Design of a Cooperative Vehicular Platoon System Based on Zynq/SoC Architecture

Yi Wang, Yi Zhou(✉), Wei Li, Gaochao Wang, Lin Ren, and Ruirui Huang

School of Computer and Information Engineering, Henan University, Kaifeng 475000, China
zhouyi@henu.edu.cn

Abstract. Different from traditional intelligent transportation systems, vehicular platoon systems pay more attention to interactive communications of vehicle-to-vehicle (V2V) and vehicle-to-road (V2R). Both V2V and V2R communications in platoon have higher demands of real-time and active safety applications, where low-latency transmission and strong perception capability are the fundamental guarantee of platoon cooperation. This paper proposed a cooperative vehicular platoon system based on Zynq-7000 all programmable SoC architecture, in which six miniature vehicles are designed through Zynq modules for evaluating the platooning performance. We use the Vivado development kit to create the system architecture, and evaluate cooperative communication and coordinated control technology of the platoon. The test results show that the Zynq architecture can improve the real-time processing and information interaction performance of cooperative platoon systems.

Keywords: Vehicular platoon · Cooperative control · Zynq/SoC · Programmable architecture

1 Introduction

As a typical application of IoT (Internet of things), Connected vehicles use embedded sensor devices to get vehicular status information, and implement the information interaction through vehicular networks [1], which enables real-time networking and information sharing between vehicles and other traffic elements (e.g. other vehicles, road-side units, infrastructures, and pedestrians) [2, 3].

With the rapid development of vehicular networking technology, the collaborative vehicular platooning networks has become a promising research field of connected vehicles [4]. Vehicular platoons based on cooperative V2 V and V2R communication technology can obtain real-time information through vehicular sensors, which can

Y. Zhou—This work was supported by National Natural Science Foundation of China (No. 61304132) and partly supported by Henan International Cooperative Program of China (No. 134300510049), the Program for Science & Technology Development of Henan Province (No. 162102210022), and CERNET Innovation Project (No. NGII20151005).

Q. Chen et al. (Eds.): ChinaCom 2016, Part II, LNICST 210, pp. 335–344, 2018.
DOI: 10.1007/978-3-319-66628-0_32

automatically adjust the speed of vehicles, and keep a relatively safe distance among vehicles [5, 6].

The research of Cooperative Vehicular Platoon mainly arises from intelligent vehicle system architecture internationally. The PATH group, from the University of California, Berkeleyin, proposed the 5-tier architecture based on the intelligent vehicle-highway system Architecture in 1991 [7], specifically including the network layer, link layer, coordination layer, control layer, and physical layer. The network layer mainly solves the routing problem while the link layer adjusts the speed of vehicles along with the real-time traffic on the road. The coordination layer selects the corresponding control strategy, and the control layer implements it subsequently. At the same time, the physical layer includes vehicle-mounted controllers and the physical structure of the vehicle [8]. A vehicular collaborative driving system structure was proposed by Tsugawa *et al.* in 2000. They analyzed the demand of vehicle cooperative driving function and designed a 3-tier system structure, including traffic control layer, vehicle management layer, and vehicle control layer [9]. In this system, the traffic control layer is positioned on the roadside while the vehicle management layer and the control layer are located on the vehicle side, which are used for the implementation of cooperative driving strategy. Hallé and Chaib-draa proposed the collaborative driving systems, which has made a detailed description of data acquisition and processing, platoon coordination control, and platoon communication in the process of cooperative driving [10].

Nowadays, as the emerging embedded applications such as system on a programmable chip (SOPC) and advanced driver assistance systems (ADAS) are developing rapidly, traditional embedded development platforms are hard to satisfy the big data demand, high computation ability and scalable development capability [11]. Xilinx officially launched the first scalable processing platform for Zynq-7000 series embedded in the General Assembly 2010 held in Silicon Valley, where the ARM® Cortex™-A9 MPCore (PS) and 28 nm low-power FPGA logic (PL) are tightly integrated together [12].

