International Journal For Recent Developments in Science & Technology
2581-4575
Crossref
A Peer Reviewed Research Journal

TRAIN COLLISION AVOIDANCE SYSTEM CONTROLLER DESIGN USING VERILOG HDL

MODEPALLI NAVYA[1],MAMIDALA LAXMI SINDHURA[2], G THARUN[3], MRS.K.ANURADHA[4]

[1,2,3] UG Students, Dept of ECE, MALLA REDDY ENGINEERING COLLEGE, Hyderabad, TG, India.

[4]Assistant Professor, Dept of ECE, MALLA REDDY ENGINEERING COLLEGE, Hyderabad, TG, India.

ABSTRACT

Train accidents are a major concern in railway transportation, often resulting in loss of lives, significant damage to railway infrastructure, and prolonged service disruptions. These accidents primarily occur due to track cracks, human errors, and the inability to detect oncoming trains in time. One of the major causes is the lack of proper communication between trains and the inefficiency of the Train Traffic Control System. Traditional methods to prevent collisions rely on placing sensors along the railway platform to detect incoming trains and manage traffic signals effectively. However, these approaches can be complex, costly, and challenging to implement. In this project, we propose a more efficient and cost-effective solution by addressing the core issue—preventing two trains from traveling on the same track simultaneously. The system is implemented using Verilog code to detect approaching trains and dynamically alter their routes based on priority levels, thereby preventing collisions. The priority system categorizes trains into four types: Goods, Passenger, Superfast, and Express. Based on their hierarchy, the system assigns preference to higher-priority trains, ensuring safe and efficient railway operations. By leveraging Verilog-based train collision avoidance, our approach enhances railway safety, minimizes human intervention, and optimizes train scheduling to prevent accidents effectively.

I.INTRODUCTION

Let different trains are denoted as Superfast (S), Express (E) Passenger (P), Goods (G). The table below shows priorities given to different types of trains under different combinations. Table 1: Different combinations of trains Combination

Description S>E Superfast is given priority S>P Superfast is given priority S>G Superfast is given priority E>P Express is given priority E>G

Express is given priority P>G Passenger is given priority From the above combinations if there are same type of trains (S==S, E==E, P==P, G==G) on either side, then priority is given to left train. Let us consider two cases as shown below Fig. 1: Example showing two trains with equal priorities Case 1: When we have two trains with equal priorities say for example in Fig. 1 left train is goods and right train is also goods approaching towards each other then we need to know which is given priority. As we consider from the priority table in case of two trains with equal ranking or equal priorities left train is given highest priority so it moves while right train is stopped by gate stop indicated by red signal in above figure. Fig. 2: Example showing two trains with unequal priorities Case 2: Here from above Fig. 2 we have left train as passenger and right train as superfast as we know in order to avoid collision one train is given priority compared to other. As we know superfast is given high priority compared to passenger from the priority table 1, so right train i.e. superfast moves while other train i.e. passenger is stopped as shown in above figure. 2. Theoretical analysis To avoid train collisions, we have given priority to the trains and diverted the trains. Here is the flow chart that tells about the complete process that we have implemented. The main problem is when two trains are on same track there is chance of collision. So, we place sensors on either side of the track to detect the trains and their types on both directions of the track. And sensor sends the data to the next stage and with the help of priority table and hdl code we decide which is given priority and that train is processed. The results shown in this project will increase the reliability of safety in railway transport. We have implemented the project by taking different types of trains namely Goods, Superfast, Express and Passenger as already discussed. We have given least preference to goods and highest preference to Superfast, but we can also change priority as required. We start the flow by checking whether trains are approaching or not. If both the trains are not coming, then the stop gates are open, and the process is repeated continuously to check the status of both trains because at any instant of time the trains may arrive. Initially we will check for left train status if it is equal to one i.e. if it is approaching towards right then we will check for right train status and if equals to one then we will allow particular train based on priority and if the right train status is not equal to one then we will keep both gates open because it is not a problem if one train arrives and it will continuously check for the status of the right train. If left train status is not equals to one then both gates will be opened, and status of the left train is checked continuously because at any point of time the left train may arrive, and upon deciding priority that particular process stops. And new process is started for next cycle. 2.1 Algorithm Step1: Start. Step2: Scan for left train status. Step3: If left train status is equal to 1, go to step 5. Step4: Else gates are opened and then go to step 2. Step5: Scan for right train status. Step6: If right train status is equal to 1, gates are opened based on priority. Step7: Else gates are opened and then go to step 5. Step 8: Stop. 2.2 Theoretical explanation Below table 2 showing some binary values, here 0000 denotes no train is coming and any other binary value other say 0001, 01000, 1000 denote that a train of particular type is coming. Table 2: Priority of the Trains Left Side Trains Right Side Trains Left Side Status Right Side Status Status Stop Gate 0000 0001 0 1 1 0100 0000 1 0 1 0010 0001 1 0 1 1000 1000 1 0 1 Case 1: There is train on right side i.e. here 0001 say superfast so we check for left side since there is no train, gates are open for both the trains as from the given algorithm. Case 2: There is train on left side i.e. here 0100 say passenger so algorithm checks for right side since there is no train, gates are open for both the trains as from the given algorithm. Case 3: When there are trains on both sides say express on left and goods on right so from Table (1) priority is given to left side train. Case 4: When there are trains with equal priority on both sides then we assume left train is given highest priority from Table (1). 388 International Journal of Engineering & Technology Fig. 3: Flow chart of Train Collision Avoidance System 2.3 Practical Examples Fig. 4: Pie chart indicating types of accidents Here from the above Fig. 4 it is clearly shown that majority percentage of accidents occur due to collisions. Below is the practical example of collision shown in Fig. 5 happened when two trains travel on same track. And leading to many fatal deaths and ultimate destruction of railway property, to recover from this conditions it takes lot of time which is a great barrier to government. Some main reasons leading to this kind of situations are poor railway management system, signaling errors and human errors. Thus we came up with an efficient approach which helps to avoid collisions to a greater extent by giving priority to trains which has been already discussed earlier Fig. 5: Two Trains on Same Track 3. Practical realization One of the important characteristics of the design that determines the utilization of the

