iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** VII **Month of publication:** July 2022

DOI: https://doi.org/10.22214/ijraset.2022.45613

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue VII July 2022- Available at www.ijraset.com

CAN FD Data Link Layer with Message Authentication Using Secure Hash Algorithm (SHA-256)

Athira M[1], Dr. Anita R[2], Dr. Yogesh G S[3]
[1]*Student, VLSI & ES, Dept. of ECE, EPCET, VTU Belagavi, Bengaluru, India*
[2]*Associate Professor,* [3]*HOD, Dept. of ECE, EPCET, VTU Belagavi, Bengaluru, India*

Abstract: CAN stands for Controller Area Network. CAN is the major protocol used in Automobile industry. Through the CAN protocol, all the critical electronics devices exchange the messages to smoothly perform the actions within the automobiles/vehicles.
As the technology is the integral part of our life, the vehicles also connected to the networks through the SW clouds or IoT applications. When any system is exposed to network, there exists the scope for cyber attack.As the vehicles use the classical CAN protocol for all the node to node communication using CAN messages, the CAN message field is the interesting part to study & improvise.
If the CAN message field is protected using Cryptography, then any attacks will change the CAN messages. But it is not possible to change the authentication filed as per the changed message since only the authorized nodes will have the proper keys or Hashes (based on the Algorithm used). So, the receiver node will neglect the messages which fails with Authentication. As the receiving node discards the message which fails in the authentication, that node will be protected although any of the messages it received might be compromised.

I. INTRODUCTION

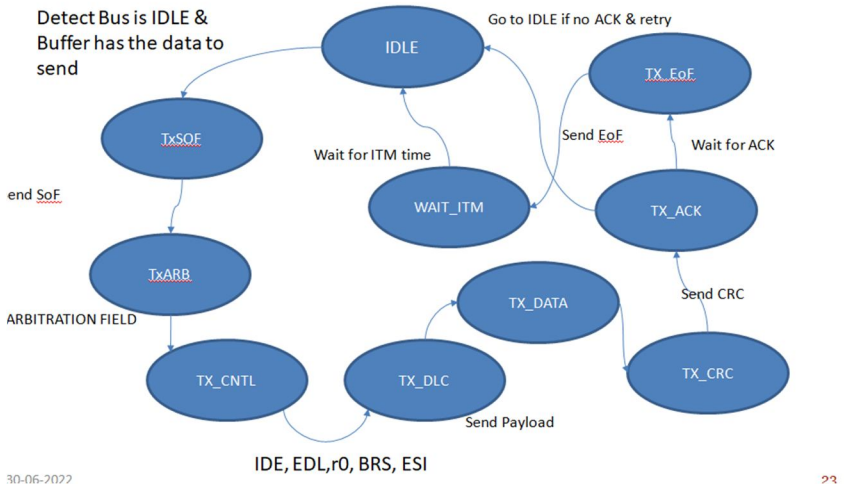
The work aims at developing a CAN controller with Message authentication algorithm. CAN is serial communication protocol used for real-time, safety critical functions inside road vehicles and other controlled applications. It is a multi-master protocol and most widely used inside vehicles.
Below are the main characteristics of the CAN protocol:

1) The maximum bitrate is 1Mbps in the classical CAN bus;
2) high speed CAN bus bitrate can vary from 125kbps to 1Mbps, while low speed CAN bus bitrate from 5kbps to 125kbps;
3) CAN Flexible Data rate (FD), the bitrate is up to 8 Mbps – with a payload size of 8 bytes in Classical CAN and up to 64 bytes in CAN FD;
4) CAN uses different frame types to carry the information among the connected nodes
a) Data Frame: contains the data payload
b) Remote Frame: used to ask for the transmission of data frame with the same identifier from another node on the bus.
c) Error frame indicates that there is an error in the bus and this frame can be used by any node.
d) Overload Frame is used when a node on the bus is too busy to receive data from another node.
5) Also, CAN message has a packet format with Headers and delimiters carrying the data/frames in between. We are going to introduce the Message authentication code (MAC) in the frames to confirm the data integrity.

