHY for transmission. |
| TXEN | I | Transmit Enable Input: Indicates that valid data is presented on the TXD[3:0] signals, for transmission. |
| TXC | O | Transmit Clock Output: 25MHz in 100Base-TX mode, 2.5MHz in 10Base-T mode. |
| RXD3/ PHYAD0 | I/O | Receive Data 3: Bit 3 of the 4 data bits that are sent by the PHY in the receiver path. PHY ADDR[0]: sets PHYADDR[0] during initialization. |
| RXD2/ PHYAD1 | I/O | Receive Data 2: Bit 2 of the 4 data bits that are sent by the PHY in the receiver path. PHY ADDR[1]: sets PHYADDR[1] during initialization. |
| RXD1/ PHYAD2 | I/O | Receive Data 1: Bit 1 of the 4 data bits that are sent by the PHY in the receiver path. PHY ADDR[2]: sets PHYADDR[2] during initialization. |
| RXD0/ DUPLEX | I/O | Receive Data 0: Bit 0 of the 4 data bits that are sent by the PHY in the receiver path. DUPLEX: sets DUPLEX during initialization. |
| RXER/ ISO | O | Receive Error: Asserted to indicate that an error was detected somewhere in the frame presently being transferred from the PHY. ISO: Sets ISO during initialization. |
| RX_DV/ CRSDV/ CONFIG2 | O | Receive Data Valid Output: Indicates that recovered and decoded data nibbles are being presented on RXD[3:0]. Carrier Receive Data Valid: Asserted to indicate when the receive medium is non-idle. CONFIG2: sets CONFIG2 during initialization. |
| RXC | O | Receive Clock Output: 25MHz in 100Base-TX mode. 2.5MHz in 10Base-T mode. |
| COL | O | Collision Detect: Asserted to indicate detection of collision condition. |
| MDIO | I/O | Management Data Input/Output: Serial management data input/output. External 4.7KOhm pull-up. |
| MDC | I | Management Clock: Serial management clock. |
| | | |
| CRS | O | Carrier Sense Output. Indicates presence of Carrier on the media. |
| INTERP | O | Interrupt Output: Programmable interrupt indication. |
| RST# | I | External Reset- input of the system reset. |
| XI/Refclk | I/O | Clock Input- 25MHz (MII) or 50MHz (RMII) external clock or crystal input. |
| XO | O | Crystal Feedback. |
| | | |
| TX+ | AO | Transmit Data Positive: 100Base-TX or 10Base-T differential transmit outputs to magnetic. |
| TX- | AO | Transmit Data Negative: 100Base-TX or 10Base-T differential transmit outputs to magnetic. |
| RX+ | AI | Receive Data Positive: 100Base-TX or 10Base-T differential transmit inputs from magnetic. |
| RX- | AI | Receive Data Negative: 100Base-TX or 10Base-T differential transmit inputs from magnetic. |
| | | |
| REXT | IO | Connects to reference resistor of value 6.49K-Ohm in parallel with a 100pF capacitor. |
| | | |
| VDDIO_3.3 | POWER | +3.3V Digital Vcc |
| VDDA3.3 | POWER | +3.3V Analog Power |
| VDD_PLL | POWER | + 1.8V analog ground |
| GND | POWER | Ground |

Table 2: Micrel KS8041NL Pin Descriptions

KS8041NL Functional Description

The key functions performed by the KS8041NL include transmit, receive, clock synthesis, management interface and power management.

100Base-TX Transmit

The 100Base-TX transmit function performs parallel-to-serial conversion, 4B/5B coding, scrambling, NRZ-to-NRZI conversion, and MLT3 encoding and transmission. The circuitry starts with a parallel-to-serial conversion, which converts the MII data from the MAC into a 125MHz serial bit stream. The data and control stream is then converted into 4B/5B coding, followed by a scrambler. The serialized data is further converted from NRZ-to-NRZI format, and then transmitted in MLT3 current output.

The output current is set by an external 6.49k Ω 1% resistor for the 1:1 transformer ratio. It has typical rise/fall times of 4 ns and complies with the ANSI TP-PMD standard regarding amplitude balance, overshoot and timing jitter. The wavelshaped 10Base-T output drivers are also incorporated into the 100Base-TX drivers.

100Base-TX Receive

The 100Base-TX receiver function performs adaptive equalization, DC restoration, MLT3-to-NRZI conversion, data and clock recovery, NRZI-to-NRZ conversion, de-scrambling, 4B/5B decoding, and serial-to-parallel conversion.

The receiving side starts with the equalization filter to compensate for inter-symbol interference (ISI) over the twisted pair cable. Since the amplitude loss and phase distortion is a function of the cable length, the equalizer must adjust its characteristics to optimize performance. In this design, the variable equalizer makes an initial estimation based on comparisons of incoming signal strength against some known cable characteristics, and then tunes itself for optimization. This is an ongoing process and self-adjusts against environmental changes such as temperature variations.

Next, the equalized signal goes through a DC restoration and data conversion block. The DC restoration circuit is used to compensate for the effect of baseline wander and to improve the dynamic range. The differential data conversion circuit converts the MLT3 format back to NRZI. The slicing threshold is also adaptive.

The clock recovery circuit extracts the 125MHz clock from the edges of the NRZI signal. This recovered clock is then used to convert the NRZI signal into the NRZ format. This signal is sent through the de-scrambler followed by the 4B/5B decoder. Finally, the NRZ serial data is converted to the MII format and provided as the input data to the MAC.

PLL Clock Synthesizer

The KSZ8041NL generates 125M z, 25M z and 20M z clocks for system timing. Internal clocks are generated from an external 25MHz crystal or oscillator. In RMII mode, these internal clocks are generated from an external 50MHz oscillator or system clock.

Scrambler/De-scrambler (100Base-TX only)

The purpose of the scrambler is to spread the power spectrum of the signal in order to reduce EMI and baseline wander.

