

A Novel Transceiver Architecture for High-Speed Parallel Ethernet Network

Rand Basil Mohammed and Dr. Roelof van Silfhout (Authors)

Abstract— The Ethernet network is currently used as the common way to transfer data between devices. As with all transceiver systems the transmission speed is the most critical factor. For high-resolution digitised images, for example, such network connections are often a bottleneck. The main objective of this paper is to create a full transceiver device as part of an embedded system, at both source and host devices that provide a high transmission speed. The main idea is to design a new system architecture, called Novel Reference Module (NRM), for both transmitter and receiver, based on four parallel Ethernet (PE) networks placed at the bottom of the system architecture, that connect to the devices at the opposite side. The four Ethernet ports connect to each other across the network media using a point-to-point network topology. Due to the high bandwidth it is not possible to use embedded processors to optimise the bandwidth for all network ports simultaneously. To this end we have developed a dedicated controller that controls the data flow and maximising the use of bandwidth. The Parallel Ethernet system builds as a software using Qsys builder, which is provided in Quartus II software from Altera, and tested using the SignalTap Analyzer tool, which is also provided by Altera. With this tool we demonstrate our embedded system using a StratixIV GX on DE4 board, which is provided by Terasic.

Keywords—High-speed transceiver; parallel ethernet networks; novel reference module; embedded system; StratixIV GX.

I. Introduction

The Ethernet network system is one of the most common ways to transfer data between devices, these transceiver systems used to have a special architecture that helps to carry data between devices. The transceiver architecture based on divided the transmitter and receiver devices into multi sub-layers, which known as reference module. The main concept for this work is building a novel reference module, as will describe later. All works had been done from other researchers with similar concept were focusing on using a regular reference module called ISO/OSI or

TCP/IP [1-3], this ISO standard is based on dividing the transmitter and receiver devices into seven layers each one has a specific purpose, as shown below in Fig. 1., this reference module requires a specific protocol for transmitter and receiver, each layer has its own, for example the transport layer based on TCP protocol, and a network layer based on the IP protocol. These protocols add a specific header and trailer information to each packet of data that read from the top layer, at the transmitter side, while at receiver side each layer removes the header and trailer information from each packet of data then forward it to the next layer.

At the bottom of the ISO reference module is the physical layer, which is fully implemented as hardware, this layer will specify the transmission speed for transferring data depending on the chosen protocol. For example when the physical layer transfers data to the Ethernet network, the protocol standard used is called IEEE802.3, within the Ethernet protocol there is a wide range of speeds supported by IEEE. For example IEEE802.3a is the standard protocol to support transmission speed up to 10Mbps, while IEEE802.3ab is a protocol that supports transmission speed up to 1Gbps[4-6].

Depending on a standard architecture and computing system, a designer has to take many steps to build it. In realizing the network appliance a designer requires to provide each layer with its architecture. One of the main benefit of following the OSI/ISO standard is, there are more than one program can read the data at receiver side, which gives opportunity to build just transmitter and use one of exciting project to read data at receiver side.

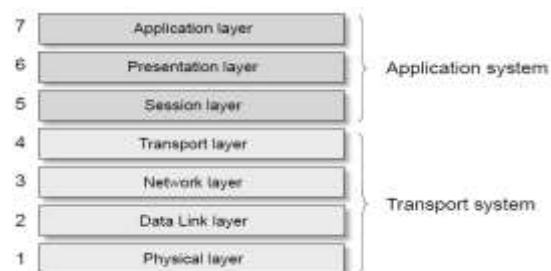


Fig.1. ISO/OSI Reference module layers [6]

Rand Basil Mohammed (Author)

School of Electrical and Electronic Engineering
The University of Manchester
Manchester, United Kingdom

Dr. Roelof van Silfhout (Author)

School of Electrical and Electronic Engineering
The University of Manchester
Manchester, United Kingdom

On the other hand it also has disadvantages; the transceiver speed is fixed by specifying a protocol and cannot increase it until built a new full system, for example the next speed for the 1G will be 10Gbps which is expensive and complicated when provided it to the system; and the data have been sent will not be secured because any random user can receive it; these two weaknesses in OSI/ISO module we are focusing on them and try to solve with this paper, by giving alternative reference module. The full description for this alternative and powerful solution will present later in this paper.

