

LIN/CAN 차량용 인터페이스와 칼만 필터 기능을 통합한 차량용 ECU 설계

Vehicle ECU Design Incorporating LIN/CAN Vehicle Interface with Kalman Filter Function

정 선 우*, 김 용 빈*, 이 성 수*[★]

Seonwoo Jeong*, Yongbin Kim*, and Seongsoo Lee*[★]

Abstract

In this paper, an automotive ECU (electronic control unit) with Kalman filter accelerator is designed and implemented. RISC-V is exploited as a processor core. Accelerator for Kalman filter matrix operation, CAN (controller area network) controller for in-vehicle network, and LIN (local interconnect network) controller are designed and embedded. Kalman filter operation consists of time update process and measurement update process. Current state variable and its error covariance are estimated in time update process. Final values are corrected from input measurement data and Kalman gain in measurement update process. Usually floating-point multiplication is exploited in software implementation, but fixed-point multiplier considering accuracy analysis is exploited in this paper to reduce hardware area. In 28nm silicon fabrication, its operating frequency, area, and gate counts are 100MHz, 0.37mm², and 760k gates, respectively.

요 약

본 논문에서는 자동차의 위치 및 자세 추정에 사용되는 칼만 필터 가속기를 내장한 차량용 ECU(electronic control unit)를 설계하고 구현하였다. 프로세서 코어는 RISC-V를 사용하였으며 칼만 필터의 행렬 연산을 수행하는 가속기, 차량 내 통신에 사용되는 CAN(controller area network) 제어기, 센서 연결에 사용되는 LIN(local interconnect network) 제어기를 내장하였다. 칼만 필터 연산은 시간 업데이트와 측정 업데이트의 두 단계로 나뉘며 시간 업데이트 단계에서는 현재 상태변수와 오차 공분산을 예측하고 측정 업데이트 단계에서는 입력값을 받아 칼만 이득을 계산하여 값을 보정한다. 보통 소프트웨어에서는 곱셈에 부동소수점 연산을 사용하지만 본 논문에서는 하드웨어 면적을 줄이기 위해 정밀도 분석을 고려한 고정소수점 곱셈기를 사용하였다. 설계된 ECU는 Verilog HDL을 이용하여 검증하였으며 28nm 실리콘 공정으로 구현하였다. 28nm 실리콘 공정으로 구현하였을 때 동작 주파수는 100MHz, 면적은 0.37mm², 게이트 수는 76만 게이트였다.

Key words : ECU, Kalman Filter, Location Estimation, Pose Estimation, LIN, CAN

* Soongsil University(Researcher, Researcher, Professor)

★ Corresponding author

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

※ Acknowledgment

This work was supported by the R&D Program of the Ministry of Trade, Industry, and Energy (MOTIE) (20008417) and Korea Evaluation Institute of Industrial Technology (KEIT). This work was also supported by the MOTIE (Ministry of Trade, Industry & Energy) (10080649) and KSRC (Korea Semiconductor Research Consortium) support program for the development of the future semiconductor device.

Manuscript received Sep. 25, 2021; revised Oct. 27, 2021; accepted Nov. 1, 2021.

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.

I. 서론

칼만 필터(Kalman filter)[1]는 제어 및 신호 처리 분야에서 가장 널리 적용되고 있는 기술 중 하나로 잡음이 포함된 측정 신호로부터 현재의 위치 상태를 최적으로 추정해준다. 칼만 필터는 각 단계에서 반복적인 측정 업데이트와 시간 업데이트를 한다. 이 단계에서 행렬 연산들이 많아 하드웨어 단에서 병렬처리를 하면 소프트웨어로 연산하는 것보다 효과적으로 시간을 단축할 수 있으며 전력 소모도 줄일 수 있다.

본 논문에서는 자동차의 위치 및 자세 추정에 사용되는 칼만 필터를 내장한 차량용 ECU(electronic control unit)를 설계하였다. 프로세서 코어는 RISC-V[2]를 사용하였으며 칼만 필터의 행렬 연산을 수행하는 가속기 IP를 내장하였다. 또한 차량 내 통신에 사용되는 CAN 제어기[3] IP와 LIN 제어기[4] IP도 내장하였다. 설계된 ECU는 Verilog HDL을 이용하여 검증하였으며 28nm 실리콘 공정으로 구현하였다.

