INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 11(1):39-50, February 2016.

A Hardware Independent Real-time Ethernet for Motion Control Systems

S. Ji, C. Zhang, T. Hu, K. Wang

Shuai Ji
Mechantronic Engineering School
Shandong Jianzhu University
Jinan, Shandong, China
jizhongzhe@126.com

Chengrui Zhang*
School of Mechanical Engineering
Shandong University,
Jinan, Shandong, China
*Corresponding author: 13969076910@126.com

Tianliang Hu
School of Mechanical Engineering
Shandong University
Jinan, Shandong, China
tlhu@sdu.edu.cn

Ke Wang
Shanghai STEP Electric Corporation
Shanghai, China
ahker534@sina.cn

Abstract: Ethernet for Manufacture Automation Control (EtherMAC) is a new kind of real-time Ethernet used in motion control systems. It adopts a line topology with a standard industrial computer based master node and field-programmable-gate-array based slave nodes. EtherMAC employs one slave node to manage cycle communication and clock synchronization, so the real-time demand for its master node can be greatly reduced and dedicated hardware is no longer mandatory. Its distributed clock compensation mechanism can get synchronization accuracy in nanosecond order. The advantages of industrial computer and field programmable-gate-array are combined with EtherMAC, so that high control performance can be achieved.
Keywords: Real-time Ethernet, cycle time, synchronization, motion control, FPGA

1 Introduction

Fieldbus has played an important role in factory automation field during the past 20 years [1]. However, it can hardly meet the development of the networked control systems any longer because of the numerous standards, low baud rate, high cost and etc [12]. At the same time, as a mature technology in office, Ethernet is introduced into field control layer for its low price, high communication rate, robustness and easy deployment. Many automation vendors begin to propose their own industrial communication solutions by modifying the original Ethernet protocol to improve its real-time performance (see [4, 13]). These types of Ethernet based on IEEE 802.3 standards and with real-time property are called real-time Ethernet (RT Ethernet) [5]. Standardization of industrial ethernet Standardization of industrial ethernet

IEC61784-2 has defined some indicators to specify the real-time performance of RT Ethernet, such as delivery time and synchronization accuracy. But the minimum cycle is a more direct

indicator to evaluate the performance of motion control system than delivery time(see [6]- [9]). The RT Ethernets whose cycle time can be smaller than 1ms with jitters less than 1ms are called isochronous RT Ethernet (see [10]- [11]), which are most likely to be used in motion control systems.

However, most of the current isochronous RT Ethernet solutions are based on modifications of the hardware [12]. Dedicated Network Interface Card (NIC) or real-time OS are mandatory for the master node in time critical applications. For example, PROFINET IRT is achieved with a special switch ASIC (see [12]); dedicated NICs are mandatory for SERCOS III and Ethernet Powerlink's master/management node in time-critical applications (see [13]); EtherCAT does not require a dedicated master node, but a hard real-time kernel is mandatory for the distributed clock synchronization, in addition the operating system can introduce jitter to the cycle time(see [14]). The dedicated hardware is closed and high cost, which is not good for the development of motion control system.

Herein, a standard hardware based real-time Ethernet named EtherMAC is proposed. EtherMAC employs the programmable gate array chips (FPGA) based slave node to manage the communication and synchronization, so that IPC master node can focus on motion control algorithms. In this way, computation capability of IPC and hard real-time property of FPGA are combined, and hard real-time performance can be achieved even the master node without dedicated NICs and hard real-time operating systems.

The rest of this paper is organized as follow. In Section 2, the protocol model of EtherMAC is introduced. In Section 3 and 4, its real-time and synchronization mechanism are described in detail. Section 5 is the slave node implementation on a FPGA chip. Then two experiments are carried out in Section 6 to test the delivery time and synchronization precision. Finally, the conclusion is given in Section 7.

2 Overview of EtherMAC

2.1 Topology of EtherMAC

Line topology network is widely used in industrial automation field for its simple structure, low cost and advantage in deterministic communication(see [11], [15]- [16]Jasperneite A Proposal for a Generic Real-TimeJasperneite A Proposal for a Generic Real-TimeJasperneite A Proposal for a Generic Real-Time). EtherMAC also takes the line topology as its fundamental topology, where the IPC with at least one standard NIC works as the master node, and the FPGA-based slave nodes with two Ethernet ports cascade with each other in a line as shown in Figure 1. All the slave nodes can be treated as a single standard Ethernet device for the master node, because they share a common Media Access Control (MAC) address.

