CAN Tx Frame Implementation using Verilog

Author: Mr. Manjunath Koti, Dept of ECE, BNMIT, Bengaluru, India

Author Affiliation: Dr. Basavaraj I Neelgar, Professor, Dept of ECE, BNMIT, Bengaluru, India

10.51201/JUSST/21/07311

Abstract: *This article discusses the concept of CAN protocol and its implementation in verilog language. Initially the CAN protocol description is given in brief with the block diagram, later its design, implementation in verilog code is presented. The CAN transmission (Tx) data Frame is realized using verilog code, this is achieved by defining individual sub-blocks verilog codes and combining these to get the CAN transmission of data frame. In the year 1986, CAN data link layer protocol was introduced in SAE conference. In 1993, CAN protocol and high speed physical layer were internationally accredited as ISO 11898. As on today it has 11898-1 to 4 standard documents. The CAN 1.0, 2.0 versions were initially had fixed data rate for the entire frame. In 2012, CAN-FD (Flexible data rate) protocol was introduced. This will allow data phase a second higher bit rate, along with this restriction of 8 bytes is extended up to 64 bytes.In this paper CAN Tx data frame is realized using Xilinx 14.7 version using verilog language.*

Keywords: **CAN protocol, Verilog, data Frame, SAE, CAN-FD**

1. INTRODUCTION

Controller Area Network, in short CAN protocol is a serial communication protocol which provides efficient support to mainly automotive real time control systems with a very high level of security, error detection and correction. CAN is multi-master i.e. any node can initiate the communication, with synchronization using bit stuffing. CAN is highly reliable, low latency timings and priorities based messages can be configured by the user. Initially Robert Bosch GmbH has developed version 1.0 later it is modified to release the 2.0 to accommodate more CAN IDs by extending the 11-bit Identifier to 29-bit identifier. Applications are ranging from automotive to various electronics world. Almost all the transportation system is equipped with CAN bus system. A simple CAN bus system is shown in fig.1.

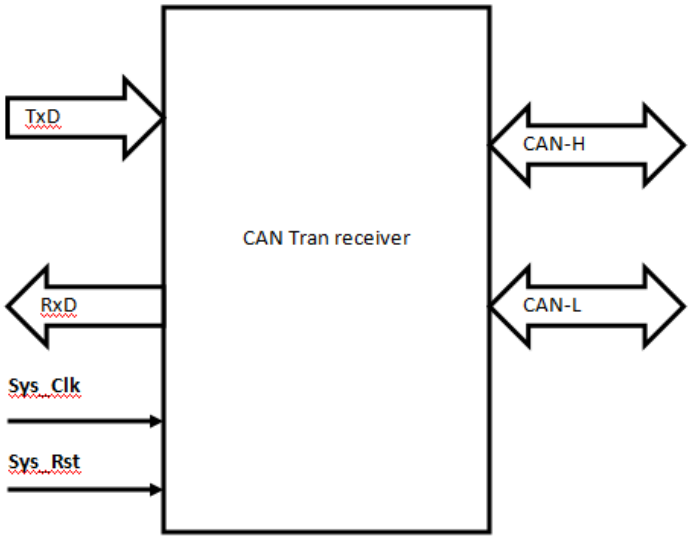


Fig. 1 CAN bus system

CAN offers CAN-FD which increases the data rate and flexibly data/messages can be sent. Meaning to achieve the data rate above 1Mbps and the data field can be up to 64 byte long and a limit of 8 bytes no more exists.

Fig. 2 shows the evolution of CAN protocol, since its inception it is less expensive and easy to implement feature has made widely used serial communication protocol. As on today, CAN-FD makes the presence and being used in may automotive applications.