II. 아키텍처 설계

본 논문에서 설계한 ECU의 아키텍처는 그림 1과 같다. 프로세서 코어 IP로는 Sifive의 E21 코어[5]를 사용하였으며 AMBA(advanced microcontroller bus architecture) AHB 버스를 통해 SRAM, 플래시 메모리 등과 통신한다. AMBA APB 버스에는

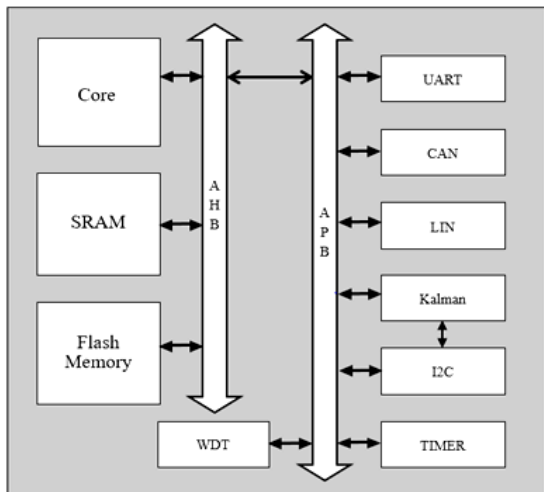


Fig. 1. ECU architecture.
그림 1. ECU 아키텍처

칼만 필터 가속기 IP, CAN 제어기 IP[3], LIN 제어기 IP[4], 기타 주변 IP가 연결되었으며 AHB-APB 브릿지를 통해 두 버스를 연결하였다.

칼만 필터 가속기는 많은 곱셈이 포함되어 있는 칼만 필터의 행렬 연산 부분을 수행한다. CAN 통신은 차량 내부의 다른 ECU와 데이터 통신을 수행하며 LIN 통신은 차량 내에 장착된 센서와 데이터 통신을 수행한다[6].

SRAM과 플래시 메모리는 메모리 컴파일러를 이용하여 16k×32비트의 크기로 설계하였다. SRAM은 코어의 주 메모리로 사용되며 플래시 메모리에는 RISC-V 코어의 부팅을 위한 코드가 저장되어 있다.

CAN 제어기는 차량 내에 산재되어 있는 여러 ECU 간의 통신을 위해 사용된다. CAN 통신은 멀티마스터 멀티슬레이브 구조로 버스 중재 기능이 있으며 본 논문의 CAN 제어기는 CAN 2.0A와 CAN 2.0B 모두를 만족한다.

LIN 제어기는 ECU와 센서, 액추에이터 사이의 통신에 사용되고 CAN 버스의 하위 버스(sub-bus)를 제공해준다. LIN 통신은 싱글마스터 멀티슬레이브의 구조로 본 논문의 LIN 제어기는 LIN 2.2A 표준을 만족하고 버스 전체를 제어하는마스터 역할을 수행한다.

칼만 필터 가속기는 관성 센서인 IMU(inertial measurement unit) 센서 등과 연결하기 위한 I2C (inter-integrated circuit) 버스 인터페이스를 가지고 있으며 코어와 연산 값을 주고받기 위한 AMBA APB 버스 인터페이스를 가지고 있다.

III. 칼만 필터 설계

칼만 필터는 선형 시스템에서 최적화된 제어기법으로 과거의 정보와 새로운 측정값을 사용하여 측정값에 포함된 노이즈를 제거해서 정확한 값을 추정하는 데 사용하는 알고리즘이다.

칼만 필터 연산은 그림 2와 같이 시간 업데이트와 측정 업데이트 단계로 나뉜다. 시간 업데이트 단계에서는 현재 상태변수(state variable)를 예측하고 오차 공분산(error covariance)을 예측한다. 측정 업데이트에서는 입력값을 받아서 칼만 이득(Kalman gain)을 계산해서 값을 보정한다.

최종 결과는 오일러 각(Euler angle)을 이용해서 위치 및 자세를 추정하지만, 오일러 각은 짐벌락

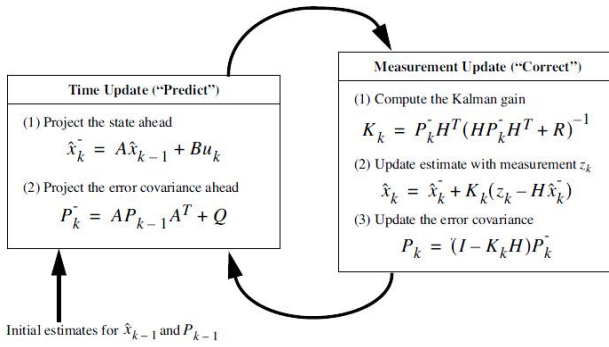


Fig. 2. Kalman filter algorithm.
그림 2. 칼만 필터 알고리즘

$$\phi(roll) = \arctan\left[\frac{c_{32}}{c_{33}}\right] = \arctan\left[\frac{2(q_2q_3 + q_0q_1)}{1 - 2(q_1^2 + q_2^2)}\right]$$

$$\theta(pitch) = \arcsin[-c_{31}] = \arcsin[-2(q_1q_3 - q_0q_2)]$$

$$\psi(yaw) = \arctan\left[\frac{c_{21}}{c_{11}}\right] = \arctan\left[\frac{2(q_1q_2 + q_0q_3)}{1 - 2(q_2^2 + q_3^2)}\right]$$