II. PROPOSED WORK

By analyzing the problem, it is understood that the basic working principle of Automobiles/vehicles is the CAN protocol. As per the default CAN protocol, the Can messages do not carry any security/authentication options. The solution to the cyber attacks could be countered by the message authentication. If at all any information is compromised, the authentication checksums/keys will protect the system from accepting such CAN messages in the receiver node.
Hence, although there is a malicious message intruded in the In-Vehicle network, but it cannot harm the system.
So, the work related to CAN protocol implementation with Message Authentication is targeted.

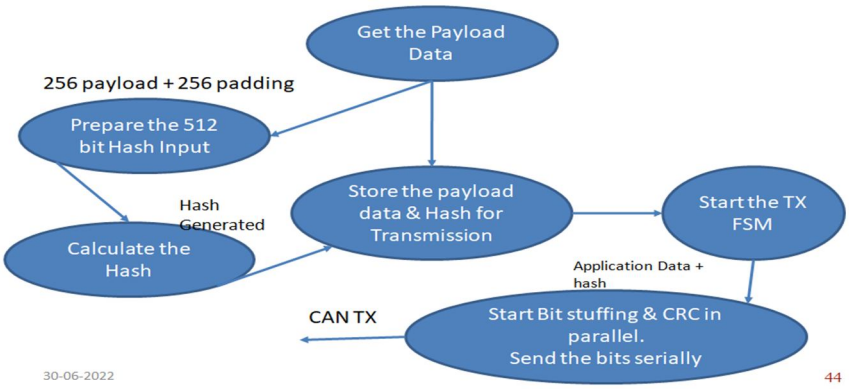
III. ARCHITECTURE



CAN FD Frame Generator

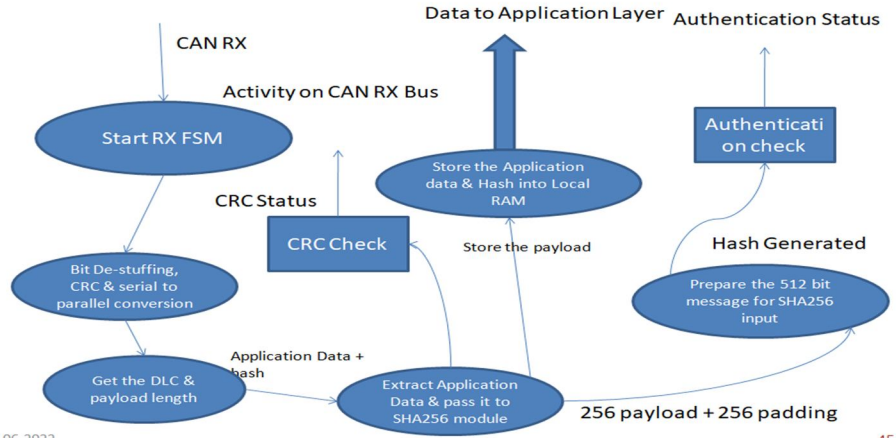
1) *IDLE:* Default state. Whenever BUS is Free & data is ready to be sent, FSM goes to TXSoF.
2) *TXSoF:* Send bit '0' . After that FSM goes to TxARB.
3) *TxARB:* Send ARBITRATION FIELD
4) *TX_CNTL:* Send CONTROL FIELD bits [In CAN FD BASE FORMAT the CONTROL FIELD consists of the bits IDE, EDL,r0, BRS, ESI]
5) *TX_DLC :* Send DATA LENGTH CODE
6) *TX_DATA :* This will send payload data + MAC /Hash data
7) *TX_CRC:* After the data. Send 21 bit CRC.
8) *TX_ACK:* Wait for the ACK bit
9) *TX_EoF :* Send End of frame. After this EoF, FSM goes to IDLE state
   Same state machine is responsible for the Parallel to serial conversion of the bits to transmit over CAN lines.
10) *TX Flow :* A top module is implemented to integrate both TX & RX logics under single module.

