

차량용 온칩 버스의 데이터 무결성을 위한 종단간 에러 정정 코드(e2eECC)의 설계 및 구현

Design and Implementation of e2eECC for Automotive On-Chip Bus Data Integrity

길은배*, 박찬*, 김주호*, 정준호*, 이주석*, 이성수**

Eunbae Gil*, Chan Park*, Juho Kim*, Joonho Chung*, Joosock Lee*, and Seongsoo Lee**

Abstract

AMBA AHB-Lite bus is widely used in on-chip bus protocol for low-power and cost-effective SoC. However, it lacks built-in error detection and correction for end-to-end data integrity. This can lead to data corruption and system instability, particularly in harsh environments like automotive applications. To mitigate this problem, this paper proposes the application of SEC-DED (Single Error Correction-Double Error Detection) to AMBA AHB-Lite bus. It aims not only to detect errors in real-time but also to correct them, thereby enhancing end-to-end data integrity. Simulation results demonstrate real-time error detection and correction when errors occur, which bolsters end-to-end data integrity of automotive on-chip bus.

요약

AMBA AHB-Lite 버스는 저전력 및 경제성 측면에서 SoC에 널리 사용되는 온칩 버스 프로토콜이다. 하지만 이 프로토콜은 종단간 데이터 무결성을 위한 에러 검출 및 정정이 불가능하다. 이로 인해 자동차와 같이 열악한 환경에서 동작하는 경우에 데이터 변질과 시스템 불안정을 일으킬 수 있다. 이러한 문제를 해결하기 위해 본 논문에서는 AMBA AHB-Lite 버스에 SEC-DED(Single Error Correction-Double Error Detection)를 적용하는 방법을 제안한다. 이는 전송 중 발생하는 데이터 에러를 실시간으로 감지하고 정정하여 종단간 데이터 무결성을 강화한다. 시뮬레이션 결과, 에러가 일어나도 실시간으로 이를 감지하고 정정하여 차량용 온칩 버스에서 종단간 데이터 무결성을 강화하는 것을 확인하였다.

Key words : Error Correction Code, SEC-DED, End-to-End Data Integrity, Bus Architecture Design, AMBA AHB-Lite

* School of Electronic Engineering and Department of Intelligent Semiconductor, Soongsil University (Student, Student, Student, Student, Professor, 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) and Korea Evaluation Institute of Industrial Technology (KEIT). (RS-2022-00154973, RS-2023-00232192). It was also supported by MOTIE and Korea Institute for Advancement of Technology (KIAT) (P0012451). The authors wish to thank IC Design Education Center (IDEC) for CAD support.

Manuscript received Mar. 5, 2024; revised Mar. 21, 2024; accepted Mar. 22, 2024.

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

자율주행은 자동차 산업에 혁명을 일으켜 새로운 패러다임을 제시하고 지속 가능하고 환경친화적인 대안으로 떠올랐다. 자율주행 기술이 급성장하면서 자동차의 디지털화 또한 가속화되고 있다. 이에 따라 자동차를 제어하는 ECU(Electronic Control Unit)의 역할이 중요해지고 있으며 ECU는 차량의 동작을 제어하는 핵심적인 역할을 수행하므로 차량의 안정성에 크게 이바지한다. 이에 따라 ECU를 구성하는 반도체에서 데이터가 전송 도중 변질되지 않고 정상적으로 전달되는 데이터 무결성(Data Integrity)이 중요해지기 시작했다.

소프트 에러(Soft Error)는 칩 내부에서 발생하는 일시적인 에러로, 외부에서 들어오는 방사선 등에 의해 순간적으로 발생하며 감지가 곤란하고 재현이 되지 않으며 대처가 어렵기 때문에 자율주행차로의 발전에 있어서 해당 문제는 반드시 해결되어야 한다[1][2].

ECU에 사용되는 AMBA 버스는 저전력과 확장성 측면에 있어서 장점을 가지고 있지만, 내장된 EDAC(Error Detection And Correction) 메커니즘이 부족하다. 이를 보완하는 자동차의 표준 소프트웨어 플랫폼인 AUTOSAR(AUTomotive Open System ARchitecture)에서는 임베디드 네트워크를 통한 ECU 간의 부정확한 데이터 통신을 방지하기 위해 CRC(Cyclic Redundancy Check) 코드를 사용하는 e2e(End-to-End) 라이브러리를 제공한다[3]. 하지만 CRC는 단순히 에러가 일어났음을 감지할 수 있을 뿐 수정까지 가능하지는 않다. 또한 이처럼 소프트웨어적으로 처리하는 것보다는 임베디드 네트워크 내에서 하드웨어적으로 처리하는 것이 속도 면에서 보다 효율적이다.