10Base-T Transmit

The 10Base-T drivers are incorporated with the 100Base-TX drivers to allow for transmission using the same magnetics. The drivers also perform internal wave-shaping and pre-emphasize, and output 10Base-T signals with a typical amplitude of 2.5V peak. The 10Base-T signals have harmonic contents that are at least 27dB below the

fundamental frequency when driven by an all-ones Manchester-encoded signal.

10Base-T Receive

On the receive side, input buffer and level detecting squelch circuits are employed. A differential input receiver circuit and a PLL performs the decoding function. The Manchester-encoded data stream is separated into clock signal and NRZ data.

A squelch circuit rejects signals with levels less than 400 mV or with short pulse widths to prevent noise at the RX+ and RX- inputs from falsely trigger the decoder. When the input exceeds the squelch limit, the PLL locks onto the incoming signal and the KSZ8041NL decodes a data frame. The receive clock is kept active during idle periods in between data reception.

SQE and Jabber Function (10Base-T only)

In 10Base-T operation, a short pulse is put out on the COL pin after each frame is transmitted. This SQE Test is required as a test of the 10Base-T transmit/receive path. If transmit enable (TXEN) is high for more than 20 ms (jabbering), the 10Base-T transmitter is disabled and COL is asserted high. If TXEN is then driven low for more than 250 ms, the 10Base-T transmitter is re-enabled and COL is de-asserted.

Auto-Negotiation

The KSZ8041NL conforms to the auto-negotiation protocol, defined in Clause 28 of the IEEE 802.3u specification. Auto-negotiation is enabled by either hardware pin strapping (pin 30) or software (register 0h bit 12). Auto-negotiation allows unshielded twisted pair (UTP) link partners to select the highest common mode of operation. Link partners advertise their capabilities to each other, and then compare their own capabilities with those they received from their link partners. The highest speed and duplex setting that is common to the two link partners is selected as the mode of operation.

If auto-negotiation is not supported or the KSZ8041NL link partner is forced to bypass auto-negotiation, the KSZ8041NL sets its operating mode by observing the signal at its receiver. This is known as parallel detection, and allows the KSZ8041NL to establish link by listening for a fixed signal protocol in the absence of auto-negotiation advertisement protocol.

MII Management Interface

The KSZ8041NL supports the IEEE 802.3 MII Management Interface (MIIM), also known as the Management Data Input / Output (MDIO) Interface. This interface allows upper-layer devices to monitor and control the state of the KSZ8041NL. An external device with MIIM capability is used to read the PHY status and/or configure the PHY settings. Additional details on the MIIM interface can be found in Clause 22.2.4.5 of the IEEE 802.3u Specification.

Power Management

The KSZ8041NL offers the following power management modes:

Power Saving Mode

This mode is used to reduce power consumption when the cable is unplugged. It is in effect when auto-negotiation mode is enabled, cable is disconnected, and register 1F bit 10 is set to 1 (default setting). Under power saving mode, the KSZ8041NL shuts down all transceiver blocks, except for energy detect and PLL circuits. Additionally, in MII mode, the RXC clock output is disabled. RXC clock is enabled after the cable is connected and link is established. Power saving mode is disabled by writing a zero to register 1F bit 10.

Power Down Mode

This mode is used to power down the entire KSZ8041NL device when it is not in use. Power down mode is enabled by writing a one to register 0 bit 11. In the power down state, the KSZ8041NL disables all internal functions, except for the MII management interface.

KS8041NL to FPGA Connectivity

There are two main choices for connectivity between the KS8041NL and the FPGA: the Media Independent Interface (MII) or the Reduced Media Independent Interface (RMII). These interfaces are standard 'busses' that are compatible among a wide range of Ethernet devices. The MII standard requires 16 pins for each MAC to PHY interface. In some applications, hubs and switches with multiple MACs and external PHYs for example, can require a very large number of pins. In order to reduce pin-count in these applications the RMII standard was developed which requires only 6 pins per MAC to PHY interface, by eliminating some functionality. In this example design the MII standard will be used to illustrate a more feature rich implementation and because pin limitations are not critical in our application.

A block diagram of the MII connectivity between the KS8041NL and the Spartan-3A FPGA is shown in Figure 1 below. The signal assignments and constraints required for the FPGA will be discussed in more detail in the FPGA Design section of this document. Important board level considerations like grounding, layout and routing constraints will be covered in the Board Design section of this document.

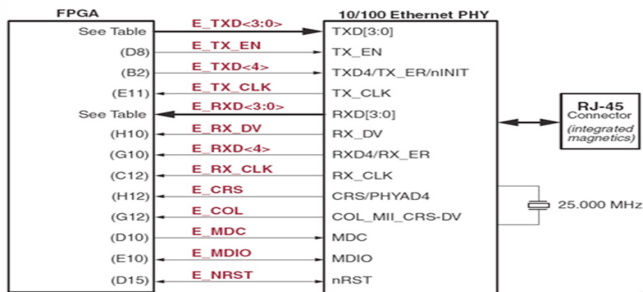


Figure 1: Interconnection between Spartan-3A FPGA and Micrel Ethernet PHY

Xilinx Spartan-3 Generation of FPGAs

The Xilinx Spartan-3 generation of FPGAs are optimized for a variety of applications. Spartan-3 devices are targeted for high density, high pin count designs and are ideal for data processing applications. Spartan-3E devices are optimized for logic intensive designs and are ideal for co-processing and embedded control applications. Spartan-3A DSP devices are optimized for DSP designs and are ideal for ultrasound, consumer video and video surveillance applications. Spartan-3A devices are optimized for IO intensive designs and are ideal for bridging, differential signaling and memory interface applications. Spartan-3AN devices are non-volatile and are ideal for any application with design security, low cost and high-volume requirements.