Fig. 3. Quaternion-Euler angle conversion.
그림 3. 쿼터니언-오일러각 변환

(gimbal lock) 문제가 있어서 실제 연산에서는 쿼터니언(quaternion) 기반으로 연산한다. 쿼터니언은 4개의 성분, 즉 벡터값 3개와 스칼라값 1개로 이루어진다.

그림 3은 쿼터니언 값을 오일러 각으로 변환하는 수식이다. 쿼터니언을 계산하기 위해 소프트웨어 연산에서는 부동소수점 곱셈을 사용하지만 본 논문에서는 하드웨어 면적을 줄이기 위해 고정소수점 곱셈기를 사용하였다. 이때 소수점이 고정되면 유효숫자 때문에 오차가 발생할 우려가 있으므로 반드시 정밀도 분석을 수행해야 한다.

IV. 시뮬레이션 및 검증

본 논문에서는 칼만 필터 알고리즘을 RTL로 구현하여 ECU에 통합되는지를 검증하였다. 정확한 값으로 연산이 수행되는지 알기 위해 먼저 IMU 센서로부터 받은 값을 오실로스코프로 확인했다. 이후 그림 4와 같이 I2C 버스를 이용하여 칼만 필터 가속기의 입력값인 Accel, Gyro, Mag 센서들의 X, Y, Z 값을 추출했다.

그림 5는 Python으로 칼만 필터 알고리즘을 모델링한 코드를 동작시켰을 때 나오는 쿼터니언 값을 기반으로 3D 프로세싱을 한 결과이다. 초기 쿼

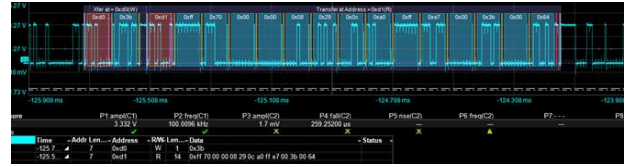


Fig. 4. IMU sensor data.
그림 4. IMU 센서 데이터

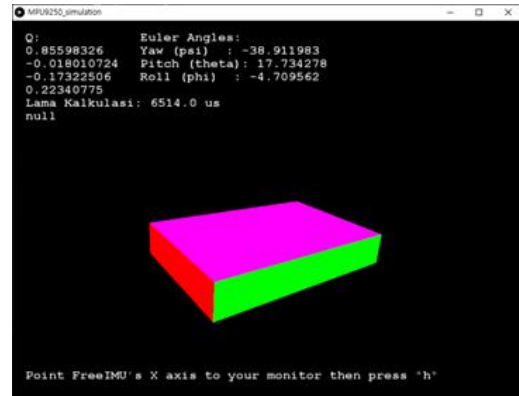


Fig. 5. Results of 3D processing.
그림 5. 3D 프로세싱 결과

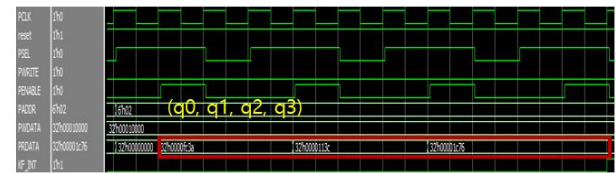


Fig. 6. Results of Kalman filter calculation.
그림 6. 칼만 필터 연산 결과값

터니언 값이(1,0,0,0) 일 때가 정지 상태의 모습이고 이 모델링을 기반으로 Verilog HDL 시뮬레이션을 진행하였다.

그림 6은 초기값으로 9개의 입력에 0001000 고정소수점 값을 넣어주었을 때 연산 결과를 버스로 보내주는 결과 파형이다. 4개의 쿼터니언 값이 나오는데 q0의 크기는 1에 가깝고 정지상태의 초기값이므로 나머지 3개의 값은 0에 근사한 값이 나왔고 모델링된 결과값과 근사한 것을 확인하였다. 입력값이 들어가고 결과값이 나올 때까지 15 us가 걸린다.