Therefore, this work proposes a new system architecture build instead of regular one OSI/ISO reference module, called Novel Reference Module (NRM); the main idea for this reference module is supporting a high transmission speed with lower cost, by dividing the transmitter architecture into two layers, one is a fully software and the other is partially software and partially hardware, the top part reads data from the source device and the lower part has four sub designs. Each sub design acts as an individual transceiver device, all the devices together will access the same source location and transfer the same amount of data at the same time, which makes the data transfers four times faster than the regular one. Each one of these devices can support speeds up to 1Gbps, which means a new reference module can provide a user with 4Gbps as a transmission speed; the new reference module is shown below in Fig.2.

The transmitted data uses no regular protocol; data will be handled by the user only when it provides the opposite side, receiver side. The receiver side has the same system architecture as transmitter and it is going to do the opposite operation. This means collecting the data from the four parts, then removing the extra information and storing them at storage side in same order that have been sent. This new system has an additional advantage, which is not offered by other systems, by giving the opportunity to the receiver to control more than one application at the same time, for example read data from one source device then each one can choose a different path which helps to do four different operations at the same time without losing any time or costing more. This will be helpful for a real-time applications; beside that this system helps to increase the transmission speed without the requirements to build a complicated system, by replicating the single system core with a number that need to support a higher transmission speed, for example replicate the 10Gbps system four times to get a transceiver system working with data rate equal to 40Gbp. Section II gives more information about the system methodology and how can achieve it, while Section III provides an information about results and discussion.

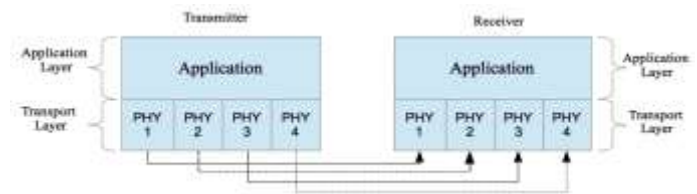


Fig.2. Novel Reference Module for Parallel Ethernet System at both Transmitter and Receiver sides.

II. Methodology

The article has two main concepts, the first one is the new system architecture, Novel Reference Module (NRM), and the second one is the Parallel Ethernet system (PE) for both transmitter and receiver. The NRM has two parts the top one is the application layer that reads data from the source devices and then stores the data in a temporary location before passing on to the next layer; the transport layer. The transport layer reads a data as a packets of 1024 long words (32bits) that are subsequently divided into four equal sub-packets with the same length but consisting of bytes. This layer also adds a header file to each packet before sending it out to the receiver through the network media. The header file helps the packet to find its direction at receiver side, by comparing the header file for each packet with the default setting at each receiver device.

The receiver system has two parts as shown in Fig.2., at the bottom the transport layer is shown, which is interfacing to the Ethernet media and takes the data from the transmitter side; inside this layer the header file of each packet of data is comparing with the default value at the receiver port settings, when the data arrives at a correct destination port, the header file removing and prepare the data to combined with the others port's data, each part of Parallel Ethernet System (PES) has 8bits as data width 8 bits. At the time of combining the data from all PES, it generates a bigger packet with 32 bits as data width with 1Kbits as length. After generates a bigger frame the data forwarding to the next layer, application layer, inside this layer the system is going to do the opposite for any operation has been applied at the transmitter side, then store the data at a storage device.

Fig.3. shows a schematic diagram for the Ethernet system, this transceiver system built and tested using Altera's software with its tools. All components, are using with this design, are built by write a Verily HDL codes then convert them to IP Components using Qsys™ builder, which is one of the Quartus II software tools.

The parallel Ethernet system is implemented by connecting four single Ethernet systems in parallel; the main purpose of using the parallel Ethernet system technique is to increase the transmission speed, which helps to carry a large amount of data during the same time period.

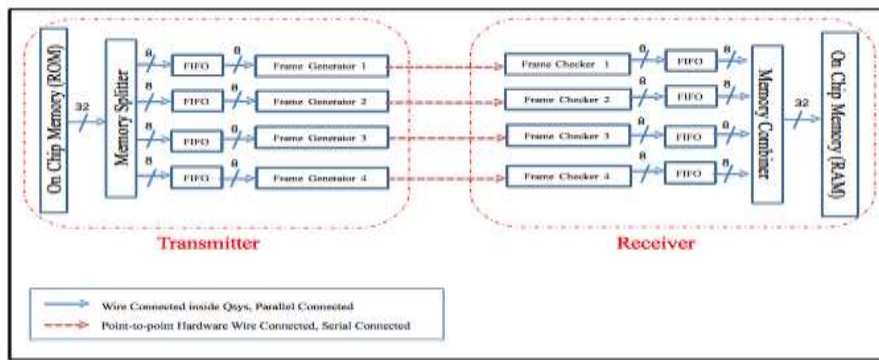


Fig.3. Parallel Ethernet System based on connecting four Single Ethernet Systems in parallel for supporting transmission speeds up to 4Gbps

The four Single Ethernet systems, shown in Fig. 3., are connected in parallel at both transmitter and receiver. Each one of these single Ethernet systems is a custom Ethernet core that supports speed up to 1Gbps.

This system contains many components such as: On-Chip Memory (at both transmitter and receiver), Splitter, Combiner, FIFO, Frame Generator and Checker, all these component, except on-chip memory and FIFO, are built as a custom IP components; below is a brief description for each component: Storage Devices, are places that contain data; the On-Chip Memory is used as a storage device at both sides, transmitter and receiver; on the transmitter side the system reads data from on-chip memory, while on the receiver side the system writes the data inside it. Compared with the Novel Reference Module architecture, the Application Layer is presented as On-Chip memory with Memory-Splitter at Transmitter side and On-Chip memory and Memory Combiner at Receiver side; while the FIFO with Frame Generator and Frame Checker presents the transport layers at both transmitter and receiver respectively.

On the transmitter side, the system will access the same memory bank that holds the 32-bit data width each clock cycle, and each of four transmitters will be responsible for carrying one byte making up the single 32-bit data. For example, a first byte has been carried by Ethernet system 1, the second byte by system 2 and so on; to divide the reading frame into four parts requires an intermediate component called a Memory Splitter inserted between the memory and generator devices.



Fig.4. Diagram describing the operation that divides 32 bit into four 8 bits on the transmitter side then combines them on the receiver side.

The main job for Memory Splitter is taking the data from memory, dividing it into four parts, and then forwards each part separately to one of the four Ethernet Generators to send it out to the network media. All outputs are connected directly to the receiver’s inputs as a point-to-point, which means every TX forwards a data packet to a specific RX; this will help to keep the order of the data at the receiver based on a component called Combiner; the Combiner collects the data from all receiver devices, then combines the data and stores them inside the receiver storage location, also the combiner will combine the data in the same order as they have been sent. Fig.4. shows a simple schematic diagram for how data have been divided on the transmitter side then recombined again at the receiver.

As shown in Fig.3., the transmitter side has another component which is called Frame Generator, this core is also created inside Qsys builder as a custom component, its main function is, takes the data from the Application Layer (in this case Memory with splitter), then add all requiring header files to the main packet, such as IP address, Mac address and port number (depending on which reference module has been used), after that the frame generator send the frame packet out to the receiver side over network media. The Frame Generator has two clocks to control its operation, as shown in Fig.5., the first clock equal to 125MHz that synchronize the input parallel data, which placed at the system side, and the second clock is high clock that equal to 1GHz, placed at the other side of the component that send a serial data to the network media, with a transmission speed of 1Gbps.

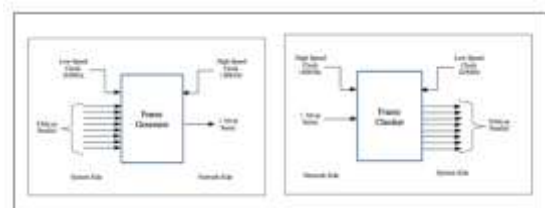


Fig.5. The requiring clocks at system and Network sides for both Frame Generator and Checker.

On the receiver side the system will perform the opposite operation: each receiver will write the specific byte in the memory bank, for example receiver 1 writes byte 1, receiver2 writes byte 2 and so on; this operation will help to keep tracking the data without losing the specific order for each byte. To do this operation correctly the system requires an additional component called Memory-Combiner placed between memory and Checker components. This component will take the data that have been received from the checker and combine them with the correct order before sending them to the memory. The Frame Checker is the first component on the receiver side, which will receive the data directly from the network media that transfer with speed equal to 1Gbps, then the frame checker removes all header files and convert a serial data that gotten from network to a 8 bits as a parallel, after that forwarding the original packet to the storage devices system; this core has been built inside Qsys builder as well. In general, the receiving speed has to be greater or equal to the transmission speed to avoid losing data in the transmission line.

iii. Results and Discussion

All systems described above have been built based on a custom Verilog core. These systems try to improve the efficiency of the design by increasing the transmission speed, reducing the requiring time for completion and using a new protocol known as Novel Protocol. All these systems are implemented on the DE4 FPGA Terasic board and tested using one of the Quartus II development tools called SignalTap II Logic Analyzer; this tool helps to run and test the system as hardware. The schematic diagram for this design is shown above at Fig.3.

This system has been tested by exporting their inputs and outputs to the HSMC connectors then connect TX and RX ports to each other using External loopback; this loopback is point-to-point connectors, which mean every TX connects directly to a specific RX; this will appear later when describing the output waveforms when running the systems with the SignalTap II Logic Analyzer.

Fig.6. shows the output waveforms when running a Single Ethernet system with the SignalTap analyser; the first group is the data that read from memory with width equal to 128 bits at clock equal to 125MHz. These reading data divided into multi packets, each packet has 8bits as a width, then all these packets has been forwarded to the Frame Generator separately, to send them out to the network media, the Transmitter side will send one bit, serial data, with each clock cycle to the receiver side, this is shown at group 2 in Fig.6.



Fig.6. Screen Shot for SignalTap II Logic Analyser, when one Ethernet Transceiver carries 128 bits.

On the receiver side, the data will be received at the Frame Checker as one bit width with each clock cycle, this appears as Group 3 in Fig.6; these receiving packets, with 8bit width, will combine again to give a full data frame of 128bits width, then the system will send the large frame directly to the memory.

For example, when the size of full packet of data is equal to 128 bits, as shown in Fig.6., the frequency of the main clock is equal to 125MHz, and the system able to transfer 8 bits of data with each rising edge from clock; this means the system requires 16 cycles to send the full packets of data to the next side of the system, $(128\text{bits (full packet's size)} / 8\text{bits (transferring size width with each rising edge)}) = 16\text{cycles}$.

Fig.7. shows the signal waveforms for the first Parallel Ethernet System that is based on connecting four single Ethernet system in parallel; the waveforms appear are divided into four groups, the first group is the data that have been read from memory, these data will be divided into four groups, prepare them to be sent out to the Frame generator, the second group shows the data that have been sent out from the transmitters to the network media, these data will be received from the Frame Checker, and appear in group three, then the received data will combine again at the Combiner and be stored in receiver memory, appearing as group four; the conclusion for this design is: the data that have been sent out from the Transmitter Memory (group 1), have been received and stored in Receiver Memory (group 4).

For instance, with the second system, parallel Ethernet system, when apply input data as a full packet of data equal to 128 bits, this packet has been divided into four smaller packets, the size for each one is 32 bits; if the clock cycles for the system side equal to 125MHz, and the amount of data transfer with each clock equal to 8bits, this means the 32 bits requires four clock cycles to send them out, $(32\text{ bits (packet's size with each transceiver system)} / 8\text{bits (amount of transferring data with edge rising edge)}) = 4\text{ cycles}$.



Fig.7. Screen Shot for SignalTap II Logic Analyser, when Parallel Ethernet Transceiver has been tested to transfer 128 bits

Comparing between two systems, single and parallel Ethernet, the same size of data (128 bits) required 16 clock cycles to send out through network media with single Ethernet system, while the similar amount of data requires just four cycles to send all data out using Parallel Ethernet system; in other words with 16 clock cycles we are able to carry only 128 bits using Single Ethernet system, but with the same time duration, 16 clock cycles, we are able to carry 512 bits using Parallel Ethernet system, which means the efficiency for parallel system is four times more than the efficiency for single Ethernet system, this is the main purpose to use Parallel Ethernet system instead of Single Ethernet system.

iv. Conclusion

Parallel Ethernet system, is one of the most efficient way to increase the transceiver speed, by connect multi transmitter systems to the multi receiver system both of them as a parallel; one of the most important thing with parallel is either all transmitters or all receivers or both of them access the same memory location, which helps to transfer as much data as can at the same time, that will improve the design efficient; in case of either all transmitters or all receivers access the same memory locations this gives opportunity to the user either to have multi receiver applications, in case of access one transmitting location, or to have multi transmitting applications all of them will send the data to one receiving application through the parallel system; this multi choices with transmitting and receiving connecting helps to improve the system efficiency and increase it's useful application. In this research, a new reference module is created that called NRM, it is based on has multi Ethernet transceiver systems, each single Ethernet system can support either 1Gbps or 10Gbps, depending on the transceiver devices that available for each FPGA; the main idea has been discussed in this research is connect multi single Ethernet system as a parallel to get higher transceiver speed based on the new reference module, the number of the transceiver system can connect together as a parallel will depend on the available FPGA device; for example with the StratixIV GX FPGA, that implement on DE4 Terasic board, it allows to connect 34-LVDS devices as a parallel, each one can support 1Gbps as a transceiver speed, also with the same FPGA device, it has 12 GXB

devices, all these devices can connect together as a parallel, to get a transceiver speed up to 30Gbps. Each single transceiver system has two sides, transmitter and receiver; both of them built as embedded system inside FPGA and connect to each other through network media as a point-to-point connector; all data used inside system are parallel, and these data convert to serial data with a very high clock when carry through network media. This parallel system has been implemented using Quartus II software and built as an embedded system inside StratixIV GX FPGA, then tested as a hardware using SignalTap II Logic Analyzer.

v. References

1. M. Kwiatkowski, M. Alsdorf, B. Dehning, W. Viango and C. Zamantzas, "A Gigabit Ethernet Link For an FPGA Based Beam Loss Measurement System", 2nd International Beam Instrument Conference, IBIC2013, Oxford, UK, 2013.
2. C. Estevez, S. Angulo, A. Abujatum and G. Ellinas, "A Carrier-Ethernet oriented Transport Protocol with a Novel Congestion Control and QoS Integration: Analytical, Simulated and Experimental Validation", IEEE International Conference On Communication, IEEE ICC 2012, Ottawa, ON, 10-15 June 2012.
3. Tao Lin, J. Tingbiao and Z. Xiangli, "A Solution for Ethernet-based real-time communication network of Distributed Numerical Control System", International Technology and Innovation Conference 2009, ITIC 2009, Xian, China, 12-14 Oct. 2009.
4. P. Anand and S. Bates, "Characterization of Alien-Next in 10GBASE-T Systems", Canadian Conference on Electrical and Computer Engineering, Saskatoon, Sask., 1-4 May 2005.
5. C. Meinel and H. Sack, "Network Access Layer (1): Wired LAN Technology," Internetworking Working, Technological Foundations and Applications, Potsdam, Germany, Springer, 2013, Ch. 4.
6. X. Chen, A. Jukan and M. Medard, "A Novel Network Coded Parallel Transmission Framework for High-Speed Ethernet", in Global Communication Conference (GLOBECOM) 2013 IEEE, Atlanta, GA., 2013, pp. 2382-2387.