International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 21 (2017) pp. 11124-11129

EMI Reduction algorithm using enhanced-HARQ Implementation for Controller Area Network

Cheolsu Han[1] and Hi Seok Kim[2]

[1,2]Electronic Engineering Department, Cheongju University, Cheongju City, South Korea.

[1]Orcid : 0000-0001-9515-4392, [2]Orcid: 0000-0001-5693-7669

Abstract

Controller Area Network (CAN) is exposed to noise and other electro-magnetic interferences (EMI) specifically in automotive industry application. Maintaining a flawless CAN controller is challenging especially if requires an immediate response in correcting this unwanted noise in the channel that lead errors in the received message. This proposed method aims to achieve an error correction capability to facilitate an optimum redundancy and to reduce the probability of error. Thereby, minimizing the retransmission operation that causes to produce an EMI will benefit the system. The proposed scheme is synthesized on the Xilinx Virtex-5 FPGA. The enhanced-Hybrid Automatic Repeat reQuest (e-HARQ) scheme implementation for CAN helps to minimize the EMI and to improve the system efficiency.

Keywords: EMI; CAN; e-HARQ

INTRODUCTION

Controller Area Network (CAN) uses a serial bus to transmit and receive message information and it requires a high level of fidelity and robust on error detection [1-2]. This CAN bus is the backbone network in managing the car system like the engine timing, mode of transmission, and brakes, etc. Nowadays, CAN is one of the most widely used communication networks in the automotive industry. This protocol reduces usage of wiring due to distributed control and ensures the system performances. Additionally, in CAN system, the traffic congestion is eliminated as the messages are transmitted based on their priority. In addition, it performs carrier sense multiple access/collision detection with arbitration on message priority (CSMA/CD + AMP) [3].

CAN provides the ability to work in a different electrical environment. However, it is exposed in Electro-Magnetic Interference (EMI) and Electro Static Discharge (ESD) particularly in harsh environments like in automotive and industrial applications [4]. This drawback, especially the EMI, occurred from a large inductive loads even in external radio frequency interference [5]. Also, one reason on how this EMI occur is through the uses the automatic repeat request (ARQ) scheme in a CAN. This disadvantage increases the probability of CAN to get corrupted. The outcome in CAN bus reduces the efficiency and introduce error-bits into the transmitted frame. These error-bits can be a single bit or burst errors depending on the severity of EMI [2]. Some major factors that affect the efficiency of the CAN bus are caused by magnetic relays and communication cables in the vehicle and from a repetitive transmission. Furthermore, inefficient stop-and-wait retransmission process and limitation in length requirements [6] of CAN influence the reduction of bus performance. One important technique used to improve the efficiency of the communication process in the CAN protocol is to verify the information sent by the transmitter.

To lessen the problem in CAN transmission, it has 6 different error management techniques such as bit monitoring, bits stuffing, acknowledgment check, message frame check, error signaling and cyclic redundancy checking (CRC) to maintain its error-free system. Several techniques have been designed to minimize or totally eliminate the EMI problem in achieving a reliable communication network [7], however, these existing schemes have their own limitations. In this paper, the implementation of enhanced hybrid automatic repeat request (e-HARQ) aims to reduce the effect of EMI on CAN bus efficiency. HARQ is a technique which utilizes Forward Error Correction (FEC), along with the error-detection detection used by the ARQ scheme. Also, HARQ offers the potential for better performance if ARQ and the FEC schemes are properly implemented.

The rest of this paper is organized as follows. In Section II, the discussion of error management technique for CAN protocol and the HARQ implementation is introduced. The proposed algorithm using e-HARQ scheme is presented in Section III. Section IV describes the experimental testing and analysis of results. And, Section V, concludes this work.