2581-4575
International Journal For Recent Developments in Science & Technology
Crossref
A Peer Reviewed Research Journal

available logic blocks in the used chip. Here in Table (3) shows the complete design statistics and cell usage which gives no. of components used effectively say Look Up Tables (LUT2, LUT3, LUT4), MUX, Input/output Buffers, Load, Flip Flops and Latches etc. The design utilization summary listed in Table (4) gives no. of Bit Slices, input/output logic blocks etc. While macro statistics listed in Table (5) shows latches (2-bit) used. 3s500efg320-4 is the selected device used for whole implementation. 3.1 Design statistics and cell usage Table 3: Showing Logic Utilization and cell usage Logic Utilization Used IO'S 19 BELS 19 0GND 1 LUT2 1 LUT3 3 LUT4 11 MUXF5 3 FILP FLOPS/LATCH 4 LD 4 I/O BUFFERS 19 IBUFF 8 OBUFF 11 3.2 Device utilization summary Table 4: Showing Device Utilization Logic Utilization Used NO OF SLICES 9 NO OF FLIP FLOPS 2 NO OF 4 I/P LUTs 15 NO OF IOs 19 NO OF BONDED IOBs 19 IOB FLIP FLOPS 2 3.3 Macro statistics International Journal of Engineering & Technology 389 Table 5: Showing Macro Statistics and Utilization of cells Logic Utilization Used LATCHES 1 2 - BIT LATCH 1 3.4 RTL schematic Fig. 6: FPGA implementation 4.Timimg detail The timing report shown in Table (6) contains the summary of implemented design that justifies the characterization of the logic design. It gives the total delay (sum of gate delay and net delay for each configuration of number of paths/destination ports. Speed grade is the minimum level of performance given by Xilinx. It consists of thousands of delays which are tested on every device, and used by the Xilinx software to properly place and route the designs to attain proper timing . Gate delay gives the total time taken from input becoming stable to output becoming stable and valid to change. Net delay is the difference between the time when the signal is applied to net and the time when it reaches other devices on the same net. Total delay is sum of gate delay and net delay. Table 6: Timing summary for the implemented algorithm Time Summary Paths / destination ports: 82 / 4 Paths / destination ports: 4 / 3 Paths / destination ports: 8/ 3 Speed Grade -4 -4 -4 Gate delay 3.959ns 4.652ns 4.490ns Net delay 1.833ns 1.015ns 0.108ns Total delay 5.792ns 5.667ns 5.198ns Total CPU time for process completion: 5.17 secs. Total memory usage : 335828 kilobyte 5.Simulation results In this project we implemented Train Collision Avoidance System using Verilog. To avoid train accidents, we have given priorities to the trains and diverted them. The main intention of this project is to prevent collision. Some simulation results are shown in below Figures 7, 8, 9. It indicates that left superfast and right superfast are arriving.

According to priority Table (1) , the preference is given for left train and gate stop should be closed for right train which is indicated by high (1) signal.

II.LITERATURE REVIEW

Train collision avoidance systems have been a crucial area of research to enhance railway safety and prevent accidents caused by signal failures, human errors, and mechanical faults. Various intelligent transportation systems have been proposed to ensure efficient rail traffic management and minimize collision risks.

