Hindawi Publishing Corporation
VLSI Design
Volume 2016, Article ID 5752080, 10 pages
http://dx.doi.org/10.1155/2016/5752080

Research Article

A Novel Time Synchronization Method for Dynamic Reconfigurable Bus

Zhang Weigong,[1,2] Li Chao,[1,3] Qiu Keni,[1,3] Zhang Shaonan,[1,3] and Chen Xianglong[4]

[1]*Beijing Engineering Research Center of Highly Reliable Embedded System, College of Information Engineering, Capital Normal University, Beijing 100048, China*
[2]*Beijing Center for Mathematics and Information Interdisciplinary Sciences, Beijing 100048, China*
[3]*Beijing Advanced Innovation Center for Imaging Technology, Beijing 100048, China*
[4]*China Academy of Aerospace Systems Science and Engineering, No. 16 Fucheng Road, Haidian District, China*

Correspondence should be addressed to Qiu Keni; qiukn@cnu.edu.cn

Received 28 September 2015; Revised 16 December 2015; Accepted 11 February 2016

Academic Editor: Spyros Tragoudas

UM-BUS is a novel dynamically reconfigurable high-speed serial bus for embedded systems. It can achieve fault tolerance by detecting the channel status in real time and reconfigure dynamically at run-time. The bus supports direct interconnections between up to eight master nodes and multiple slave nodes. In order to solve the time synchronization problem among master nodes, this paper proposes a novel time synchronization method, which can meet the requirement of time precision in UM-BUS. In this proposed method, time is firstly broadcasted through time broadcast packets. Then, the transmission delay and time deviations via three handshakes during link self-checking and channel detection can be worked out referring to the IEEE 1588 protocol. Thereby, each node calibrates its own time according to the broadcasted time. The proposed method has been proved to meet the requirement of real-time time synchronization. The experimental results show that the synchronous precision can achieve a bias less than 20 ns.

1. Introduction

With the development of science technology, embedded systems have been widely used in the automotive electronics, aerospace, financial areas, and so forth, which require high reliability, precision, and real-time performance [1–3]. As the link of various components in embedded systems, the bus plays an important role in guaranteeing the system's real-time response and reliability. However, in aerospace scenarios which need high reliability and timeliness, traditional buses such as RapidIO [4] and SpaceWire [5] have several shortcomings:

(1) There are lots of communication resources statically redundant during working. The system is not equipped fault tolerance function, and the communication rate is quite low. The system could not work if any one of the links fails.

(2) Routers or repeaters are needed for interconnection between multiple devices, which is not friendly to system miniaturization. On the other hand, these routers and repeaters also have influence on system reliability and bring a lot of uncertain factors for system communication, which further impact the real-time performance.

In order to solve these problems, we devised UM-BUS, a dynamically reconfigurable high-speed serial bus. UM-BUS is built on multipoint low voltage differential signaling (MLVDS) [6] technology. It allows 2 to 32 lanes to transmit data concurrently. Devices can interconnect directly, with no need of routers. UM-BUS supports data communication between multiple master nodes, and the maximum communication rate can reach 6.4 Gbps. The bus could tolerate up to 31 channels fault in real time.

UM-BUS can be configured as single-master mode and multimaster mode. In the multimaster mode, the bus supports direct interconnections up to eight master nodes and multiple slave nodes, but only one master node can occupy

Table 1: Comparison to the other bus technologies.

	UM-BUS	Space Wire	TTE	1553 B
Bus rate	400 Mbps~6.4 Gbps	2~400 Mbps	1 Gbps	1 Mbps
Communication mode	Multimaster, command/response	Limited full-duplex	Full-duplex	Signal bus control, command/response
Connection mode	Bus	Star	Star	Bus
Dynamic fault tolerant	Yes	No	Yes	Bus switching
Synchronization accuracy	<20 ns	<9 us	<100 ns	<2 us

the bus and initiate communication request in any time. UM-BUS arbitrates bus using the slots token rotary method, which requires all nodes to be kept in the same time system and guarantee the bus arbitration running orderly.

The previous time synchronization algorithms cannot meet the demands of the UM-BUS system due to their complicated hardware or software design. The problems exposed by those algorithms have become the motivation of the proposed novel time synchronization method for UM-BUS. In this paper, a method of time synchronization with transmission time correction on UM-BUS is presented. The contributions of this paper are as follows:

(1) An improved Precision Time Protocol (PTP) time synchronization algorithm is proposed and realized using hardware language, VHDL.

(2) The novel nodes "proofread time" mechanism is designed based on time broadcast packet.

(3) Time synchronization [7] can be completed in one clock count cycle. The cost on bus bandwidth is very low.

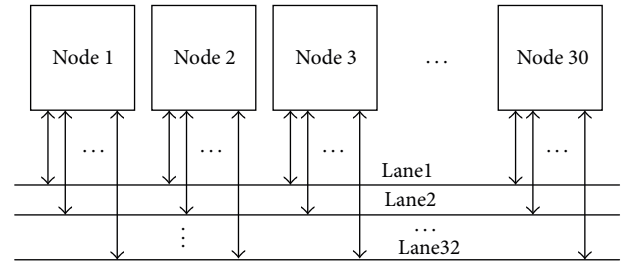


Figure 1: UM-BUS topology.