The Spartan-3A DSP family builds on the success of the Spartan-3A FPGA family by increasing the amount of memory per logic and adding XtremeDSP™ DSP48A slices. New features improve system performance and reduce the cost of configuration. These Spartan-3A DSP FPGA enhancements, combined with proven 90 nm process technology, deliver more functionality and bandwidth per dollar than ever before, setting the new standard in the programmable logic and DSP processing industry.

Spartan-3A and Spartan-3A DSP FPGAs

The Spartan-3A DSP FPGAs extend and enhance the Spartan-3A FPGA family. The XC3SD1800A and the XC3SD3400A devices are tailored for DSP applications and have additional block RAM and XtremeDSP DSP48A slices. The XtremeDSP DSP48A slices replace the 18x18 multipliers found in the Spartan-3A devices and are based on the DSP48 blocks found in the Virtex®-4 devices. The block RAMs are also enhanced to run faster by adding an output register. Both the block RAM and DSP48A slices in the Spartan-3A DSP devices run at 250 MHz in the lowest cost, standard -4 speed grade. Because of their exceptional DSP price/performance ratio, Spartan-3A DSP FPGAs are ideally suited to a wide range of consumer electronics applications, including broadband access, home networking, display/projection, and digital television equipment. The Spartan-3A DSP family is a superior alternative to mask programmed ASICs. FPGAs avoid the high initial cost, lengthy development cycles, and the inherent inflexibility of conventional ASICs. Also, FPGA programmability permits design upgrades in the field with no hardware replacement necessary, an impossibility with ASICs.

Selecting the Spartan-3A DSP Device

The selection of the target Spartan-3A DSP device depends on both the IO requirements and the logic requirements. For our target application let's assume we need over 400 IOs and over 15K logic slices. Let's also assume that our DSP48A requirements are around 64, based on a quick review of the DSP oriented functions we need to implement. Table 3 below shows the range of capabilities of the Spartan-3A DSP devices. The XC3SD1800A device is the smallest that meets all of our key requirements. (During the design process we will have the option to design the interface in such a way that we can migrate to another device with the same package footprint, with a different amount of logic capacity. This can be useful if the design requirements might change late in the design process.)

Device Selection Summary

We have decided on the key components from which to construct our Ethernet subsystem. We chose the Micrel KS8041NL Ethernet PHY for its low cost, flexibility, advanced features, power saving capabilities and small footprint. We selected the Xilinx Spartan-3A DSP XC3SD1800A for its advanced DSP capabilities, abundant IO resources and low cost.

Now that we have selected our target devices we can begin the design of the FPGA portion of the subsystem. This involves configuring the Ethernet Controller that resides in the Spartan-3A DSP FPGA. This process is described in detail in the next section of this document.

| Device | Slices | DSP48As | Block RAM Bits | IOs |
|------------|--------|---------|----------------|-----|
| XC3SD1800A | 16,640 | 84 | 1512K | 519 |
| XC3SD3400A | 23,872 | 126 | 2268K | 469 |

Table 3: Spartan-3A DSP Logic Function and IO Capabilities

FPGA Design

Once the key devices have been selected the FPGA stage of the design can be started. During this phase the designer will need to create an Ethernet interface subsystem, inside the FPGA, to control the external PHY. There are a few choices available from Xilinx to implement this function in a Spartan-3A DSP Family device. The OPB 10/100 Ethernet MAC Lite is available as part of the Embedded Development Kit and

is optimized for embedded applications. It has an OPB interface to make it easy to connect to on-FPGA embedded processor solutions like MicroBlaze™ and is a light weight implementation that leaves out some of the more complex Ethernet features not required in simple embedded designs. The Ethernet MAC 10/100 controller is a more full featured implementation and is well suited for a variety of higher end networking applications where a more full featured Ethernet implementation is required. The Tri-Mode Ethernet Media Access Controller (TEMAC) implements 10/100/1000 Ethernet and includes Gigabit Ethernet support for high speed bridging and routing applications. For our target application we will use the OPB 10/100 Ethernet MAC Lite version since it has all the functionality we require and has a small gate count which will help keep our design footprint small and will help reduce cost and power.

Xilinx Ethernet Controller IP Core

The Ethernet OPB 10/100 MAC Lite is primarily intended for use with MicroBlaze applications. The Ethernet Lite MAC controller core uses fewer FPGA resources and is ideal for applications that do not require support for interrupts, back-to-back data transfers, and statistics counters. You can refer to the OPB Ethernet MAC data sheet included in the reference section of this application note for additional functional details. This document will focus primarily on the functions of the input and output signals.

The Ethernet OPB 10/100 MAC Lite IP Core is available as a hardware evaluation version. The hardware evaluation version of the Ethernet MAC core operates for approximately eight hours in silicon before timing out. This is sufficient to prove out and test a design prior to the purchase of the production version. To order the production version of the Ethernet MAC IP core, visit the Xilinx IP Center website at: <http://www.xilinx.com/ipcenter/index.htm> and search for "Ethernet MAC".

Xilinx Ethernet Controller IP Core Overview

The Ethernet MAC Lite IP Core interfaces directly to the Micrel PHY via the FPGA I/O pins and to the FPGA user logic via the on-chip OPB interface. A detailed list of the OPB Ethernet MAC IP Core I/O signals is given in Table 4 below. The signals in the first part of the table are connected directly to the external PHY device. The signals in the second part of the table are the inputs and outputs internal to the FPGA that connect the IP Core to the OPB interface used to connect the user logic to the Ethernet MAC IP Core. The last signals in the table are the system clocking and reset controls.