On the TX Side, it integrates TX FSM, SHA256, CRC generator & Bit stuffing module along with local RAM to store the intermediate data.Once the Payload (32bytes) is available at the Input RAM, the process starts.Initially the payload data is passed to the SHA256 module to get the 256 bit Hash.Once the Hash is received, the Hash value is also stored in the RAM. FSM sends the EOF & continues to send other fields as per the data link layer message format. In the payload filed, 1st the 32bytes of payload is transmitted serially followed by the 256bits of Hash. For both payload & SHA256 hash,, CRC will be keep updating.Once the last bit of hash is sent over serial, the 21 bit CRC value will be sent followed by EOF.At the output, bit stuffing logic will be always active.



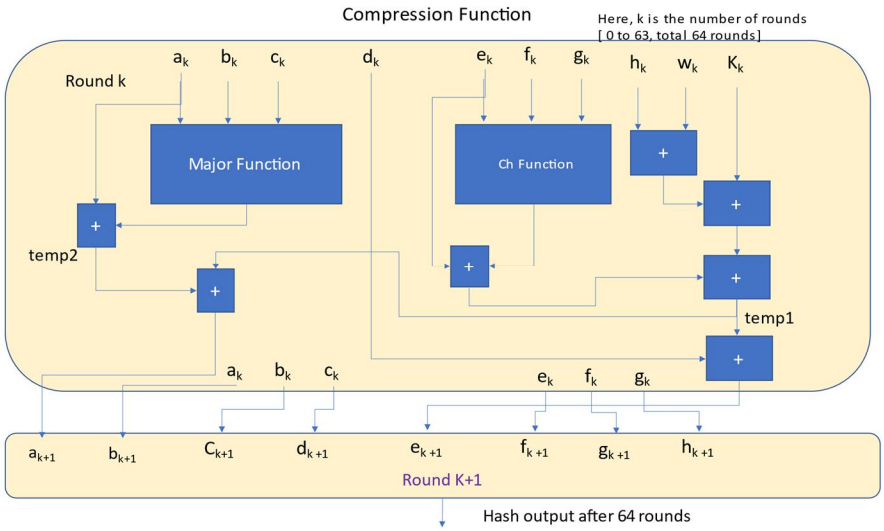
Transmitter Flow Diagram

11) *RX Flow:* On the RX Side, it integrates RX FSM, SHA256, CRC generator & checker & Bit de-stuffing module along with local RAM to store the intermediate data.In the RX side, whenever the FSM detects a SoF condition, the FSM goes to next state & receives the fields as per the CAN message format.If any field is invalid, the error status will be set.When the payload starts, the FSM makes the parallel payload byte & stores it into a RAM.Once all the 32bytes are received, the FSM stores next 32bytes as the hash value.The received payload goes to the SHA256 module once again to generate the hash locallyOnce the RX side generates the hash, it compares with the received hash.If both generated & received hash matches authentication is known. Else, authentication test fails.Parallely, CRC calculation will also happen & received & calculated hash will be compared to check the data integrity of the physical layer.



Receiver Flow Diagram

SHA256 implementation



SAH256 Compression function flow

The hash generated from Verilog code must match with the hash generated by any third party tool with same set of inputs. This ensures that the Verilog module which is written is completely functional.