2. Dynamic Reconfigurable Bus (UM-BUS)

2.1. Relation Work. There exist several bus technologies with time synchronization function. (1) 1553B [8] bus supports command/response and "broadcasting" communication modes. The bus controller can use this "broadcasting" mode and send time synchronization messages to the other RT (Remote Terminal). Basically, 1553B realizes the time synchronization in a software way. However the accuracy of the synchronization is not satisfactory. It can only reach μs order of magnitude. (2) TTEthernet [9, 10] adopts a "centralized-calculation-feedback" algorithm to realize the global time synchronization. Firstly, the Compression Master (CM) centralized Protocol Control Frame (PCF), which sent from the other Synchronization Masters (SMs). Then, the CM calculates a compromise time and sends PCF including the calculated time to SM. Finally these SMs modify their own local time according to the PCF sent by CM. In this way, TTEthernet can reach high time synchronization accuracy and achieve ns order of magnitude. But its time synchronization algorithm is complicated. It is not easy to implement in hardware. (3) SpaceWire bus network's time synchronization is implemented by the time code mechanism. Usually, the timing unit activates the TICK-IN signal provided by time host interface periodically. Then the time host inserts time code into the package and sends the package to the SpaceWire network. However, since the SpaceWire network needs the support of router, the delay error will accumulate over time. The time synchronization accuracy can only reach μs order of magnitude.

The corresponding architecture and the minimized high-speed bus are required to support the high performance, miniaturization, standardization, and high reliability development of embedded systems. We developed the UM-BUS [11] to follow the development tendency of embedded systems. The comparisons among the UM-BUS and the commonly used buses are shown in Table 1.

2.2. UM-BUS Protocol. As shown in Figure 1, UM-BUS is characterized the bus topology with multichannel transmitting data concurrently. The nodes can be directly connected to the bus without routers or repeaters. Those nodes are classified into master nodes, slave nodes, and monitor nodes according to their different functions. If some lanes fail during communication, the bus controller can detect the channel state in real time and mask those fault lanes. Information can be transmitted through other active channels dynamically.

Figure 2 shows the protocol model of UM-BUS which has three logical layers: the Transaction layer, the Data Link layer, and the Physical layer. The top layer of the model is the Transaction layer which is mainly responsible for the management of the entire bus system, exchanging information with upper application or peripherals. The middle layer is the Data Link layer, which contains of MAC sublayer and the Transport sublayer. The main task of MAC sublayer is link detection. A table named Link Condition Table (LCT) can be formed, which can identify the lane status for data transmission. The Transport sublayer realizes data transmission and recombination base on LCT. The bottom layer is the Physical layer, which includes the Logical Submodule and the Electrical Submodule. The Logical Submodule is responsible for the transmission data 8 b/10 b encoding and

Table 2: Bus protocol data packet.

Header (16 B)								Data (1025 B)	
1 B	1 B	1 B	6 B	4 B	1 B	1 B	1 B	1024 B	1 B
Dest Num	Src Num	Cmd Frame	Addr Offset	Short Data	Ack Cmd	Ack Sta	Cmd CRC	Data	Data CRC

Note that Cmd, CRC and Data CRC are calculated separately.

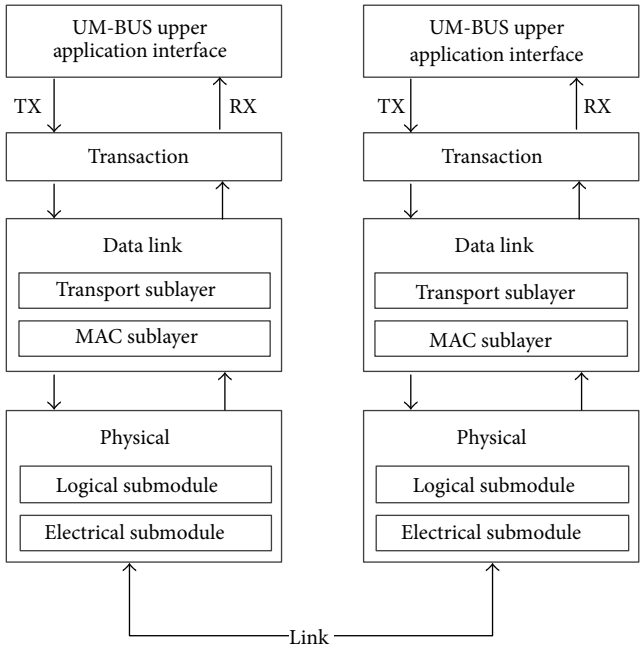


Figure 2: UM-BUS protocol model.