본 논문에서는 메모리에서 사용하는 ECC(Error Correction Code)인 SEC-DED(Single Error Correction-Double Error Detection)를 AMBA AHB-Lite 온칩 버스에 적용하여 즉각적인 에러 수정이 가능한 방안을 제안하고 이를 Verilog HDL로 구현하였으며 IDEC에서 제공한 설계 툴을 이용하여 시뮬레이션으로 검증하였다.

II. AMBA 버스와 SEC-DEC 코드

AMBA(Advanced Microcontroller Bus Architecture) AHB-Lite는 ARM 사가 개발한 시스템 칩 설계에서 사용하는 버스 아키텍처이며[4] 마이크로컨트롤러 장치를 넘어 ASIC과 SoC 분야에서 광범위하게 사용되고 있다.

AMBA 버스는 저전력과 확장성이 용이하게 설계되어 ECU와 같은 마이크로컨트롤러 설계에 적합하다. AHB-Lite는 그림 1과 같은 구조로 단일 Master 구조를 가진 버스 아키텍처이다.

AHB Master는 Address Phase, Data Phase 두 가지의 Phase를 통해 데이터를 주고받는다. Address Phase에서는 그림 1의 Address and Control에 표시된 신호들을 사용하고, Data Phase에서는 R/W(Read/Write) 동작에 따라 HRDATA 또는 HWDATA를 사용한다. 그림 1에서와 같이 AHB-Lite Master 신호들을 사용하여 R/W 동작을 결정하고, Data Phase에서 해당 동작에 맞는 데이터를 R/W 한다.

그림 2에서의 Decoder는 HADDR을 통해 HSEL을 제어한다. 각각의 HSEL은 할당된 Slave와 연결되고 이를 통해 HWDATA의 Slave에 대한 Write 여부를 결정한다. 또한 Multiplexer select를 통해 HADDR에 따른 HRDATA를 Multiplexer에서 선택하여 Master의 입력으로 보낸다.

에러 감지와 정정에 사용할 SEC-DED는 Hamming Code에 기반한 ECC로 1비트 에러 정정과 2비트 에러 검출을 할 수 있는 에러 정정 코드이다. SEC-DED 코드에 대해 알아보려면, Hamming Code에 대해 알아봐야 한다.

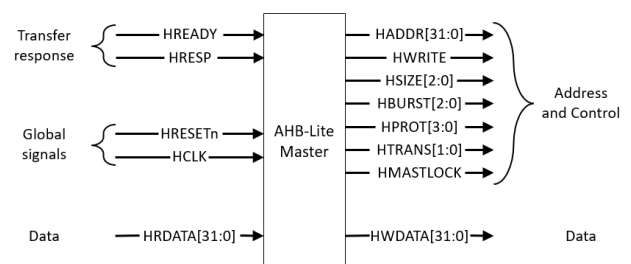


Fig. 1. Signals of AHB-Lite bus master.

그림 1. AHB-Lite 버스 마스터 신호

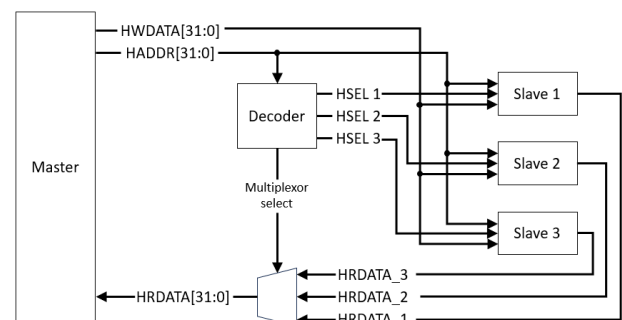


Fig. 2. AHB-Lite bus block diagram.