IV. SIMULATION RESULTS

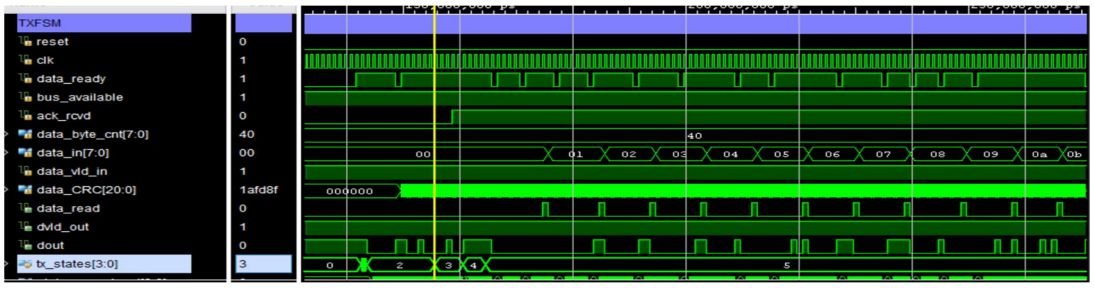
Coding used is Verilog: Standardized as IEEE 1364, is a hardware description language (HDL) used to model electronic systems
Validation & Simulation Results Error case Generation
Inject_error = 1 ; This injects a single bit error in the transmitted data
Inject_error = 0 ; This does not inject any bit error in the transmitted data

```
//inst
    can_top can_top_inst (
        .clock           (clock           ),
        .reset           (reset           ),
        .bus_available   (bus_available   ),
        .data_input      (data_input +1   ),
        .data_vld_input  (data_vld_input  ),
        .data_read_req   (data_read_req   ),
        .dbyte_cnt_in    (dbyte_cnt_in    ),
        .parallel_data   (parallel_data   ),
        .rx_rd_addr      (rx_rd_addr      ),
        .msg_auth_done   (msg_auth_done   ),
        .dbyte_cnt_out   (dbyte_cnt_out   ),
        .crc_error       (crc_error       ),
        .EOF_error       (EOF_error       ),
        .ack_dlmtr_error(ack_dlmtr_error),
        // .inject_error    (1'b1),//Enable if simulation is for Error case, else comment
        .inject_error    (1'b0),//Enable if simulation is for non-Error case, else comment
        .can_txout       (can_txout), //loopback
        .can_rxin        (can_txout)
    );
```

CAN top module instantiation @ Testbench

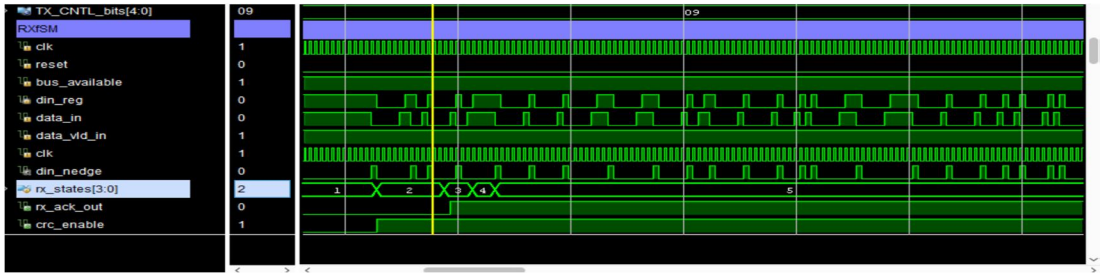
Testbench : Select Can_top_tb as the top module & run the simulation.The testbench will contain a clock generation module & reset generation module.Also, the testbench generates incremental data as the payload dataAdd the signals from TX FSM, RX FSM & signals from relevant logical modules into wave window.Run the simulation for 1msCheck the TX FSM behaviour & RX FSM behaviour

