# *Altera FPGA final laboratory*

## *XGMII Interface Implementation*

The FPGA performs some of the Multi-Protocol Label Switching (MPLS) packet processing. The MPLS FPGA connects to a MAC device on the PHY side. The XGMII block provides a simple ingress/egress interconnection between the MAC sub-layer (XAUI converters) and the PHY (FPGA) at rate of 10Gbps in each direction. We deal with receive side only.

- XGMII interface works at 32bit@156.25 MHz DDR.
- The XGMII block interface to the FPGA fabric works at 128bit@150 MHz.
- In the receive direction the block removes the packet preamble, Start-of-frame-delimiter and the End-of-frame-delimiter bytes.

Data frames transmitted through the XGMII shall be transferred within the XGMII data stream. The data stream is a sequence of bytes, where each byte conveys either a data octet or control character. The parts of the data stream are shown in Figure 1.

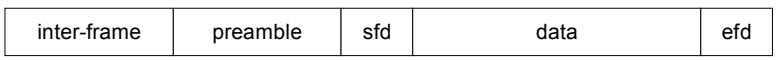



**Figure 1 – XGMII data stream**

***Inter Frame***: The inter-frame period on an XGMII transmit or receive path is an interval during which no frame data activity occurs. The inter-frame corresponding to the MAC inter-packet gap begins with the Terminate control character, continues with idle control characters and ends with the idle control character prior to a Start control character. The length of the inter-packet gap may be changed between the transmitting MAC and receiving MAC by one or more functions (e.g., RS lane

alignment, PHY clock rate compensation, or 10GBASE-W data rate adaptation functions). The minimum IPG at the XGMII of the receiving RS is five octets.

***Preamble and start of frame delimiter***: The preamble begins a frame transmission by a MAC and when generated by a MAC consists of 7 octets with the following bit values:

**10101010 10101010 10101010 10101010 10101010 10101010 10101010**

The Start control character indicates the beginning of MAC data on the XGMII. On transmit, the RS converts the first data octet of preamble transferred from the MAC into a Start control character. On receive; the RS will convert the Start control character into a preamble data octet. The start control character is aligned to lane 0 of the XGMII by the RS on transmit and by the PHY on receive.

The start of frame delimiter (sfd) indicates the start of a frame and immediately follows the preamble. The bit value of sfd at the XGMII is the bit sequence:

**10101011**

The preamble and SFD are shown previously with their bits ordered for serial transmission from left to right. As shown, the left-most bit of each octet is the LSB of the octet and the right-most bit of each octet is the MSB of the octet. The preamble and SFD are transmitted through the XGMII as octets sequentially ordered on the lanes of the XGMII. The first preamble octet is replaced with a Start control character and it is aligned to lane 0, the second octet on lane 1, the third on lane 2 and the fourth on lane 3, and the four octets are transferred on the next edge of TX_CLK. The fifth octet is assigned to lane 0 with subsequent octets sequentially assigned to the lanes with the SFD assigned to lane 3. The XGMII preamble and sfd are:

| **Lane 0** | **Lane 1** | **Lane 2** | **Lane 3** |
|---|---|---|---|
| Start | 10101010 | 10101010 | 10101010 |
| 10101010 | 10101010 | 10101010 | 10101011 |

***Data***: The data in a well-formed frame shall consist of a set of data octets (8 bit).

***End of frame delimiter***: Assertion of TXC with the appropriate Terminate control character encoding of TXD on a lane constitutes an end of frame delimiter efd for the transmit data stream. Similarly, assertion of RXC with the appropriate Terminate control character encoding of RXD constitutes an end of frame delimiter for the receive data stream. The XGMII shall recognize the end of frame delimiter on any of the four lanes of the XGMII.

In the receiver side, XGMII format packets enter from 32bit @ 156.25 MHz DDR bus. The 32bit bus is converted to 64bit @ 156.25 SDR bus. The conversion is done by using the DDR capabilities of the FPGA IOs.

The XGMII overheads from packets are removed and the packet is written into 64bit-to128bit width-conversion DPRAM. From the 128bit port of the DPRAM, packets are transferred to the FPGA fabric at rate of 150 MHz.