Using real cars to carry out the experiment will bring great difficulty to the research and design in real environments, and the complexity of the experiment and maintenance will greatly increase the cost. Therefore, we take smart miniature vehicles as the experimental platform of vehicular platoon and design a cooperative vehicular platoon system based on six Zrobot-III modules, which are built by Zynq-7000 embedded SoC architecture. The remainder of this paper is as follows: Sect. 2 introduces the system model of the cooperative vehicular platoon. Section 3 describes the detailed design of the cooperative vehicular platoon system, followed by a brief of key algorithms in Sect. 4. Section 5 presents the analysis of the experimental results and Sect. 6 concludes this paper.

2 System Model

2.1 Zynq/SoC Architecture

As shown in Fig. 1, The Zynq extensible processing platform (EPP) products consists of a SoC style integrated processing system (PS) and programmable logic (PL), which

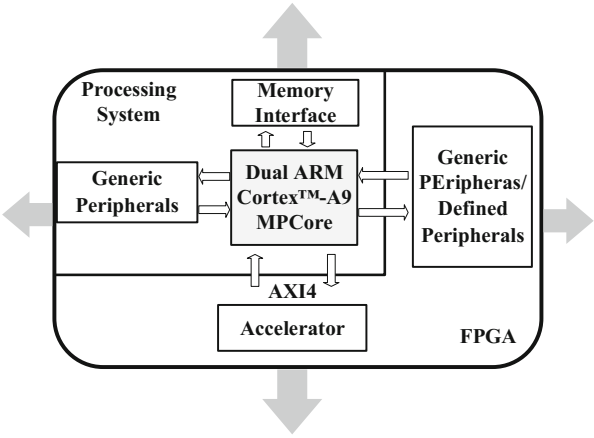

Fig. 1. Zynq/SoC architecture.

provides an extensible and flexible SoC solution on a single chip [13]. On the single chip, the PS includes the dual ARM Cortex-A9 processor, which comes with a dedicated NEON co-processor and a double-precision floating-point arithmetic unit. The PS is connected to the PL resources through multiple AXI (Advanced Extensible Interface) ports.

The PL features Xilinx 7 Series FPGA logic from the Artix and Kintex families which consists of configurable logic blocks (CLBs) and block random-access memories with high performance and ultralow power consumption. AMBA (Advanced Microcontroller Bus Architecture) bus specification is an open standard interconnection specification, which is used for the connection and management of the system function modules on the chip. ARM processor is able to control the design of the function module via AXI bus in accordance with the design of using FPGA to customize function module. Compared to a single ARM Cortex A9 board or a single Xilinx FPGA board, Zynq series products not only integrate different technology characteristic processor and FPAG on a single chip, but also build the high- performance connection between the processors and FPGA.

2.2 Cooperative Vehicular Platooning Model

This vehicle platoon system mainly consists of one miniature leading-vehicle and five miniature following-vehicles, where each vehicle is designed based on the Zynq/SoC architecture. To realize vehicular platooning cooperation, we have added some necessary sensor modules to the miniature vehicles. Table 1 lists all the peripheral devices used in each vehicle of the system.

Figure 2 shows the proposed cooperative vehicular platooning framework. The platoon moves forward in accordance with the safe distance between the vehicles, while the ultrasonic module can ensure the safe distance in real time.

The PL is mainly responsible for processing the received information and collecting the data of the vehicle through sensor modules. Then, the data is processed and

Table 1. Peripheral settings in the system.

Device name	Number
Ultrasonic sensors	3
Infrared photoelectric sensors	3
Brushless DC motors	2
Holzer velocity measurement units	2
USB camera	1
Gyroscope module	1

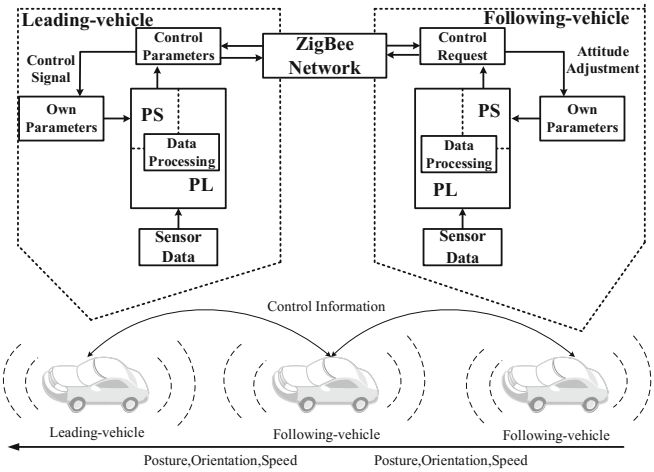

Fig. 2. System architecture.