A. TX FSM Signals



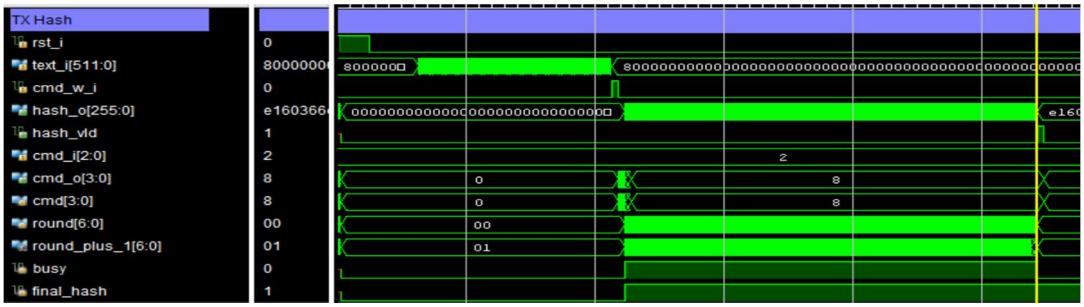
TX FSM signals while conducting Integrated Simulation

B. RX FSM Signals



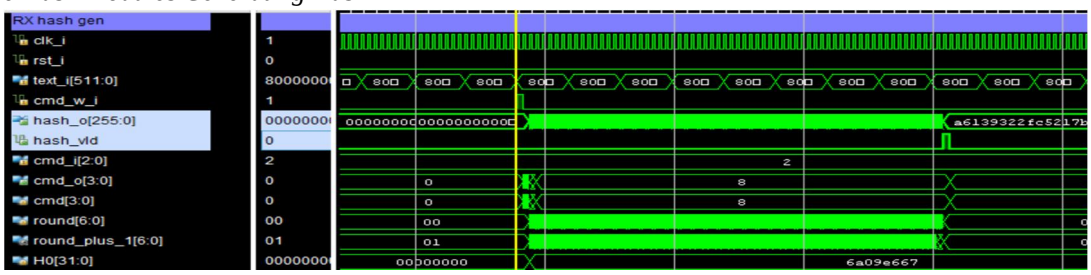
RX FSM signals while conducting Integrated Simulation

