User Manual to the Gigabit Ethernet Controller

Written by Ryan Rivera
rrivera@fnal.gov
Last Updated: 12/17/2007

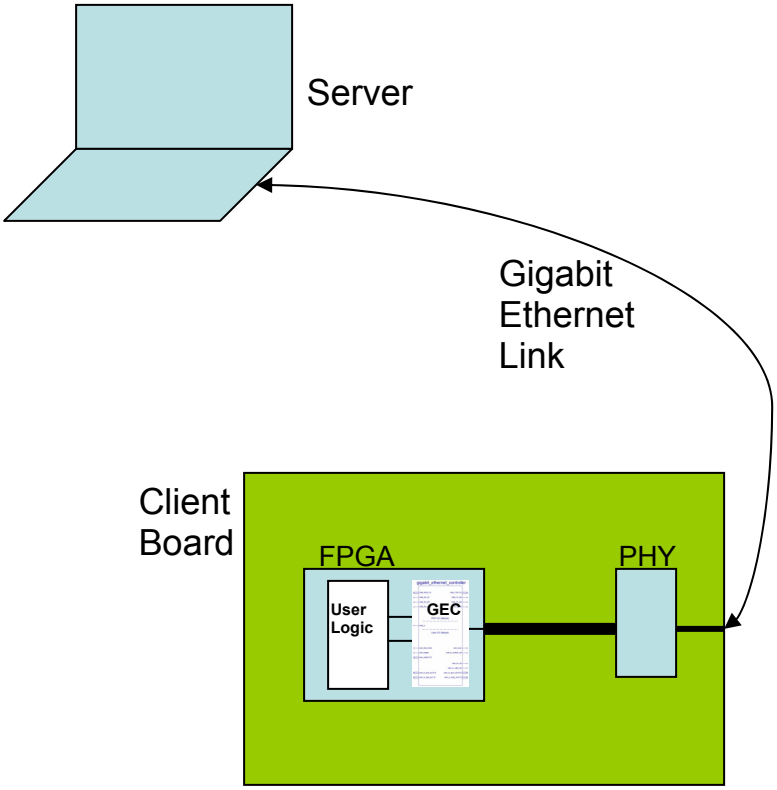


Fig. 1. System overview of a gigabit Ethernet solution containing the Gigabit Ethernet Controller (GEC).

Introduction to the GEC

The Gigabit Ethernet Controller (GEC) is a firmware block that provides the user with a simplified interface to a GMII capable PHY without utilizing any vendor specific FPGA components. The GEC was designed such that only UDP packets can be sent and received – this lends to the simplicity of the GEC. This also places the burden of packet order verification and retransmission at the application level.

The user need only interface to the 12 *User I/O* ports of the GEC. They are described in the next section. The remaining ports interface directly with the PHY using the GMII protocol. The GEC operates on the assumption that communication only takes place with a single server entity. Therefore it does not input a destination address; the GEC gets the destination address for outgoing packets from incoming UDP packets.