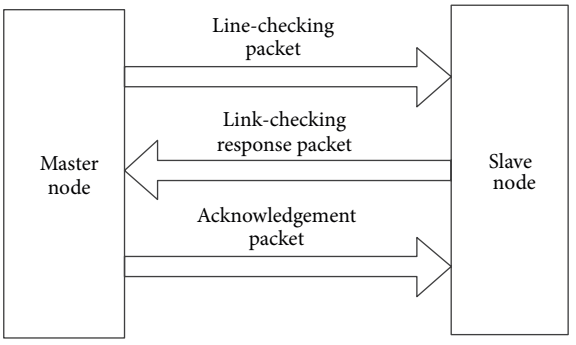


Figure 3: Link-checking process.

decoding, clock recovery, and transmission mode switched between serial and parallel. The Electrical Submodule defines the physical characteristics of the bus.

UM-BUS transfers data between various protocol layers using the data packet format as presented in Table 2. It employs the "master command and slave response" mode to complete data transfer. The master nodes send the bus protocol data packet to the slave node on the bus. The slave node executes the corresponding read or write command and then returns the status or data back to the master node. The bus protocol data packet contains the Short Packet Data (SPD) and the Long Packet Data (LPD), as shown in Table 2. The SPD only has the 16 B header, which responds to the control commands or short length data transfer. The LPD can be used to transfer large amount of data.

All the nodes on the bus will conduct a line-checking in every lane during the system initialization or fault detection. The maximum number of the lanes is 32. The line-checking process, as shown in Figure 3, sets up LCT through a three-handshake process. If the master node receives the link-checking response packets from the target slave node, LCT will be set in the master node and the line will be marked as available for data transfer. If the slave node receives the link-checking acknowledgement packet from the master node, LCT of the target node will be set. In this way, all nodes interconnected to the bus should keep a LCT. The Transport sublayer in each node will allocate or recombine data dynamically according to LCT.

3. A Novel Time Synchronization Method for UM-BUS

In this paper we select one master node in the UM-BUS system as the Time Master Node (TMN). The TMN broadcasts Time Packet (TP) periodically to the other Time Slave Nodes (TSNs) where time codes are included. When TSN receives TP from MSN, it will calibrate its own time according to the TP. The time synchronization of the UM-BUS contains the following main aspects.

(1) Transmission Delay Measurement. The transmission delay includes time deviation between TMN and TSN, circuit delay in transceiver, and the data processing delay (coding delay). The circuit delay and the coding delay can be known when the components are selected and they are the same to all nodes. However, the transmission delays are different from each other due to the different location of each node.

(2) The Time Packet (TP) Design. The TMN periodically broadcasts time to the other TSN. In order to meet the time-accuracy requirement of the token arbitration, a reasonable TP format is needed.

(3) Time Synchronization. The TSN that receives TP from TMN uses the time code in TP and the measured transmission delay to calibrate its own time. Furthermore, we need to design a rational broadcast cycle to satisfy the time precision requirement in UM-BUS.

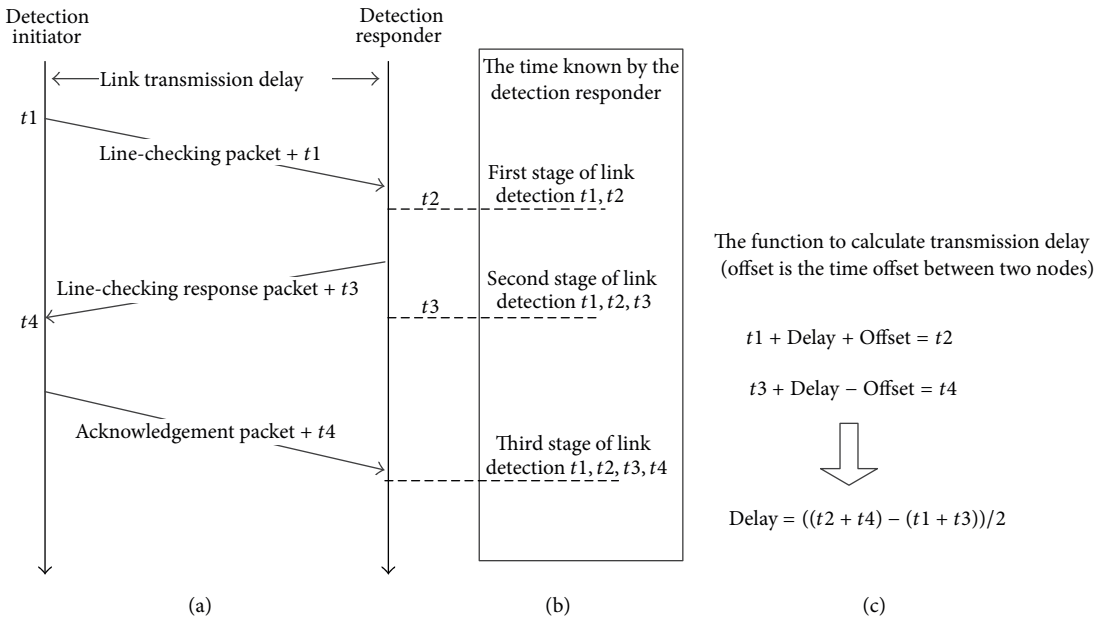


Figure 4: Measurement of delay.