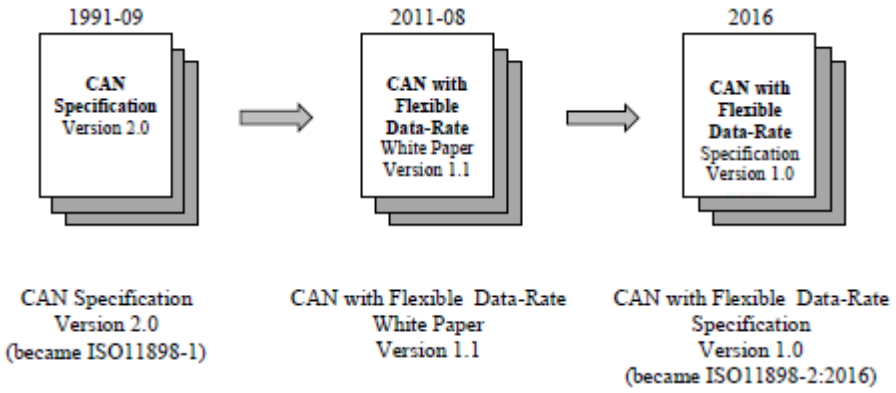


Fig. 2 Evolution of CAN protocol

2. Design and Methodology

CAN data frame can be defined as in the fig.2 below. It consists of start of frame (SOF), arbitration field, control field, data field, CRC field, ACK field and end of frame (EOF). This entire bits are been simulated in our work in verilog language.

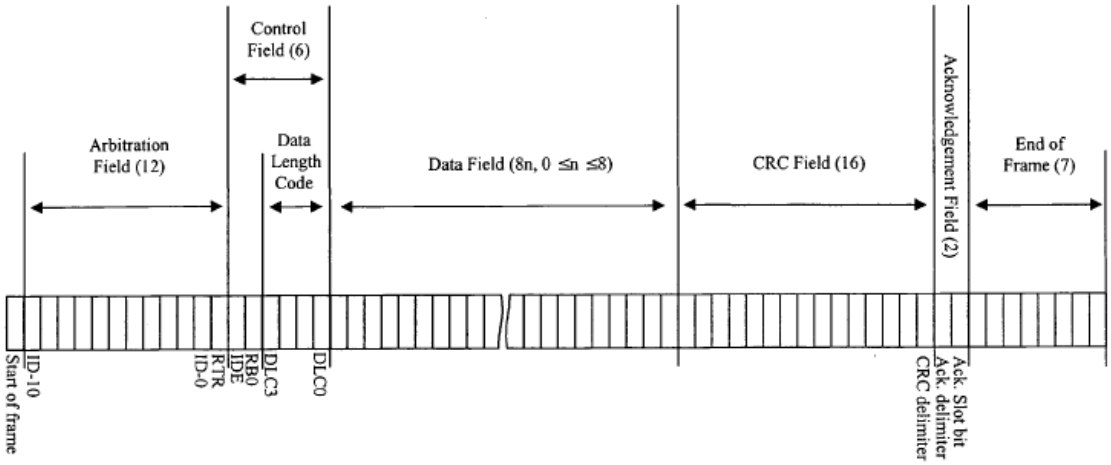


Fig.3 CAN data frame format

CAN Frame is transferred as electrical signals in physical layer. Twisted pair cable is used to transfer CAN_H and CAN_L signals. The specification of physical layer can be decided by the designer and signal level can be optimised for their application. NRZ signal format or any other format can be used by the designer depending on the application.

Messages

When the bus is in idle state, any CAN node can initiate the transfer of information on the bus. This has a fixed format these are called messages. These message formats have different name as data frame, remote frame error frame etc.

Information Routing

In CAN environment each node will not use any system related information i.e. it is independent of embedded system being used and it has many consequences as below.

System Flexibility

CAN transreceiver nodes can be added like a line replaceable units in aircraft. I.e. plug and play without changing its hardware or software of any node in the entire system or the application.

Message Routing

The CAN Identifier is used as a unique number though which message routing is possible. This ID indicates the meaning of data, so that all other nodes in the network can able to decide by MESSAGE FILTERING the information is required or not.

Multicast

As a result of MESSAGE FILTERING, where any node can access the data which it requires opens the data and any number of nodes can access the data simultaneously.

Data Consistency

CAN network ensure that CAN Tx frame sent message is received by all nodes which requested or no node is received. So the CAN network system ensures the data consistency across all the timings. Hence the multicast and error handling can be ensured.

Bit rate

In CAN number of bits transmitted per second can be different for different application system. But this speed will be uniform across all nodes in the given CAN network system.

Priorities

The CAN IDENTIFIER ensures to get the priority been assigned to each CAN Node, which defines the static priority to access the CAN bus.

Remote Data Request

CAN Node requests the data frame by sending remote frame across the CAN bus. The Identifier is unique for both data frame and remote frame.