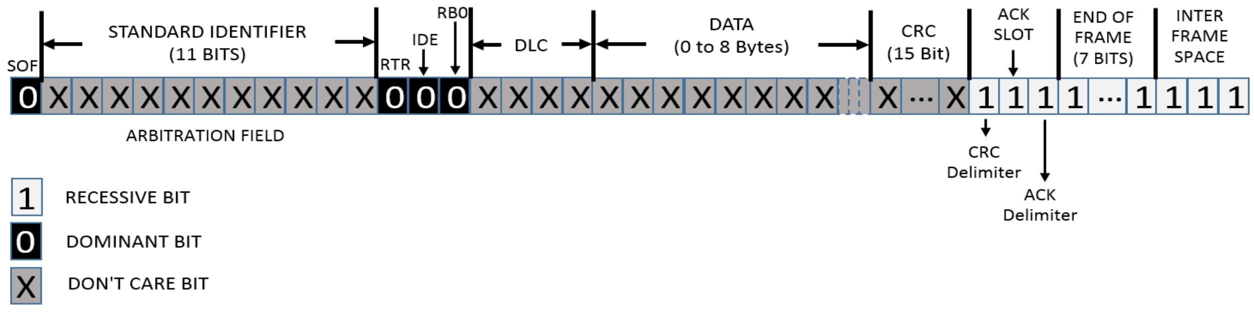

Figure 1: CAN's Standard Data Frame Format

DIFFERENT ERROR MANAGEMENT TECHNIQUES OF A CAN, HARQ SCHEME, AND RELATED WORKS.

The main objective of the CAN is to prevent any occurrence of errors in able to maintain a reliable transmission communication network. The CAN protocol utilizes an error-management techniques to suppress any manifestation of errors and to evaluate the CAN bus performance.

A. Error Detection Techniques

The error-handling schemes aim to detect errors in the transmitted message frames over the bus and request for retransmission. Any node that detects an error raises an error flag, halting the transmission on the bus [8]. The five different error detection mechanisms used by CAN protocol is listed below [9].

1. *Bit Monitoring.* This method compares the signal sent with the signal seen on the bus line excluding the arbitration phase during the transmission of the identifier. The transmitter sends an error frame whenever the signal on the bus line is different from the signal sent.
2. *Bit Stuffing.* It inserts a complementary bit at the sixth-bit position whenever five consecutive identical bits is detected. Moreover, this rule only applies between in SOF and CRC frame.
3. *Message Frame Check/Form.* Whenever a fixed format field in a received frame does not conform to the protocol specification, the receiver sends an error frame and does not accept the received frame.
4. *Acknowledgement Check***.** It monitors a dominant bit in acknowledging slot both for the data and remote frame.
5. *Cyclic redundancy check (CRC).* It computes the 15-bit CRC bits sequence used for error checking between the frame fields of the start of frame (SOF) to data field [10], figure 1 shows the CAN's standard data frame format.

B. Hybrid Automatic Repeat Request (HARQ) Algorithm

Part of CAN protocol is the use of Automatic Repeat Request (ARQ). It is widely used error control technique in many communication systems. This technique includes the parity or redundant bits to the transmitted information stream [11] and relies on error detection scheme code like CRC. It aims to make a reliable data link. Before sending it into the noisy medium, it split the information stream into smaller packets and encoded using the error-detection coding technique. The receiver computes for the syndrome if the result is zero, it means that the packet is error free and the receiver will accept the packet. Then a positive acknowledgment will be sent by the receiver confirming a successful reception. However, if no acknowledgment is received, the transmitter waits for a pre-determined amount of time and a retransmission of packet will be processed. A negative acknowledgment will be sent to the transmitter requesting for retransmission once the received syndrome is not equal to zero. The receiver discards the corrupted packet and all further packets until the corrupted packet is correctly received. An erroneous packet is delivered to the destination node only when the error-detection scheme fails to detect the error. In other words, if the information arrived properly without any erroneous packet, the receiver is ready to receive and process new data. However, if the information arrived with some problem, corrupted, the receiver must request that the transmitter sent the packet again or needed for retransmission. Figure 2 shows the basic ARQ scheme in sending an acknowledgment feedbacks uses by the receiver to inform whether the transmission was successful or not. The transmitter sends a packet 1 when the receiver detects no erroneous information, it responds to a positive acknowledgment signal "ACK" message, informing the transmitter that the transmitted packet 1 has no error or uncorrupted. The transmission continues, and packet 2 is sent. If packet 2 arrives with error, the receiver sends a negative acknowledgment signal "NACK".