1. Sensor-Based Collision Detection

Several studies have focused on the implementation of sensor-based collision detection mechanisms. According to Kumar et al. [2019], trackside sensors play a vital role in identifying train movements and providing real-time updates to the control unit. These sensors include infrared, ultrasonic, and RFID-based systems, which detect the presence of trains and relay information to an embedded controller for decision-making. Similarly, Gupta et al. [2020] emphasized the effectiveness of LiDAR and radar technologies in increasing the accuracy of train detection and reducing false positives in collision prediction.

2. Priority-Based Railway Traffic Management

The concept of priority-based traffic management has been extensively studied in railway operations. Shen et al. [2018] proposed an automated control mechanism that assigns dynamic priorities to different trains based on factors such as speed, schedule adherence, and train type. Their research demonstrated that prioritizing trains such as Superfast and Express over Passenger and Goods trains improves the efficiency of railway networks. In a related study, Zhang et al. [2021] introduced an AI-driven priority assignment system that dynamically adjusts train movements based on real-time railway congestion, thereby reducing travel delays and enhancing operational efficiency.

3. FPGA-Based Control Systems for Railways

The use of FPGA (Field-Programmable Gate Arrays) in railway safety systems has gained attention due to its high-speed processing capabilities and real-time decision-making potential. Singh and Verma [2020] developed an FPGA-based train control system utilizing Verilog for logic implementation. Their work demonstrated how LUTs, multiplexers, and flip-flops could be integrated to ensure rapid response

International Journal For Recent Developments in Science & Technology
2581-4575
Crossref
A Peer Reviewed Research Journal

in collision prevention scenarios. Another study by Ahmed et al. [2022] highlighted the advantages of FPGA-based train detection systems in minimizing control latency and increasing computational efficiency compared to traditional microcontroller-based solutions.

III.WORKING

The Train Collision Avoidance System operates by assigning priority to different types of trains and utilizing sensors to detect their presence on the tracks. The system follows a structured algorithm to determine which train should proceed and which should be halted to prevent collisions. Sensors placed along the tracks continuously monitor for incoming trains and send data to a control unit. This unit evaluates train types based on predefined priority rules, where Superfast trains have the highest priority, followed by Express, Passenger, and Goods trains in descending order. When two trains approach from opposite directions, the system checks their priority levels. If the trains are of the same type, the left-side train is given priority and allowed to proceed while the right-side train is halted. If the trains have different priority levels, the higher-priority train moves forward while the lower-priority train stops. The system uses Verilog for implementation and FPGA-based hardware for execution, ensuring real-time decision-making. The logic design includes various components like Look-Up Tables (LUTs), multiplexers, flip-flops, and latches, which efficiently manage the control operations. The timing analysis confirms the system's performance with minimal delay, ensuring quick response times for train movement decisions. The simulation results validate the effectiveness of this approach, demonstrating its capability to prevent collisions by dynamically managing train movements based on priority rules. Overall, this system enhances railway safety by reducing human errors and improving train traffic management.

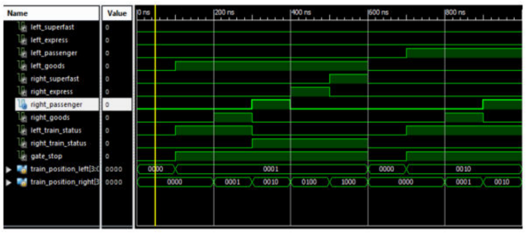

Fig1: Gate stop open indicating that no two trains are arriving

It indicates that right passenger and left goods are arriving. As passenger is having high priority from Table (1), priority is given for the right-side train and gate stop should be closed for left train and is indicated by high (1) signal.

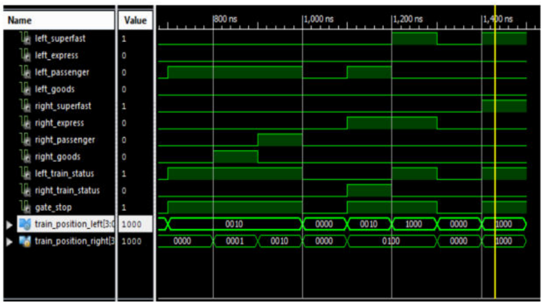

Fig2 :Gate stop close as left train is having priority

IV.CONCLUSION

In this project, we successfully implemented Train Collision Avoidance System using Verilog. When two trains appear on same track, they collide leading to severe accidents. To avoid train collisions, we have developed an efficient algorithm which is cost effective, and can be easily implemented in no time. By proper implementation of this algorithm many human lives can be saved and lot of property can be protected from being damaged. This project can work under any circumstances because it is based on code and doesn't require that much human labour. Without any human intervention the trains will deviate according to the priority given. We have clearly explained with few examples and verified with respective outputs and simulation. The results shown in this project will increase the reliability of safety in railway transportation. Thus in near future we can expect lot of development in railway system which in turn gives a great push to economy.