I/O Descriptions

Inputs	
reset_n	Synchronous active low reset. Must reset block before use to initialize GEC.
user_test_mode	When '1' the GEC is in *Test Mode*. When '0' the GEC is in *User Mode*.
user_trigger	If *user_busy* is '0', a rising *user_trigger* edge causes the GEC to send either a test or user UDP packet, depending on the mode.
user_addrs(7:0)	In *User Mode*, *user_addrs* determines the lowest order byte of the firmware's IP and MAC address.
user_tx_size_in(10:0)	In *User Mode*, *user_tx_size_in* sets the amount of data in bytes to be sent as the payload of a UDP packet.
user_tx_data_in(7:0)	In *User Mode*, *user_tx_data_in* is the byte-by-byte payload for the UDP packet to be transferred.
Outputs	
user_busy	When '1', the GEC will not respond to *user_trigger* edges.
user_tx_enable_out	In *User Mode*, a '1' indicates to the user that byte-by-byte data must be present on *user_tx_data_in* at the next rising clock edge for the most recently triggered UDP packet. This enable signal will only be '1' for the *user_tx_size_in* number of clock periods that user data is expected.
user_crc_err	If a CRC error is detected in a received packet, this signal will pulse high for one cycle no later than 22 clock cycles after the *user_rx_valid_out* signal returns low. A CRC error indicates to the user that the received data may contain bit errors.
user_rx_valid_out	'1' indicates that a UDP packet has been received and data is ready. The size of the received packet is immediately present on *user_rx_size_out* and the byte-by-byte data will be valid at each rising clock edge while *user_rx_valid_out* is high. This duration will be equal to *user_rx_size_out* clock cycles.
user_rx_size_out(10:0)	When *user_rx_valid_out* is '1', *user_rx_size_out* is the size in bytes of the received UDP packet.
user_rx_data_out(7:0)	When u*ser_rx_valid_out* is '1', *user_rx_data_out* is the byte-by-byte data of the received UDP packet.

The Basics

Before any packets can be sent or received there are a couple fundamental items to understand. First, where does the user clock originate? The clock that synchronizes all processes actually comes from the PHY as an input to the FPGA: the *GMII_RX_CLK*. It

is a 125 MHz clock. The very same clock is also returned to the PHY via the *GTX_CLK*. So either of those two clocks (they are actually the same) may be taken by the user as the system clock.

Second, what are the addresses of the system components? The value of *user_addrs* becomes the lower byte of both the user IP and MAC addresses. Therefore the complete user IP and MAC addresses are *192.168.133.X* and *00.80.55.EC.00.X* respectively, where *X* is the value of *user_addrs*.

The *user_addrs* values of 0 and 255 are invalid. In *Test Mode* the *user_addrs* is internally forced to the value of *3*. The server's IP address must always be set to *192.168.133.1.*

After reset and once the addresses are set, the GEC is ready to start sending and receiving UDP packets.

Receiving Packets

Immediately after reset, in the presence of a stable *user_addrs*, the GEC is ready to receive UDP packets and ARP requests from the server. At least one UDP packet must be received before a valid UDP packet can be transmitted. ARP requests are handled automatically and are not the responsibility of the user. Packets of types other than ARP or IPv4 are ignored.

When a valid UDP packet is received with the destination address matching that of the GEC, the *user_rx_valid_out* signal will transition from low to high. On *user_rx_valid_out*'s rising edge, *user_rx_size_out* has the size in bytes of the received packet and is guaranteed to be valid until *user_rx_valid_out* clears.

On the first rising clock edge after *user_rx_valid_out* is set, the first received payload byte can be latched from *user_rx_data_out*. Byte by byte the received data will be available on rising clock edges while *user_rx_valid_out* is high. The user should use *user_rx_size_out* to determine when the last byte has been received.

No later than 22 clock cycles after the last byte has been received the error status of the received packet is revealed to the user in the form of a pulse on *user_crc_err* as shown in Figure 2.

The packet receiving process is the same in *Test Mode* as in *User Mode*.

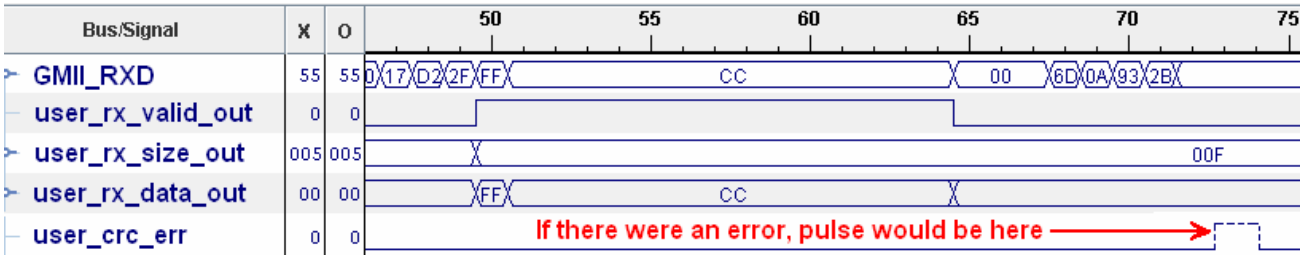


Fig. 2. The figure shows the timing of the user signals with regard to a received UDP packet. This is a snapshot of an actual 15 byte packet being received. The payload's first