3.1. Transmission Delay Measurement. The most important delay that needs to be measured is the transmission delay. We calculate the transmission delay and time deviation via three handshakes during self-checking and channel detection, referring to the IEEE 1588 protocol [8]. The calculation process is shown in Figure 4. We work out the UM-BUS line transmission delay through the record of time sending or receiving in the detection packet:

(1) When UM-BUS channel starts fault detection, the detection initiator (only bus master node) sends line-checking packet with its sending time ($t1$) as the last byte to the detection responsor (the other node).

(2) The detection responsor records each channel's receiving time ($t2$) when the line-checking packet is received.

(3) According to the UM-BUS channel detection protocol, the detection responsor sends back line-checking response packet with its sending time ($t3$) to the detection initiator from all channels.

(4) The detection initiator records each channel's receiving time ($t4$) when the line-checking response packet is received and sends acknowledgement packet with this recorded time value to the detection responder.

(5) The detection responder uses each channel's $t1$, $t2$, $t3$, and $t4$ to calculate the line transmission delay and records this value in a two-dimensional table organized by the channel number and the detection initiator's device number. The line transmission delay and offset between those two nodes can be calculated by the follow equations:

$$\begin{aligned} t1 + \text{Delay} + \text{Offset} &= t2, \\ t3 + \text{Delay} - \text{Offset} &= t4. \end{aligned} \tag{1}$$

In the equation set, Delay denotes the line transmission delay between the detection responder and the detection initiator. Offset represents the time offset between two nodes. Delay can be worked out via solving this equation set:

$$\text{Delay} = \frac{((t2 + t4) - (t1 + t3))}{2}. \tag{2}$$

Since the position of the node will not be changed once the UM-BUS system is established, the measured line transmission delay is invariant. However, it is various for different lines due to the various lengths of cables. In this paper, we set up a two-dimensional delay timetable in the MAC sublayer, which records the transmission delay between this node and the other node after channel detection.

3.2. The Design of the Time Packet (TP). When all the nodes complete their channel detection and work out the transmission delay, the Time Master Node (TMN) periodically sends the TP to other TSNs via those channels. The TP format is shown in Table 3. Each part of the TP is explained as follows.

(1) Control Word. The packet uses 8 b/10 b encoding's K28.4 to indicate a Time Packet (TP).

(2) Command Word. The first 5 bits are the device ID of TMN and the following 3 bits consist of the timing command which can represent 8 commands. Timing command 0 is defined as the time synchronization command. TSN uses the receiving time value from TMN to synchronize its nanoseconds, microseconds, and milliseconds counter. Timing command 1 is defined as the setting command of second counter. TSN

Table 3: Broadcast time packet.

Control word								Command word								Time value 1								Time value 2								Time value 3								Time value 4								Check word							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
K28.4								DEV ID					CMD			Time 1								Time 2								Time 3								Time 4								Check word							