Multi-master

CAN Network system ensure the any node can initiates the frame transmission. So whenever the bus is free, node initiates the transfer of data. So the property as Multi-master.

Arbitration

CAN Tx frame starts the frame transmission when the bus is free. Suppose if 2 or more ECU's or units try to send the message on CAN bus, then it is handled by CAN identifier using bitwise arbitration. This ensures that message/information or time instinct is not lost during communication. Data frame has the precedence over the Remote frame, if simultaneously triggered. Every ECU transmits the signal level, this is compared against the CAN bus signal level. If the signal sent from the unit is recessive and observed dominant at the CAN bus level, then the unit looses the arbitration. The unit which has the dominant bit wins the arbitration and continue transmitting the message.

Safety

Safety is the primary concern of data communication protocols, CAN offers error detection, signalling and self-checking these errors. Below are some of the mechanisms implemented to achieve the safety of data transmission.

Error Detection

To detect the errors, following measures have been taken:

- Monitoring the CAN bus level (transmitting unit compares the bit level to be transmitted with the bit level detected on the CAN bus)
- CRC (Cyclic Redundancy Check)
- Bit Stuffing (Consecutive 5 bits)
- Message Frame format Check

Error Signalling and Recovery Time

Data corrupted will be noticed by the CAN node for the error, then such messages are re-transmitted automatically. The time is approximately 29 bits of times, provided if there is no further data corruption.

Fault Confinement

Failure of CAN node is detected, if it seems faulty and CAN protocol switches OFF the respective CAN node. Additionally it distinguishes the short disturbance/error with the permanent errors.

Connections

There no limit for the number of CAN nodes connected to the network system. But practically delay times and electrical loads on the CAN bus line decides the number of CAN nodes in a system.

Single Channel

The CAN bus physically consists of single channel, which carries the bits being transmitted serially. This implementation is not fixed, it might be E.g., single wire (plus ground), two differential wires, optical fibres etc.

Bus values

The CAN bus have two complementary logical values: dominant and recessive signal level. During the simultaneous transmission resulting singal level will be dominant. Ex: In case of Wired-AND logic implementation dominant level will be defined as logical “0” and recessive level will be by logical “1”.

Acknowledgment

CAN receivers will address the data consistency i.e. data received by acknowledging in the CAN bus.

More clearly, you can see below the fig. 3 bit representation of CAN data frame. This figure elaborates on the number of bits been present in each field. Each of these bit fields have significance in its job and responsibility.

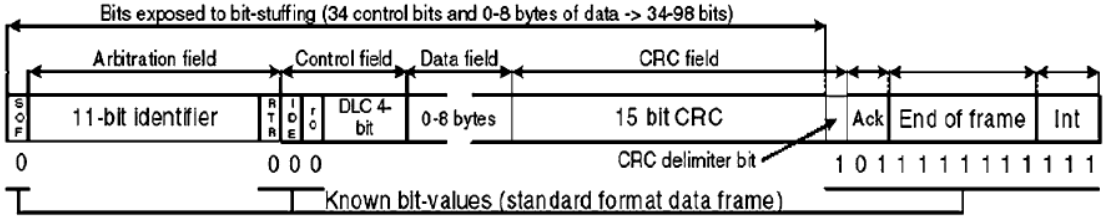


Fig.4 CAN data frame bit representation

3. CAN Tx Design

In this paper the CAN Tx data frame is designed as shown in the below fig. 4. This is simple representation of inputs and outputs in a block structure.