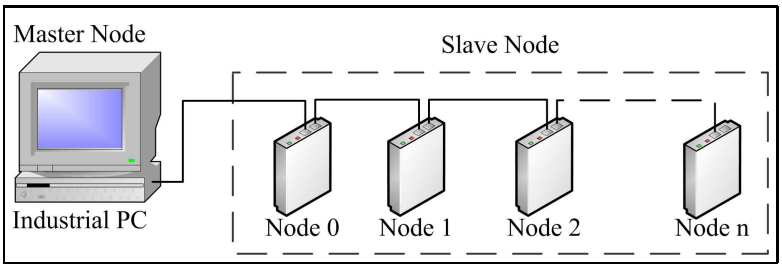

Figure 1: Topology of EtherMAC

EtherMAC employs summation frames in Figure 2 to convey application data, which is similar to INTERBUS [17] and EtherCAT [18]. The configuration and control data are conveyed

down by a summation frame in the descending way, and the state data of the slave nodes are collected by another summation frame in the ascending way. In this manner, all the nodes can be updated in a single communication cycle. This is more efficient than the other RT Ethernets employing individual frame(see [8], [11]). In addition, the control and state data are separated in the descending and ascending ways, so bandwidth of Ethernet can be used more efficiently.

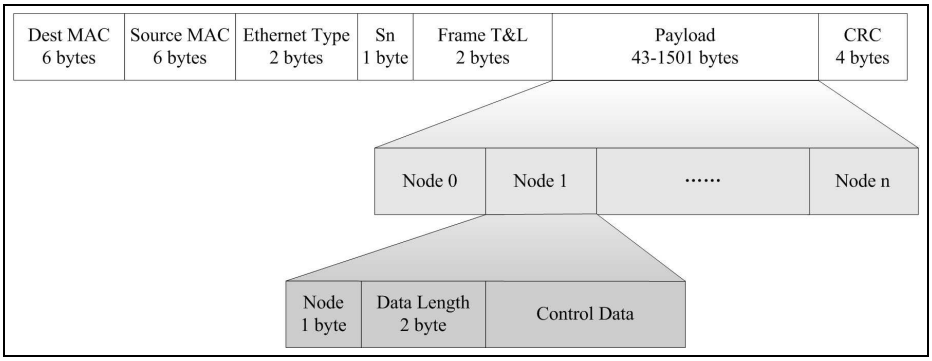

Figure 2: Frame structure of EtherMAC

2.2 Model of EtherMAC

There are three solutions for Ethernet to get real-time communication. The first class is named "on top of TCP/IP", which keeps the TCP/UDP/IP protocols unchanged and concentrates all real-time modifications in the application layer. In the second solution, application layer programs bypass the TCP/UDP/IP protocols and operate the Ethernet directly. In the third one, medium access mechanism and hardware structure of Ethernet are modified to achieve isochronous real-time performance (see [8], [12]). The architecture of EtherMAC is shown in Figure 3. The master node is implemented on a standard IPC, and TCP/IP protocols are bypassed. The slave nodes are implemented on FPGA chips with modified Data Link Layer (DLL).

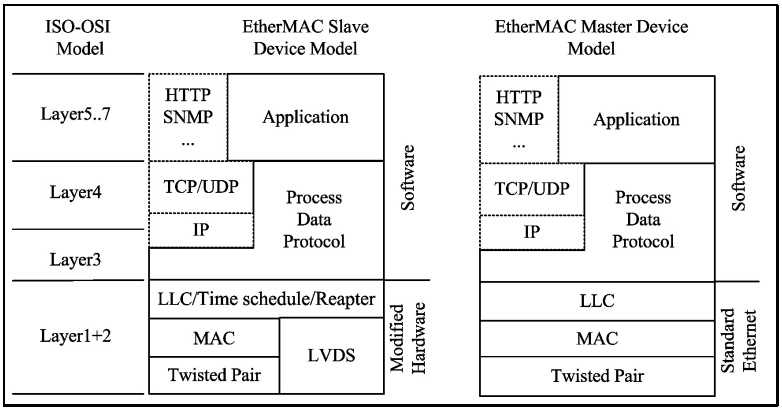

Figure 3: Protocol Suite

The modified DLL of the slave node has the following characters:

- A time schedule module is added to DLL to manage the communication cycle and synchronize the distributed clocks.
- A repeater is designed to receive and forward the summation frame on line, so that small forward delay can be achieved.

3 Real-time communication

The master-slave structure is most used in RT Ethernet, where the master node sends out the control data periodically and the slave nodes carry out the received commands. The cycle duration accuracy is determined by the master node in this structure, as a result dedicated NIC hard real-time or operating system is necessary for the master node in time critical applications. However, on one hand dedicated NIC solution is closed architecture and high cost, on the other hand the response time jitter of real-time operating system can reduce the cycle accuracy.

EtherMAC turns its time-critical tasks over to the FPGA-based slave nodes, while master node is designed to complete the configuration and complex control algorithms. The slave nodes of EtherMAC are implemented on FPGA chips, which can provide high timing precision because of their hardware property.