그림 2. AHB-Lite 버스 블록 다이어그램

Bit position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
Encoded data bits	p1	p2	d1	p3	d2	d3	d4	p4	d5	d6	d7	d8	d9	d10	d11	p5	d12	d13	d14	d15	
Parity Bit Coverage	p1	0	0		0		0		0		0		0		0		0		0		
	p2		0	0			0	0			0	0			0	0			0	0	
	p3				0	0	0	0					0	0	0	0					0
	p4								0	0	0	0	0	0	0	0					
	p5															0	0	0	0	0	

Fig. 3. Hamming code table.

그림 3. 해밍 코드 관계표

Hamming Code는 리처드 해밍이 고안해 낸 알고리즘으로 선형 부호 이론의 기반이 되었다[5]. 해당 코드는 다음과 같은 방식으로 구성된다. 전송하고자 하는 데이터 비트가 k 개, 최종적으로 전송될 부호화된 비트가 n 개라고 가정할 때, k 개의 데이터 비트로부터 r 개의 패리티 비트가 생성되며 식 (1), (2)를 만족한다[6][7]. 그림 3을 통해 데이터 비트와 패리티 비트의 관계성을 파악할 수 있다.

$$n = k + r \tag{1}$$

$$2^r \geq k + r + 1 \tag{2}$$

예를 들어 데이터 비트(k)가 3비트, 패리티 비트(r)가 3인 (6,3) 해밍 코드가 있으면 패리티 비트(p)는 식 (3)-(5)와 같이 XOR 연산을 통해 정해진다. 이때 최종적으로 부호화된 데이터(n)는 식 (6)과 같이 정해진다.

$$p_1 = k_1 \oplus k_2 \tag{3}$$

$$p_2 = k_1 \oplus k_3 \tag{4}$$

$$p_3 = k_2 \oplus k_3 \tag{5}$$

$$n = [p_1, p_2, k_1, k_2, p_3, k_3] \tag{6}$$

해당 코드는 그림 3을 통한 복호화를 거쳐 원래 데이터로 정정된다. 해당 패리티 비트를 연산할 때 사용된 데이터 비트와 그 패리티 비트를 다시 XOR 연산을 하여 에러가 일어났는지 점검한다. 이 연산의 결과를 신드롬(Syndrome)이라고 하며, 에러가 일어난 위치를 찾는 데 사용된다. (6,3) 해밍 코드를 예시로 들면, 신드롬(s)는 식 (7)-(9)를 통해 생성된다.

$$s_1 = k_1 \oplus k_2 \oplus p_1 \tag{7}$$

$$s_2 = k_1 \oplus k_3 \oplus p_2 \tag{8}$$

$$s_3 = k_2 \oplus k_3 \oplus p_3 \tag{9}$$

에러가 발생하면 해당 신드롬은 1을 값으로 가진다. 신드롬을 통해 2진수 값을 얻을 수 있고, 이를 통해 에러가 발생한 비트의 위치를 파악할 수 있다. 예를 들어, $s_3=0, s_2=0, s_1=1$ 인 경우, 첫 번째 비트에 에러가 일어남을 의미한다.

SEC-DED 또한 Hamming Code와 부호화와 복호화 과정은 대부분 동일하다. 다만 SEC-DED는 Hamming Code의 부호화된 데이터 전체에 대한 XOR 연산을 통해 1비트의 패리티 비트를 추가로 만든다는 차이가 존재한다. 추가적인 1비트를 통해 이중 비트 에러를 감지할 수 있다.

III. AMBA AHB-Lite 온칩 버스의 데이터 무결성을 위한 SEC-DED 코드의 적용

제안하는 방법에서는 그림 1에서 Address Phase에서 사용하는 신호들과 Data Phase에서 사용하는 신호들을 분리하여 부호화한다. 또한 확장성을 위해 1바이트씩, 즉 8비트 단위로 데이터를 부호화한다. 따라서 (12,8)의 Hamming Code를 사용하고 이에 1비트를 추가한 (13,8) SEC-DED를 통해 부호화한다. 실제 AMBA AHB-Lite에서 사용하는 신호들은 8비트보다 많지만 이를 8비트 단위로 끊어서 부호화한다면 소모 전력과 지연 시간을 줄일 수 있다[8].

그림 4에서처럼 Master가 동작하기 위해 버스로 내보내는 신호들은 Encoder를 통해 부호화가 진행된다. Master가 Read Operation을 진행하기 위해 읽어드린 HRDATA는 그림 5에서 Slave의 Encoder를 통해 부호화된 HRDATA이다.