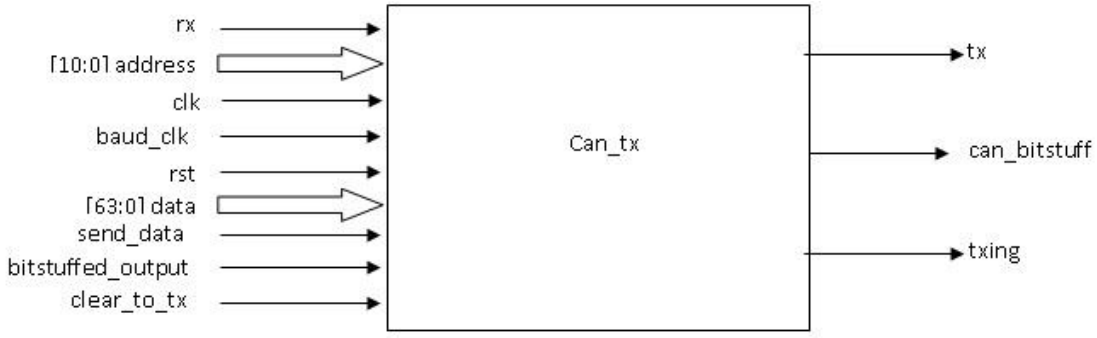


Fig. 5 CAN Tx Design

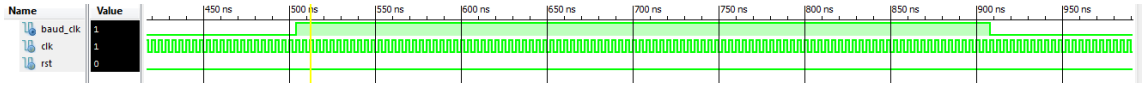
This entire design consists of following sub-blocks, namely

- Baudrate Generation block
- CRC block
- OneShot block

The individual sub-blocks carry different functions and combined effect is the transmission of Tx data frame.

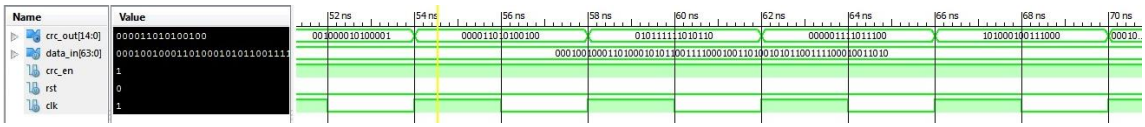
Baudrate Generation block:

This block generates the baudrate pulse out of clock signals. It means it binds the transmission of data frame within this baudrate pulse.



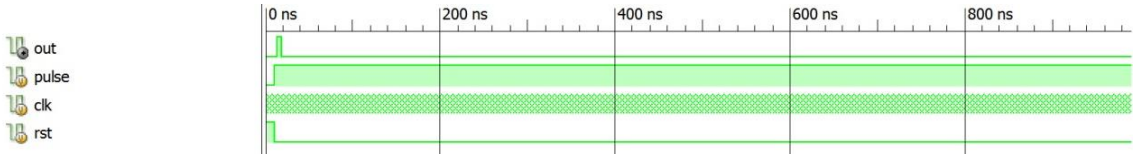
CRC block:

In this block Cyclic Redundancy Check algorithm will be calculated 15-bit CRC will be calculated and it is enclosed within the frame transmission.



OneShot block:

This block triggers the calculation for a single pulse, which is responsible in turn to start the CRC calculation.



The I/O signal description of CAN tx design is shown below in the table 1.

Table 1 Signal Description of CAN tx

Name	Width (bits)	Direction	Description
Rx	1	IN	Receiver enable signal
address	10	IN	CAN Identifier
clk	1	IN	Clock signal
Baud_clk	1	IN	Baud rate clock signal
rst	1	IN	System reset signal
data	64	IN	CAN data/message
Send_data	1	IN	Command to send data
Bitsuffed_output	1	IN	Bitstuffing enable signal
Clear_to_tx	1	IN	Clear the previous transmitted data
Tx	1	OUT	Transmitted data
Can_bitstuff	1	OUTPUT	CAN bitstuff result

4. Simulation Results

The verilog code is simulated in Xilinx 14.7 version software and simulate results are as below for CAN tx data frame. The result can be viewed as a different signals and their bit value representation. Fig. 6 shows the 2 frames being transmitted one after the other, whereas fig. 7 has a single data frame being shown.

Can_bitstuff indicates the bit stuffing is enabled for that entire range of bits whenever the signal is from low to high otherwise it is disabled. Txing will say that transmission of data frame is happening at the HIGH state. Rest all are inputs such as CAN identifier(11-bits length), data(64-bit length) and other control signals. Signal tx will indicate the entire data frame starting from start of frame (SOF) to End of frame(EOF).