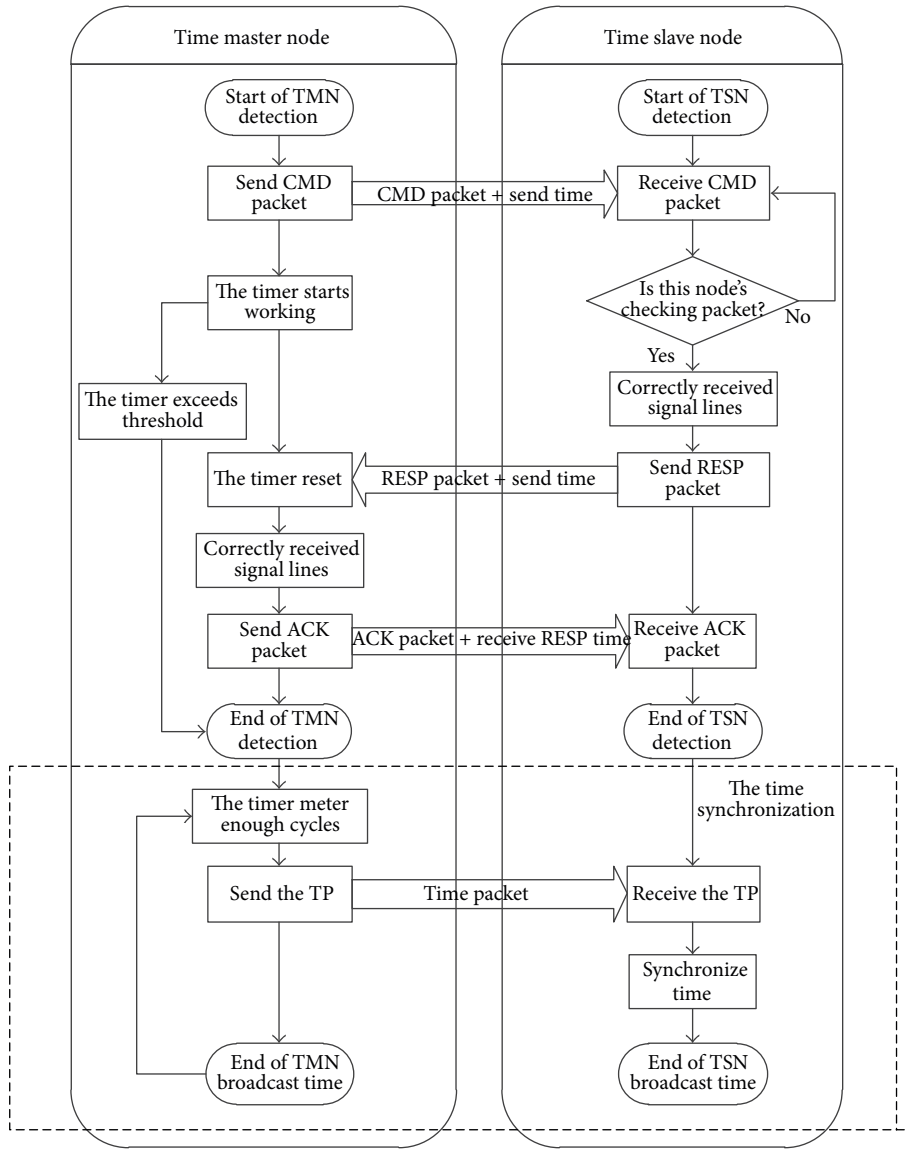


Figure 5: Time synchronization.

uses this receiving time value to set its second counter. The other timing commands are reserved.

(3) Time Code. The time code (T) is composed of four parts. For timing command 1, T indicates the current second count of TMN. For timing command 0, T consists of 10 bits of milliseconds count value (ranging from 0 to 999), 10 bits microseconds of count value (ranging from 0 to 999), and 8 bits of nanoseconds count value (ranging from 0 to 255). The others are reserved.

(4) Check Word. This byte is accumulated by the mentioned four parts of time code (T) and the command word in binary mode.

3.3. Time Synchronization. After transmission delay detection, TSN can calibrate its own time by TP, which is broadcasted periodically by TMN. The synchronized time is calculated by the following: $T\text{node} = T + \text{Delay} + Td$, where Td includes the data processing delay and transceiver delay. They are known beforehand. When working out the Tnode value, TSN uses this value to reset its corresponding counter. The process is demonstrated in Figure 5.

The top part of Figure 5 represents the channel detection and the measurement of delay. The bottom part in the dashed box completes the time synchronization. TMN broadcasts TP every 2.5 ms. Note that the period of broadcast can be determined based on the synchronization precision requirements. TSN synchronizes its time when TP is received.

In this paper, the broadcast period is assigned as 2.5 ms. When the accuracy deviation of UM-BUS clock is between 20 ppm and 50 ppm, the clock will generate an error within 20 ns~50 ns every 1 ms. In the multimaster arbitration protocol, the maximum precision deviation is allowed to be 150 ns

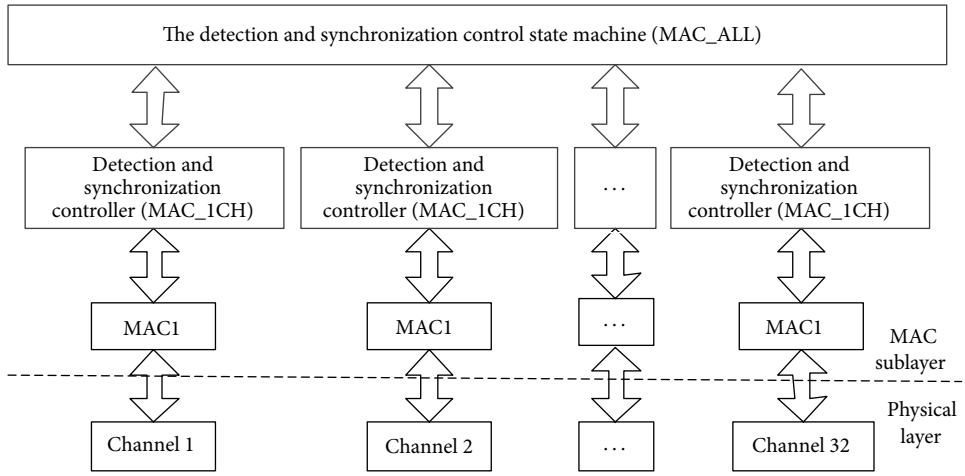


Figure 6: UM-BUS detection and synchronization model.

and the left 90 ns is needed to be eliminated. Therefore we select 2.5 ms as the TMN broadcast period.

In order to avoid single point failure, every master node that has the time synchronization function can replace the failed TMN and become the new TMN. TMN broadcasts TP through all the channels every 2.5 ms before transferring of synchronization control, so that TSN can calibrate its own time.