3.1 Communication process

The communication process of EtherMAC is shown in Figure 4. The slave nodes in the networks carry out a self-checking program after powered on. If there are no problems with the hardware, the slave nodes will get into the idle state and wait for commands from master node.

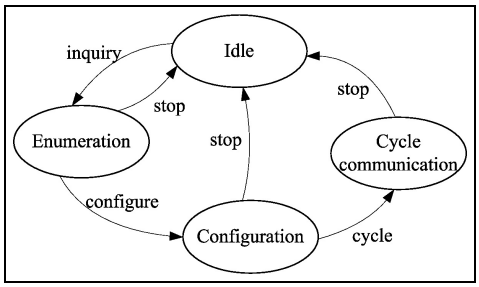

Figure 4: State machine of EtherMAC working process

1)Enumeration. The slave nodes are assigned a unique sequence number and feedback their inherent information in this stage. The detailed process is shown in Figure 5.

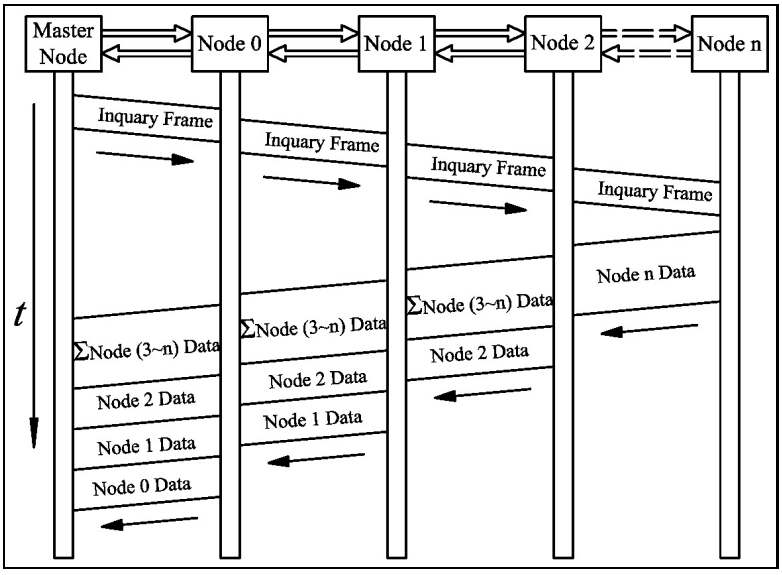

Figure 5: Enumeration Process

First, master node sends an inquiry frame to the slave nodes. Every slave node takes the data in certain place of the frame as its sequence number, increases its value by one and forwards the frame the next. The slave nodes get their sequence number in this way.

Then, the last slave node stops the forwarding process and starts to feedback after receiving the inquiry frame. It packets its inherent information into the feedback frame and sends the frame to the second last slave node. All the previous slave nodes attach their inherent information to the end of the feedback frame and forward it.

At last, the frame with the information of all slave nodes is transmitted to the master node by slave node 0. The master node compares the received information with the network configuration file to check the configuration state of the network.

Figure 5 The Inquiry process

2)Configuration. The master node starts the configuration stage if all the slave nodes in the network have been checked well in enumeration stage. Master node delivered the configuration frame to slave nodes, and slave nodes return their configuration results. Then the master node prepares the configuration data of the next cycle based on the feedback information. The forward delays measurement is also carried out in this stage, which will be introduced in the Section 4.

3)Cycle communication. The master node sends out a frame to start the cycle communication stage after the network configure done. The management right of the communication is transferred to the last slave node from then on. Different from the last two stages, the communication is started by the last slave node instead of the master node in this stage. The detailed process is shown in Figure 6.

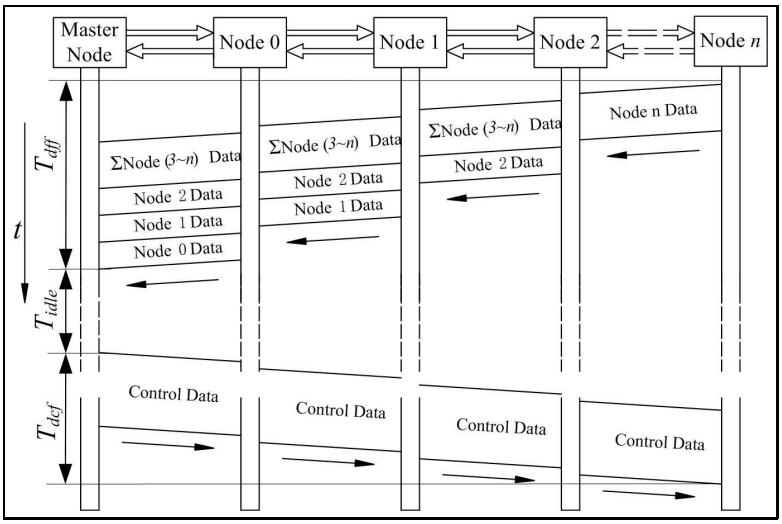

Figure 6: Cycle Communication Process

The last slave node starts a new communication cycle by sending a frame back with the set communication period. The slave nodes update their state data to master node by attaching their data to the end of the feedback frame. As soon as receives the feedback frame, master node computes the commands of the next period and sends it to slave nodes. Therefore, real-time communication can be achieved if the master node can finish all the operations before the last slave node starting the next communication cycle.