Fig.6 CAN data frame transmission

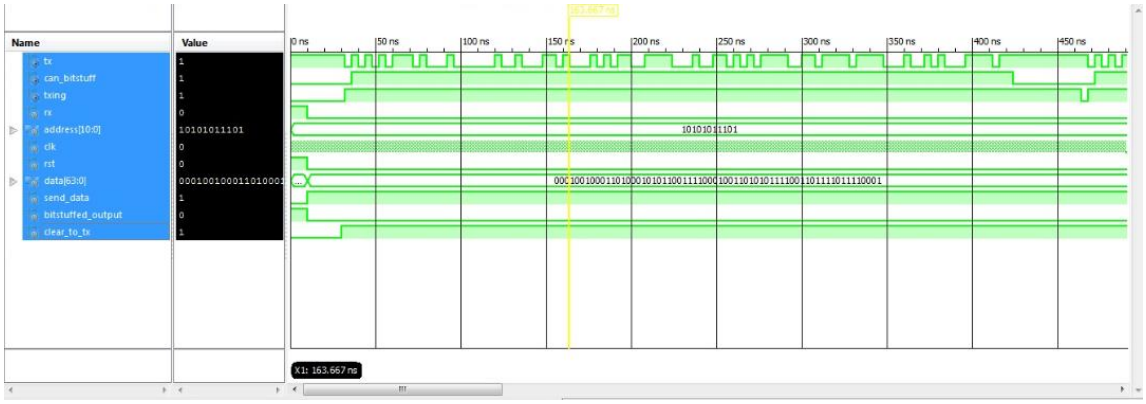


Fig. 7 CAN single data frame

5. Conclusion

In this paper Controller Area Network (CAN) data frame is realized using Xilinx 14.7 version software. Simulation results are verified against the CAN 2.0 specification. Future work can be included to realize the CAN-FD protocol.

REFERENCES

- [1] BOSCH, CAN Specification, 1991, Robert Bosch GmbH, Postfach 50, D-7000 Stuttgart 1
- [2] Jose E. 0. Reges, Edval J P. Santos, A VHDL CAN MODULE FOR SMART SENSORS, in: IEEE, 2008.
- [3] Manjunath Savadatti, Prof. Dr. Meghana Kulkarni, Design and Implementation of IP Core for CANProtocol, IJSTE - International Journal of Science Technology & Engineering, June 2016
- [4] D.Sridhar1, N.Mallika2 and Chirivella Anjaneyulu3, IMPLEMENTATION OF INTER AND INTRA VEHICULAR COMMUNICATION SYSTEM, IJSTE - International Journal of Science Technology & Engineering, Oct 2012
- [5] Robby Andersson, FPGA Design of a Controller for a CAN Controller Linkopings Universitet – 2003
- [6] Vaibhav Bhutada1, Shubhangi Joshi2, Tanuja Zende3, Design and Implementation of CAN Bus Controller on FPGA, IJSTE - International Journal of Science Technology & Engineering, Oct 2017
- [7] Rugved D. Katyarmal, Prof. P. Daigavane, Design of Controller Area Network for SensorNetwork Application using Verilog-HDL, IJSTE - International Journal of Science Technology & Engineering, Oct 2014
- [8] Milind Khanapurkar1, Dr. Preeti. Bajaj2, Sandesh Dahake3 Hemant Wandhare4, Design approach for VHDL and FPGA Implementation of Automotive Black Box using CAN Protocol, IJCSNS International Journal of Computer Science and Network Security - 2008
- [9] Fabiano Costa Carvalho, Ingrid Jansch-Pôrto, Edison Pignaton de Freitas, The TinyCAN: An Optimized CAN Controller IP for FPGA-Based Platforms, 2005 - IEEE
- [10] 1Pardeep Kaur and 2Chanpreet Kaur Toor , An FPGA Implementation of CAN Protocol in Automobiles, AUSTRALIAN JOURNAL OF BASIC AND APPLIED SCIENCES – 2017.

BIOGRAPHIES:

Manjunath Koti Received his B.E. degree in Electronics and communication Engineering From BLDEA's institute of Engineering and technology college Bijapur, Karnataka India in 2007. Presently pursuing in MTech.(Specialization in VLSI Design and Embedded system) from BNM institute of Technology, Bangalore, Karnataka, India from 2019-2021.

Dr. Basavaraj I Neelgar is presently working as an Professor, Department of ECE, BNM institute of Technology Bangalore, Karnataka, India.