Since the TMN broadcasts its time code to the other TSNs every 2.5 ms, considering the link delay of the lane and some protocol overhead, a TSN will receive a PT at least more than 2.5 ms. So we set a time counter in the node which has the time synchronization and broadcasting function. The time counter works as a watchdog and the length of different node's time counter is increasing by the size of node ID. For example, TMN1's time counter length is 3 ms, TMN2's time counter length is 6 ms, and TMNn's time counter length is "3n" ms. Note that the initial time master node is TMN0. If the other TMNs in the bus system except TMN0 receive a TP during its timing period, then it will reset the time counter and restart timing. If a TMN does not receive any TP at the end of the timing period, it will believe that the original TMN has failed. Then this TMN will stop the timer counter and start the time synchronization function. This node will become the new TMN and will send TP every 2.5 ms.

4. Simulation and Experimental Result Analysis

4.1. Design of Time Synchronization. The design of time synchronization is located in the MAC sublayer. The UM-BUS has 2 to 32 lanes where data can be transmitted in parallel. In order to improve synchronization efficiency and reduce the cost on bandwidth during time synchronization, this paper proposes a parallel method to conduct the time synchronization. When the channel transmission delay is calculated through the channel detection, TP is broadcasted on all the channels simultaneously. In addition, this paper uses a hierarchical control model with "centralized control and independent broadcast" to simplify the time synchronization logic and reduce the resource consumption. As shown in Figure 6, each communication node uses the detection and synchronization control state machine (MAC_ALL) to control all channels' detection and synchronization controllers (MAC_1CH).

Table 4: Resource usage comparison.

	The proposed method	The traditional method	Improvement (prop./trad.)
LUT	18950	27071	30%
Register	11577	12059	6%
Frequency	96 MHz	101 MHz	

MAC_ALL is the main control section for detection and synchronization. Detection starts when the detection command packet or the upper layer's detection command is issued. The link transmission delay is calculated at the same time and the delay timetable is also updated. MAC_ALL also controls TP sending and receiving in each channel's MAC_1CH and realizes the time synchronization.

MAC_1CH of each channel performs TP packing or unpacking under the control of MAC_ALL. It sends and receives TP on all channels, so as to complete bus time synchronization.

Table 4 presents the resource usage comparison of the proposed hierarchical control model to the previous single-level control model with purely separated broadcast in each channel. We can see that the proposed method consumes much less resources than the traditional one.

4.2. Simulation Results Analysis. In this paper, we set up a 16-channel UM-BUS simulation system. One TMN and four TSNs are interconnected inside the system to verify the time synchronization method. The frequency of TMN's MAC clock is set to be FM, while the frequency of TSN-1's MAC clock is set to be FM + 20 ppm, the frequency of TSN-2's MAC

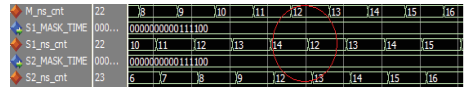
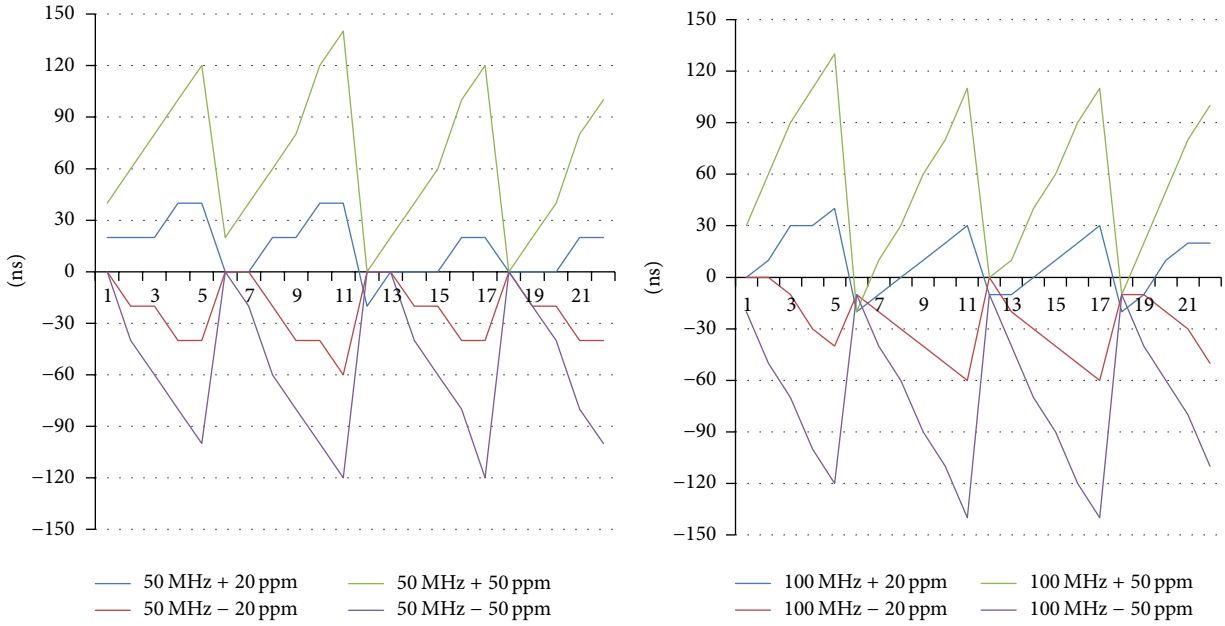


Figure 7: Simulation result.



