Dynamic Message Id Allocation and Arbitration In Can Architecture

Keshav Verma[1], Raman Jeet Singh[2]
[1] Student, [2]Lecturer,
[1]ECE Department, Ludhiana Collage of Engineering and technology, Katanikalan
[2]ECE Department Ludhiana Collage of Engineering and technology, Katanikalan

***Abstract* - On board communication on FPGA is an issue of interest for many researchers in recent years. Various protocols have been proposed in the literature to provide communication between central unit and peripherals. Control Area Network is also an On board protocol for communication of FPGA or microcontrollers with different peripherals. Traditionally the message id allocation in CAN is static which means only one id is given to each application and it remains same throughout the communication. This leads to biasness for various applications while others always access the bus at the time of arbitration. In this paper Dynamic approach of message id allocation from the message id window is proposed which increases the randomness and thereby decreases the biasness. The results discussed in the coming sections also prove the same.**

***Keywords* - CAN, Bus Arbitration, Dynamic Message Scheduling.**

I. INTRODUCTION

The Controller area Network may be a serial bus prescript developed by Bosch within the early Eighties [1]. It defines a regular for economical and reliable communication between detector, controller, actuator and different nodes in period of time applications. It is the de facto customary during a giant form of networked embedded management systems. The first development was primarily supported by the vehicle industry: which is found during the form of cars, trucks, boats, spacecraft, and different kinds of vehicles [2]. The protocol is additionally wide used these days in industrial automation and different areas of networked embedded management, with applications in numerous products like production machinery, medical instrumentality, building automation, weaving machines, and wheelchairs [1].

The will protocol standardizes the physical and electric circuit layers, that square measure the 2 lowest layers of the open systems interconnect communication model. For many systems, higher-layer protocols square measure required to modify economical development and operation [3]. Such protocols square measure required for outlining however they will protocol ought to be employed in applications, as an example, a way to talk to the configuration of identifiers with relation to application messages, a way to package application messagesinto frames, and the way to affect start-up and fault handling.

Description

A bus can manage data field and overhead, like symbol and management fields [4]. Since the appliance processes generally square measure asynchronous, the bus features a mechanism for resolution conflicts. For CAN, it's supported a non-destructive arbitration method. The will protocol thus belongs to the category of protocols denoted as carrier sense multiple access/collision shunning, which implies that the protocol listens to the network so as to avoid collisions [5]. CSMA/CD protocols like local area network have instead a mechanism to affect collisions once they're detected. It can also include numerous strategies for error detection and error handling. The communication rate of a network supported will depends on the physical distances between the nodes [5]. If the gap is a smaller amount than forty m, the speed is up to one Mbps.

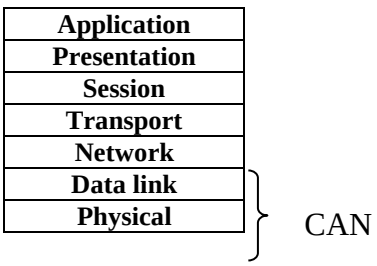

Figure: 1: Vehicle Applications of Controller Area Network

Message formats

CAN distinguishes four message formats: information, remote, error, and overload frames. Here we tend to limit the discussion to info frame [10]. The knowledge frame begins with the start-of-frame bit. It's followed by associate eleven-bit symbol and also the

remote transmission request bit. The symbol and also the RTR bit kind the arbitration field [6]. The management field consists of six bits and indicates what percentage bytes of information follow within the data field. The info field is zero to eight bytes. The info field is followed by the cyclic redundancy verification field that permits the receiver to envision if the received bit sequence was corrupted [7]. The two-bit acknowledgment field is employed by the transmitter to receive associate acknowledgment of a legitimate frame from any receiver.

S O F	Identifier	RTR	Control	Data	CRC	AC K	E O F

Figure 2: Three nodes connected through a CAN bus

Arbitration

Arbitration is that the mechanism that handles bus access conflicts. Whenever the will bus is free, any unit will begin to transmit a message [8]. Attainable conflicts, as a result of quite one unit getting down to transmit at the same time, square measure resolved by bit-wise arbitration victimisation the symbol of every unit [8]. Throughout the arbitration section, every transmission unit transmits its symbol and compares it with the extent monitored on the bus [9]. If these levels square measure equal, the unit continues to transmit. If the unit detects a dominant level on the bus, whereas it had been attempting to transmit a recessive level, then it equal transmission.

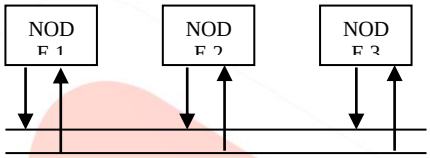

Figure 3: CAN message frame

II. PROPOSED METHODOLOGY

The problem of message scheduling in Control Area Network can be solved by simply making the ids correspond to every message dynamic. This simply means that with every message the id window changes with every message. These message ids selected from a pre defined message id window. The selection of message ids from that window is also dynamic, means there is no fixed order of selection. So that any id can request the bus access any time can have random id, if they are not a part of emergency messages. Besides this if any message belongs to emergency message category then it must be given access as soon as it is available or enabled.