V. ASIC 칩 구현

본 논문에서 설계한 차량용 ECU를 IDEC에서 지원한 설계 툴을 사용하여 28nm RFCMOS 공정에서 설계 및 검증을 진행하였다. 프론트엔드 설계 단계에서는 Design Compiler, Prime Time Tool을

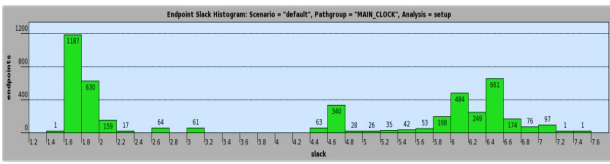


Fig. 7. Results of timing verification.
그림 7. 타이밍 검증 결과

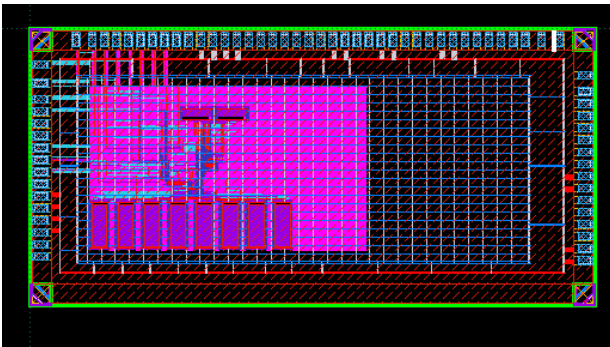


Fig. 8. ECU layout.
그림 8. ECU 레이아웃

Type of Check	Total	Met	Violated
setup	8805	7963 (90%)	0 (0%)
hold	8805	7963 (90%)	0 (0%)
recovery	2570	2560 (100%)	0 (0%)
removal	2570	2560 (100%)	0 (0%)
min_period	320	10 (3%)	0 (0%)
min_pulse_width	41188	30908 (75%)	0 (0%)
clock_gating_setup	7	4 (57%)	0 (0%)
clock_gating_hold	7	4 (57%)	0 (0%)
out_setup	10	10 (100%)	0 (0%)
out_hold	10	10 (100%)	0 (0%)
All Checks	64292	51992 (81%)	0 (0%)

Fig. 9. Timing coverage report.
그림 9. 타이밍 커버리지 리포트

사용하여 합성 및 타이밍 검증을 수행하고 합성 후 기능에 필요한 셀들이 최적화되어 사라지지 않았는지 Formality Tool로 Equivalence를 체크하였다.

그림 7은 설계된 칩의 타이밍을 Prime Time으로 검증한 결과이며 100MHz에서 동작할 수 있다는 것을 보여주고 있다.

백엔드 설계 과정에서는 셀 라이브러리를 이용하여 IC Compiler II를 사용하여 설계했고 완성된 ECU의 칩 레이아웃은 그림 8과 같다.

그림 9는 PNR을 완료한 후 Post STA의 결과이며 모든 타이밍을 만족한 것을 알 수 있다. 설계된 ECU의 총 면적은 376,785.06um²이고 라이브러리의 2NAND 셀 면적이 0.5um²이므로 대략 76만 게이트가 사용된 것으로 추정된다.

VI. 결론

본 논문에서는 Sifive E21 RISC-V 코어를 기반으로 하여 칼만 필터 가속기를 내장한 차량용 ECU를 설계하였다. 설계한 ECU는 32비트 RISC-V 코어, LIN, CAN 등의 차량 통신 제어기, SRAM, 플래시 메모리 등의 메모리 및 인터페이스, SPI, UART, I2C 등의 I/O 인터페이스를 내장하였다. 28nm RFCMOS 공정에서 구현한 결과 동작 주파수 100MHz, 면적 0.37mm², 게이트 수 약 76만 게이트로 나타났다.

References

[1] R. Kalman, "A New Approach to Linear Filtering and Prediction Problems," Journal of Basic Engineering, vol.82D, pp.34-45, 1960.

[2] RISC-V International, <https://riscv.org>

[3] J. Lee and S. Lee "Design and Verification of Automotive CAN Controller," J.inst.Korean.electr.electron.eng., vol.21, no.2, pp.162-165, 2017.

[4] J. Lee and S. Lee "Design and Verification of Automotive LIN Controller," J.inst.Korean.electr.electron.eng., vol.20, no.3, pp.333-336, 2016.

[5] Sifive, "E20 Core Complex Manual," <https://www.sifive.com/documentation>

[6] J. Choi and S. Lee "Design of Electronic Control Unit for Parking Assist System," J.inst.Korean.electr.electron.eng., vol.24, no.4, pp.1172-1175, 2020.