In cycle communication, the control frames are forwarded directly and without any modification in the descending way, and the feedback frames are processed online in ascending way, so that small forward delay can be achieved. This means small delivery time for line topology network. For one slave node, the total forward delay in a communication cycle is about 520ns.

3.2 Performance analysis

The delivery time of EtherMAC can be derived from its communication mechanism and the theoretical analysis in(see [6]- [8])

$$T_d = T_{frame} + T_{cable} + \sum_{i=0}^{n-1} \Delta T(i)_{delay} \tag{1}$$

where T_{cable} is propagation delay from cable, which is about 5ns/m for the category 5 cable. T_{frame} is the frame transmission time, which is in linear with the frame size. $\Delta T(i)_{delay}$ is the frame forward delay of slave node i, which is about 520ns with proposed the forward mechanism.

There are two different types of Ethernet frames in the cycle communication stages, so the minimal cycle time T_{cycmin} can be derived

$$T_{cycmin}h = T_{dcf} + T_{dff} + T_{ifg} + T_{idle} \tag{2}$$

where T_{dcf} is the delivery time of control frame, T_{dff} is the delivery time feedback frame, T_{ifg} is the inter frame gap, because the full duplex communication is employed there is only one T_{ifg}. T_{idle} is the idle time when there is no communication in the network.

In this way, the react time jitters of master node does not influence cycle time accuracy any more, and isochronous real-time communication can be achieved even the master node only with soft real-time property. The minimal cycle time can be as small as $20\mu s$ when there are only 3 slave nodes and payload is no more than 48 bytes length.

4 Synchronization of distributed slave nodes

Synchronization accuracy of RT Ethernet is its capability to synchronize the actions of the distributed devices [19]. RT Ethernet demands the synchronized actions must be repeated periodically with strict jitter bounds [3]. Poor synchronization of relevant axes in motion control systems means diminished dimensional accuracy of the work-piece or even unusable products (see [20]- [21]). Many synchronization methods have been proposed for networked control systems(see [22]- [24]), of which Precision Time Protocol (PTP) described in IEEE 1588 is one of the most used. However, PTP is difficult to implement, which consumes too many hardware and bandwidth resources in practice. Therefore, a concise synchronization algorithm is proposed to get high synchronization accuracy.

4.1 Factors result to asynchronization

Every slave nodes in the network has its own oscillator. The frequency of each oscillator cannot be exactly the same, besides the clock frequency drifts with environmental temperature and time. As a result, synchronization mechanisms are necessary for networked control systems. Broadcast frame is often used as the synchronization signal in star topology networks, but it can hardly work in line topology networks thanks to the forward delay between cascaded slave nodes. Equations in (4.1) show the time difference of the slave nodes.

$$\begin{aligned} T_1 &= T_0 + \Delta T^1_{delay} \\ T_2 &= T_1 + \Delta T^2_{delay} = T_0 + \Delta T^1_{delay} + \Delta T^2_{delay} \\ &\cdots\cdots \\ T_{n-1} &= T_0 + \sum_{i=0}^{n-1} \Delta T(i)_{delay} \end{aligned} \tag{3}$$

where T_i is the time a frame arriving at the slave node i in descending way. T^i_{delay} is the forward delay of salve node i, which accumulates with the node number and cable length. The forward delay of every slave node is not constant because of the phase separation of receiving and transmitting clock. Thus, it is impossible to get the exact value beforehand and compensate it in cycle communication.

4.2 Synchronization mechanism of EtherMAC

The distributed clocks synchronization can be divided into two stages: delay measurement and clock compensation.

Delay measurement is only carried out once at the first time of network configuration. The detailed process is shown in Figure 7. Master node sends the measure frame cyclically, and the last slave node starts the feedback process as soon as receives the frame. Every slave node records the time between it receives the measure frame and feedback frame. The measurement begins at the time of the frame arriving at the media independent interface, so the influence of software stack can be ignored. The ascending and descending way can be symmetrical by compensating the forward FIFO depth. Here the measured delay includes the slave node forward delay and cable delay. Take the phase separation of the transmitting and receiving clock into account, the average value of hundreds of record delay time is used as delay time.