Id1	Id2	Id3	Id4	Id5	Id6

Figure 4: Message Id Window

Figure 4 shows the message id window from where the ids are fetched randomly and are different for every type of applications for arbitration.

Pseudo Code:

1: Start

2: Assign each application a unique message id from id window

3: Emergency messages are assigned with minimum id (decimal value)

4: Enable 2 or 3 applications to check bus arbitration

(assigned ids act as weights to the equation

$$access\ grant = \sum_{i=1}^{N} w_i * application_i \quad (1)$$

5: Check for weights for the selected enables

6: $if\ (w_i < w_{i+1})\ \ i \in N\{..\}$ (covers both normal and emergency messages)

w_i is granted the permission to access the bus

7: $end\ if$

8: End

III. RESULTS AND DISCUSSIONS

The proposed methodology is simulated using the Xilinx Virtex4 FPGA in Xilinx ISim and the coding is done in Xilinx ISE environment using VHDL language. Figure 6 shows the top module implementation of the proposed methodology and figure 7 shows the simulation of the top module.

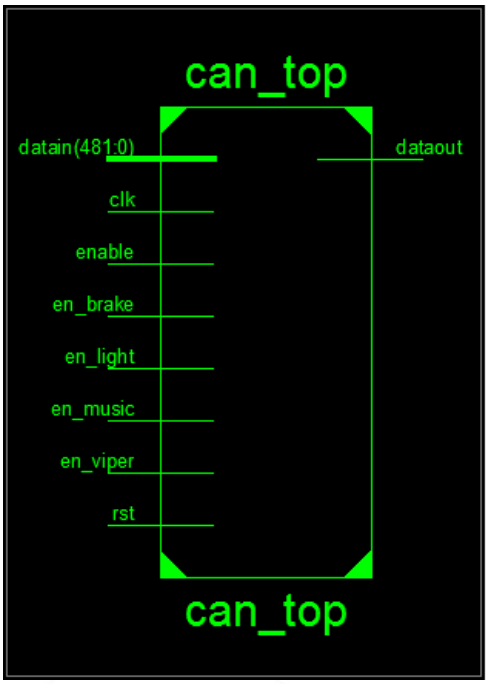

Figure 6: CAN Top Module

The worst case reaction time of the system to each message is given by

$$W_M^{n+1} = \min(\alpha, Fixed\ Id) + \sum \frac{W_M^n + \tau_{bit} + D_M}{T_M} \qquad (2)$$

$$R_M = W_M^{n+1} + C_M$$

Here R_M is the worst case response time. $\alpha$is the number of bits, τ_{bit} is the time required to transfer 1 bit,D_M is the deadline time and T_M is the period of the message, W_M is the step value for Message M and C_M is the normalized bit value. Table 1 shows the considered values of period and deadline of the message. Table 2 shows the actual order of processing of the message based on the number of bits and the fixed or assigned id to each message.

