A New System-On-Programmable-Chip (SOPC) for Smart Car Braking Control

Ying-Hao Yu, N. M. Kwok, and Q. P. Ha
School of Electrical, Mechanical and Mechatronic Systems
Faculty of Engineering and Information Technology
University of Technology, Sydney
PO Box 123 Broadway NSW 2007 Australia

Abstract— **This paper presents a new system- on-programmable chip (SOPC) design for vehicle braking control systems. Unlike the conventional control theoretic or fuzzy logic control approaches, where control signals are mostly computed on general purpose computers, the control here is much simpler and can be implemented reliably based on a single chip processor. We propose a braking nervous factor (BNF) to predict the braking responses of a driver for arbitrary distance and relative speed of two vehicles moving in a straight line. The proposed BNF control system can learn from the braking experience from the driver, and thus a database can be created for autonomous driving to offer the driver both safety and comfort.**

I. INTRODUCTION

Transpostation systems play a crucial role in all aspects of today's globalized world. There has been significant research effort devoted to improving safety, efficiency, and embedded intelligence to these systems. With various types of sensors, CPUs, software, communication capacities, monitoring and control systems, modern vehicles with sensor-networked road infrastructures are increasingly contributing to reduce the risks of accidents, increase capacity, enhance driver's comfort and performance by enabling intelligent driver assistance systems, and allowing for new levels of vehicle autonomy [1]. Control architecture has been developed for improvement of highway capacity and safety with full automation at the single vehicle level and platoon level [2]. To this end, interesting inputs can be obtained from robotic formation research, where an essential requirement is the cooperative control of vehicles moving in desired geometrical patterns, see, e.g. [3]. Although experimental platoon tests on highways have proved successful, as remarked in [4], due to financial and practical limitations, the research focus has recently emphasized more on driver assist systems that can be independently implemented in modern cars without costly modifications in the infrastructure. On the other hand, while "fly-by-wire" systems have been applied in aviation for over two decades, "drive-by-wire" systems for automotive vehicles are still a research topic due to high levels of required safety, reliability, and cost-effectiveness. In this regard, driving functions such as automated parking, speed adjustment and lane keeping during highway travel have been interesting topics for networked driver assist systems.

In intelligent transport systems, once having acquired sufficient information of the vehicle state with respect to other vehicles and the road, control actions are issued to either assist the driver in controlling the vehicle or autonomously take over the command of the vehicle itself. Thus, automatic control is playing an increasingly significant role in control of all modes of transportation, extending from vehicle levels to much higher levels and more complex functions [5]. The lower-level controllers for a vehicle involve the control of its engine, brakes, and steering while a higher-level controller determines the vehicle desired motion and trajectory from its kinematics. Conventionally, the vehicle level controllers are designed on the dynamics of the vehicle. This would require accurate models of the engine, throttle, driveline, tyre and brake for longitudinal control, and steering, slip models for lateral control.

To design braking control systems, a comprehensive model for the brake is presented in [6], depending on various parameters and uncertainties such as the brake temperature and tire-road friction. For this, a variable structure system approach has been proposed for nonlinear vehicle brake control in collision warning/collision avoidance applications [7]. Different compression brake control techniques were designed for longitudinal control of heavy trucks [8]. Full vehicle active brake control using a rigid ring tyre model is proposed in [9], where the criterion is braking as fast as possible. These controllers are model-based in nature, and subject to nonlinearities, system uncertainties, and external disturbances. System reliability may then be an issue particularly under limited communication and computation capacity [10]. Another factor that should be considered is the natural feeling and psychological behavior of the driver and passengers in the vehicle [11]. Addressing these concerns, this paper aims to design an intelligent braking system that can handle reliably these tasks in a separate unit. To avoid the access to a large external storage device when running the whole control program and acquiring information from outside [12], we propose to design the System-on-Programmable-Chip (SOPC) using the Field Programmable Gate Array - Programmable Logic Devices (FPGA-PLD) technology. For this, a braking nervous factor (BNF) is incorporated in the control design, considering the driver's comfort.

The remainder of this paper is organized as follows. After the introduction, Section II introduces the BNF method and compares its advantage over a traditional fuzzy control. In Section III, the SOPC chip design is described with timing sequences shown in the simulation results. Some discussion on the development is given in Section IV. Finally, a conclusion is drawn in Section V.

II. BNF CONTROL

The basic braking problem of smart cars may generally be solved by using fuzzy logic control [13]. Fig. 1 shows an example of the basic braking control system by the fuzzy PD controller [14]. The braking system therein requires the vehicle to stop exactly in front of the target by issuing a safe stop distance, D_s, with appropriate membership functions for the distance error e_d, target speed error e_v, and the braking positions. Accordingly, the control rules are formulated in a standard way to design the fuzzy controller for braking.

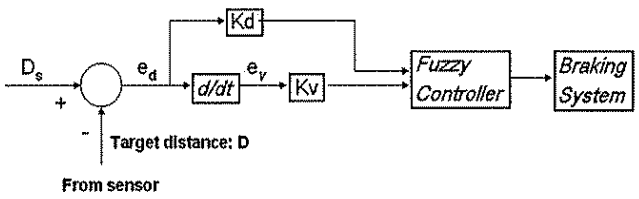


Fig. 1. Fuzzy PD braking control system.

Although the fuzzy PD control can simplify the calculation effort to design complex control systems, fuzzy sets are not flexible to predict an arbitrary value which is outside its membership universe of discourse [13]. Following the fuzzy braking control system shown in Fig. 1, the car velocity is limited by the safe stop distance, D_s. Therefore, the vehicle will bump to the target if initially it is too fast and impossible to stop within D_s. In other words, a braking system using fuzzy logic control might need a great number of rules for arbitrary speed and distance, which obviously depend on computational resources. On the other hand, as the general purpose computer system usually relies on external storage devices and information acquired from sensors, any error occurring in the system operation or storage devices will deteriorate the braking control performance and could affect the driver's safety. Therefore, a control system implemented on a single chip will be more reliable and also commercially viable. To consume less computational resources, the driver's braking behavior for arbitrary target distance and relative speed is proposed in lieu of the fuzzy controller in Fig. 1.

The stop distance for a vehicle traveling on a straight line depends on the square of its speed as given in [15]. The safe stop distance D_s thus satisfies:

$$D_s(v) \geq \frac{v^2}{2g\mu}, \tag{1}$$

where μ is the friction constant around 0.8 in good weather, g is the constant gravitational acceleration, 9.8 m/sec^2, and v is the car relative speed, m/sec, to the front target. At a certain value of D_s, m, the full braking nervous factor (BNF), N_f, is proposed as:

$$N_f(v, D_s) = \frac{v^2}{2g\mu D_s}. \tag{2}$$

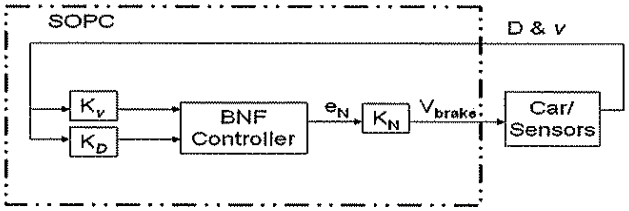
Under the situation of N_f., the driver will press the brake to its bottom position, B_f. Thus, an arbitrary position B of the brake will normally correspond to a BNF level N, which is a function of vehicle's speed v and distance D to the front target:

$$B = \frac{B_f N}{N_f}, \tag{3}$$

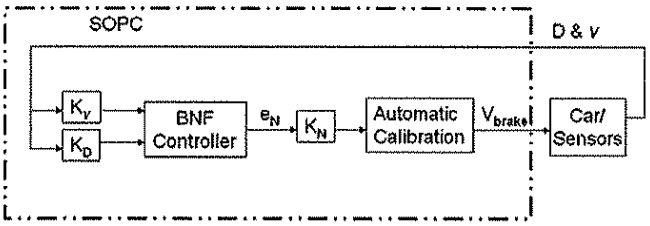
and

$$N(v, D) = \frac{v^2}{2g\mu D}. \tag{4}$$

The control model can now be shown on the Fig. 2 (a), where feedbacks from the vehicle's distance to front target D and speed v are obtained from sensors.



(a) BFN control model in SOPC.



(b) BFN control model in SOPC with braking calibration function.

Fig. 2. BNF control using SOPC.

In addition, although the vehicle is stopped or decelerated abruptly at full brake B_f, the driver and passengers may not feel comfortable. Also, repeated full braking will exhaust the brake quickly and increase the risk of brake sticking. Taking these observations into account, a calibration function is augmented as shown in Fig. 2 (b). Thus, the braking algorithm is proposed as:

$$B_{(i+1,1)} = \frac{B_f N_{i+1}}{N_f}, \text{ if } N_{i+1} \leq N_i \tag{5}$$

or else

$$B_{(i+1,1)} = B_{(1,1)} + \sum_{j=1}^{n} \Delta B_{(j,1)}, \text{ if } N_{i+1} > N_i \quad (6)$$

where i=1,2,…,*n* and *n* is the number of active intervals for a brake length.

As BNF adopts human behavior in pressing the vehicle's brake at different values of speed and distance, the proposed control system can handle braking efficiently with less memory space and computational tie while increasing driver comfort.

III. DESIGN WITH SOPC

The new SOPC chip is designed to implement the proposed BNF and braking logic control, which is an auxiliary function for calibration of the braking system to adopt behavior of the driver. Here, the software is transferred into the hardware circuit in order to save the memory space from programming and to achieve flexibility for parallel operations. The memory space in the SOPC is left entirely for the system information acquisition. The architecture of the SOPC chip is described as following.

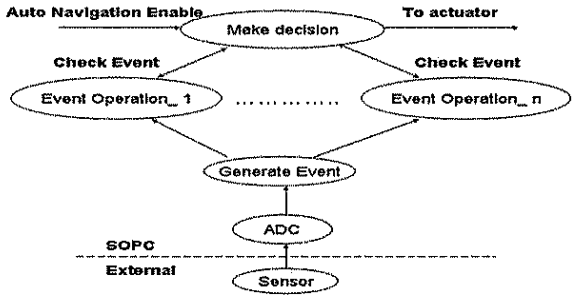


Fig. 3. Hierarchical structure of SOPC.

A. *Basic hardware structure*

In contrast to the general purpose computer which is designed to process numerical data by the Arithmetic and Logic Unit (ALU) [16], our SOPC chip design is mainly designed for logical operations as in the software programming [17].

The chip logic circuit is classified into several cores (groups) for different functions, one core deals with one function. The cores are presented in a hierarchical structure as shown in Fig. 3, wherein the operation between cores can be synchronous or asynchronous control. The SOPC cores can handle functions like the Analog to Digital Converter (ADC) interface and the output interface.

A logical operation delivered between two cores is called an *event*. In our system, events are defined as the nervous factor *N* and the relative braking strategy *B*. The intelligent braking behavior operates by judging the event logic and then obtains the correct strategy for the braking system.

B. *SOPC flow chart*

Figure 4 shows the flow of SOPC logical operations. Here, the SOPC chip is not only used for BNF braking control but it can also complement braking behavior of the driver.

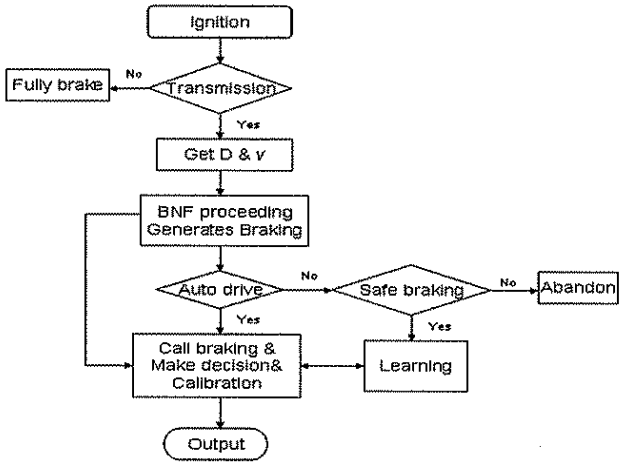


Fig. 4. Operation flow of SOPC.

Table 1. Simulated scenarios of SOPC.

Transmission position: 0: off 1: on		Distance Relations: Not Change: 11 Leaving: 10 Closing: 01 Car Stop: 00	Both Car distance: (m) (Arbitrary)	Car Relative Speed: (m/sec)	Driver Braking Position
1bit		2bits	6bits	5bits	6 bits
0	(a)	00	111110	00000	000000
1	(b)	11	111110	00000	000000
1		01	111110	00001	000010
1		01	111110	00101	000011
1	(c)	01	111110	01001	001000
1	(d)	01	111110	01001	000010 (Wrong Braking)
1	(e)	01	111110	01001	000000 (Auto drive)
1	(f)	01	111110	00111	000000 (No experience)

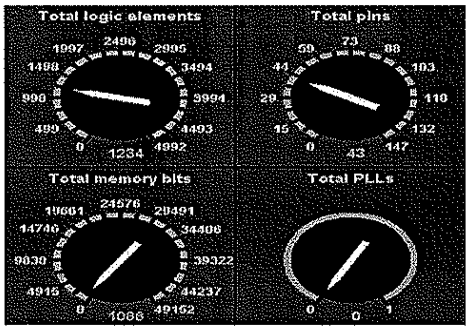
The proposed SOPC initially learns from the driver's experience to formulate its BNF control strategies. As soon as the driver's braking behavior results in a lower braking position (higher speed), the system will stop learning the driver's behavior and switch to updating the BNF braking strategy in the database of driver's experience stored in the chip. In other words, if experience from the driver is not efficient for braking, the chip can control the system by commanding appropriate brake positions from BNF control strategies in the autonomous driving mode.

C. *On-chip simulation*

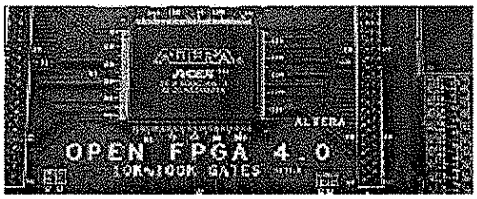
We simulated the chip by using the input scenarios that have been saved in its inbound Read Only Memory (ROM).

The scenarios for simulation on SOPC chip are shown on Table 1. Signals for the transmission position, braking response, relative car distance, relative speed, and brake positions represent all SOPC responses from the ignition of the vehicle by the driver to its autonomous driving mode.

We designed the chip on the basis of the FPGA & PLD test kit with the Altera EP1K100QC208-1 platform by using the Quartus II design tool. The SOPC chip has inbound ROM and 49152-bit RAM and 4992 logic elements. The resource usage is presented in Fig. 5(a). The chip designed is shown in Fig. 5(b), implementing the proposed BNF braking control only consumed nearly one third of the total logic gates. That gives us a great opportunity to embed more intelligent functions for autonomous driving in the future.



(a) SOPC resource usage on Quartus II design tool.



(b) The proposed SOPC chip

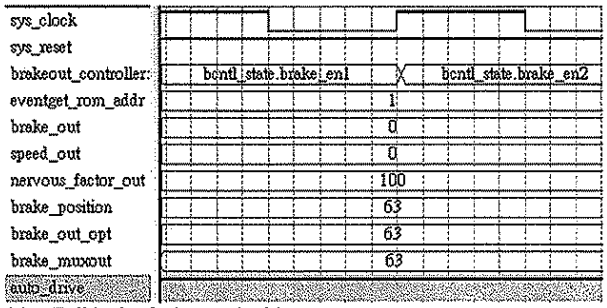
Fig. 5. SOPC resource usage and chip designed.

The simulated timing sequences are shown in Fig. 6 with simulated scenarios without the brake calibration function taken from Table 1. On these sequence charts, the nervous factor is scaled from 0 to 100% and its value is set in the decimal form for reading convenience. Corresponding to Table 1, cases for starting the car with full brake, normal run at a steady speed (nervous factor 0), learning driver's response at position 8 of the brake, bad braking response at position 2 of the brake, calling the experience database for autonomous driving, and no more experience learned from the driver are represented by the timing sequence charts shown respectively in Figures from 6(a) to 6(f). The signals used therein are defined as:

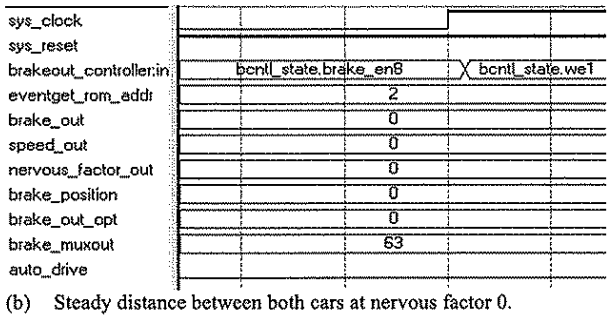
System clock:	clock source of chip.
System reset:	reset signal of chip.
Eventget_rom_addr:	scenario address in ROM.
Brake_out:	brake behavior of driver.
Speed_out:	relative speed to front target.
Nervous_factor_out:	BNF value.
Brake_position:	brake position by BNF.
Brake_out_opt:	brake learning selection between the BNF and driver.
Brake_muxout:	braking strategy output, only valid by Auto_drive.
Auto_drive:	autonomous driving trigger.

Figure 7 illustrates the braking scenarios when two cars are running at a small intervehicle distance. These include cases when the brake position increases in accordance with the increasing nervous factor (a), stops increasing as the nervous factor is not changing (b), keeps the current status as the nervous factor starts decreasing (c), and lastly when the BNF controller is released as the two cars stop closing (d). Figures from 7(a) to 7(d) shows respectively in the corresponding timing sequence charts. They justify the need for inclusion of the braking calibration function described in Fig. 2(b). Definitions of the additional signals of Fig. 7 are as follows:

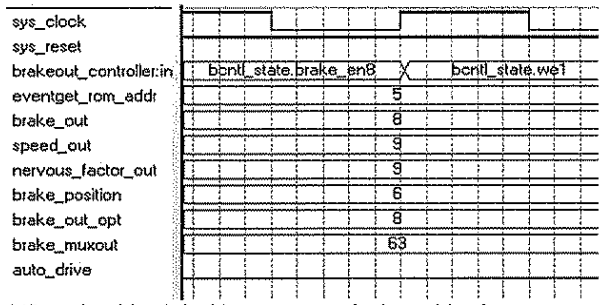
Clock_nervous:	clock signal for calibration logic.
Reset_nervous:	reset signal for calibration logic.
Nout1:	nervous signal N_{n+1}.
Nout2:	nervous signal N_n.
AA:	braking position B_n from "brake_position".
BB:	braking position B_{n+1} from "brake_position".
Nadd:	value of $B_{(n+1,1)}$.
Relation_out0:	distance relation of both car, bit 1.
Relation_out1:	distance relation of both car, bit 0.
Brake_calibration:	calibrated braking output.



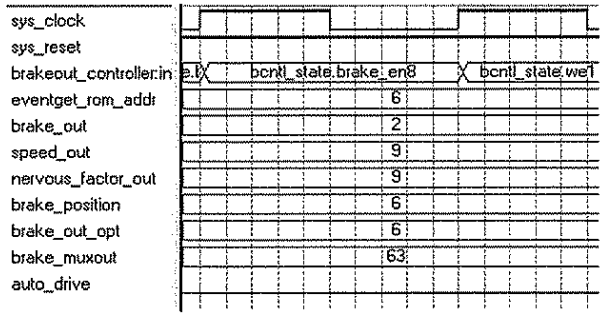
(a) Full brake during car ignition.



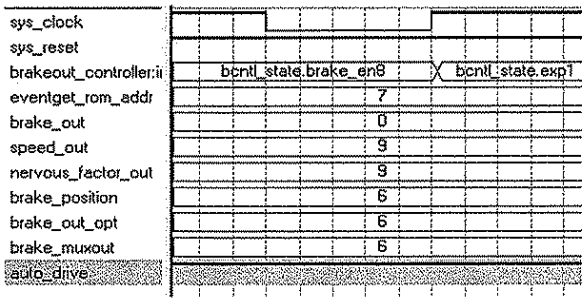
(b) Steady distance between both cars at nervous factor 0.



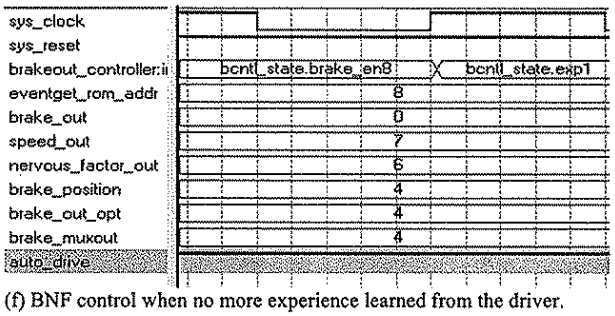
(c)Learning driver's braking response at brake position 8.



(d) Risky situation at brake position 2, updating BNF database.

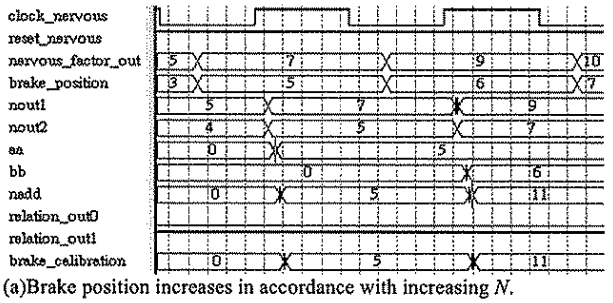


(e) Calling for autonomous driving.

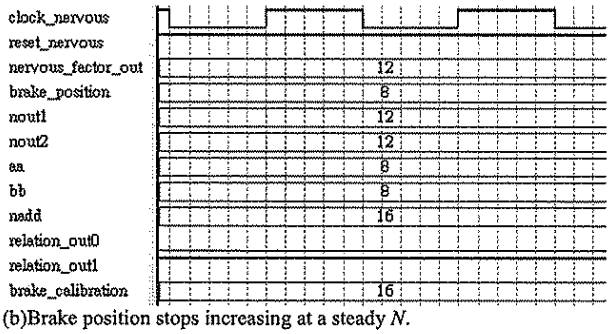


(f) BNF control when no more experience learned from the driver.

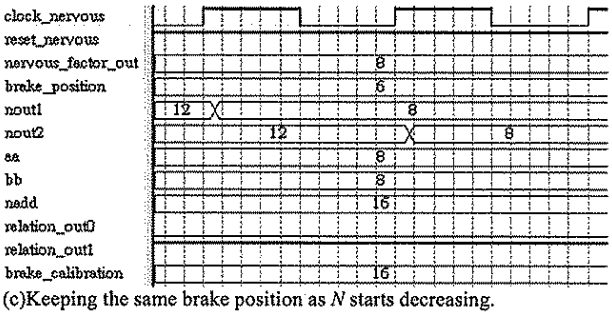
Fig. 6. SOPC simulation: time sequences.



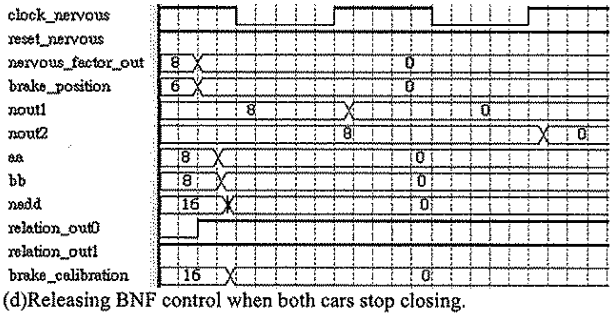
(a)Brake position increases in accordance with increasing *N*.



(b)Brake position stops increasing at a steady *N*.



(c)Keeping the same brake position as *N* starts decreasing.



(d)Releasing BNF control when both cars stop closing.

Fig. 7. SOPC simulations with braking calibration functions.

IV. Discussion

The BNF method for vehicle's braking control proposed in this paper illustrates the advantage of the SOPC implementation: enhanced safety, reliability and increasing comfort. While the traditional fuzzy control using look-up table rules takes more memory capacity, the BNF control adopting the nervous level from the relative car speed can issue control actions with very simple calculations and be implemented in a single chip.

In addition, the general purpose computer usually faces unexpected threats from hardware failures and computer system errors, thus increasing the potential risk for high speed driving.

According to the usage of SOPC resource as shown in Fig. 5, the BNF control system and the new chip architecture only consumed less than 50% the logic gates in the small FPGA chip. It means that more complex logic designs can be implemented as the software programming to build an intelligent system in a single chip.

V. Conclusion

In this paper, we have presented a new system- on-programmable chip design for vehicle braking control systems. A braking nervous factor is proposed to predict the braking responses of a driver for arbitrary values of the safe stop distance to the front vehicle and relative speed of the two vehicles moving in a straight line. The proposed SOPC and BNF control system can learn the braking behavior of the driver to create an experience database for the autonomous driving mode. This will enhance the driver's safety and comfort. The SOPC design is performed on the platform with Altera FPGA-PLD chip using the Quartus II development tool. Simulations of the chip timing sequence charts are shown to illustrate the design validity.

VI. Acknowledgments

This work is supported by the Centre of Excellence programme, funded by the Australian Research Council (ARC) and the New South Wales State Government.

References

[1] F.Y. Wang, D. Zeng, and L. Yang, "Smart cars on smart roads: an IEEE Intelligent Transportation Systems Society update," *Pervasive Computing*, pp. 68-69, 2006.

[2] R. Horowitz and P. Varaiya, "Control Design of an Automated Highway System," *IEEE Proceedings of the IEEE*, Vol. 88, No. 7, pp. 913-925, 2000.

[3] N.M. Kwok, Q.P. Ha, and G. Fang, "Motion Coordination for Construction Vehicles using Swarm Intelligence," *International Journal of Advanced Robotic Systems*, Vol. 4, No. 4, pp. 469-476, 2007.

[4] A. Vahidi and A. Eskandarian, "Research advances in Intelligent Collision Avoidance and Adaptive Cruise Control," *IEEE Trans. Intelligent Transportation Systems*, Vol. 4, No. 3, pp. 143-153, 2003.

[5] U. Kiencke, L. Nielsen, R. Sutton, K. Schilling, M. Papageorgiou, and H. Asama, "The impact of automatic control on recent developments in transportation and vehicle systems," *Annual Reviews in Control*, Vol. 30, pp. 81-89, 2006.

[6] H. Raza, Z. Xu, B. Yang, and P. Ioannou, "Modelling and control design for a computer-controlled brake system," *IEEE Trans. Control System Technologies*, Vol. 5, pp. 271-296, 1997.

[7] K. Yi and J. Chung, "Nonlinear brake control for vehicle CW/CA systems," *IEEE/ASME Trans. Mechatronics*, Vol. 6, pp. 17-25, 2001.

[8] M. Druzhinina, L. Moklegaard, and A. Stefanopoulou, "Speed gradient approach to longitudinal control of heavy-duty vehicles equipped with variable compression brake," *IEEE Trans. Control System Technologies*, Vol. 10, pp. 209-220, 2002.

[9] J. P. Pauwelussen, L. Gootjes, C. Schroder, K. U. Kohner, S. Jansen, and A. Schmeitz, "Full vehicle ABS braking using the SWIFT rigid ring tyre model," *Control Engineering Practice*, Vol. 11, pp. 199-207, 2003.

[10] I. Kabashkin, "Reliability Model of Intelligent Transport Systems," *Proc. Intelligent Transport Systems*, Sophia Antiplolis, France, pp. 1- 4, Jun, 2006.

[11] J. C. Miles and A. C. Walker, "The potential application of artificial intelligence in transport," *IEE Proc. Intelligent Transport Systems*, vol. 153, no. 3, pp. 183- 198, 2006.

[12] T. Shanley, *The Unabridged Pentium 4: IA32 Processor Genealogy*, B. Colwell (Ed.), Addison Wesley: USA, 2004.

[13] J. Jantzen, *Foundations of Fuzzy Control*, John Wiley & Sons: UK, 2007.

[14] B.-F. Wu, T.-W. Chang, J.-W. Perng, H.-H. Chiang, C.-J. Cheng, T.-Y. Liao, S.-J. Wu, and T.-T. Lee, "Design and Implementation of the Intelligent Stop and Go System in Smart Car, TAIWAN iTS-1," *IEEE Intl. Conf. on Systems, Man and Cybernetics*, Taipei, Taiwan, vol. 3, 8-11. pp. 2075- 2080, 2006.

[15] Y. Liu and Z. Wu, "A smart car control model for driver's comfort of car following," *IEEE Proc. Intelligent Vehicles Symposium*, Las Vegas, USA, pp. 833-839, June, 2005.

[16] E. O. Hwang, *Digital Logic and Microprocessor Design with VHDL*, Thomson-Engineering: USA, 2005.

[17] S. Kilts, *Advanced FPGA Design*, John Wiley & Sons: USA, 2007.