transmitted to the PS. When the leading-vehicle meets the obstacle, the PS calculates the expected direction and speed through data from the infrared photoelectric sensors and ultrasonic sensors, and sends the signal to the following- vehicles through ZigBee networks. Following-vehicles integrate its self-perception parameters and the received data to extract characteristic and adjust attitude, and then follow the head vehicle to make the corresponding action. Self-perception parameters mainly include the current position, the distance to the vehicle ahead, the direction angle, and the speed. When a vehicle plans to leave the platoon, it will send the request commands to the vehicle platoon through the ZigBee network and perform the corresponding action when receiving the confirmation. After the vehicle leaves the platoon, the following vehicles keep up with the forward vehicle smoothly. If any vehicle intends to join in the platoon, it will send the request parameters to the leading-vehicle, and follow the last vehicle in the platoon with permission.

3 Cooperative Vehicular Platoon System

3.1 Hardware Platform

The proposed system is built on ZedBoard, which is a development board with high performance based on Zynq-7000. ZedBoard provides 512M DDR3, 256M four bit SPI FLASH and 4 GB SD card and contains an OTG USB, a USB serial port and Gigabit Ethernet port. In addition, five Pmod interfaces and a FMC (FPGA Mezzanine Card) extension connection are offered as well. We rebuild ZedBoard through integrating necessary peripheral modules to set up the miniature platooning vehicles.

Figure 3 illustrates the whole underlying hardware architecture of the miniature vehicle. Each miniature vehicle is composed of five components—power module, communication module, display module, attitude acquisition module and obstacle avoidance module.

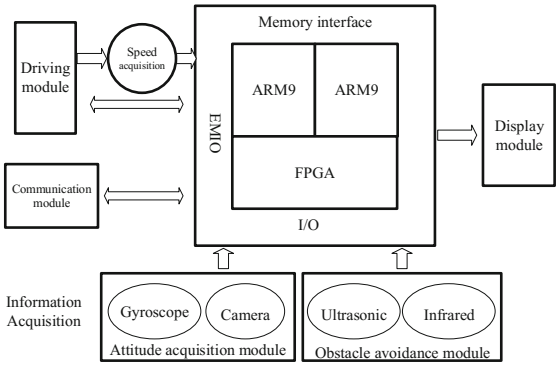

Fig. 3. Hardware architecture of miniature vehicle.

3.2 Design of PS and PL

We built an embedded Linux operating system on the PS, using Linux to manage all peripheral interfaces of the cooperative vehicular platoon system. The peripheral interfaces include the human-machine interface, acceptance of control signals, and call of IP cores. When the system is powered on, BootROM starts to control the whole initialization process. FSBL (First Stage BootLoader) fulfills the initialization of the PS, and then uses the bitstream file to configure the PL. Finally, FSBL loads U-boot into ARM to complete hardware initialization, and Linux kernel sets up the compile environment. Based on the IP module address (shown in Fig. 4), Linux driver calls the corresponding modules by the address.

Pmod on the ZedBoard has four interfaces: JA, JB, JC, JD. PWM signal is interfaced to the JA port of Pmod to control the motor's speed. Speed signal is connected to the JB port of Pmod to obtain the speed of vehicle, acceleration, and the current direction. Ultrasonic signal is linked with JD port of Pmod, which collects the distance with the front vehicle and identifies the obstacles to maintain the normal inter-vehicle

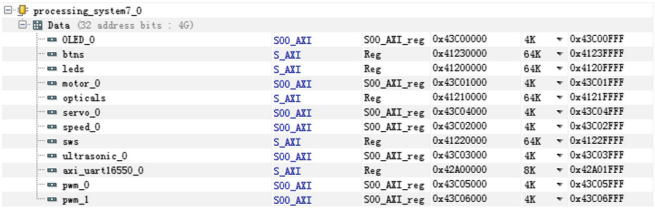

Fig. 4. Module address.

distance. Optical signal and UART signal are jointed with the JC port of Pmod for signal transmission.