부호화된 데이터는 그림 6의 제안하는 SEC-DED 디코더를 통해 복호화된다. Decoder의 Syndrome Generator는 원본 데이터인 Input Data Bits와 Parity Bits를 통해

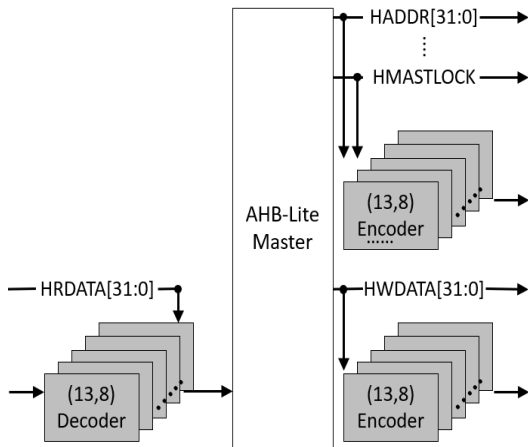


Fig. 4. AHB-Lite bus master with SEC-DED.
그림 4. SEC-DED가 추가된 AHB-Lite 버스 마스터

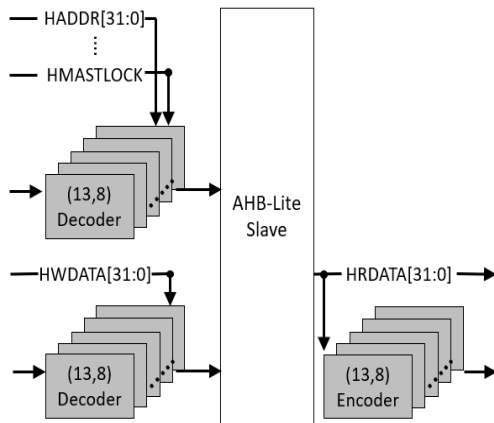


Fig. 5. AHB-Lite bus slave with SEC-DED.
그림 5. SEC-DED가 추가된 AHB-Lite 버스 슬레이브

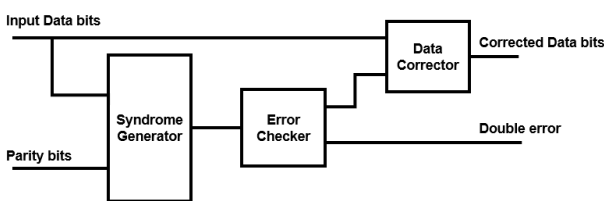


Fig. 6. SEC-DED decoder.
그림 6. SEC-DED 디코더

신드롬을 만들어 낸다. 해당 신드롬을 통해 Error Checker에서 단일 및 이중 비트 에러가 일어났는지 판단한다. 또한 에러가 일어난 비트의 위치를 파악하고 Data Corrector에서 이를 반영하여 Corrected Data bits로 정정하여 원본 데이터를 복구한다.

IV. 시뮬레이션 및 검증

HADDR, HWDATA, HRDATA에서 1비트 에러가 발

생한 경우와 HWDATA에 2비트 에러가 있는 경우의 시뮬레이션 결과는 다음과 같다.

1. HADDR에서 1비트 에러 발생

그림 7은 HADDR에서 1비트 에러가 발생한 경우의 에러 감지 및 정정을 보여주고 있다. 그림 7의 A는 HADDR_err가 0으로 에러가 일어나지 않은 상태이다. Master가 버스로 내보내는 HADDR_i, Slave의 Decoder가 받는 HADDR_toSLV, Slave의 Decoder에서 최종적으로 Slave가 입력으로 받는 신호인 DCD_HADDR, 모두 0x0000fa90으로 동일한 것을 확인할 수 있다. 그림 7의 B에서는 HADDR_err가 1로 에러가 일어나 HADDR_i와 HADDR_toSLV가 서로 다를 수 있다. 그러나 Slave의 Decoder의 입력인 HADDR_toSLV가 에러가 발생한 입력을 받았음에도 불구하고 Decoder의 출력인 DCD_HADDR은 에러가 일어난 경우에도 정상 데이터인 HADDR_i와 값과 동일하여 에러가 정정되었음을 확인할 수 있다.

