10 Gigabit Ethernet TCP Frame Data Registration Algorithm Based on FPGA

Xia Yang, Chen YueYang*

Beijing Institute of Technology, Beijing, 100081, P.R. China

Keywords: 10GigE, TCP, FPGA, data registration.

Abstract. Using FPGA to achieve full hardware protocol stack in 10 Gigabit Ethernet (10GigE) has the following advantages, such as low cost, wide range of application, high efficiency, etc. Compared with gigabit Ethernet, 10GigE adopts wider data width, which makes hardware face the difficulty of data registration for handling intra-frame data. This paper designed “FIFO form circular-queue handling TCP frame data registration algorithm”, solved the problem of data registration when FPGA device receives, stores and sends TCP frame data and adopted FPGA to build practical system to realize TCP through hard core coding. At the same time, this algorithm can also be used for the data registration of other protocol in 10GigE environments.

Introduction

10GigE has such advantages as high transmission efficiency and compatibility. With the continuous development of network technology, various costs of Ethernet have greatly reduced and the range of application has gradually enlarged. Currently, Ethernet has been widely applied in several industries, such as communication, finance, education, broadcast, etc.[1]

In 10GigE network environment, using FPGA to realize full hardware protocol has the following advantages: 1. 10-gigabit Ethernet poses great pressure on general PC. Strong PC handling capacity and storage performance are required to handle and analyze the protocol stack of 10-gigabit Ethernet. However, FPGA has integrated functions of receiving, analyzing, handling and transmitting protocol stack, which greatly saves hardware cost. 2. FPGA is programmable and configurable, which can be applied in different network environment. 3. Hard core coding realizes protocol stack, which can reduce the amount of chip and has higher implementation efficiency.

TCP (Transmission Control Protocol) is a connection-oriented, reliable, byte stream based transmission layer communication protocol, which is commonly used in network transmission. These mechanism, such as window confirmation ensure the reliability of TCP [2]. The implementation of full hardware TCP is an important part in the implementation of full hardware protocol.

This paper describes the technological difficulty resulting from the application of 64-bit data width in 10GigE. The design of “FIFO forming circular-queue handling TCP frame data registration algorithm” solves the stitching problem of TCP intra-frame data. Practical system is built and full hardware TCP is implemented on the FPGA device of Xilinx.

Technological difficulty

Gigabit network is for transmission 8bit data per clock cycle. 10GigE is for transmission 64bit data per clock cycle. In 10GigE network environment, TCP data frame format is as shown in the following Figure.
Algorithm Analysis

Key Problem. The 8-byte data received within a clock cycle is stitched and stored with the previous data, or the stored 8-byte data is split and sent.

Algorithm Introduction. In FPGA device, FIFO(First Input First Output) is often adopted as the temporary storage unite of data. Take storing TCP frame data as an example, form circular queue by using 8 8-bit FIFOs and store the 8-bit data received within a clock cycle. Set a “write data start position” pointer $p$ for controlling the start number of stored FIFO. Set a “residual frame length” variable $len$ for recording the length of data required to be stored. Set a “width to be written” variable $n$, denoting the quantity of FIFO to be written this cycle. The block diagram of algorithm is as shown in following Figure.
After system initialization is completed, set write data start position pointer \( p \) at FIFO1, when FPGA device receive a TCP data frame, set the residual frame length equal to data length read from TCP frame header.

If the current cycle is the first cycle of TCP frame data part, name the width to be written \( n=2 \) (in this cycle the first 6 bytes is the tail of TCP frame header and the last 2 bytes is the head of the data parts), writing the 2 bytes data into FIFOP-FIFOP+2 successively, setting the pointer \( p \) at \( p+2 \), if the current cycle is not the first cycle of data part, judge the radial frame length.

If the radial frame length is greater or equal to 8, the width to be written \( n \) is 8, write 8 8-bits(totally 8 bytes) data into FIFOP-FIFOP+1 successively in a circular manner, keep the location of pointer unchanged, \( len=len-8 \). If \( len \) is less than 8, then \( n \) equals to \( len \), write \( n \) 8-bits(totally \( n \) bytes) data into FIFOP-FIFOP+N successively, set the pointer at \( p+n \) and reset \( len \). During the implementation of the algorithm, the operation of pointer \( p \) shall ensure that the value is less than or equal to 8 (if the value is more than 8, 8 shall be taken from the value) to ensure circular queue.

The transmission and receiving principle of TCP data is in the same mode. The algorithm for sending is performed by using read data start bit pointer, residual transmission length and the length to be read.

This algorithm is designed for the stitching problem of the data received, stored and transmitted by TCP frame. In fact, for stitching the other data in 10GigE, the storage method of this algorithm can be adopted and the stitching has been completed for read and store multi-frame data part successively; for frame header transformation between different protocols, the transmission method of this algorithm can be adopted, after new protocol frame header is generated, based on the residual data width beyond new frame header, read the specific-width data. Then the transformation of frame header can be completed. Namely, this algorithm can be applied for solving the stitching problem of various protocols in 10GigE.
Verification of Algorithm

**Verification of System Building.** PC and FPGA adopt 10GigE fiber module (AFBR-703SDZ 10Gbps fiber module manufactured by Avago) as access network interface and transmit transmission signal to FPGA system by using 10GigE interface (88X2011 chip manufactured by Marvell). The transmitted differential signal is converted into 64-bit digital signal by FPGA coding. Then after the availability of the data frame is verified by GMAC module, the digital signal becomes the data form which can be operated by using FPGA coding.

![Block Diagram of the algorithm verification system.](image)

**Verification Results.** By writing practical hard core code, we design two versions of FPGA code with and without data registration algorithm. After comparison tests, we find that by using the algorithm, the function of the TCP realization on FPGA can be fully achieved, which improve the communication efficiency between PC and FPGA.

<table>
<thead>
<tr>
<th></th>
<th>The limit of frame length</th>
<th>Sending and receiving mode</th>
<th>retransmission timeout function</th>
<th>window confirmation function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Without algorithm</td>
<td>Only specified frame length</td>
<td>Single frame transmission</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>With algorithm</td>
<td>Arbitrary frame length</td>
<td>Continuous multi frame transmission</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Under the same verification system, this algorithm is adopted to the further coding on FPGA. And the UDP frame data stitching and inter-conversion between UDP and TCP frame header are completed.

**Conclusion**

By using “FIFO forming circular-queue for handling TCP frame data registration algorithm”, the following problems are solved, such as the receiving, storing, stitching and transmission of TCP frame data in 10GigE and various functions specified by TCP stack can be completely realized, which makes it possible for the FPGA device to implement full hardware TCP in 10GigE environment. Solutions can also be provided for the data stitching of other protocols and different protocol frame header conversion.

**References:**