3.3 IPcore

Zynq/SoC architecture is focused on IP based system realization, and IPcores have become the key technology of SoC design. In the process of design and development, we can define our own IP cores according to the specific requests. Using vivado development tools, the VHDL program will be packaged as user IPcore, which includes configurable register group, clock, reset, and interrupt port. This system designed motor, speed, UART, ultrasonic and other IPcores. The hardware architecture is illustrated in Fig. 5, where we use self-defined IPcores in Vivado.

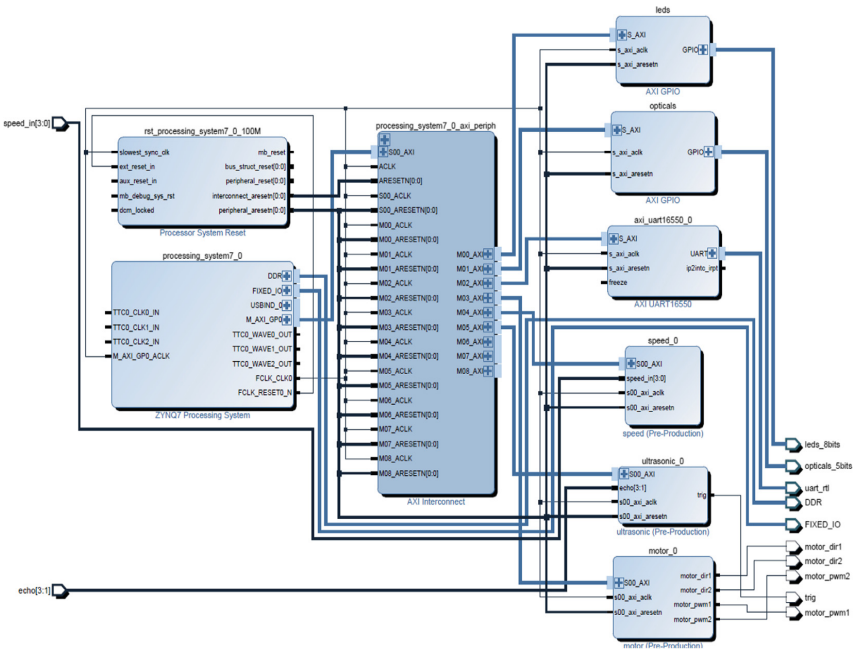

Fig. 5. Systems block configuration.

4 Design of Key Algorithms

4.1 Cooperative Sensing

The collaborative sensing requires various sensors to deal with complicated environments. Before making a control decision, we must process the different sensing data in different formats to get effective information, and then transmit to the PS according to the specific format predefined. In the process, the multi-source information fusion is necessary for cooperative sensing. As depicted in Fig. 6, the multi-source information fusion model is divided into two progressive stages. The sensor modules mainly include camera, ultrasonic sensor, gyroscope, and infrared sensor. In the first stage, the raw data obtained from the sensors are processed and transformed to characterize surrounding environments, extracting features transmitted to ARM processors (PS) in Zynq. In the second stage, the system gets the control parameters and evaluates the results after information was processed, and then sends control signals to the platoon.

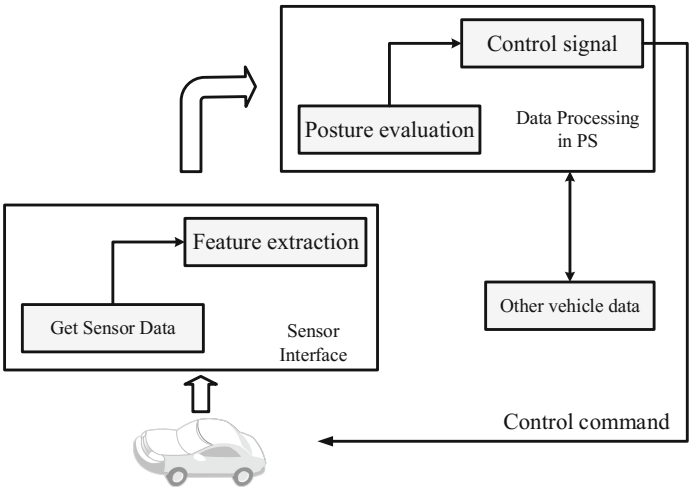

Fig. 6. Multi-source information fusion model.