In the cycle communication stage, every salve node initializes its local clock with half of the average delay time at the time of receiving the feedback frame, so that the slave nodes can share a common zero point. In this way, the distributed clock of the slave nodes can be synchronized.

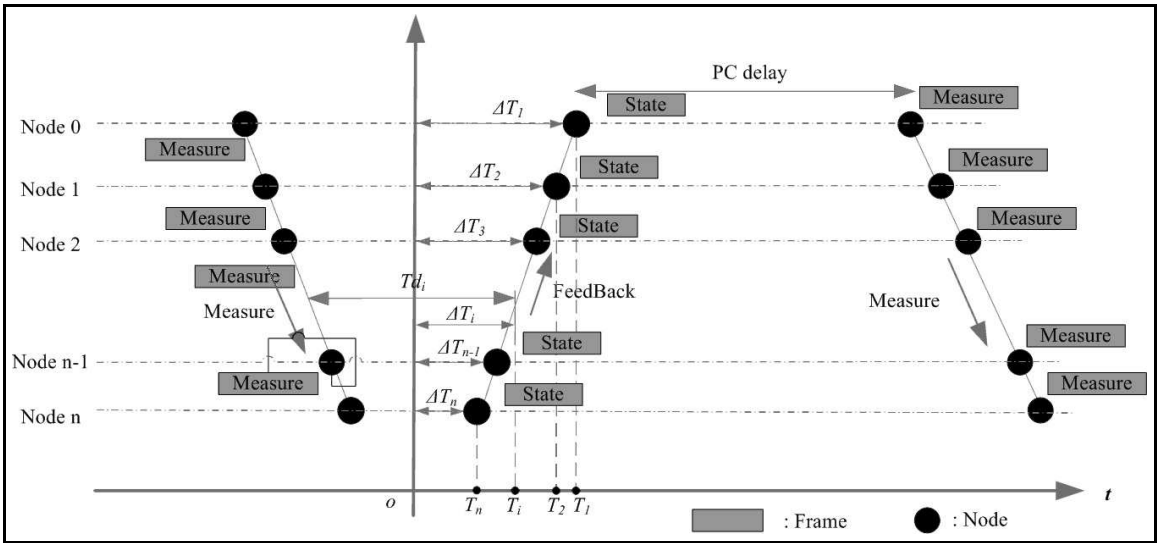

Figure 7: Synchronization Process

The forward delay and clock drift can be compensated cyclically because the feedback process is carried out every communication cycle, so high synchronization accuracy between slave nodes can be achieved in this way.

5 Implementation of slave node

Master node of EtherMAC can be implemented on any controller with standard Ethernet interfaces, and no dedicated hardware in needed. Real-time OS is necessary for some time critical applications, but the real-time demand can be greatly reduced with the above mentioned

communication mechanism. Therefore, only the slave node implementation is introduced in this paper. The slave nodes are implemented on FPFA chips as shown in Figure 8.

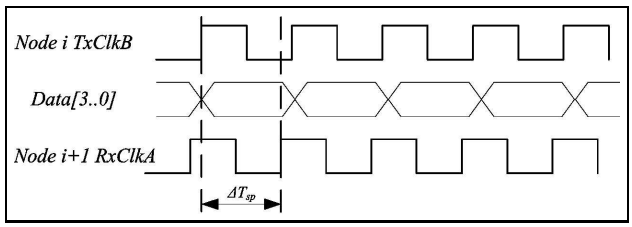

Figure 8: Implementation of EtherMAC slave node

FPGA is more than some programmable resource with a certain number of logical cells today. The loadable soft cores enable the user to create a microcontroller within an FPGA chip and equip it with ultra fast peripherals [25]. It allows the developers to customize peripherals to meet the specific requirement.

All the functions of the EtherMAC are packaged into an IP core. The application layer is implemented as software in Nios II a soft processor designed specifically for Altera FPGA chips. IO, Servo Control and the other peripherals are mounted on the Avalon bus. A Static Random Access Memory (SRAM) chip is used to running application programs, and a flash chip is used to store the configuration data of FPGA. The schematic diagram of EtherMAC IP core is shown in Figure 9.

There are two concurrent channels in descending way. One is for protocol analysis, and the other one is for frame forward. In protocol analysis channel, application data belong to the current slave node are extracted and stored in a dual-port-RAM AppCtrl Data, and the operation information including data address and length are stored in a dual-port-RAM Operation Info at the same time. After receiving the whole frame, the signal RxDone holds high for twelve system clocks to inform the upper layer program that application data have been updated. Then the application programs can get the commands of this cycle from AppCtrl Data.