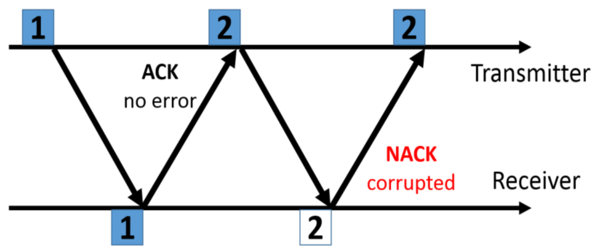

Figure 2: ARQ transmission scheme.

Hybrid Automatic Repeat Request (HARQ or hybrid ARQ) is a combination of FEC and ARQ error-control and this technique called as soft combining [12] which do not discard the received corrupted data. With the soft combining, the data packets that are not properly decoded are not discarded. The received signal is stored in a buffer and will be combined with the next retransmission. Figure 3 show this technique of HARQ in using the soft combining scheme. The buffer served as saving storage of the corrupted received signal.

Therefore, in standard ARQ, redundant bits are added to data to be transmitted like in CRC. Receivers will request a new message from the sender once a corrupted message is detected. However, in HARQ, the original data is encoded with an FEC code, and the parity bits are either immediately sent together with the message or only transmitted upon request whenever a receiver detects an erroneous message.

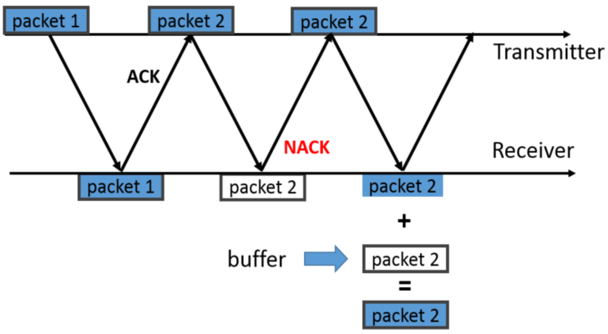

Figure 3: HARQ transmission scheme.

The error-detecting code like CRC can be omitted when a code used that can perform both FEC like Reed-Solomon code. The FEC code is chosen to correct an expected subset of all errors that may occur, while the ARQ scheme is used as correct errors using retransmission. As a result, HARQ performs better than the conventional ARQ especially in harsh conditions such as in automotive industry applications.

The three types of HARQ schemes are Type-I, II and III. Type-I keeps on retransmitting the same information and discards the previously received information even if there are some important information with it resulting in an inefficient method. Type-II uses a retransmission sequence containing the redundant bits instead of resending the original bits sequence. These resent redundant bits will be recombined with the recently received data and it can result to a higher robustness of FEC. While in Type-III, the packet is self-decodable. It can recover the missing message sequence.

C. Related Works

Several solutions are presented to minimize or totally eliminate the EMI problem in the CAN. A technique like [5] presents by combining error-detection scheme and differential bus twisted-pair and/or shielded cables. However, there is a code which appropriates and a bit effective in correcting burst errors which is Reed-Solomon (RS) code. Reference [13] only discussed the theory and relevance of HARQ), there is no experimental testing has been presented. Although, the presentation on [14] is energy efficient with the implementation of HARQ, however, there is no real simulation conducted to verify the theoretical presentation.

PROPOSED IMPLEMENTATION

A. Controller Area Network (CAN) Architecture

Reconfiguration is one key concept in maximizing the potential of any FPGA-based device and its software or hardware is easier to modify. Therefore, to implement this proposed e-HARQ scheme, necessary modifications of the sub-block under the CAN protocol is needed. This sub-block is called as a Bit stream processor (BSP). The CAN block diagram is shown in Figure 4. This processor performs several medium access control (MAC) and/or logical link control (LCC) during transmission and reception of CAN messages, and insertion of CRC bits and other pre-defined field. The other function blocks are defined below..