2. HWDATA에서 1비트 에러 발생

그림 8은 HWDATA에서 1비트 에러가 발생한 경우의 에러 감지 및 정정을 보여주고 있다. 그림 8의 A는 HWDATA_err가 0으로 에러가 일어나지 않은 상태이다. 원본 데이터인 HWDATA_i과 Decoder를 통해 복호화되어 최종적으로 Slave가 입력으로 받는 신호인 DCD_HWDATA이 서로 동일함을 확인할 수 있다.

하지만 HWDATA_err가 1인 그림 8의 B에서는 HWDATA_i는 정상 데이터인 0x00002408이지만 Decoder의 입력인 HWDATA_toSLV는 0x00006408로 에러가 발생하였다. 그러나 최종적으로 Slave가 받는 데이터인 DCD_HWDATA에서 원본인 정상 데이터와 동일한 0x00002408이므로 에러가 정정되었음을 확인할 수 있다.

3. HRDATA에서 1비트 에러 발생

그림 9에서는 Master가 입력으로 받는 HRDATA에서 1비트 에러 감지 및 정정이 완료됨을 확인할 수 있다. HRDATA_err가 0으로 에러가 일어나지 않은 상태인 그림 9의 A에서는 Slave에서 버스로 출력되는 HRDATA_i, Master의 Decoder가 입력으로 받는 HRDATA_toMAS, 최종적으로 Master가 입력으로 받는 DCD_HRDATA 세 가지의 데이터가 모두 0x0000672a로 동일함을 확인할 수 있다.