The Syn signal holds high for twelve system clocks when the distributed clocks time to the set point. Upper layer programs and other peripherals modules can carry out the synchronous actions and latch the state of slave node at this moment. The state data have been latched can be upload to the master by writing them into the dual-port-RAM AppState Data.

6 Experiment

The system in Figure 10 is employed to evaluate the real-time performance of EtherMAC. The system has one master node and eleven slave nodes. The master node is an IPC with an Intel Atom D510 1.66GHz CPU and Windows 7 OS. The Network Driver Interface Specification (NDIS) layer of the system is modified to improve its real-time performance [26]. The slave nodes are connected with a backplane, which hides in the steel rail. 100BASE-T Ethernet cable is adopted when the adjacent nodes are far away from each other as in Figure 10.

Two experiments are carried out to measure the delivery time and synchronization accuracy respectively. In the first experiment, delay time Tdi in Figure 7 is measured and delivered to master. Here, the time duration between slave node 0 receiving the measurement and feedback frame is twice of the delivery time, while half of the time difference between adjacent nodes is the forward delay of a slave node. As shown in Figure 11, the delay time is linear with the sequence number of slave nodes and the forward delay of a single slave node is about 520ns.

In the second experiment, an oscilloscope is used to watch the synchronization signals. For convenience, the synchronization signals are assigned as output pins of the FPGA chips. Slave