| Name | Type | Description |
|------------------|--------|--|
| PHY_rx_data(3:0) | Input | Ethernet Receive Data |
| PHY_tx_data(3:0) | Output | Ethernet Transmit Data |
| PHY_dv | Input | Ethernet Receive Data Valid |
| PHY_rx_er | Input | Ethernet Receive Error |
| PHY_tx_en | Output | Ethernet Transmit Enable |
| PHY_tx_clk | Input | Ethernet Transmit Clock |
| PHY_rx_clk | Input | Ethernet Receive Clock |
| PHY_crs | Input | Ethernet Carrier Sense Input |
| PHY_col | Input | Ethernet Collision Input |
| PHY_rst_n | Output | Ethernet PHY reset |
| Sin_DBus(0: n) | Output | Slave Output Data bus. Configurable width. |
| Sin_xferAck | Output | Slave Transfer Acknowledge |
| Sin_Retry | Output | Slave Retry |

| Name | Type | Description |
|----------------|--------|---------------------------------------|
| Sin_ErrAck | Output | Slave Error Acknowledge |
| SinToutSup | Output | Slave Timeout Suppress |
| OPB_ABus(0: m) | Input | OPB Address Bus. Configurable Width |
| OPB_BE(0: k) | Input | OPB Byte Enables. Configurable Width. |
| OPB_DBus(0: m) | Input | OPB Data Bus. Configurable Width |
| OPB_RNW | Input | Read Not Write |
| OPB_select | Input | Master has taken control of the bus. |
| OPB_seqAddr | Input | OPB Sequential Address |
| OPB_Clk | Input | System Clock |
| OPB_Rst | Input | System Reset |
| IP2INTC_Irpt | Output | System Interrupt |

Table 4: Ethernet MAC IP Core I/O Signal Definitions

The user interface to the Ethernet MAC IP Core is a very simple register free implementation. All communication between the Ethernet MAC and the user logic is done via a 4K byte block of dual port memory. Optional ping-pong buffers can be included when configuring the IP Core to improve transmit and receive performance. In our example design we will not use these buffers since the increased performance is not a requirement of our design. There are a variety of other configurable options including the FPGA device family, the bus clock period, base addresses, maximum address, OPB address and data bus widths. (For a more detailed description of these parameters refer to the data sheet for the OPB Ethernet MAC Lite MAC data sheet given in the Reference section of this document.)

In our example design we will use the default parameters for the Ethernet MAC Lite IP Core used in the Xilinx Spartan-3A DSP Reference Kit. These parameters will automatically configure the IP Core for our target Spartan-3A DSP device, the MicroBlaze processor, the correct bus widths and timing and the correct addresses used for the FIFOs and OPB resources. Once XPS has been run with the Spartan-3A DSP Reference Kit as the target and all the defaults are selected the generated MHS file contains a list of the key parameters used by the Ethernet MAC Lite IP Core (along with all the other IP Cores included in the Reference Kit).

The listing in Figure 2 below shows the various PARAMETER, BUS_INTERFACE and PORT definitions created by XPS. The Xilinx XPS User Guide, referenced at the end of this document, contains a more complete description of how to create various processor based designs using any Xilinx Processor or IP Core.

```

BEGIN xps_ethernetlite
PARAMETER INSTANCE = Ethernet_MAC
PARAMETER HW_VER = 1.00.a
PARAMETER C_SPLB_CLK_PERIOD_PS = 16000
PARAMETER C_BASEADDR = 0x81000000
PARAMETER C_HIGHADDR = 0x8100ffff
BUS_INTERFACE SPLB = mb_plb
PORT PHY_col = fpga_0_Ethernet_MAC_PHY_col
PORT PHY_crs = fpga_0_Ethernet_MAC_PHY_crs
PORT PHY_rst_n = fpga_0_Ethernet_MAC_PHY_rst_n
PORT PHY_rx_clk = fpga_0_Ethernet_MAC_PHY_rx_clk
    
```

```
PORT PHY_dv = fpga_0_Ethernet_MAC_PHY_dv
PORT PHY_rx_data = fpga_0_Ethernet_MAC_PHY_rx_data
PORT PHY_rx_er = fpga_0_Ethernet_MAC_PHY_rx_er
PORT PHY_tx_clk = fpga_0_Ethernet_MAC_PHY_tx_clk
PORT PHY_tx_en = fpga_0_Ethernet_MAC_PHY_tx_en
PORT PHY_tx_data = fpga_0_Ethernet_MAC_PHY_tx_data
END
```

Figure 2: XPS Port Definitions for Xilinx Ethernet MAC Lite IP Core

Using the Xilinx Ethernet Controller IP Core

The Ethernet Lite MAC Core requires design constraints to guarantee performance. These constraints should be placed in a UCF file for the top level of the design. The following example of the constraint text is based on the port names of the Ethernet Lite MAC core. If these ports are mapped to FPGA pin names that are different, the FPGA pin names should be substituted for the port names in the examples in Figure 3 below. Note that this list includes some constraints that don't apply to our target device. These are listed for completeness and the specifics are given in the notes at the end of the signal list.

```
NET "phy_rx_clk" TNM_NET = "RXCLK_GRP";
NET "phy_rx_clk" TNM_NET = "RXCLK_GRP";
TIMESPEC "TSTXOUT" = FROM "TXCLK_GRP" TO "PADS" 10 ns;
TIMESPEC "TSRXIN" = FROM "PADS" TO "RXCLK_GRP" 6 ns;
NET "opb_rst" TIG;
NET "phy_rx_clk" USELESSKEWLINES; Note 1
NET "phy_tx_clk" USELESSKEWLINES; Note 2
NET "phy_tx_clk" MAXSKEW= 2.0 ns; Note 2
NET "phy_rx_clk" MAXSKEW= 2.0 ns; Note 2
NET "phy_tx_clk" MAXSKEW= 1.0 ns; Note 3
NET "phy_rx_clk" MAXSKEW= 1.0 ns; Note 3
NET "phy_rx_clk" PERIOD = 40 ns HIGH 14 ns;
NET "phy_tx_clk" PERIOD = 40 ns HIGH 14 ns;
NET "phy_rx_data<3>" NODELAY; Note 1
NET "phy_rx_data<2>" NODELAY; Note 1
NET "phy_rx_data<1>" NODELAY; Note 1
NET "phy_rx_data<0>" NODELAY; Note 1
NET "phy_rx_data<0>" NODELAY; Note 1
NET "phy_dv" NODELAY; Note 1
NET "phy_rx_er" NODELAY; Note 1
NET "phy_crs" NODELAY; Note 1
NET "phy_col" NODELAY; Note 1
NET "phy_rx_data<3>" IOBDELAY=NONE; Note 3
NET "phy_rx_data<2>" IOBDELAY=NONE; Note 3
NET "phy_rx_data<1>" IOBDELAY=NONE; Note 3
NET "phy_rx_data<0>" IOBDELAY=NONE; Note 3
NET "phy_dv" IOBDELAY=NONE; Note 3
NET "phy_rx_er" IOBDELAY=NONE; Note 3
NET "phy_crs" IOBDELAY=NONE; Note 3
NET "phy_col" IOBDELAY=NONE; Note 3
```