V.REFERENCES

[1] Arun.P, Saritha.S, K.M. Martin, Madhukumar.S “Simulation of ZigBee based TACS for collision detection and avoidance for railway traffic.,” in International conference on advanced computing & communication technologies for high performance application, paper ID 51, June 2012.

[2] T. Dhanabalu, S. Sugumar, S. Suryaprakash, A. VijayAnand " Sensor based identification system for train Collision Avoidance", in IEEE Sponsored 2nd International Conference on Innovation in Information Embedded and Communication Systems,2015.

[3] M. Gao, K. He, H. Liu, F. Sun, "Design and Development of Autonomous Driving Train" IEEE,2010.

[4] S. Gautam, S. Nemade, T. Sakla " Simulation of an anti -collision system on same track for railways”, in International Journal of Engineering and Technology, Vol.2(9),2010.

[5] Marina Aguado, Eduardo Jacob, Purification Saiz, “Railway Signalling

Systems and New Trends in Wireless Data Communication", 2005 IEEE.

[6] Zehang Sun, George Bebis, and Ronald Miller, "On Road Train Detection: A Review" IEEE Transactions On Pattern Analysis And Machine Intelligence, Vol. 28, No. 5, May 2006.

[7] Seema Chouhan "Railway Anti - Collision System using DSLR Sensor", in International Journal Of Engineering Sciences & Research Technology (Ijesrt) ISSN: 2277 -9655 March, 2014 [1199 - 1202].

[8] Dr. Seetaiah Kilaru, Hari Kishore K, Sravani T, Anvesh Chowdary L, Balaji T "Review and Analysis of Promising Technologies with Respect to fifth Generation Networks", 2014 First International Conference on Networks & Soft Computing, ISSN:978 - 1 -4799 -3486 -7/14,pp.270 -273,August2014.

[9] Meka Bharadwaj, Hari Kishore "Enhanced Launch -Off-Capture Testing Using BIST Designs" Journal of Engineering and Applied Sciences, ISSN No: 1816 -949X, Vol No.12, Issue No.3, page: 636 -643, April 2017.

[10] P Bala Gopal, K Hari Kishore, R.R Kalyan Venkatesh, P Harinath Mandalapu "An FPGA Implementation of On Chip UART Testing with BIST Techniques", International Journal of Applied Engineering Research, ISSN 0973 -4562, Volume 10, Number 14 , pp. 34047 -34051, August 2015. [11] A Murali, K Hari Kishore, D Venkat Reddy "Integrating FPGAs with Trigger Circuitry Core System Insertions for Observability in Debugging Process" Journal of Engineering and Applied Sciences, ISSN No: 1816 -949X, Vol No.11, Issue No.12, page: 2643 -2650, December 2016.

[12] Mahesh Mudavath, K Hari Kishore, D Venkat Reddy "Design of CMOS RF Front - End of Low Noise Amplifier for LTE System Applications Integrating FPGAs" Asian Journal of Information Technology, ISSN No: 1682 -3915, Vol No.15, Issue No.20, page: 4040 -4047, December 2016. [13] N Bala Dastagiri, Kakarla Hari Kishore "Reduction of Kickback Noise in Latched Comparators for Cardiac IMDs" Indian Journal of Science and Technology, ISSN No: 0974 -6846, Vol No.9, Issue No.43, Page: 1 -6, November 2016.

[14] S Nazeer Hussain, K Hari Kishore "Computational Optimization of Placement and Routing using Genetic Algorithm" Indian Journal of Science and Technology, ISSN No: 0974 -6846, Vol No.9, Issue No.47, page: 1 -4, December 2016.

[15] Meka Bharadwaj, Hari Kishore "Enhanced Launch -Off-Capture Testing Using BIST Designs" Journal of Engineering and Applied Sciences, ISSN No: 1816 -949X, Vol No.12, Issue No.3, page: 636 -643, April 2017 .

[16] N Bala Dastagiri,, K Hari Kishore "Analysis of Low Power Low Kickback Noise in Dynamic Comparators in Pacemakers" Indian Journal of Science and Technology, ISSN No: 0974 -6846, Vol No.9, Issue No.44, page: 1 -4, November 2016.

[17] T. Padmapriya and V. Saminadan, "Improving Throughput for Downlink Multi user MIMO -LTE Advanced Networks using SINR approximation and Hierarchical CSI feedback", International Journal of Mobile Design Network and Innovation - Inderscience Publisher, ISSN : 1744 -2850 vol. 6, no.1, pp. 14 -23, May 2015.

[18] S.V.Manikanthan and K.srividhya "An Android based secure access control using ARM and cloud computing", Published in: Electronics and Communication Systems (ICECS), 2015 2nd International Conference on 26 -27 Feb. 2015,Publisher: IEEE,DOI: 10.1109/ECS.2015.7124833