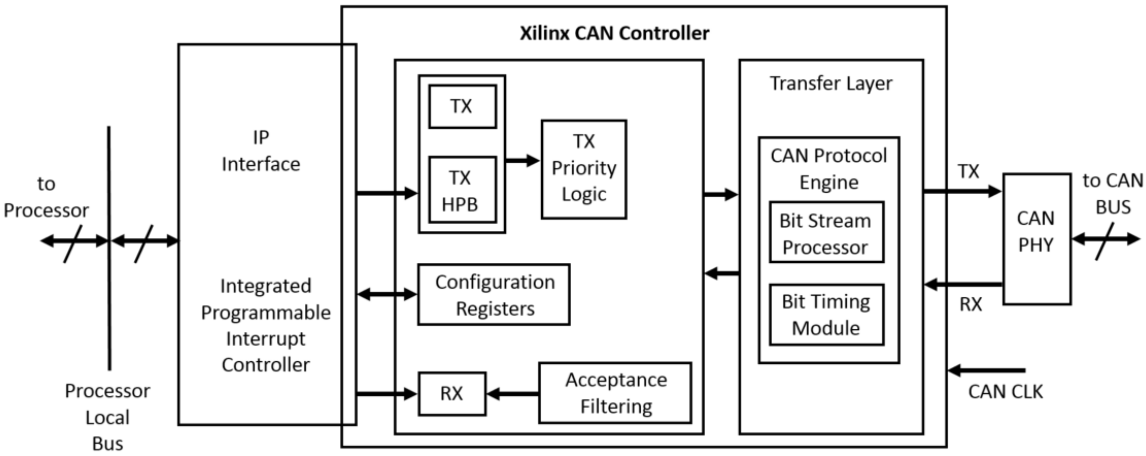

Figure 4: CAN Block Diagram.

The configuration register handles for read and write access to the registers through the external micro-controller interface. Transmit (TX) and Receive (RX) modules are separate storage buffer for transmitting and receiving messages using the first in-first out (FIFO) scheme. Each has a maximum depth of up to 64 messages that are individually configurable. The transfer high priority buffer (TX HPB) provides storage for one transmit message. Any message written on this buffer has a high priority for transmission. Acceptance filters handle the sorting of incoming messages with a user-defined acceptance mask and ID registers to determine whether to save in the RX buffer or discard them. Also, under the CAN protocol engine is the bit timing logic (BTL). This module provides the synchronization, generates sampling clock for the BSP module state machine

B. Enhanced-Hybrid Automatic Repeat Request (e-HARQ)

The technique in forward error correction (FEC) is by adding controlled redundancy to the data for the receiver to correct the errors. Thus, it demands more bandwidth. Moreover, in real time application, certain delay is not allowable. The methods used by CAN bus in handling errors does not effectively maintain its data rates especially in the environments where the interference is severe.

As shown in Figure 1, the conventional CAN uses CRC to detect errors. This proposed scheme uses a modified HARQ scheme (e-HARQ) which uses the column-row parity matrix and the frame is redesign to reduce the frame's bit overhead that can improve the performance of the CAN bus. The data bits in CAN protocol are encoded into frame format, this method uses a fixed size matrix. The concept is, to construct only the data bits (only the data frame) into a matrix form, the maximum data bits is 64 bits. Therefore, the 64 frame bits can be constructed on an 8 by 8 matrix. The initial testing was simulated on the maximum number of data bits. The proposed algorithm is defined below.

An example for explanation purposes, a random set of 64-bit was set for simulation. It was denoted as *D(x)* = 1835111023 in base 10.

1st step. The matrix form of *D(x)* converted in base 2. The most significant bit (MSB) is encoding on the first row-first column position to the least significant bit (LSB) on the last row-last column potion for proper reference as shown in Figure 5.

MSB 0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	1	1	0	1	1	0	1
0	1	1	0	0	0	0	1
1	0	0	1	0	0	1	0
0	1	1	0	1	1	1	LSB 1

Figure 5: Matrix size (8 by 8) allocated for CAN frame with 64-bits data frame.

2nd step. Since the maximum data bits can be 64 bits and a possible number of all 1s on any row is 8, it can be set that the number row parity corrector bits are 4 (shaded in light blue), and the column parity corrector is only 1 (shaded in light green). Therefore, the total matrix size for the maximum data bits is 9-bit row and 11-bit column, as shown in Fig. 6. The shaded in the darker blue color is the position of row parity corrector bits.