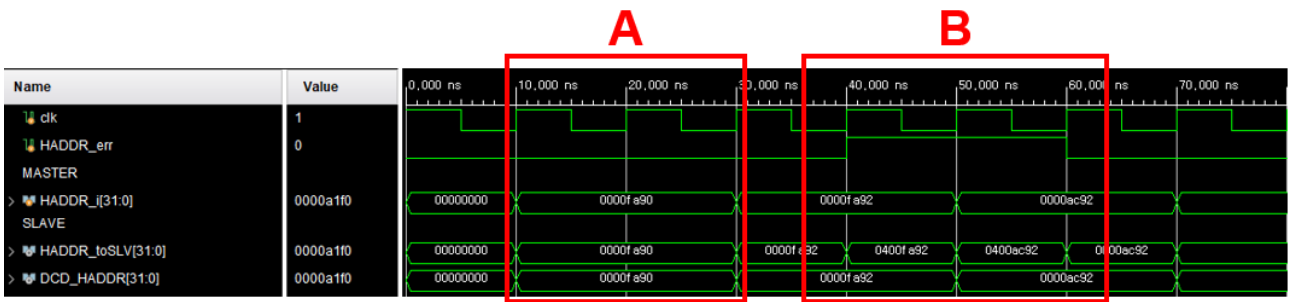


Fig. 7. 1 bit error in HADDR.
그림 7. HADDR에 1비트 에러가 발생한 경우

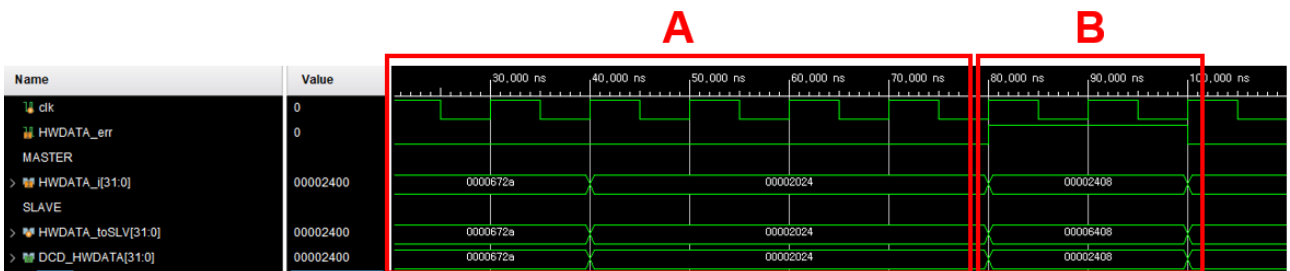


Fig. 8. 1 bit error in HWDATA.
그림 8. HWDATA에 1비트 에러가 발생한 경우

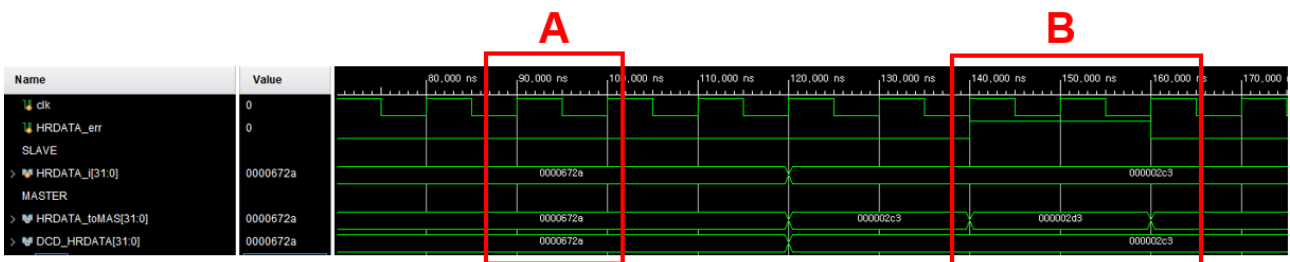


Fig. 9. 1 bit error in HRDATA.
그림 9. HRDATA에 1비트 에러가 발생한 경우

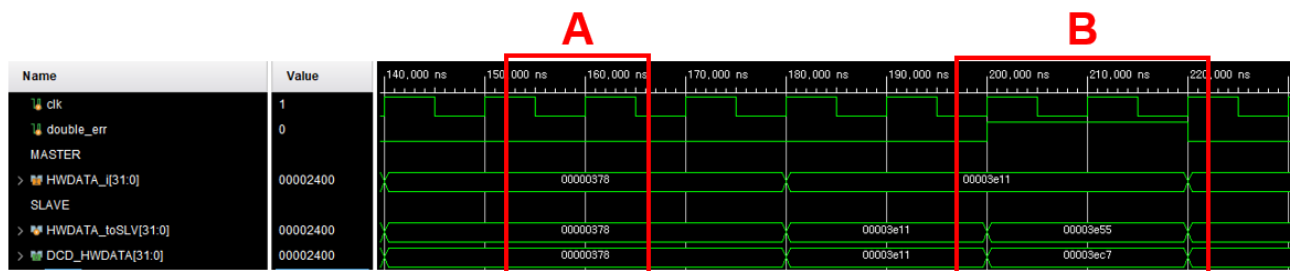


Fig. 10. 2 bits error on the bus.
그림 10. 버스에 2비트 에러가 발생한 경우

하지만 HRDATA_err가 1인 그림 9의 B에서는 1비트 에러가 발생하였다. Slave에서 버스로 출력된 데이터인 HRDATA_i에 에러가 발생하여 Master의 Decoder가 비정상 데이터인 0x000002d3을 입력으로 받았다. 하지만 Decoder에서 에러 정정 동작을 수행하여 Master

가 받는 데이터인 DCD_HRDATA는 정상 데이터인 0x000002c3을 수신함을 확인할 수 있다.

4. 2비트 에러 발생

그림 10에서는 Master의 출력 데이터인 HWDATA_i

가 버스 상에서 2비트 에러가 발생한 경우를 나타낸다. 그림 10의 A에서는 에러가 발생하지 않아 모든 데이터가 0x00000378로 같은 것을 확인할 수 있다. 그러나 2비트 에러가 발생한 그림 10의 B에서는 모든 데이터가 다름을 확인할 수 있다. 특히, 최종적인 Slave의 입력인 DCD_HWDATA의 0x00003ec7과 원본 데이터인 HWDATA_i의 0x00003e11과 다르다. 즉, SEC-DED의 Decoder가 2비트 에러가 발생했을 때 정상적으로 에러 정정이 가능하지 않음을 의미한다.

V. 결론

본 논문에서는 AHB-Lite 버스에서 에러가 일어난 경우, SEC-DED에 기반한 ECC를 통해 중단간 데이터 전송에서 실시간 에러 정정을 수행하는 방법을 제시하였으며 저전력과 지연시간을 고려한 SEC-DED Encoder 및 Decoder를 설계하여 1비트 에러 정정이 가능함을 확인하였다.