4.2 Information Interaction

V2V communications play an important role in the vehicle platooning system. ZigBee is mainly used for data transmission between miniature vehicles in short distance and low power consumption, which can be used to transmit periodic data and intermittent data in low reaction time, and also effectively to ensure the transmission of control data. In this system, ZigBee module is connected to the UART port of ZedBoard, which forms a mobile ad-hoc network.

ZigBee on the head-vehicle is served as the coordinate node to supervise the whole network. In that way, ZigBee modules on other vehicles are responsible for

communication as routing nodes, Zynq chip calls UART's IPcore, function pthread_create(&tid[0], &attr[0], thread_serial_ttyPS1, NULL) gets ZigBee network data, and packages vehicle information into a data packet in accordance with the custom format, while the packet is sent to other vehicles through the UART. The communication data packet format is defined as follows:

```
StartByte ->1B (represent start byte of packet)
PropertyId->1B (represent PropertyId)
NodeId->1B(represent device node Id)
PacketLength->2B (represent the length of the package)
PrivateData->XB (represent load content)
EndByte->1B (represent end byte of packet)
```

5 Experimental Analysis

The development and design of system algorithms is based on Vivado suite, which provides a new integrated engine, IP and hardware and software integration. After the system is completed, we create the system_wrapper, and add the pins and timing constraints. At the end of the system compilation, Vivado will generate a compile report. From the utilization report in Fig. 7, it indicates that the main consumption of LUT (logical unit table) is about 2/3 and memory LUT only consumes 9%.

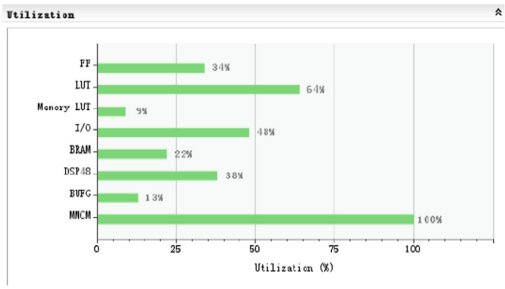

Fig. 7. Utilization report.

After the completion of code writing and simulation debugging, we make the boot. bin image which runs on the miniature vehicle. When the system is powered on, each module on the vehicle operates normally. The camera of leading-vehicle collects surrounding information to confirm and identify the obstacles ahead assisted by ultrasonic distance measuring module. The PL analyzes the underlying dense data stream of image, motion control, and other typical applications in the system, and the data is reported back to the PS, allowing the vehicle make corresponding movements of obstacle avoidance, turning and so on, and also transferring the data to the follow - vehicles behind. As shown in Fig. 8, follow-vehicles travel stably at the back of leading car according to a safety distance at a constant speed.

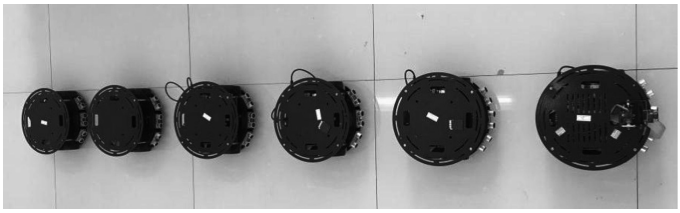

Fig. 8. Platoon prototype.

Under the normal running of the system, image distortion and dislocation will occur, making it unable to show the processing results, which influences the identification of obstacles. Since the image is displayed completely, after excluding the problem of image transmission width, the problem of unstable IPcore work of image pre-processing is found. During the design of IPcore, in hardware project, the clock of processor, data bus, VDMA transmission channel related to IPcore should be set uniformly, because the reference clock in HLS will influence the compiling results of IPcore, and the PS works normally after the problems are solved. As depicted in Fig. 9, when the front vehicle turns or deviates direction, the system can successfully detect the characteristic points on the vehicle. After that, the PS establishes the coordinate system according to the characteristic points, and calculates the deflection angle of the front vehicle.