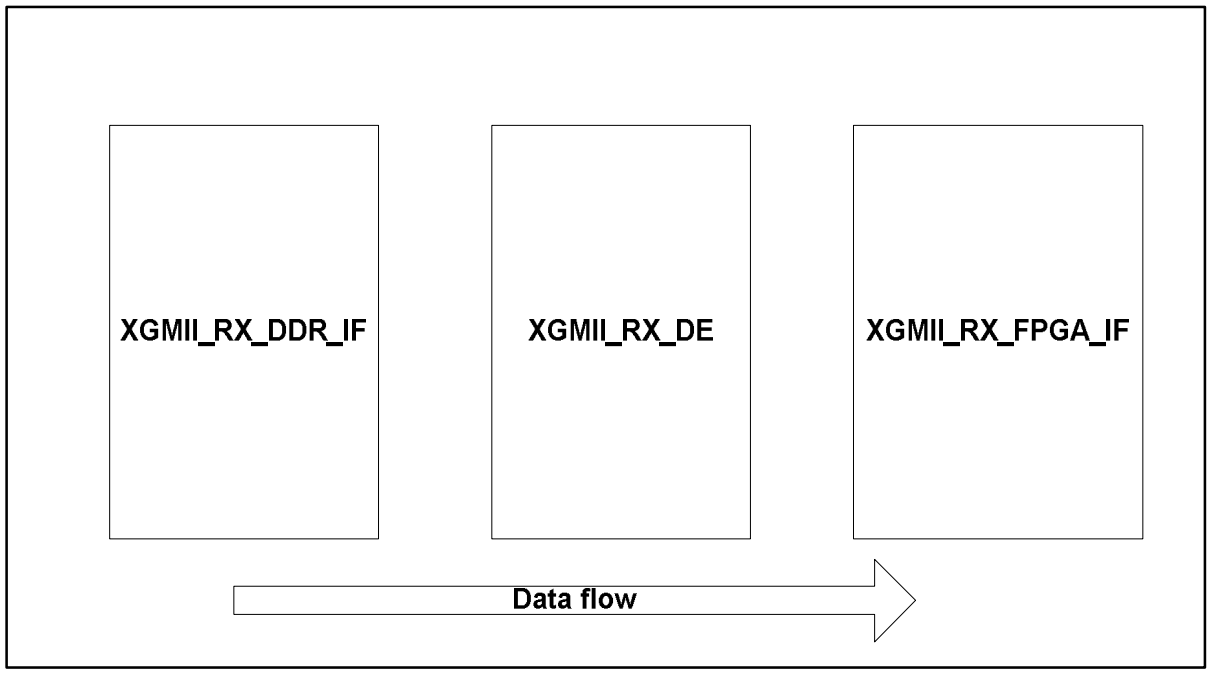



**Figure 2 – XGMII data stream**

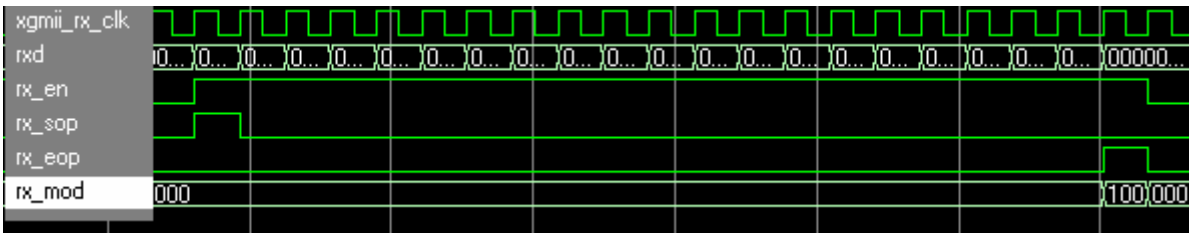
We deal with 64 bit  to 128 bit converter block.

This block receives the De-encapsulated data words in 64bit width and writes then into 64bit-to-128bit DPRAM. Due to the variable nature of the packets, a packet descriptor is written for each packet. The descriptors are store in separate FIFO.

The packet descriptor contains information about the number of 128bit words that the packet occupies in the DPRAM (rounded to the nearest 64bit word that will align the last write to 128bit word boundaries) and the number of valid bytes within the last 128bit word.

Write side interface:

The signal RX_EN – indicates when RXD[63:0] contains the packet payload. The signal RX_SOP is asserts for one clock period when RX_EN is asserted. The signal RX_EOP is asserts when RX_EN = '1' and the first all-lines idle sequence (into the next data word) detected. I.e., RXD[63:0] = 0x0707070707070707. The signal RX_MOD[2:0] loads the value of the number of valid bytes within the last 64bit word.




Read side interface:

Packet_mod[3:0] – The number of valid bytes within the last 128bit word.

| Packet_mod[3:0] | Valid 8bit lanes |
|---|---|
| 0000 | [127:120],[119:112],[111:104],[103:96],[95:88],[87:80],[79:72];[71:64]<br>[63:56],[55:48],[47:40],[39:32],[31:24],[23:16],[15:8],[7:0] |
| 0001 | [127:120],[119:112],[111:104],[103:96],[95:88],[87:80],[79:72];[71:64]<br>[63:56],[55:48],[47:40],[39:32],[31:24],[23:16],[15:8] |
| 0010 | [127:120],[119:112],[111:104],[103:96],[95:88],[87:80] |

| | |
|---|---|
| | ,[79:72];[71:64]<br>[63:56],[55:48],[47:40],[39:32],[31:24],[23:16] |
| 0011 | [127:120],[119:112],[111:104],[103:96],[95:88],[87:80]<br>,[79:72];[71:64]<br>[63:56],[55:48],[47:40],[39:32],[31:24] |
| 0100 | [127:120],[119:112],[111:104],[103:96],[95:88],[87:80]<br>,[79:72];[71:64]<br>[63:56],[55:48],[47:40],[39:32] |
| 0101 | [127:120],[119:112],[111:104],[103:96],[95:88],[87:80]<br>,[79:72];[71:64]<br>[63:56],[55:48],[47:40] |
| 0110 | [127:120],[119:112],[111:104],[103:96],[95:88],[87:80]<br>,[79:72];[71:64]<br>[63:56],[55:48] |
| 0111 | [127:120],[119:112],[111:104],[103:96],[95:88],[87:80]<br>,[79:72];[71:64]<br>[63:56] |
| 1000 | [127:120],[119:112],[111:104],[103:96],[95:88],[87:80]<br>,[79:72];[71:64] |
| 1001 | [127:120],[119:112],[111:104],[103:96],[95:88],[87:80]<br>,[79:72] |
| 1010 | [127:120],[119:112],[111:104],[103:96],[95:88],[87:80] |
| 1011 | [127:120],[119:112],[111:104],[103:96],[95:88] |
| 1100 | [127:120],[119:112],[111:104],[103:96] |
| 1101 | [127:120],[119:112],[111:104] |
| 1110 | [127:120],[119:112] |
| 1111 | [127:120] |

The others signals of write side is eop- end of packet, ena- packet enable, sop – start of packet and data[127:0].

The implementation based on DPRAM, FIFO and PLL which one created by MegaWizard Plug-in Manager.