하지만 2비트 에러가 발생하는 경우에는 여전히 에러 정정이 이루어지지 않는다는 한계점이 존재한다. 그러나 복잡한 회로로 구성된 프로세서 코어나 회로 자체가 소프트웨어 에러에 취약한 메모리와는 달리 온칩 버스는 다수의 소프트웨어 에러가 동시에 발생할 확률이 낮다고 알려져 있다. 그러나 자동차와 같이 데이터 무결성이 매우 중요한 어플리케이션의 경우, 실제 소프트웨어 에러 발생을 측정할 확률 데이터에 기반하여 동시에 2비트 에러가 발생할 확률을 계산하고 이로부터 2비트 에러를 걱정할 필요가 없다는 점을 확인하든지 아니면 2비트 에러까지 정정할 수 있도록 기술을 개선하든지 하는 추가 연구가 장기적으로는 필요하다.

References

- [1] H. Jun, "SEC-DED-DAEC codes without mis-correction for protecting on-chip memories," *Journal of the Korea Institute of Information and Communication Engineering*, vol.26, no.10, pp. 1559-1562, 2022.
DOI: 10.6109/jkiice.2022.26.10.1559
- [2] J. Kim, S. Yang, and S. Lee, "Design of a Delayed Dual-Core Lock-Step Processor with Automatic Recovery in Soft Errors," *Korean.electr. elctron.eng.*, vol.27, no.4, pp.683, 2023.

DOI: 10.7471/ikeee.2023.27.4.683

[3] AUTOSAR, *Specification of SW-C End-to-End Communication Protection Library*, 2010.

[4] *ARM_IHI0033A_AMBA_AHB-Lite_SPEC*, ARM, 2001.

[5] R. Hamming, "Error Correcting and Error Detecting Codes", *The Bell System Technical Journal*, vol.29, no.2, pp.147, 1950.

DOI: 10.1002/j.1538-7305.1950.tb00463.x

[6] A. Neale and M. Sachdev, "A New SEC-DED Error Correction Code Subclass for Adjacent MBU Tolerance in Embedded Memory," *IEEE Transactions on Device and Materials Reliability*, vol.13, no.1, pp.223, 2013. DOI: 10.1109/TDMR.2012.2232671

[7] H. Moon, W. Lee, and H. Kal, "Study on Structure and Principle of Linear Block Error Correction Code", *Journal of the Korea Institute of Electronic Communication Sciences*, vol.13, no.4, pp.721, 2018.

DOI : 10.13067/JKIECS.2018.13.4.721

[8] S. Shamshiri, A. Ghofrani, and K. Cheng, "End-to-end error correction and online diagnosis for on-chip networks," *Proceedings of IEEE International Test Conference*, 2011.

DOI: 10.1109/TEST.2011.6139156

BIOGRAPHY

Eunbae Gil (Member)



2018~ : Candidate for BS degree in Electronic Engineering, Soongsil University.

⟨Main interest⟩ Security SoC, Automotive SoC, Processor SoC

Chan Park (Member)



2017~ : Candidate for BS degree in Electronic Engineering, Soongsil University

⟨Main Interest⟩ Automotive SoC, AI SoC

Juho Kim (Member)

2022 : BS degree in Electronic Engineering, Soongsil University.
 2022~ : Candidate for MS degree in Electronic Engineering, Soongsil University.
 <Main Interest> Automotive SoC, Processor SoC, AI SoC

Seongsoo Lee (Life Member)

1991 : BS degree in Electronic Engineering, Seoul National University.
 1993 : MS degree in Electronic Engineering, Seoul National University.

1998 : PhD degree in Electrical Engineering, Seoul National University.

1998~2000 : Research Associate, University of Tokyo

2000~2002 : Research Professor, Ewha Womans University

2002~Now : Professor in School of Electronic Engineering, Soongsil University

<Main Interest> AI SoC, Automotive SoC, Security SoC, Processor SoC, Power Management SoC, Battery Management SoC, Reliability and Safety

Joonho Chung (Member)

2018~ : Candidate for BS degree in Electronic Engineering, Soongsil University
 <Main Interest> Automotive SoC, AI SoC, Security SoC, Processor SoC

Joosock Lee (Member)

1983 : BS degree in Electronic Engineering, Sogang University.
 1985 : MS degree in Electronic Engineering, Korea University.
 1999 : PhD degree in Electrical Engineering, Korea University.
 1985~1995 : Senior Engineer, LG Central Laboratory

2004~2005 : CTO, MtekVision Ltd

2006~2010 : Chief of SoC Center, Chungbuk Technopark

2022~Now : Professor in School of Electronic Engineering, Soongsil University

<Main Interest> AI SoC, Power Management SoC, Battery Management SoC