In-Line CRC Calculation and Scheduling for 10 Gigabit Ethernet Transmission

Tomas Henriksson
Department of Electrical Engineering, Linköpings universitet, SE-581 83 Linköping
e-mail: tomhe@isy.liu.se, phone +46-13-288956, fax: +46-13-139282

ABSTRACT

CRC calculation is a core part of the Ethernet processing. In 10 Gigabit Ethernet, which uses the extended gigabit media independent interface the CRC calculation in-line with the frame transmission creates some data alignment problems. A CRC calculation unit has been designed, which handles all control symbols and alignment of them and the data in accordance with the specification. Area is always critical for Ethernet circuitry, since it is only part of a chip. The unit occupies 0.51 mm^2 silicon area in a 0.35 micron technology.

1. INTRODUCTION

Computer networks keep evolving at a non-decreasing pace. The next standard to be adapted is 10 Gigabit Ethernet (10 GE). 10 GE can be used for local area networks as well as for wide area networks, for which until now Ethernet has not been used.

For both scenarios, the CRC calculation is a core part of the ethernet processing. CRC is used to generate a checksum over the whole frame at transmission and to check that the frame is still correct at reception. Since 10 GE uses full duplex bidirectional links each side of the link must be capable of computing two CRC checksums at the same time, one for the received frame and one for the transmitted frame.

There are some differences in the requirements on these two CRC calculation units. The reception unit receives the data via the 32 bit wide XGMII (eXtended Gigabit Media Independent Interface) and since the checksum is included in the end of the frame the output is just a one bit flag that tells if the frame can be accepted or must be discarded. The transmission unit, on the other hand, receives data from the MAC (medium access control) as it is transmitted and has to generate the CRC checksum in-line with short latency and then append it to the end of the frame. The interface from the MAC is 64 bits of data, since that is the preferred data width in 10 GE equipment. The output on the other hand is 32 bit wide XGMII.

In a previous study we have designed a CRC calculation unit for 8 bit input in parallel, which avoids the controlling circuitry and reduces the size of the core CRC calculation unit [1]. However, to run this unit at the required frequency of 1.25 GHz is not acceptable in systems of today.

In Fast Ethernet (100 Mb/s) a technique was used that could append the CRC in-line, but most problems did not occur in that environment. First of all for Fast Ethernet, MII (Media Independent Interface) was used, which is 4 bit wide. Also, the data was handled in 8 bit symbols in the MAC, which is the smallest data unit in Ethernet. Although the interface widths have increased with newer versions of Ethernet, the minimum data unit has not changed. This introduces the problem that the size of a frame must not be divisible with 32 bits and thus the last word on the interface may contain 8, 16, 24, or 32 bits of data. This is solved by having 4 control signals together with the 32 bits of data. Each control bit announces the presence of valid data on the corresponding data lane (8 bits wide). More over, the first two words on the XGMII interface contain only start of packet symbol, preamble symbols and a start frame delimiter symbol, so they must not be included in the CRC calculation. After the frame, including the appended CRC, an end-of-packet symbol must be inserted on the next free lane. Whenever no data is transmitted on the XGMII all lanes carry idle symbols and if an error occurs during transmission an error symbol is inserted by the physical layer on the receiver side [2]. This paper describes how to create the required XGMII format on the Ethernet frame in-line with the CRC calculation.

It is desirable to have as short latency as possible through the CRC calculation unit. This reduces the number of flip-flops needed for alignment of the data and thereby decreases the silicon area for the CRC calculation unit.

This paper starts with discussing the system perspectives in section 2, then section 3 describes the implementation. In section 4 the results are presented and discussed and finally the conclusions are drawn in section 5.

2. SYSTEM PERSPECTIVE

The system environment for a 10 GE CRC calculation unit is either a port in a switch or router, a network card in a desktop or laptop computer, or a specialized system for example an IP phone connected to the 10 GE. In all cases the silicon area is important since the unit will be integrated on one silicon die together with many other cores. The throughput requirement is standardized and an implementation has to do a trade-off between silicon area and power consumption. Both are limiting the maximum possible functionality that can be integrated on the chip and therefore the system constraints will dictate which trade off to use.

Figure 1 gives an example of how the CRC calculation unit for the transmission part of the MAC fits into the environment in a network card in a desktop computer. All interfaces consist of data lines in combination with control signals.