byte is *0xFF* and the remaining 14 are *0xCC*. The timeline ticks indicate the rising edge of the system clock.

Sending Packets

The GEC can send UDP packets anytime after the first UDP packet is received. This limitation is necessary to acquire the server's IP/MAC address and port to fill the destination field of outgoing packets.

There are two modes available with the GEC. There is *Test Mode* and *User Mode*. *Test Mode* allows the user to receive UDP packet payloads of sizes ranging from 1 to 1472 bytes, but transmit only test packets. When in *Test Mode*, the only active user input is *user_trigger*. The firmware's IP address is forced to *192.168.133.3* and its MAC address to *00.80.55.EC.00.03*.

If a rising edge is input on the *user_trigger* line, a five byte UDP packet is sent containing *0x41 42 43 44 45*, which are the ASCII hexadecimal values for the character sequence, *ABCDE*.

User Mode allows the user to receive and transmit UDP packet payloads of sizes ranging from 1 to 1472 bytes. The user also may specify the address of the firmware. When in *User Mode* the *user_addrs* line specifies the lower byte of both the IP and MAC addresses.

To send a user specified packet, the user must first setup *user_tx_size_in* with the number of bytes to be sent. Then a rising edge on the *user_trigger* line will initiate the transmission of the packet. Once the *user_tx_enable_out* signal is high, the user should provide data, byte-by-byte through *user_tx_data_in*, on each rising clock edge starting with the first edge after the rise of the enable signal. This is done until all of the bytes have been processed. A timing example is shown in Figure 3 for five bytes of user data.