(a) 50 MHz clock frequency

(b) 100 MHz clock frequency

Figure 8: Results under different clock frequency synchronization.

clock as FM − 20 ppm, the frequency of the TSN-3's MAC clock as FM + 50 ppm, and the frequency of the TSN-4's MAC clock as FM − 50 ppm. These settings are used to simulate the tiny crystal precision deviation of different nodes in real situations.

We use MODELSIM to simulate the time synchronization method. The simulation result is shown in Figure 7, where FM of TMN is set as 50 MHz. MASK_TIME is the transmission delay calculated by the channel detection. M_NS_CNT represents nanoseconds value of TMN. S1_NS_CNT and S2_NS_CNT represent two TSN's nanoseconds. Figure 7 indicates the time count values of different nodes with a little bit deviation before synchronization (as shown by the red circle in the figure). After the time synchronization, TSN's time count values may repeat (TSN's clock frequency is faster than TMN) or jump (TSN's clock frequency is slower than TMN). But they approximate to TMN's time count values. The synchronization deviation is less than 20 ns.

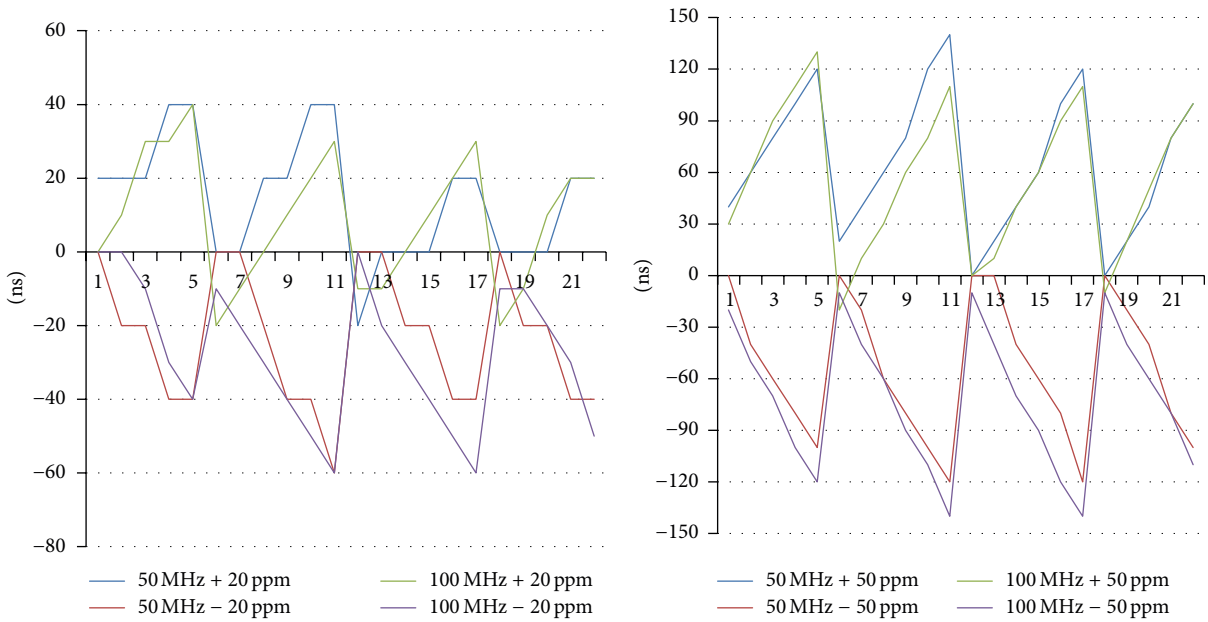
In order to observe the synchronization effect, we present the difference between the TMN and the four TSNs under different MAC clock frequencies (50 MHz and 100 MHz) and different precision deviations (20 ppm and 50 ppm). The ordinate represents the ns time difference between TSN and TMN while the abscissa represents the sampling time interval, which is set to be 500 us.

The time differences of TSN and TMN under different MAC clock frequencies are shown in Figures 8(a) and 8(b). We can see that the difference value will be close to 0 which means that TSN's time is close to TMN. We can also figure out that the smaller the clock frequency is, the smaller the convergence is.

The time value difference of TSN and TMN under different precision deviation is shown in Figures 9(a) and 9(b). We can see that TSNs can synchronize with TMN. The larger the precision deviation is, the larger the convergence is.

In summary, the method proposed in this paper can realize all nodes' time synchronization for a large range of clock frequencies with different precision deviation. This method works well in UM-BUS and achieves time synchronization in one synchronous period. The synchronization deviation is less than 20 ns.