C. Tx Side Hash Modules Generating Hash



TX Side Hash module behavior in Simulation

D. Rx Side Hash Modules Generating Hash

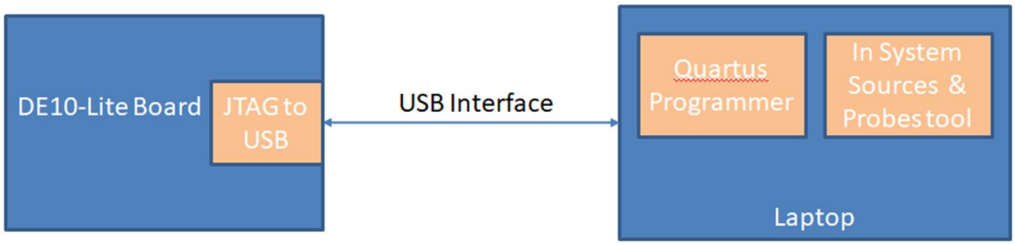


RX Side Hash module behavior in Simulation

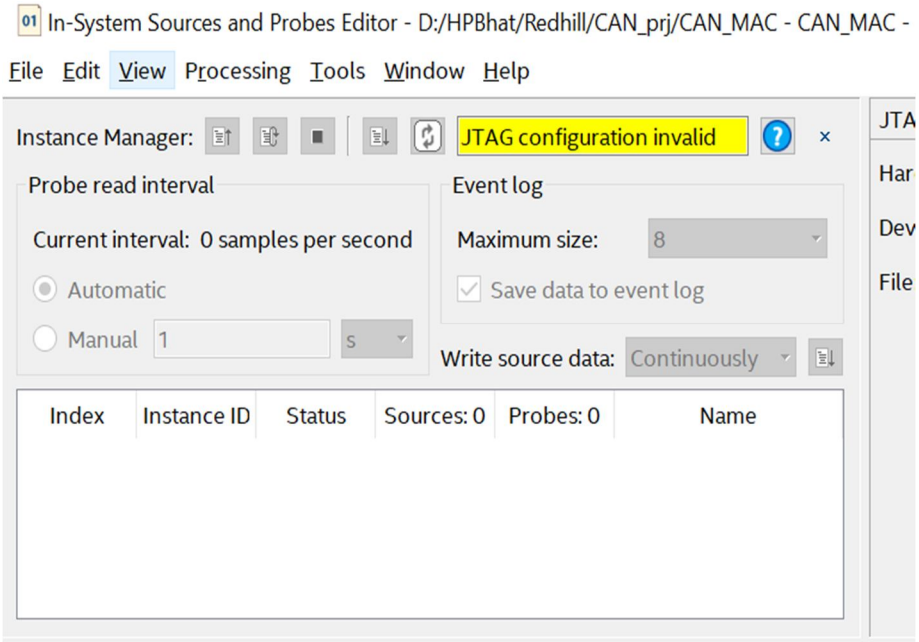
V. CONCLUSION

DE10-Lite Board is used t o validate the design on the on Board MAX 10 FPGA.

The inputs are fed from the In System probes & Sources interface using on board JTAG Interface. Same interface is used to verify the outputs of the modules as well



Hardware Testing Set up



Hardware Testing Set up – Interface to User inputs & output monitoring

In this project, the CAN data link layer is built with Message Authentication Feature. The implemented design is well validated in Simulation as well as on the hardware using MAX 10 FPGA.

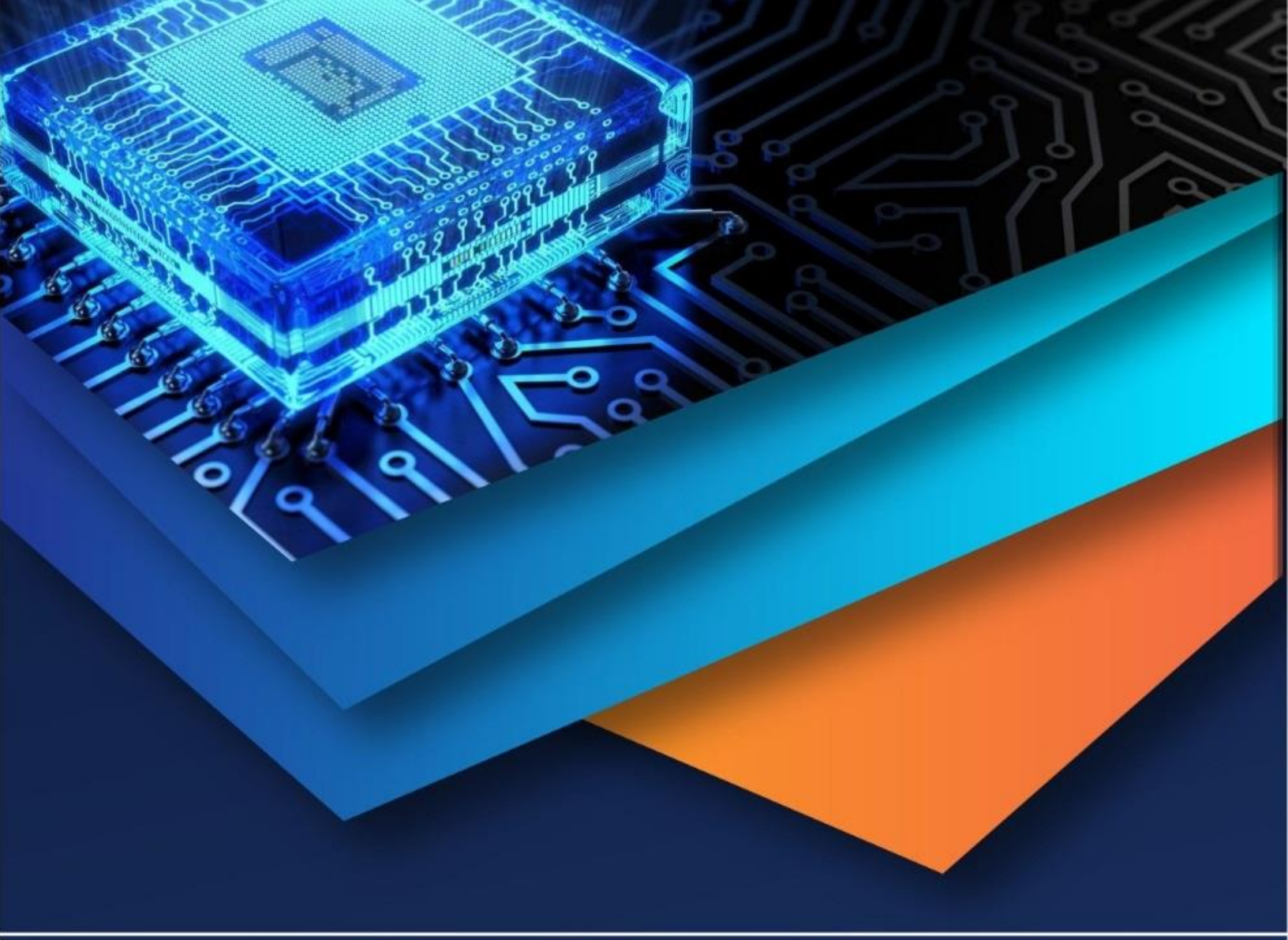
Further the VLSI properties of the design have been checked & the characteristics like resource utilization, Power & Timing analysis has been done.