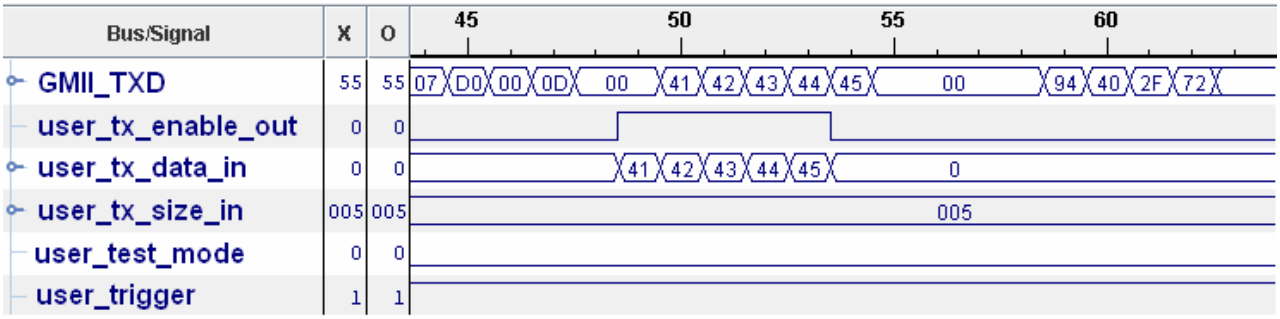
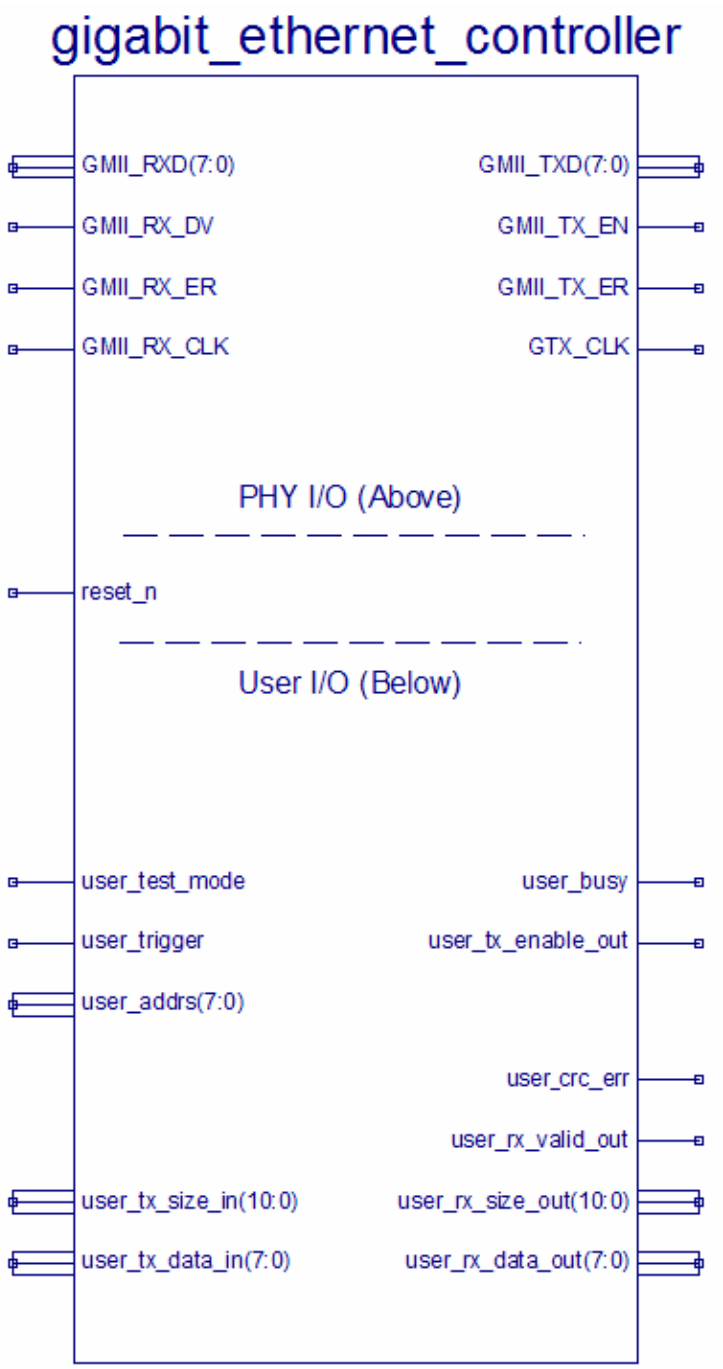


Fig. 3. The figure shows the timing of the user signals with regard to a transmitted UDP packet in *User Mode*. The *user_trigger* rising edge and setup of *user_tx_size_in* occurred earlier than what is shown. The timeline ticks indicate the rising edge of the system clock.

Appendix A: Xilinx ISE GEC Symbol



Appendix B: Getting Started with Xilinx ISE 9.2

The following steps should be taken to get the sample project up and running:

1. Download the GEC zip folder. http://www-ese.fnal.gov/DIG_Test_Stand/firmware/GEC_files.zip. Extract all.
2. Open Xilinx ISE 9.2 and select File → New Project…
3. Enter a Name and Location for your Project. Set Top-Level Source Type to Schematic. Click Next.
4. Select the proper Device and Design Flow. The Synthesis tool must handle VHDL/Verilog. Click Next.
5. Click Next through Create a New Source.
6. In Add Existing Sources click Add Source. Select all the files in the folder *GEC_files*. Make sure the Copy to Project box is checked for all. Click Next.
7. Click Finish.
8. *TOP_LEVEL* is a schematic. Replace the *fake_user_data* block and add receive buffers and logic as desired.