- Notes:
1. Use for Virtex, Virtex-E, Spartan-II, or Spartan-IIe devices only.
 2. Use for all devices except Virtex-4.
 3. Use for Virtex-4 only.

Figure 3: Ethernet MAC Lite Constraints File Listing

Once the Ethernet MAC has been successfully configured and integrated in to the users design it is time to move on to the Board Design portion of the project. The key Board Design considerations are given in the next section of this application note.

During the board design portion of the Ethernet subsystem design process there are several key design considerations that need to be addressed. These considerations can be grouped as follows: Pin Assignment considerations, Board Layout/Signal Layout Considerations, Power Supply/Decoupling Considerations, Board Layout and Component Placement Considerations. Following the discussion of these considerations is an example Ethernet subsystem design showing details of the device placement, layout and signal routing used in the design. This illustrates some of the trade-offs involved in implementing a real-world system and shows how the designer can use the board layout considerations given in this document to help simplify the process of evaluating various design trade-offs.

FPGA Pin Assignment Considerations

The Xilinx IP Core generator tool can use pin assignments supplied by the user, making it easy to simplify the board layout process. If board layout isn't a concern the pin assignment can be left up to the Xilinx software and the assignment will be made based on the design requirements of the logic and interconnect inside the FPGA. It is a fairly simple process to make pin assignments for the important MII signals between the FPGA and the Micrel PHY. Simply select pins of the side of the FPGA that is closest to the Micrel PHY device as determined by your board layout. An example pin list, in Figure 4 below, is taken from the Xilinx Spartan-3A DSP Reference Kit design and can be used a starting point for any design using the Spartan-3A XC3SD1800A in the 484 BGA package.

| Name | Signal Name | FPGA Ball No. |
|--------------------------|--------------|---------------|
| IO58_L35N_1 | MDIO | E20 |
| IO13_L8P_1 | RXDV | V22 |
| IO14_L8N_1/VREF1_1 | COL | U21 |
| IO29_L18P_1/RHCLK0 | REFCLK | N18 |
| IO31_L19P_1/RHCLK2 | MDC | N21 |
| IO32_L19N_1/TRDY1/RHCLK3 | TXC | M20 |
| IO33_L20P_1/RHCLK4 | RXC | L21 |
| IO49_L30P_1/A18 | CRS | G19 |
| IO50_L30N_1/A19 | TXD3 | F20 |
| IO51_L32P_1 | TXD2 | J19 |
| IO52_L32N_1 | TXD1 | H20 |
| IO53_L33P_1 | TXD0 | F22 |
| IO54_L33N_1 | TXEN | E22 |
| IP9_L27P_1 | RXD0/DUPLEX | J21 |
| IP10_L27N_1 | RXD1/AD2 | J22 |
| IP11_L31P_1/VREF7_1 | RXD2/AD1 | H21 |
| IP12_L31N_1 | RXD3/AD0 | G20 |
| IO47_L29P_1/A16 | INTRP | K16 |
| IO56_L34N_1 | RESET# | G18 |
| IO60_L36N_1/A21 | RXER/ISOLATE | E19 |

Figure 4: Pin Locations for Ethernet MAC PHY on XC3SD1800A

Simultaneous Switching Output Considerations

Ground bounce must be controlled to ensure proper operation of high-performance FPGA devices. When multiple output drivers change state at the same time, power supply disturbance occurs. These disturbances can cause undesired transient behavior in output drivers, input receivers, or in internal logic. These disturbances are often referred to as Simultaneous Switching Output (SSO) noise. The SSO limits govern the number and type of I/O output drivers that can be switched simultaneously while maintaining a safe level of SSO noise. SSO of an individual FPGA IO bank is calculated by summing the SSO contributions of the individual I/O standards in the bank.

The SSO contribution is the percentage of full utilization of any one I/O standard in any one bank. The Weighted Average SSO (WASSO) calculation is the done by combining the SSO contributions of all I/O in a bank into a single figure.

A Microsoft Excel-based spreadsheet entitled "WASSO Calculator" has been created by Xilinx to automate these calculations. The WASSO calculator uses PCB geometry, such as board thickness, via diameter, and breakout trace width and length, to determine board inductance. It determines the smallest undershoot and logic-Low threshold voltage among all input devices, calculates the average output capacitance, and determines the SSO allowance by taking into account all of the board-level design parameters mentioned in this document. In addition, the WASSO calculator performs checks to ensure the overall design does not exceed the SSO allowance.

The WASSO Calculator with support for Spartan-3A DSP (Rev 1.0) is available as part of xapp689 (rev 1.2). Simply download the calculator, refer to xapp689 (rev 1.2) for a description of each entry and enter the specifics for your PCB Design Parameters, the devices being driven by the FPGA, the parameters for the FPGA output loading and the FPGA bank allocation of resources. The calculator will determine what percentage of the WASSO limit is being utilized for each bank. If your design meets the WASSO requirements of the target device the green OK cells will be highlighted in the spreadsheet.

Power Supply/Decoupling Capacitor Considerations

Power supply design and the selection of decoupling components can be of critical importance to any subsystem with a combination of analog and digital functions like Ethernet. The key considerations can be grouped into these categories: Power Supply, PCB Decoupling Capacitors, PCB Bypass Capacitors and PCB Bulk Capacitors.

Power Supply

- Ensure adequate power supply ratings. Verify that all power supplies and voltage regulators can supply the amount of current required.
- Power supply output ripple should be limited to less than 50 mV.
- Noise levels on all power planes and ground planes should be limited to less than 50 mV.
- Ferrite beads should be rated for 4 – 6 times the amount of current they are expected to supply. Any de-rating over temperature should also be accounted for.

PCB Decoupling Capacitors

- Every high-speed semiconductor device on the PCB assembly requires decoupling capacitors. One decoupling cap for every power pin is necessary.
- Decoupling capacitor value is application dependent. Typical decoupling capacitor values may range from 0.1 µF to 0.001 µF.
- The total decoupling capacitance should be greater than the load capacitance presented to the digital output buffers.
- Typically, Class II dielectric capacitors are chosen for decoupling purposes. The first choice would be an X7R dielectric ceramic capacitor for its excellent stability and good package size vs. capacitance characteristics. Low inductance is of the utmost importance when considering decoupling capacitor characteristics.
- Each decoupling capacitor should be located as close as possible to the power pin that it is decoupling.
- All decoupling capacitor leads should be as short as possible. The best solutions are plane connection vias inside the surface mount pads. When using vias outside the surface mount pads, pad-to-via connections should be less than 5 – 10 mils in length.
- Trace connections should be as wide as possible to lower inductance.

PCB Bypass Capacitors

- Bypass capacitors should be placed near all power entry points on the PCB. These caps will allow unwanted high-frequency noise from entering the design; the noise will simply be shunted to ground.
- Bypass capacitors should be utilized on all power supply connections and all voltage regulators in the design.
- Bypass capacitor values are application dependent and will be dictated by the frequencies present in the power supplies.
- All bypass capacitor leads should be as short as possible. The best solutions are plane connection vias inside the surface mount pads. When using vias outside the surface mount pads, pad-to-via connections should be less than 5 – 10 mils in length. Trace connections should be as wide as possible to lower inductance.

PCB Bulk Capacitors

- Bulk capacitors must be properly utilized in order to minimize switching noise. Bulk capacitance helps maintain constant DC voltage and current levels.
- Bulk capacitors should be utilized on all power planes and all voltage regulators in the design.
- All bulk capacitor leads should be as short as possible. The best solutions are plane connection vias inside the surface mount pads. When using vias outside the surface mount pads, pad-to-via connections should be less than 5 – 10 mils in length. Trace connections should be as wide as possible to lower inductance.
- Whenever a ferrite bead is implemented, bulk capacitance must be used on each side of the ferrite bead.

Board Layout and Component Placement Considerations

Proper board layout and component placement considerations are important to any high-speed mixed analog/digital design. The following methodology was used on the layout of the Nu Horizons Spartan-3A DSP Reference Kit. Included with the purchase of the Reference Kit is a complete documentation package that includes details on component placement and signal routing. This is an excellent way to get a head start on any new Spartan-3A DSP design.

The requirements for the layout of the signals connecting the Xilinx Spartan-3A DSP FPGA and the Micrel PHY are shown in Figure 5 below. These requirements are used to drive the layout of specific signals and must be checked against actual board signal routing results.

| Group # | Group Name | Signal Name | Z0 (ohms) | Preferred layers | Spacing within group(Air Gap)(mils) | Spacing with other traces (Air Gap)(mils) |
|-------------|-------------------|------------------------------------|-----------|----------------------------------|-------------------------------------|---|
| MII Signals | Transmit Data | O_MII_TXD[3:0]+ ST_MII_TXD[3:0] | 50 | Any layer with solid reference | 15 | 25 |
| | | CLK_I_MII_TXC | | | | |
| | Receive Data | ST_MII_RXD[3:0]+I_MII_RXD[3:0] | 50 | Any layer with solid reference | 15 | 25 |
| | | CLK_I_MII_RXC | | | | |
| MDI Signal | Differential Pair | MII_RX_P/N and MII_TX_P/N | 100 | S1/S2 layer with solid reference | 15 | 25 |

Notes:

1. Provide solid ground reference
2. Match transmit data with respect to transmit clock and receive data with respect to receive clock with in 100 mils
3. A single ground plane is recommended. Do not split ground plane into separate planes for analog, digital and power pins.
4. Route high speed signals above a continuous unbroken ground plane.
5. Route differential pair on the same PCB layer

Figure 5: Layout Recommendations for Ethernet PHY to Spartan-3A DSP Connections

Micrel has published an excellent application note on General PCB Design and Layout Guidelines, listed in the reference section of this document. These guidelines can be used to drive the PCB layout strategy.

PCB Layer Strategy

The following PCB layer guidelines when used as part of a comprehensive PCB layout strategy will help insure that high-speed signal integrity is preserved. These are not the only guidelines that should be used, since other considerations (cost, time to market, fabrication capabilities, etc) could dominate in any specific application. These are good examples however of general guidelines which could be a starting point for a comprehensive strategy.

General Rules

- Place components so as to avoid long loop traces.
- Choose a metal box to shield the printed circuit board.
- Use a ferrite core on the DC power cord to reduce EMI.
- Follow the guidelines to layout differential pairs, the ground plane, and high-speed signals.
- Provide controlled impedance on all clock lines and high-speed digital signals traces with right termination schemes to prevent reflection and ringing.
- Ensure that the power supply is rated for the application and optimized with decoupling capacitors.
- Keep power and ground noise under 50mV peak-to-peak.
- Ensure that the switching DC-DC converter is filtered and properly shielded as the DC-DC power converter can produce a great deal of EMI noise.
- Avoid via and pad in the path on any critical signal as via and pad will induce unwanted capacitance and inductance which can cause reflection and distortion.

Power Ground Rules

- Do not split the ground plane into separate planes for analog, digital, power pins. A single and contiguous ground plane is recommended.
- Route high-speed signals above a solid and unbroken ground plane.
- Fill copper in the unused area of signal planes and connect these coppers to the ground plane through vias.
- Stagger the placement of vias to avoid creating long gap in the plane due to via voids.

Analog VCC Plane

- Place and route analog components within the Analog VCC plane.

Digital VCC plane

- Place and route digital components within the Digital VCC plane.

Signal Ground

- The signal ground region should be one continuous and unbroken plane.
- Both analog (AGND) and digital (DGND) grounds should be directly connected to the signal ground plane.

Chassis Ground

The chassis ground and magnetics serve two purposes: they help to reduce EMI noise emissions from the signal ground plane to the PCB's external environment and also act as a shield to protect the PCB components from ESD.

- Place the chassis ground on all PCB layers and use connection mounting holes to join the chassis ground on different PCB layers.
- If the chassis ground on the PCB is directly connected to the

metal shield of equipment through the connection mounting holes, use a trench/moat to isolate the chassis ground plane from the signal ground plane.

- The chassis ground region should extend from the front edge of the PCB board (RJ45 connectors) to the magnetics and around the edge of the board.

Magnetic Noise Zone

- Void both power and ground planes on all PCB layers directly under the magnetics.
- Chassis ground should extend from the magnetics to the RJ45 connector.
- Do not route any digital signals between the PHY and RJ45 connector.

Differential Signal Layout

- Differential pair (TX+/- or RX+/-) should be routed away from all other signals and close together to use 5-mil trace width and 5-mil trace space in same length as possible with 100 ohms controlled trace.
- Keep both traces of each differential pair as identical to each other as possible.
- Route each differential pair on the same PCB layer.
- Route both TX+/- and RX+/- pairs as far as away from each other at least four times of 5-mil trace space as shown in Figure 6.

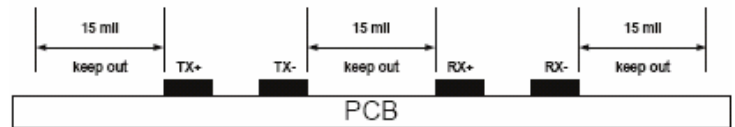


Figure 6: MDI Signal Layout Requirements

Additional PCB and layout guidelines are given in the Micrel application note. Refer to that note for additional suggestions that include PCB stacking, clock layout and ESD protection.

Micrel has published a detailed Design Kit for the KS8041NL that includes a board design with gerber files, IBIS models, Land Patterns and a schematic. A complete board user guide walks you thru the board design. The board's primary function is as an evaluation and test platform for the KS8041NL but some layout information is useful when comparing to an actual application. A picture of the Micrel evaluation test board is shown in Figure 7, below.

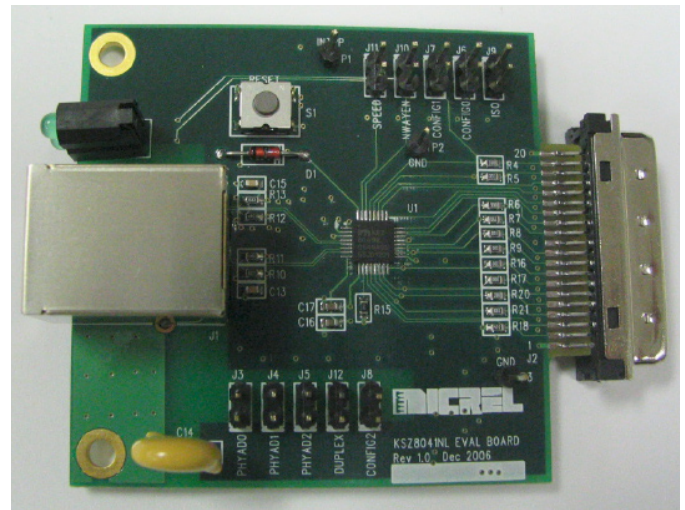


Figure 7: Micrel Evaluation Test Board

Real World Layout Example

The Nu Horizons Spartan-3A DSP Reference Kit has an excellent real world example layout for the Spartan-3A DSP xc3SD1800A device and the Micrel KS8041NL PHY device. Some post layout signal routing results are given in Figure 8 below. The routes between the PHY and the FPGA are allowed to be of reasonable distance. The key consideration is to route using equal lengths, using serpentine paths if necessary.

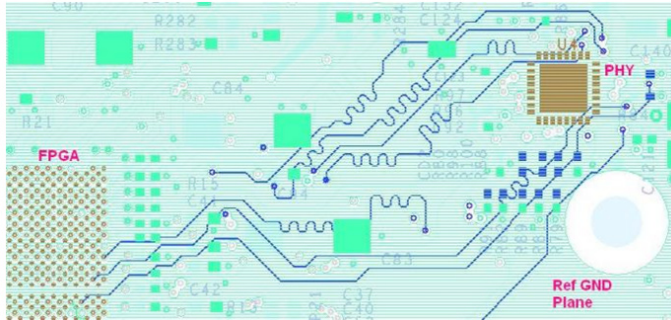


Figure 8: PHY to FPGA Signal Routing

The layout implementation between the PHY and the MDI component, illustrated in Figures 9 and 10 below, follows the guidelines by routing the differential signal pairs closely together with the same signal length. The keep-out signal distances recommended in Figure 6 have been used as constraints to make sure signal integrity is preserved. The use of vias was minimized to also maintain good signal integrity, as advised in the guidelines. The post layout check shows the following results:

- 1) The transmit data is length matched w.r.t transmit clock within +/- 70 mils
 - a. Requirement was within 100mils
- 2) The receive data is length matched w.r.t receive clock within +/- 70 mils.
 - a. Requirement was within 100mils
- 3) The number of vias used is 4. One at the BGA (FPGA), two at the series resistor and the other at the Ethernet PHY.

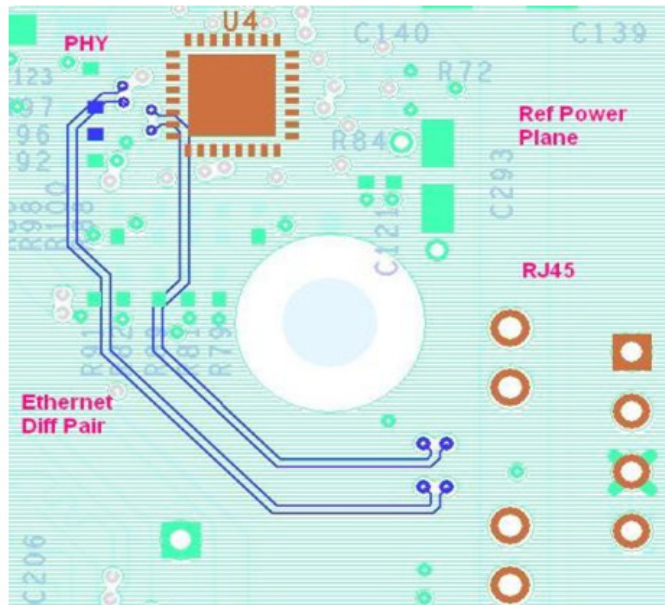


Figure 9: PHY to MDI Routing Layer 1

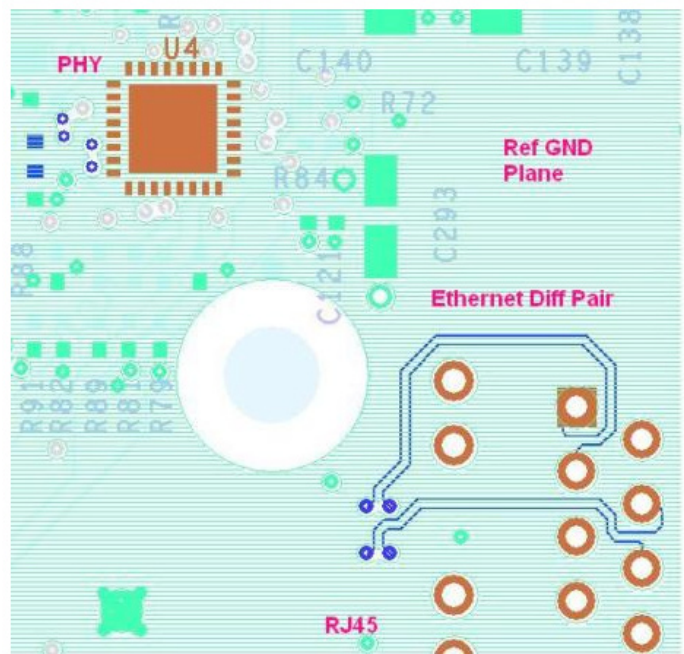


Figure 10: PHY to MDI Routing Layer 2

A final layout check will identify any errors between the constraints and final layout results. Extra care taken prior to building a board always pays off in the later stages of development.

Conclusion

This application note has provided a step-by-step guide to designing an Ethernet subsystem using Xilinx Spartan-3A DSP FPGAs, and a Micrel Ethernet PHY. The reader should be able to more easily implement these types of interfaces and better understand the major design considerations and concerns present in these types of designs. It is hoped that the reader can also extending the concepts presented in this app note to other Xilinx FPGAs and Micrel PHY devices and simplify those designs as well.

For the reader who wants additional levels of detail on the Xilinx, and Micrel products used in this app note a list of valuable documents is given in the Reference section at the end of this document. For a more hands on approach to learning more the reader is referred to the Nu Horizons Spartan-3A DSP Reference Kit. This kit is a complete design development platform and features a Xilinx xc3SD1800A FPGA, and Micrel PHY as well as a host of other useful interfaces (DDR memory, audio, video, radio, camera, and Digital to Analog Converters. You can use this platform to quickly implement and test your application and leverage a wide range of example designs, application notes and IP included with the kit. Point your browser to <http://www.nuhorizons.com/xilinx/boards/spartan3a/starterkit.asp> to learn more about this powerful development platform.

References

- 1) Xilinx Spartan3-A DSP Data Sheet (DS610 June 2, 2008)
http://www.xilinx.com/support/documentation/data_sheets/ds610.pdf
- 2) Nu Horizons Spartan-3A DSP Reference Kit
- 3) Xilinx WASSO Calculator and application note (xapp689, Rev 1.2)
http://www.xilinx.com/support/documentation/application_notes/xapp689.pdf
- 4) Xilinx XPS Ethernet Lite Media Access Controller Product Specification (v1.00a, Oct 31, 2006)
http://www.xilinx.com/support/documentation/ip_documentation/xps_ethernetlite.pdf
- 5) Micrel Ethernet PHY Data Sheet (April 2007, Rev 1.1)
http://www.micrel.com/page.do?page=product-info/fastether_trans.jsp
- 6) Micrel General PCB Design and Layout Guidelines (AN-111, Feb 2007)
http://www.micrel.com/_PDF/Ethernet/app-notes/an-111.pdf
- 7) Micrel Design Kit for KS8041NL
http://www.micrel.com/page.do?page=product-info/fastether_trans.jsp

Notice Of Disclaimer

Nu Horizons is disclosing this Application Note to you "AS-IS" with no warranty of any kind. This Application Note is one possible implementation of this feature, application, or standard, and is subject to change without further notice from Nu Horizons. You are responsible for obtaining any rights you may require in connection with your use or implementation of this Application Note. Nu Horizons makes no representations or warranties, whether express or implied, statutory or otherwise, including, without limitation, implied warranties of merchantability, noninfringement, or fitness for a particular purpose. In no event will Nu Horizons be liable for any loss of data, lost profits, or for any special, incidental, consequential, or indirect damages arising from your use of this application note.