REFERENCES

[1] Quartus User guide : https://www.intel.com/content/www/us/en/support/programmable/support-resources/design-software/user-guides.html
[2] Modelsim user guide : 1. ModelSim* - Intel® FPGA Edition Simulation Quick-Start (...
[3] DE10-Lite: Boardhttps://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=218&No=1021&PartNo=2
[4] https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=218&No=1021&PartNo=4#contents
[5] SHA256 : SHA-2 - Wikipedia
[6] https://eprint.iacr.org/2011/037.pdf
[7] CAN FD Spec : can_fd_spec.pdf
[8] CYBERATTACKS AND COUNTERMEASURES FOR IN-VEHICLE NETWORKS - arXiv:2004.10781v1 [cs.CR] 22 Apr 2020
[9] Ki Dong Kang, Youngmi Baek, Seonghun Lee, and Sang Hyuk Son. An Attack-Resilient Source Authentication Protocol in Controller Area Network. Proceedings - 2017 ACM/IEEE Symposium on Architectures for Networking and Communications Systems, ANCS 2017, pages 109–118, 2017.
[10] Giampaolo Bella, Pietro Biondi, Gianpiero Costantino, and Ilaria Matteucci. TOUCAN: A proTocol to secUre Controller Area Network. AutoSec 2019 - Proceedings of the ACM Workshop on Automotive Cybersecurity, co-located with CODASPY 2019, pages 3–8, 2019.
[11] Yujing Wu, Yeon Jin Kim, Zheyan Piao, Jin Gyun Chung, and Yong En Kim. Security protocol for controller area network using ECANDC compression algorithm. ICSPCC 2016 - IEEE International Conference on Signal Processing, Communications and Computing, Conference Proceedings, pages 1–4, 2016.
[12] Samuel Woo, Hyo Jin Jo, In Seok Kim, and Dong Hoon Lee. A practical security architecture for in-vehicle CAN-FD. IEEE Transactions on Intelligent Transportation Systems, 17(8):2248–2261, 2016.

Author's biography

Athira M is an under-graduate in Electrical and Electronics Engineering and is currently undergoing post graduation in VLSI and ES at East Point College of Engineering and Technology, Bangalore
Dr. Anita R- Associate Professor, Dept. of ECE, EPCET, Bengaluru.
Dr. Yogesh G S- Professor and HOD, Dept. of ECE, EPCET, Bengaluru



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY