

## 주차 보조 시스템을 위한 ECU 설계

# Design of Electronic Control Unit for Parking Assist System

최진혁\*, 이성수\*

Jin-Hyuk Choi\* and Seongssoo Lee\*

### Abstract

Automotive ECU integrates CPU core, IVN controller, memory interface, sensor interface, I/O interface, and so on. Current automotive ECUs are often developed with proprietary processor architectures. However, demands for standard processors such as ARM and RISC-V increase rapidly for software compatibility in autonomous vehicles and connected cars. In this paper, an automotive ECU is designed for parking assist system based on RISC-V with open instruction set architecture. It includes 32b RISC-V CPU core, IVN controllers such as CAN and LIN, memory interfaces such as ROM and SRAM, and I/O interfaces such as SPI, UART, and I2C. Fabricated in 65nm CMOS technology, its operating frequency, area, and gate count are 50MHz, 0.37mm<sup>2</sup>, and 55,310 gates, respectively.

### 요약

차량에 사용되는 ECU에는 CPU 코어, 차량통신 컨트롤러, 메모리 인터페이스, 센서 인터페이스, I/O 인터페이스 등이 집적되어 있다. 현재 사용되는 차량용 ECU는 대부분 자사만의 독점적 프로세서 아키텍처로 개발하였으나, 최근 자율주행자동차 및 커넥티드카에서 소프트웨어 범용성을 위해 ARM, RISC-V와 같은 표준 프로세서를 기반으로 한 차량용 ECU의 수요가 급증하고 있다. 본 논문에서는 명령어 집합이 무료로 공개된 RISC-V를 기반으로 하여 주차 보조 시스템에 사용하기 위한 차량용 ECU를 설계하였다. 개발된 ECU는 32b RISC-V CPU 코어, CAN, LIN 등의 IVN 컨트롤러, ROM, SRAM 등의 메모리 인터페이스, SPI, UART, I2C 등의 I/O 인터페이스를 내장하였다. 65nm CMOS 공정에서 구현한 결과는 동작 주파수 50MHz, 면적 0.37mm<sup>2</sup>, 게이트 수 55,310개였다.

*Key words* : ECU, CPU, RISC-V, CAN, LIN, SPI, UART, I2C

\* School of Electronic Engineering, Soongsil University  
(Student, Professor)

★ Corresponding author  
E-mail : sslee@ssu.ac.kr, Tel : +82-2-820-0692

※ Acknowledgment

This work was supported by the IT R&D Program (20003771) and Industrial Technology Challenge Track (20012624) of the Ministry of Trade, Industry and Energy (MOTIE) / Korea Evaluation Institute of Industrial Technology (KEIT).

Manuscript received Dec. 08, 2020; revised Dec. 14, 2020; accepted Dec. 14, 2020.

This is an Open-Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

## 1. 서론

차량용 ECU(Electronic Control Unit)는 안전성, 신뢰성 등의 문제로 인해 몇몇 제조사만이 개발하고 있으며 이들은 자사가 보유한 독점적 프로세서 아키텍처에 기반하여 ECU를 설계하기 때문에 소프트웨어도 해당 제조사만을 따라갈 수밖에 없다. 최근 자율주행자동차 및 커넥티드카에서 소프트웨어 개발 및 호환성 문제로 인해 ARM[1], RISC-V[2][3] 등 사실상 표준(De Facto Standard)인 CPU(Central Processing Unit)의 수요가 급격히 늘고 있으며, 이에 따라 많은 반도체 회사들이 자사가 기존에 보유하고 있던 ARM, RISC-V CPU 코어에 각종 주변 IP(Intellectual Property)를 결합하여 차량용 ECU를 개발하기 시작하였다. 특히 RISC-V의 경우 ARM과는

달리 명령어 집합(ISA : instruction set architecture) 이 오픈소스로 공개되어 있을뿐만 아니라 성능 측면에서도 상용화에 충분하도록 높은 수준의 사양을 갖추어서 많은 반도체 회사들이 사용하고 있다.

본 논문에서는 RISC-V CPU 코어를 기반으로 주차 보조 시스템에 사용하기 위한 차량용 ECU를 설계하고 Verilog HDL을 이용하여 구현하였다. 또한 65nm CMOS 공정으로 제작하였다.

## II. 아키텍처 설계

본 논문에서는 32b RISC-V CPU 코어를 기반으로 차량통신 컨트롤러, 메모리 인터페이스, I/O 인터페이스 등을 내장한 차량용 ECU를 그림 1과 같은 아키텍처로 설계하였다.

주차 보조 시스템에 필요한 ECU는 명령어의 종류가 많지 않고 실시간 제어에 적합해야 하기 때문에 CISC(Complex Instruction Set Computer)보다는 RISC(Reduced Instruction Set Computer)가 더 적합하다. 본 논문에서는 RISC 구조를 가지는 CPU 중에서 ARM에 비해 아키텍처 및 명령어 집합이 무료로 공개되어 있고 표 1과 같이 면적, 전력, 성능이 우수한 RISC-V를 사용하였다[4]. RISC-V 코어는 32비트 또는 64비트 중에서 선택이 가능한데, AMBA(Advanced Microcontroller Bus Architecture) 버스로 연결할 다른 IP와의 호환성을 위해 32비트 코어인 RISC-V E20 CPU[5]를 선택하였다.

또한 주차 보조 시스템은 초음파 센서, 온도 센서 등 다양한 센서로부터 데이터를 취득하고 이를 차량 전체에 전달해야 하기 때문에 여기에 사용되는 ECU도 다양한 I/O와 차량통신을 지원해야 한다. 본 논문에서는 LIN(Local Interconnect Network) [6][7], CAN(Controller Area Network) [8]-[10] 등의 차량통신 컨트롤러, ROM, SRAM 등의 메모리 인터페이스, SPI(Serial Peripheral Interface), UART(Universal Asynchronous Receiver/Transmitter), I2C(Inter-Integrated Circuit) 등의 I/O 인터페이스를 내장하였다.

ROM 인터페이스 영역에는 RISC-V CPU의 부팅을 위한 코드가 저장되어 있으며, SRAM 인터페이스에는 ECU가 지원하는 LIN, CAN 등의 통신 프로토콜에 대해 테스트하기 위한 내용들이 저장되어 있다. 버스 브릿지를 이용하여 AMBA AHB

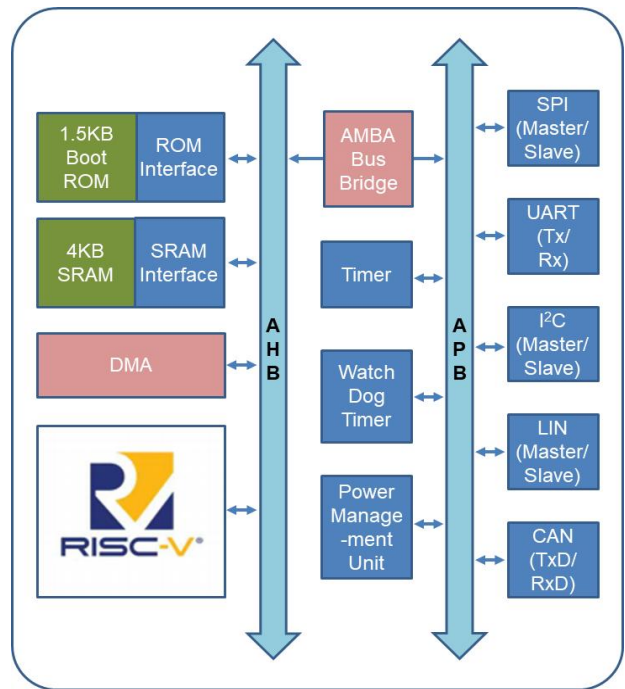


Fig. 1. Architecture of the automotive ECU based on RISC-V in this paper

그림 1. 본 논문에서 구현한 RISC-V 기반 차량용 ECU 아키텍처

(Advanced High-Performance Bus) 버스 및 AMBA APB(Advanced Peripheral Bus) 버스와 연결이 가능하며, 통신 프로토콜 중 LIN은 LIN 2.2A 표준을, CAN은 CAN 2.0A와 CAN 2.0B를 모두 만족하도록 설계하였다.

Table 1. Performance comparison between RISC-V and ARM[4].

표 1. RISC-V와 ARM의 성능 비교[4]

Parameter	RISC-V Rocket	ARM Cortex-A5
Instruction set	64-bit RISC-V v2	32-bit ARM v7
Architecture	Single-Issue In-Order	Single-Issue In-Order
Performance	1.72 DMIPS/MHz	1.57 DMIPS/MHz
Fabrication	TSMC 40GPLUS	TSMC 40GPLUS
Area(w/o Cache)	0.14 mm <sup>2</sup>	0.27 mm <sup>2</sup>
Area(w/ 16K Cache)	0.39 mm <sup>2</sup>	0.53 mm <sup>2</sup>
Performance per area	4.41 DMIPS/MHz/mm <sup>2</sup>	2.96 DMIPS/MHz/mm <sup>2</sup>
Clock	> 1GHz	> 1GHz
Power consumption	0.034mW/MHz	< 0.08mW/MHz

## III. 시뮬레이션 및 검증

설계된 ECU를 검증하기 위해서 CAN 통신 프로토콜을 이용하여 2개의 테스트 시나리오를 구성하

였다. 첫 번째 테스트 시나리오는 그림 2와 같이 버스 상에 3개의 CAN 노드 A, B, C가 연결되어 있을 때 ①과 같이 RISC-V의 SRAM 영역에 저장한 4개의 32비트 데이터를 CAN 노드 A가 노드 B로 전송하고 ②와 같이 전송을 마친 후 일정 시간을 대기하였다가 ③과 같이 다시 노드 A가 노드 C에게 다른 8개의 32비트 데이터를 전송하는 시나리오이다. 두 번째 테스트 시나리오는 그림 3과 같이 4개의 CAN 노드가 버스에 연결되어 노드 C가 CAN ID 0x63을 받았을 때 ①과 같이 노드 C가 가지고 있던 8개의 32비트 데이터 값을 송신하고 ②와 같이 노드 A와 노드 D가 수신받는 시나리오이다.

본 논문에서 설계한 차량용 ECU는 RISC-V를 기반으로 Verilog HDL을 이용하여 구현하였으며, IDEC (IC Design Education Center)에서 제공한 ModelSim을 이용하여 시뮬레이션을 진행하였다. 그림 4와 그

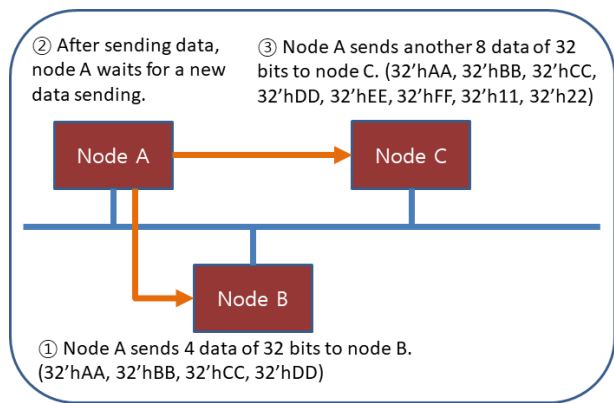


Fig. 2. Test scenario 1: Data transmission between 3 CAN nodes on CAN bus.

그림 2. 테스트 시나리오 1: CAN 버스에 연결된 3개의 CAN 노드끼리 데이터를 전송하는 과정

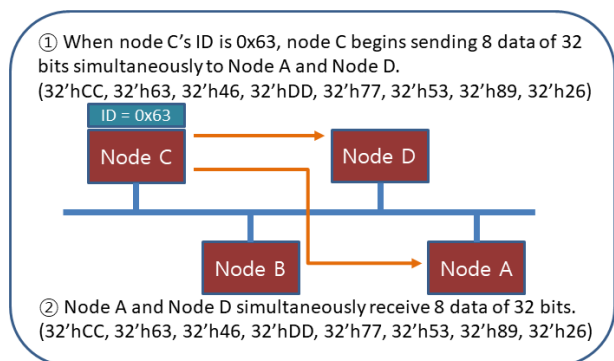


Fig. 3. Test scenario 2: Data transmission between 4 CAN nodes on CAN bus.

그림 3. 테스트 시나리오 2: CAN 버스에 연결된 4개의 CAN 노드끼리 데이터를 전송하는 과정

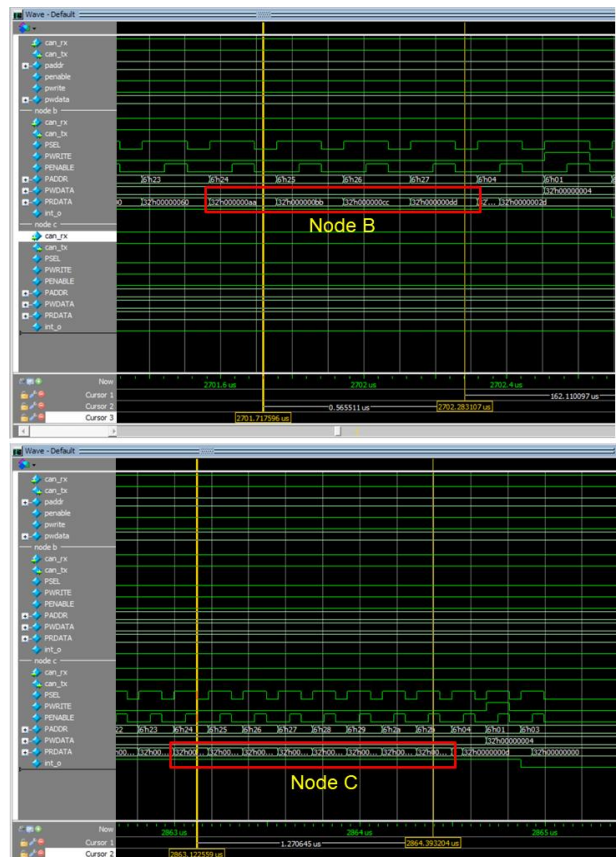


Fig. 4. Simulation results of test scenario 1.

그림 4. 테스트 시나리오 1의 시뮬레이션 결과

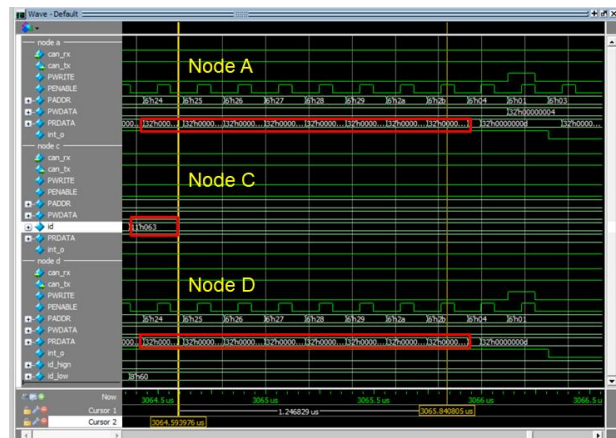


Fig. 5. Simulation results of test scenario 2.

그림 5. 테스트 시나리오 2의 시뮬레이션 결과

림 5는 각각 그림 2와 그림 3의 테스트 시나리오에 대한 시뮬레이션 결과이며, 테스트 시나리오대로 전송이 이루어지는 것을 확인할 수 있다.

#### IV. ASIC 칩 구현

본 논문에서 설계한 주차 보조 시스템용 차량용

ECU를 65nm CMOS 공정을 사용하여 검증하였다. Front-End Design 단계에서는 주로 칩의 동작을 위한 타이밍 부분을 중점적으로 검증하였다. 그림 6은 차량용 ECU 칩의 타이밍 검증 결과이며, IDEC에서 제공한 Primitime을 이용하여 50MHz 기준으로 20ns가 넘지 않도록 타이밍을 검증하였다. Back-End Desing 과정에서는 공정사로부터 제공받은 Standard Cell과 I/O Cell을 배치하는 과정으로, IDEC에서 제공한 IC Compiler 툴을 사용하여 파워 및 그라운드 라인을 배치하고 DRC (Design Rule Check) 및 LVS(Layout Versus Schematic) 만족 여부를 검증하였다. 그림 7은 설계한 차량용 ECU 칩의 최종 레이아웃으로, 동작 주파수, 칩 면적, 게이트 수는 각각 50MHz, 0.37mm<sup>2</sup>, 55,310 게이트였다.

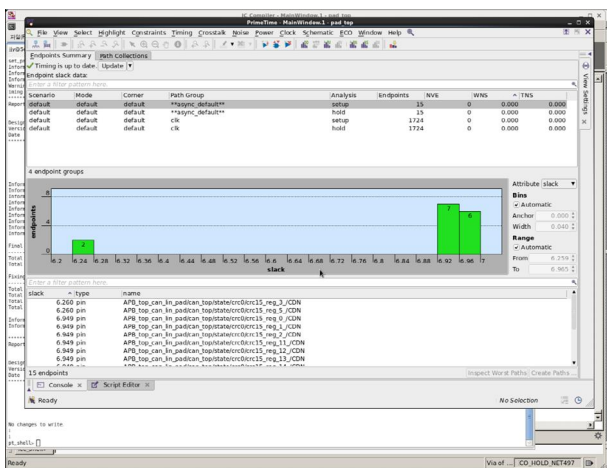


Fig. 6. Timing verification results of the ECU chip.  
그림 6. ECU 칩의 타이밍 검증 결과

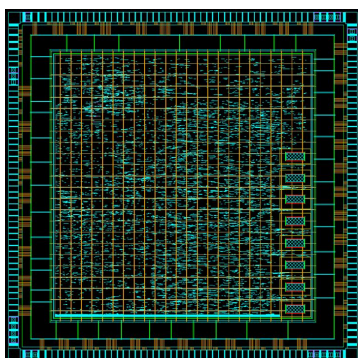


Fig. 7. Layout of the ECU chip.  
그림 7. ECU 칩의 레이아웃

## V. 결론

본 논문에서는 RISC-V를 기반으로 한 차량용 ECU

를 설계하였다. 개발된 ECU는 32b RISC-V CPU 코어, CAN, LIN 등의 차량통신 컨트롤러, ROM, SRAM 등의 메모리 인터페이스, SPI, UART, I2C 등의 I/O 인터페이스를 내장하였다. 65nm CMOS 공정에서 구현한 결과는 동작 주파수 50MHz, 면적 0.37mm<sup>2</sup>, 게이트 수 55,310개였다.

## References

- [1] ARM, "ARM Solutions Automotive," <https://www.arm.com/solutions/automotive>
- [2] RISC-V Foundation, "Vol.1, Unprivileged Spec v.20191213," <https://riscv.org/technical/specifications>
- [3] RISC-V Foundation, "Volume 2, Privileged Spec v.20190608," <https://riscv.org/technical/specifications>
- [4] Y. Lee et al, "Raven: A 28nm RISC-V vector processor with integrated switched-capacitor DC-DC converters and adaptive clocking," *Proceedings of IEEE Hot Chips Symposium*, pp.1-45, 2015. DOI: 10.1109/HOTCHIPS.2015.7477469
- [5] Sifive, "E20 Core Complex Manual," <https://www.sifive.com/documentation>
- [6] ISO 17987-1:2016, "Road vehicles-Local Interconnect Network (LIN)-Part 1: General information and use case definition," [https://www.iso.org/iso/home/store/catalogue\\_tc/catalogue\\_detail.htm?csnnumber=61222](https://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnnumber=61222)
- [7] J. Lee and S. Lee "Design and Verification of Automotive LIN Controller," *J.inst.Korean. electr. elctron.eng.*, vol.20, no.3, pp.333-336, 2016. DOI: 10.7471/ikeee.2016.20.3.333
- [8] ISO 11898-1:2015, "Road Vehicles-Controller Area Network (CAN)-Part 1: Data Link Layer and Physical Signalling," <https://www.iso.org/standard/63648.html>
- [9] J. Lee and S. Lee "Design and Verification of Automotive CAN Controller," *J.inst.Korean. electr. elctron.eng.*, vol.21, no.2, pp.162-165, 2017. DOI: 10.7471/ikeee.2017.21.2.162
- [10] J. Lee and S. Lee "Implementation and Verification of Automotive CAN-FD Controller," *J.inst.Korean.electr.elctron.eng.*, vol.21, no.3, pp. 240-243, 2017. DOI: 10.7471/ikeee.2017.21.3.240