0	0	0	0	0	0	0	0				
0	0	0	0	0	0	0	0				
0	0	0	0	0	0	0	0				
0	0	0	0	0	0	0	0				
0	1	1	0	1	1	0	1				
0	1	1	0	0	0	0	1				
1	0	0	1	0	0	1	0				
0	1	1	0	1	1	1	1				

Figure 6: The total matrix size (9 by 11) including the row and column parity corrector.

3rd step. In obtaining the row corrector parity bits, simply determine the number of 1s in each row, i.e., in row 1, the total number of 1 is 0, hence, the row parity bit corrector is 0000_2 (base 2), the same with row 2 to 4, while in row 5, the row parity corrector is 0101_2 because the total number of 1s is 5. Then 0011_2, 0011_2 and 0110_2 for row 6, 7 and 8, respectively as shown in figure 7.

0	0	0	0	0	0	0	0	**0**	**0**	**0**	**0**
0	0	0	0	0	0	0	0	**0**	**0**	**0**	**0**
0	0	0	0	0	0	0	0	**0**	**0**	**0**	**0**
0	0	0	0	0	0	0	0	**0**	**0**	**0**	**0**
0	1	1	0	1	1	0	1	**0**	**1**	**0**	**1**
0	1	1	0	0	0	0	1	**0**	**1**	**1**	**1**
1	0	0	1	0	0	1	0	**0**	**0**	**1**	**1**
0	1	1	0	1	1	1	1	**0**	**1**	**1**	**0**

Figure 7: The results of column parity bits equivalent of each row.

4th step. In determining the row parity bits, XOR process is applied on each column including the column parity bits. The result is shown in figure 8.

5th step. In able to determine if there is an error occurred during transmission, the receiver decodes the received message. A column is invalid if the bits are not in parity.

6th step. Errors in row are detected if the decimal value in row parity bit is not equivalent in each row transmitted.

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	0	1	1	0	1	0	1	0	1
0	1	1	0	0	0	0	1	0	1	1	1
1	0	0	1	0	0	1	0	0	0	1	1
0	1	1	0	1	1	1	1	0	1	1	0
1	**1**	**1**	**1**	**0**	**0**	**0**	**1**	**0**	**1**	**1**	**1**

Figure 8: The results of row parity bits equivalent of each column.

7th step. If no error is found in column or row, message will be accepted.

All detected erroneous columns and rows are recorded. Each erroneous column and row have easily traced both ways. Figure 9 shows the algorithm procedure before transmission. Once the data frame is cleared from error, it is ready to attach the remaining frames before transmission.

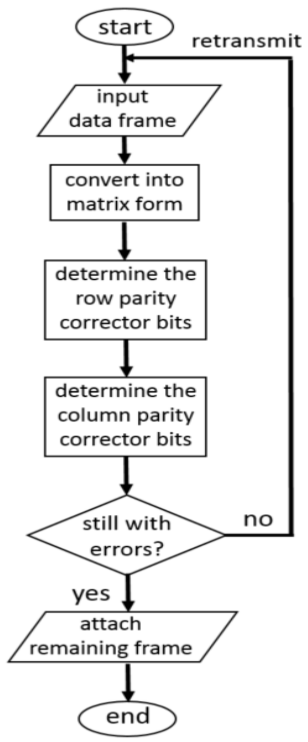

Figure 9: Flowchart of proposed e-HARQ.

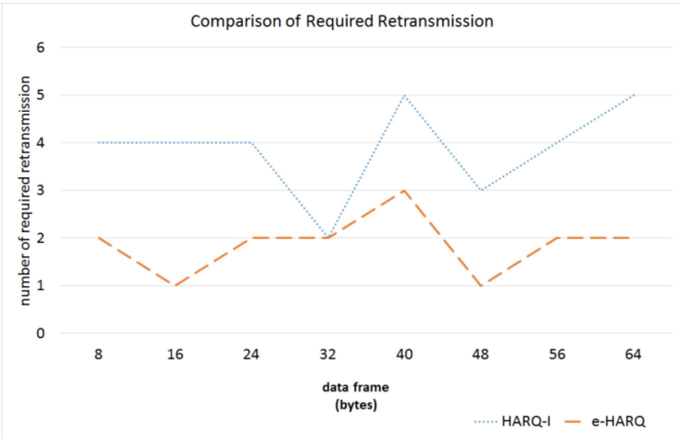

Figure 10: Test result between HARQ Type-1 and e-HARQ.