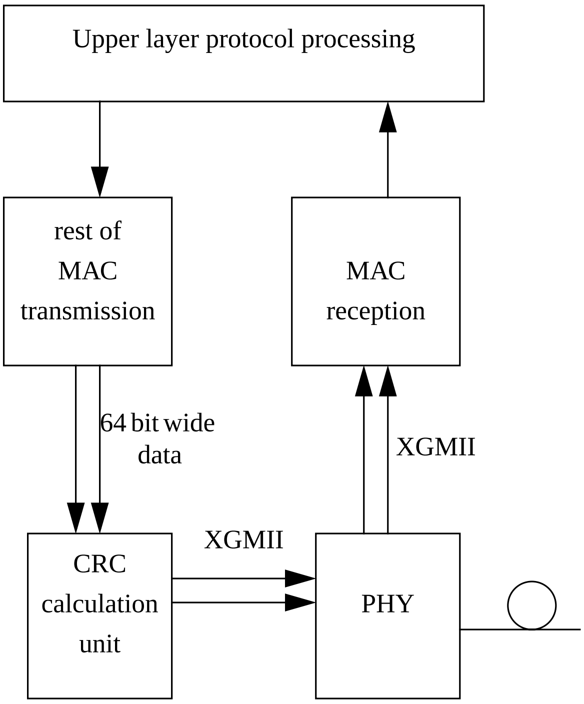
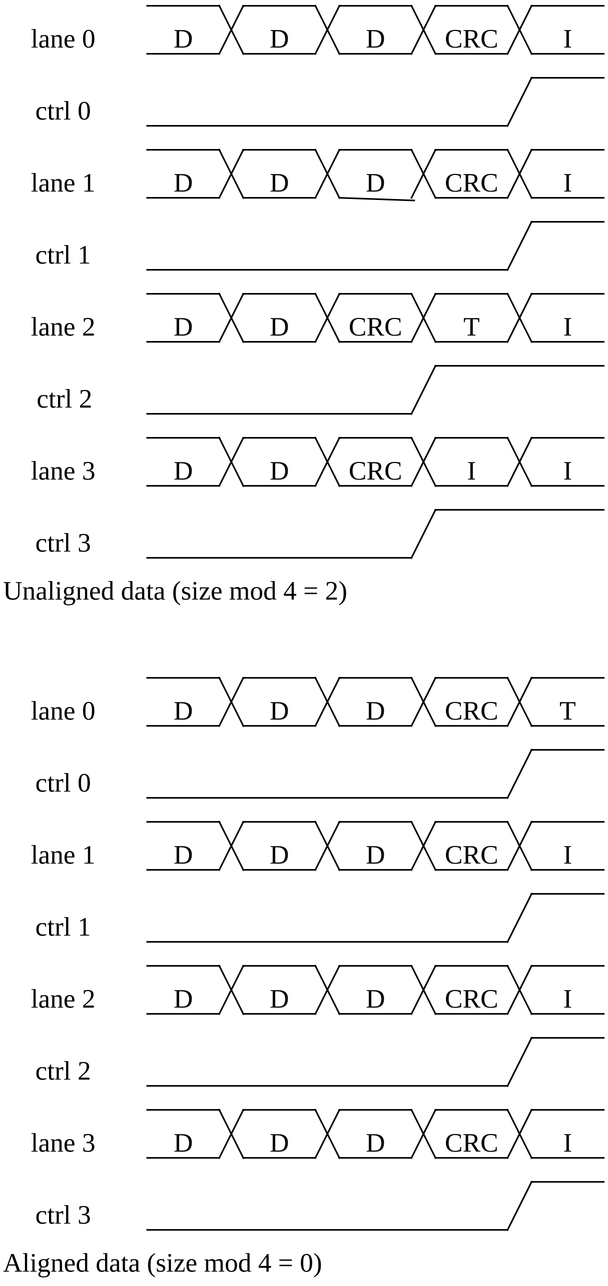


Figure 1: Network Card

3. Implementation

The CRC calculation unit has been implemented in VHDL at register transfer level (RTL). The core CRC calculation part is based on galois fields in order to simplify the logic equations [3]. By doing this there is a need to apply 32 bits of zeros at the end of the calculation and the crc register must be reseted to a vector 46AF6449 in hexadecimal numbers. The main part of the CRC calculation is performed 64 bits per clock cycle, but since the Ethernet frame may have any number of bytes, the last word computes only as many bytes that are available and possibly filling out with zeros. If the last word contains more than 4 bytes of data an extra clock cycle has to be spent in order to include the additional zeros in the calculation. The core calculation part will be presented in another paper and is not described in detail here. Instead, the focus of this paper is the integration and data alignment for the transmitter.