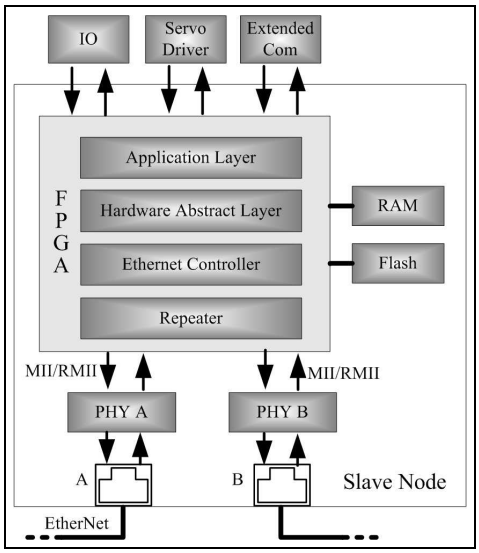

Figure 9: Hardware Architecture of EtherMAC

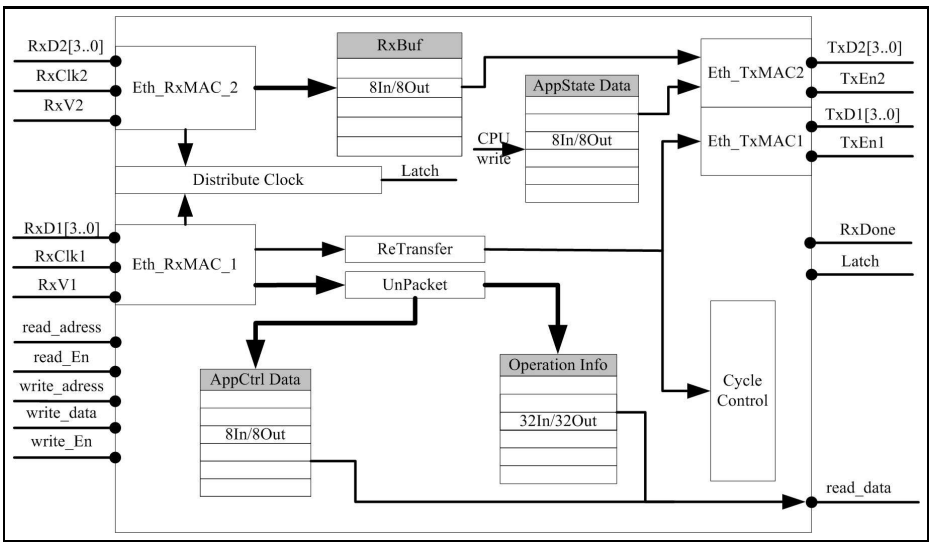

Figure 10: Distributed slave nodes

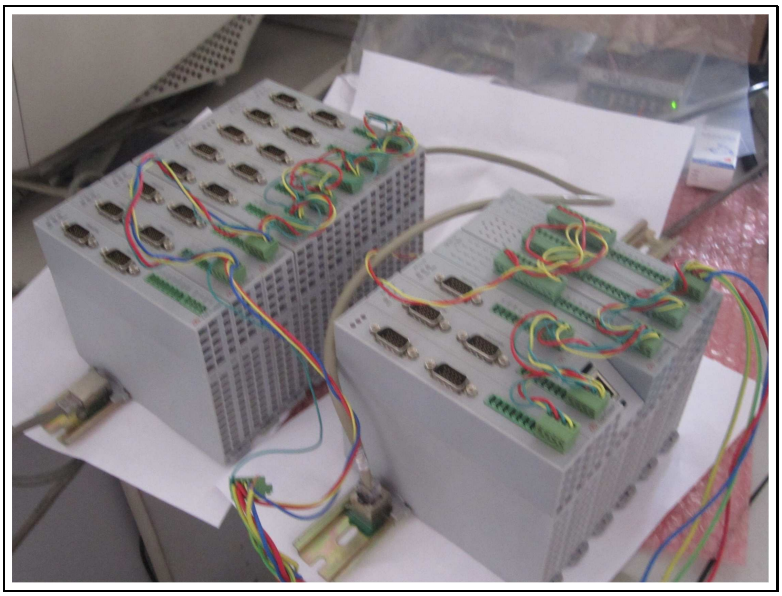

Figure 11: Delay time and jitters of Salve nodes

node 0, 5 and 10 are tested, of which slave node 0 works as the reference signal. All the triggered wave forms can superpose and stay on the screen by setting the oscilloscope works in infinite persistence mode. The results after the system worked for 30 minutes are shown in Figure 12. It is shown that the deviation of these slave nodesĄŻ synchronization signal is about 100ns with jitters no more than 100ns.

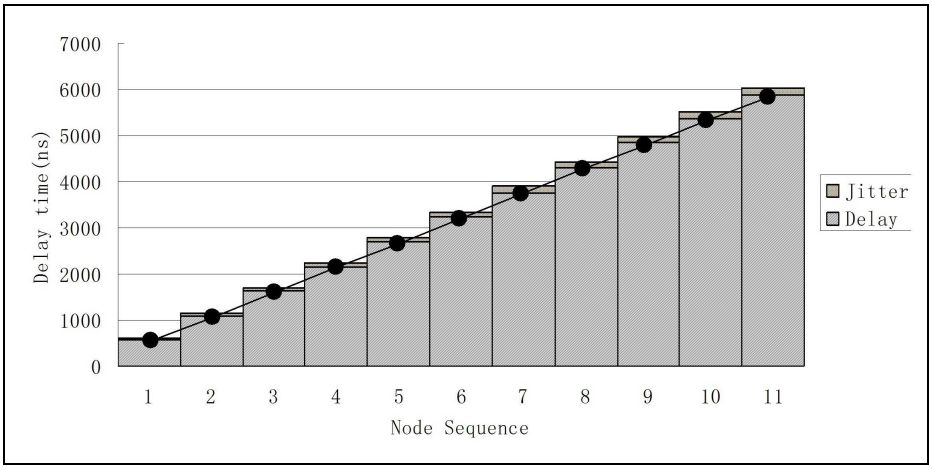

Figure 12: Jitter of the synchronization signal

7 Conclusion

RT Ethernet can help the motion control systems to get a concise architecture and good control performance [27]. However, many RT Ethernet solutions need dedicated NIC to manage communication in time critical applications. This goes against the open architecture motion control systems. Therefore, the hardware independent based EtherMAC is proposed in this paper. As most of the time critical tasks of EtherMAC are transferred to the FPGA based slave

nodes, master node can focus on the implementation of motion control algorithms. In this way, a software based motion control system with open architecture and high performance can be achieved.

Bibliography

[1] M. P. Thomesse(2005); Fieldbus technology in industrial automation, *Proceedings of the IEEE*, ISSN 0018-9219, 93(6):1073-1101.

[2] M. Felser and T. Sauter(2004); Standardization of industrial ethernet - The next battlefield?, *IEEE International Workshop on Factory Communication Systems, Proceedings, 2004*, ISBN 0-7803-8734-1, 413-420.

[3] J. D. Decotignie(2005); Ethernet-based real-time and industrial communications, *Proceedings of the IEEE*, ISSN 0018-9219, 93(6): 1102-1117.

[4] P. Neumann(2007); Communication in industrial automation's what is going on? *Control Engineering Practice*, ISSN 0967-0661, 15(11): 1332-1347.

[5] Digital Data Communications for Measurement and Control, Part 2: Additional Profiles for ISO/IEC 8802-3 Based Communication Networks in Real-Time Applications, *IEC 61784-2, 65C/350/CD circulated for comments.*

[6] L. Seno, S. Vitturi, L. Peretti, M. Zigliotto, and C. Zunino (2011); Real-time Ethernet networks for motion control, *Computer Standards & Interfaces*, ISSN 0920-5489, 33(5): 465-476.

[7] L. Seno, S. Vitturi, and C. Zunino(2009); Real time Ethernet networks evaluation using performance indicators, *Emerging Technologies & Factory Automation, 2009. ETFA 2009. IEEE Conference on*, ISSN 1946-0759, 1-8.