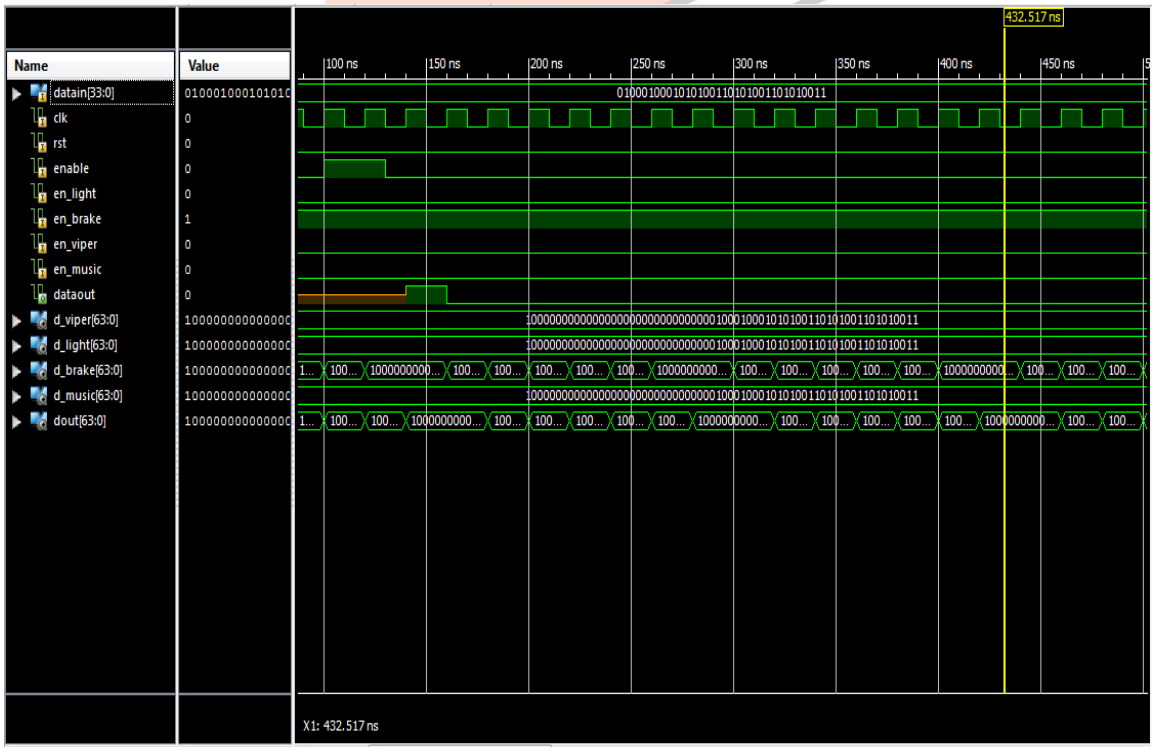

Figure 7: Simulation of Top Module

Table 1: Period and Deadline of the message

Message M	M1	M2	M3	M4	M5	M6
T_M/ms	10.0	5.0	5.0	10.0	10.0	10.0
D_M/ms	2.0	5.0	5.0	5.0	5.0	7.0

Message M	M7	M8	M9	M10	M11	M12
T_M/ms	10.0	10.0	10.0	10.0	10.0	20.0
D_M/ms	8.0	10.0	10.0	10.0	10.0	15.0

Table 2: Worst case Response Time and Order of Message

Message M	M1	M2	M3	M4	M5	M6
o(M)	1	4	9	2	8	6
R_M/ms	0.13	0.26	0.58	0.18	0.55	0.51
α [bit]	128	132	420	158	512	412
Fixed Id	2	396	132	12	42	100

Message M	M7	M8	M9	M10	M11	M12
o(M)	11	3	12	5	7	10
R_M/ms	0.65	0.19	0.91	0.49	0.55	0.63
α [bit]	324	154	448	240	264	308
Fixed Id	1282	11	1803	204	266	452

IV. CONCLUSION AND FUTURE SCOPE

The message id allocation in CAN is done statically which results in biased bus arbitration. So firstly dynamic message id allocation is tested which shows improved and unbiased message id allocation and bus arbitration. Then for emergency messages dynamic message id allocation is tested and it is proved from the simulation results that whenever there is any emergency message then it immediately grants access to that message.

In future the proposed methodology must be tested for a number of other application specific signals. Mainly for those applications in which there are two or more emergency messages wants to access the resource or are involved in bus arbitration.

REFERENCES

[1] L. Cauffriez, J. Ciccotelli, B. Conrardc, and M. Bayartc, "Design of intelligent distributed control systems: A dependability point of view," Reliab. Eng. Syst. Safety, vol. 84, pp. 19–32, 2004.

[2] S. Kelkar and R. Kamal, "Control area network based quotient remainder compression-algorithm for automotive applications," in Proc. 38th Annu. IEEE IECON, Montreal, QC, Canada, Oct. 2012, pp. 3030–3036.

[3] M. J. Short and M. J. Pont, "Fault-tolerant time-triggered communication using CAN," IEEE Trans. Ind. Informat., vol. 3, no. 2, pp. 131–142, May 2007.

[4] J. R. Pimentel and J. A. Fonseca, "FlexCAN: A flexible architecture for highly dependable embedded applications," in Proc. 3rd Int. Workshop Real-Time Netw., Italy, 2004.

[5] T. Fuhrer, B. Muller, W. Dieterie, F. Hartwich, R. Hugel, and H. Weiler, "Time triggered communication on CAN," in Proc. 7th Int. CAN Conf., Amsterdam, Netherlands, 2000.

[6] H. A. Hansson, T. Nolte, C. Norstrom, and S. Punnekkat, "Integrating reliability and timing analysis of CAN-based systems," IEEE Trans. Ind. Electron., vol. 49, no. 6, pp. 1240–1250, Dec. 2002.

[7] B. Jiang, Z. Mao, and P. Shi, " H∞-filter design for a class of networked control systems via T-S fuzzy-model approach," IEEE Trans. Fuzzy Syst., vol. 18, no. 1, pp. 201–208, Feb. 2010.

[8] I. Broster, A. Burns , G. Rodríguez-Navas, "Probabilistic Analysis of CAN with Faults", In Proceedings of the 23rd IEEE Real-Time Systems Symposium (RTSS'02), pp. 269-278, December, 2002.

[9] M. Di Natale, "Evaluating message transmission times in Controller Area Networks without buffer preemption", In 8th Brazilian Workshop on Real-Time Systems, 2006.

[10] J. Ferreira, A. Oliveira, P. Fonseca, J. A. Fonseca. "An Experiment to Assess Bit Error Rate in CAN".In Proceedings of 3rd International Workshop of Real-Time Networks (RTN2004), pp. 15-18, Cantania, Italy. June 2004.