Once the last part of the data is sent on the XGMII the CRC has to be appended immediately. Depending on the size of the data there are four cases. Two of them are shown in figure 2. In order to be able to shift the CRC in this way multiplexers are needed on all outputs, so data can be chosen from either of the two data bytes in the 64 bit input that go to each lane, from any of the four CRC bytes or from any XGMII control code or preamble or start frame delimiter. The start of packet (S) always uses lane 0 and the start frame delimiter always uses lane 3, all other control codes can be present on any lane. Figure 3 gives an overview of how the output multiplexing is performed and figure 4 shows in detail the configuration for lane 0. The big-



Unaligned data (size mod 4 = 2)

Aligned data (size mod 4 = 0)

Figure 2: XGMII with CRC appended

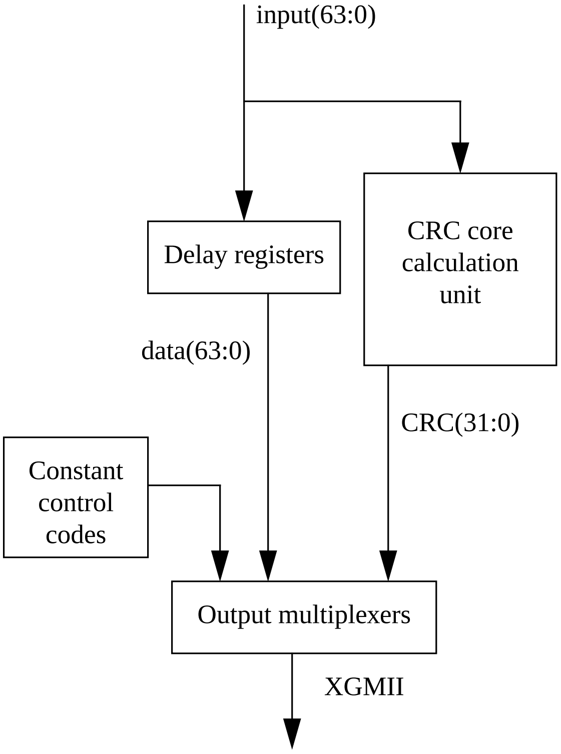


Figure 3: Output multiplexer overview

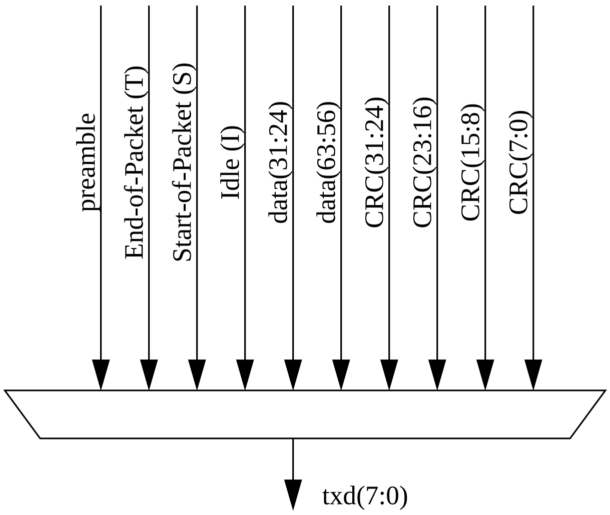


Figure 4: txd(7:0) multiplexer

gest implementation challenge is to create the control signals to these multiplexers. That was done by having a finite state machine (FSM) to keep track of the end of the data stream and combine its state with the current control signals. The FSM has different states dependent of how many bytes of valid data there are in the last 64 bit word. This simplifies the multiplexer construction.

There is also a data rate conversion, from 64 bits at 156.25 MHz to 32 bit at 312.5 MHz. This is managed by letting XGMII data change both on positive and negative clock edges, as the standardization procedure has suggested and most likely will be the case in the final standard. This is again done in the output multiplexers, which take the clock state into consideration when choosing which signal to forward to the output.

4. Results and Discussion

The CRC calculation unit for 10 GE transmission has been implemented in VHDL and synthesized with a 0.35 micron library as target technology. The area is estimated to 0.51 mm^2 when a clock uncertainty of 0.2 ns is used. This has been proven an adequate value, since similar constructions have resulted in a maximum clock skew of about 0.15 ns after complete layout. The clock frequency is 156.25 MHz according to the specification. The critical path resides in the core CRC calculation unit. This proves that the complex output multiplexers do not constrain the overall design.

The latency of the CRC calculation unit is 6.4 ns. That is the time from that the data is received on the input port until the start of packet symbol and the first three preamble symbols appear on the XGMII.

5. Conclusion

A CRC calculation unit for 10 Gigabit Ethernet transmission has been designed and implemented. The silicon area is estimated to 0.51 mm^2 in a 0.35 micron technology and the latency is 6.4 ns. The CRC calculation part is besides calculating the CRC also responsible for shaping the XGMII format on the Ethernet frame.

References

[1] T. Henriksson, H. Eriksson, U. Nordqvist, P. Larsson-Edefors, D. Liu, “VLSI implementation of CRC-32 for 10 gigabit ethernet”, ICECS 2001, vol. 3, pp. 1215-1218

[2] H. Frazier, “10Gig MII update”, on the www, http://grouper.ieee.org/groups/802/3/10G_study/public/nov99/index.htm

[3] R. J. Glaise and X. Jacquart, “FAST CRC CALCULATION”, IEEE International Conference on Computer Design: VLSI in Computers and Processors, Cambridge, MA, USA, 1993, pp. 602-605.