EXPERIMENTAL RESULTS

Simulation is done for verification purposes of the proposed implementation. This scheme and configuration is tested on the 2.0 A CAN frame on a random data bits streams and injected with a burst errors to determine the number of required retransmission. Figure 10 shows the test results between HARQ Type-1 and e-HARQ. It shows that the e-HARQ reduces the required retransmission of the CAN system. Moreover, it was synthesized to Xilinx Virtex 5 FPGA using Xilinx ISE at 125 kbps of baud rate to 1 to 8 bytes of data.

CONCLUSION

As shown in figure 10, reducing the need for retransmission minimizes the source of burst error in the CAN system. The proposed algorithm achieved in aiming the target to decrease the use of retransmission process. This e-HARQ increases the reliability and efficiency of a communication channel and can help to defend against any packet loss. Moreover, it permits a rapid retransmission of erroneous message sequence.

REFERENCES

[1] Patil, A. P., Ashtekar, S., and Chintawar, A., 2017, "Implementation of Controller Area Network Data Compression Algorithm for Automobile Applications," International Journal of Advanced Computational Engineering and Networking, 5(3), pp. 9 – 13.

[2] Serfa Juan, R. O., Jeong, M. W., and Kim H. S., 2016, "Development of Burst Error Reduction Algorithm for CAN using Interleaver Method," Proc. 2016 International SoC Design Conference, pp. 165-166.

[3] Vojayan, T., 2012, "Controller Area Network in Modern Home Automation," International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, 1(2), pp. 61-65.

[4] Serrao, V., Lidozzi, L, Di Napoli, A., 2008, "EMI Characterization and Communication Aspects for Power Electronics in Hybrid Vehicles," Proc. of 2007 European Conference on Power Electronics and Applications, 1-10.

[5] Ren, F, Zheng, Y. R., Zawodniok, M., and Sarangapani, J., 2007, "Effects of Electromagnetic Interference on Control Area Network Performance," Proc. of IEEE Region 5 Technical Conference, pp. 199-204.

[6] Kim, H. S., 2017, "FPGA Implementation of Overhead Reduction Algorithm for Interspersed Redundancy Bits using EEDC," Journal of Institute of Korean Electrical and Electronics Engineers, 21(2), pp. 130-135.

[7] Gomez, A., Junior, E.P., and Nascimento, I., 2005, "EMC-EMI Optimized High Speed CAN Line Driver," Proc. of 18th Symposium on Integrated Circuits and Design, 9-14.

[8] KVASER. "The CAN Protocol Tutorial."

[9] Charzinski, J., 1994. "Performance of the Error Detection Mechanisms in CAN," Proc. of the 1st International CAN Conference, pp. 20-29.

[10] Bosch, "CAN Specifications Version 2.0," 1991, The Bosch IC Design Center, Available online: esd.cs.ucr.edu/webres/can20.pdf

[11] Faruque, S., 2016, "Automatic Repeat Request (ARQ)," Radio Frequency Channel Coding Made Easy, pp. 17-35.

[12] Dinis, R., Carvalho. Martins, J., 2008, "Soft Combining ARQ Techniques for Wireless Systems Employing SC-FDE Schemes," Proc. of International Conference on Computer Communications and Networks – 2008, pp. 1-5.

[13] Lott, C., Milenkovic. O, and Soljanin, E., 2007, "Hybrid ARQ: Theory, State if the Art and Future Directions," Proc. of 2007 IEEE Information Theory Workshop on Information Theory for Wireless Networks, pp. 1-5.

[14] Yue. W., and Hongwen, Y., 2015, "Energy efficient HARQ scheme for wireless communication over block fading channel," The Journal of China Universities of Posts and Telecommunications, 22 (3), pp. 35-40.