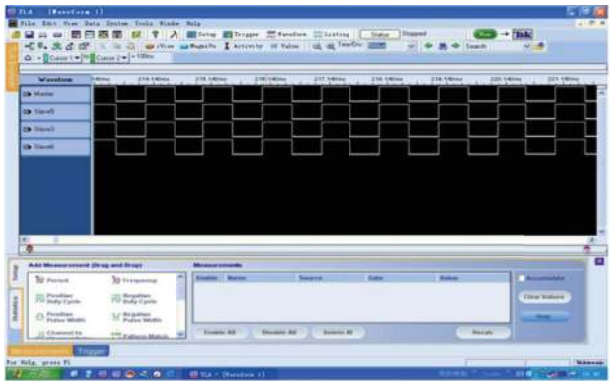
4.3. Tests on UM-BUS Platform. We built the UM-BUS testbed using VIRTEX-5 FPGA (xc5vlx85t). In the 16-lane experimental setup, the single lane's speed can reach up to 100 Mbps. The bus controller uses 11577 registers, 18950 LUTs, and 336 RAM&SRL in FPGA, rating of 22%, 36%, and 2%, respectively. We used the TEK (TLA7012 Logic Analyzer) to verify the effect of the time synchronization. We tested five nodes' time synchronization effect. Those five nodes output a pulse every 500 us. Then we compared the pulse edge of the



(a) 20 ppm precision deviation

(b) 50 ppm precision deviation

Figure 9: Results under different precision deviation synchronization.



(a) Coarse-grain time synchronization results

(b) Enlarged fine-grain time synchronization results

Figure 10: Time synchronization results on real FPGA testbed.

five nodes. Figure 10(a) shows the results of the five nodes in a coarse-grained mode. When we enlarged the figure to only one pulse edge as shown in Figure 10(b), we can find that the deviation is less than 20 ns in the best case.

5. Conclusions

This paper proposes a novel time synchronization method for a dynamic reconfigurable bus, UM-BUS. The method works as follows. First, a three-stage method "command-response-acknowledgement" is adopted to calculate and record the transmission delay between TMN and TSN. Second, TMN packs its time code and sends TP to other TSNs. Finally, TSNs that receive the TP can calibrate its own time according to the broadcast time. The time synchronization of UM-BUS can therefore be solved, and all the nodes can be maintained in the same time system. The simulation results show that the synchronization deviation is less than 20 ns with the clock frequency of 50 MHz. The time synchronization method not only provides technical support to the improvement of time certainty, but also meets the precision requirements of the bus arbitration.

The proposed time synchronization method works well in UM-BUS system and meets the time precision requirement of the dynamic reconfigurable bus. The cost of this method is low; meanwhile high reliability can be guaranteed.

Conflict of Interests

The authors declare no conflict of interests.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (no. 61170009, no. 61472260, no. 61402302, and no. 61502321) and the Project of Construction

of Innovative Teams and Teacher Career Development for Universities and Colleges under Beijing Municipality (no. IDHT20150507).

References

[1] L. W. Condra, S. J. Meschter, D. A. Pinsky, and A. J. Rafanelli, "The challenge of lead-free electronics for aerospace electronic systems," in *Proceedings of the IEEE Custom Integrated Circuits Conference (CICC '09)*, pp. 355–362, IEEE, San Jose, Calif, USA, September 2009.

[2] S. Saponara, E. Petri, M. Tonarelli, I. Del Corona, and L. Fanucci, "FPGA-based networking systems for high data-rate and reliable in-vehicle communications," in *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition*, pp. 1–6, Nice, France, April 2007.

[3] P. Šimoník, T. Mrověc, and J. Takác, "Principles and techniques for analysis of automotive communication lines and buses," in *Proceedings of the ELEKTRO 10th International Conference*, pp. 500–503, Rajecke Teplice, Slovakia, May 2014.

[4] Rapid-IO Trade Association, *Rapid-IO™ Interconnect Specification Rev2.0*, Rapid-IO Trade Association, Austin, Tex, USA, 2008.

[5] S. M. Parkes and P. Armbruster, "SpaceWire: a spacecraft onboard network for real-time communications," in *Proceedings of the 14th IEEE-NPSS Conference on Real Time*, pp. 6–10, IEEE, Stockholm, Sweden, June 2005.

[6] Texas Instruments, "M-LVDS signaling rate versus distance," TI Application Report SLLA127, 2003.

[7] IEEE Standard, "IEEE standard for a precision clock synchronization protocol for networked measurement and control systems," IEEE Standard 1588-2002, 2002.

[8] US Department of Defense, "MIL-STD-1553B: Aircraft Internal Time Division Command/Response Multiplex Data Bus," 1978.

[9] H. Kopetz, A. Ademaj, P. Grillinger, and K. Steinhammer, "The time-triggered ethernet (TTE) design," in *Proceedings of the 8th IEEE International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC '05)*, pp. 22–33, Seattle, Wash, USA, May 2005.

[10] A. Ademaj and H. Kopetz, "Time-triggered ethernet and IEEE 1588 clock synchronization," in *Proceedings of the IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication (ISPCS '07)*, pp. 41–43, Vienna, Austria, October 2007.

[11] X. Zhu, W. Zhang, J. Wang, Q. Duan, and S. Liu, "The design of high reliable serial system BUS," in *Proceedings of the International Conference on Computer Design and Applications (ICCDA '10)*, vol. 4, pp. V4-14–V4-17, IEEE, Qinhuangdao, China, June 2010.