[8] J. Jaspemeite, M. Schumacher(and K. Weber 2007); Limits of increasing the performance of industrial Ethernet protocols, *12th IEEE International Conference on Emerging Technologies and Factory Automation*, ISBN 978-1-4244-0825-2, 17-24.

[9] J. Robert, J.-P. Georges, E. Rondeau, and T. Divoux(2012); Minimum cycle time analysis of Ethernet-based real-time protocols, *International Journal of Computers, Communications and Control*, ISSN 1841-9836, 7(4): 743-757.

[10] Performance analysis of industrial ethernet networks by means of timed model-checking D Witsch, , B Vogel-Heuser, JM Faure(2006); Performance analysis of industrial ethernet networks by means of timed model-checking, *Information control problems in manufacturing 2006*, ISBN 978-0-08-044654-7, 99 C104.

[11] M. Schumacher, J. Jasperneite, and K. Weber(2008); A new approach for increasing the performance of the industrial Ethernet system PROFINET, *IEEE International Workshop on Factory Communication Systems*, ISBN 978-1-4244-2349-1, 159-167.

[12] M. Felser(2005); Real-time ethernet - Industry prospective, *Proceedings of the IEEE*, ISSN 0018-9219, 93(6): 1118-1129.

[13] J. D. Decotignie(2009); The many faces of industrial ethernet [past and present], *Industrial Electronics Magazine*, ISSN 1932-4529, 3(1): 8-19.

[14] S. Potra and G. Sebestyen(2006); EtherCAT Protocol Implementation Issues on an Embedded Linux Platform, *2006 IEEE International Conference on Automation, Quality and Testing, Robotics*, ISBN 1-4244-0360-X, 420-425.

[15] S. Ruping, E. Vonnahme, and J. Jasperneite(1999); Analysis of Switched Ethernet networks with different topologies used in automation systems, *Fieldbus Technology: Systems Integration, Networking, and Engineering*: ISBN 978-3-211-83394-0, 9:351-358.

[16] J. Jasperneite, J. Imtiaz, M. Schumacher, and K. Weber(2009); A Proposal for a Generic Real-Time Ethernet System, *IEEE Transactions on Industrial Informatics*, ISSN 1551-3203, 5(2): 75-85.

[17] H. Li, H. Zhang, and D. Peng(2009); Research and application on INTERBUS operator terminal, *Computer Science and Information Technology, 2009. ICCSIT 2009. 2nd IEEE International Conference on*, ISBN 978-1-4244-4519-6, 309-312.

[18] D. Jansen and H. Buttner(2004); Real-time Ethernet: the EtherCAT solution, *Computing and Control Engineering*, ISSN 0956-3385, 15(1): 16-21.

[19] P. Ferrari, A. Flammini, S. Rinaldi, and G. Gaderer(2008); Evaluation of clock synchronization accuracy of coexistent Real-Time Ethernet protocols, *2008 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication*, ISBN 978-1-4244-2274-6, 87 - 91.

[20] D. Sun(2003); Position synchronization of multiple motion axes with adaptive coupling control, *Automatica*, ISSN 0005-1098, 39(6): 997-1005.

[21] M. Tomizuka, J.-S. Hu, T.-C. Chiu, and T. Kamano(1992); Synchronization of Two Motion Control Axes Under Adaptive Feedforward Control, *Journal of Dynamic Systems, Measurement, and Control*, ISSN 1528-9028, 114(2): 196-203.

[22] X. L. Zhang, X. Q. Tang, and J. H. Chen(2008); Time synchronization of hierarchical real-time networked CNC system based on ethernet/internet, *The International Journal of Advanced Manufacturing Technology*, ISSN 0268-3768, 36(11-12):1145-1156.

[23] X. Xu, Z. Xiong, J. Wu, and X. Zhu(2013); High-precision time synchronization in real-time Ethernet-based CNC systems, *The International Journal of Advanced Manufacturing Technology*, ISSN 0268-3768, 65(5-8):1157-1170.

[24] L. Zhang, Z. Liu, and C. Honghui Xia(2002); Clock synchronization algorithms for network measurements, in INFOCOM 2002. *Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, ISSN 0743-166X, 1(1):160-169.

[25] B. S. Michael Samuelian(2007); A Universal Approach for implementing Real-Time Industrial Ethernet. *White Paper: IXXAT and Altera* [Online].

[26] K. Wang, C. Zhang, X. Xu, S. Ji, and L. Yang(2012); A CNC system based on real-time Ethernet and Windows NT, *The International Journal of Advanced Manufacturing Technology*, ISSN 0268-3768, 65(9-12):1383-1395.

[27] G. Morel, P. Valckenaers, J.-M. Faure, C. E. Pereira, and C. Diedrich(2007); Manufacturing plant control challenges and issues, *Control Engineering Practice*, ISSN 0967-0661, 15(11): 1321-1331.