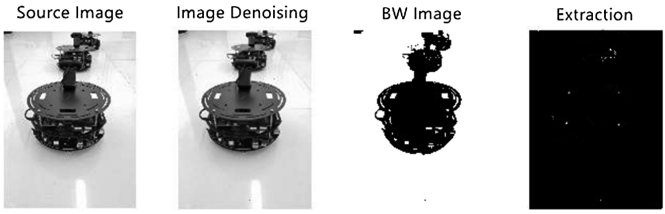

Fig. 9. Image processing result.

6 Conclusions

Communication technology is the key support of cooperative vehicular platoon system, which is the future development direction of connected vehicles. Zynq is a high-performance processing platform with low power, and it is a flexible and scalable solution. We designed a cooperative vehicular platoon system based on Zynq-7000 embedded SoC architecture, which can take place of the real car to emulate the scene of vehicular platoon and help to verify the cooperative algorithm as well. It is shown through experiments that the Zynq architecture can improve the real- time processing and information interaction performance of cooperative platoons. In the future, we will mainly focus the cooperative control algorithm embedded into Zynq, and optimize V2V interaction algorithm, which can give full play to the framework advantage of PS + PL of Zynq/SoC.

References

1. Ning, L., Nan, C., Ning, Z., Xuemin, S., Jon, W.M.: Connected vehicles: solutions and challenges. IEEE Internet Things J. **1**(4), 289–299 (2014)
2. Kaiming, R., Jizhou, L., Lingyan, L., Wenying, S.: Development status and tendency of IoV communication technology. J. Commun. Technol. **48**, 507–513 (2015)
3. Jianqiang, W., Chenwen, W., Xiaojun, L.: Research on architecture and key technologies of internet of vehicles. J. Micro Comput. Inf. **27**, 156–158 (2011)
4. Brandon, S., Michael, S.: A survey of public opinion about connected vehicles in the U.S., the U.K., and Australia. In: International Conference on Connected Vehicles and Expo, pp. 687–692 (2014)
5. Shigen, G., Hairong, D., Bin, N., Roberts, C., Lei, C., Xubin, S.: Cooperative adaptive bidirectional control of a train platoon for efficient utility and string stability. Chin. Phys. B **24**, 161–170 (2015)
6. Le, W.Y., Ali, S., George, Y.G., Abhilash, P., Wei, H.Z.: Control of vehicle platoon for highway safety and efficient utility: consensus with communications and vehicle dynamics. J. Syst. Sci. Complex. **27**, 605–631 (2014)
7. Varaja, P., Shladover, S.E.: Sketch of an IVHS systems architecture. In: Vehicle Navigation and Information Systems Conference, vol. 2, pp. 909–922 (1991)
8. Hedrick, J.K., McMahon, D., Narendran, V.K., Swaroop, D.: Longitudinal vehicle controller design for IVHS systems. In: Proceedings of 1991 American Control Conference, vol. 3, pp. 3107–3112 (1991)
9. Tsugawa, S., Kato, S., Matsui, T.: An architecture for cooperative driving of automated vehicles. In: 2000 IEEE Intelligent Transportation Systems Conference Proceedings, Dearborn (MI), pp. 422–427 (2000)
10. Hallé, S., Chaib-Draa, B.: A collaborative driving system based on multiagent modelling and simulations. J. Transp. Res. Part C Emerg. Technol. **13**, 320–345 (2005)
11. Sakaguchi, T., Uno, A., Kato, S.: Cooperative driving of automated vehicles with inter-vehicle communications. In: Proceedings of IEEE Intelligent Vehicles Symposium, Dearborn (MI), USA, pp: 516–521 (2000)
12. Xilinx Inc.: Zynq-7000 all Programmable SoC Technical Reference Manual. Xilinx Inc. (2013)
13. Roland, D., Lukas, S.: Image filter evolution on the Xilinx Zynq platform. In: 2013 NASA/ESA Conference on Adaptive Hardware and Systems (2013)