International Journal of Engineering & Technology, 7 (2.8) (2018) 386-391

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research Paper

A VLSI implementation of train collision avoidance system using Verilog HDL

Fazal Noorbasha, K. Hari Kishore, P. Phani Sarad, A. Renuka, SK . Meera Mohiddin, K. Jagadeesh Babu, B V S. Phanindra, M. Manasa

Department of ECE, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh, India-522502
**Corresponding Author Email: [1]fazalnoorbasha@kluniversity.in*

Abstract

Now a days we see many train accidents that occur in railways. These accidents occur mainly due to cracks in the track, human errors and not identifying the opposite train at the right time. When the train meets with the accident lot of people lose their lives and huge amount of railway property is destroyed and it also takes lot of time to hold back to the normal situations. Most of the accidents happen due to human error and due to lack of communication between the trains and irregularity of Train Traffic Control System. Normally to prevent these accidents we place sensors on either side of the platform to identify the train at right time and to receive traffic signals at the platform properly. Here we came with some different approach which is easy to manage and implement and cost effective. Normally collision occurs when two trains approaching in opposite directions on same track. So, if we manage to prevent two trains travel on the same track then collision can be avoided. Here in this project we have implemented Verilog code to solve this problem. The purpose of this project is to write a Verilog code to detect the opposite train and deviate the train based on priority of the trains thus avoiding collision. In this project we have chosen four different types of trains namely Goods, Passenger, Superfast, Express and we have implemented train collision avoidance using Verilog code by giving priority to each type of train and preference is given to one train to avoid collision.

Keywords: *Collision avoidance, Sensors, Train Traffic Control System (TTCS), Verilog, FPGA.*

1. Introduction

Electronic devices play an important role in our daily life. Each of us in one or other way are related directly or indirectly to electronics. Electronics deals with electric circuits and electronic devices. Electronic devices are components that control the flow of electric current through them for information processing thus can be used in wide variety of applications that help man in day to day life. Today electronics had gained much more importance due to its variety of applications in various industries. These electronic devices mostly perform functions like control, amplification and rectification [1,2]. In the last few years electronics have witnessed a great revolution in the industry than any other technology in invention. The current trending technologies include Verilog, VHDL, Mems and Nems. As we know most of the people travel by trains as it is easiest and cheapest way of transport. Annually 12 million people travel by train all over the globe but sadly neary 10% people meet with train accidents leading to severe injuires and fatal deaths [3,4].

Every hour on an average at least 8-10 persons die in train accidents and at least at least 2-3 million people are affected seriously which was shown from the recent survey. It is also estimated that the medical bill, damaged property due to accidents and other nearly cost up to 2-3 percent of world's gross domestic product. Train accidents may occur due to cracks on railway track or due to short circuiting or ineffective Train Traffic Control System that is failing to find the train approaching in opposite direction on the same track leading to collision. Recently most of the train accidents occur due to train collision [5,6].

Due to these accidents this global world is facing many fatal deaths [7]. Here we have implemented on of the best and easy solution to avoid such collisions. Earlier there are many implementations like anti collision train system etc. which is not effective all the time and maintaining cost is high. And on road train detection is one of the recent implementation which is not up to mark as it is complex. And one of the finest implementation is railway signalling using wireless communications which is efficient and effective and works fine. But in our project, we have decided to find an effective solution to avoid the collisions. In our project we are implementing the train collision avoidance system using Verilog programming language. It is a Hardware Description Language (HDL) which is used for modelling electronic systems. It is like other programming languages and mostly similar to C programming language, so it is easy to learn for beginners. To implement this program, we use software called XILINX ISE (Integrated Synthesis Environment). It is a software tool developed by Xilinx for synthesizing and analyzing of HDL designs. The software is also used to synthesize the designs, performance timing analysis, examining the RTL designs and configuring the target device with programmer. It is primarily used for circuit synthesis and its design used for System-level testing. The other components that comes with Xilinx ISE include Embedded Development Kit (EDK), Software Development Kit (SDK) and a chip scope probe.

Generally, when a single train is moving there is no point of collision. The problem arises when two train approaching towards each other on a single track. In our project we worked on identifying the status of the opposite train and processing only one train based upon their importance or priority. When two trains come on same track there comes the chance of a collision. To avoid these accidents, we allow only one train based upon their priority. Here in this project we had considered 4 types of trains on each side i.e. Right side and left side . The trains are categorized as left superfast, left express, left passenger, left goods, right superfast, right express, right passenger, right goods. And priority is given to a train see from Table1.

Copyright © 2018 Authors. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1.1 Priority Table

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

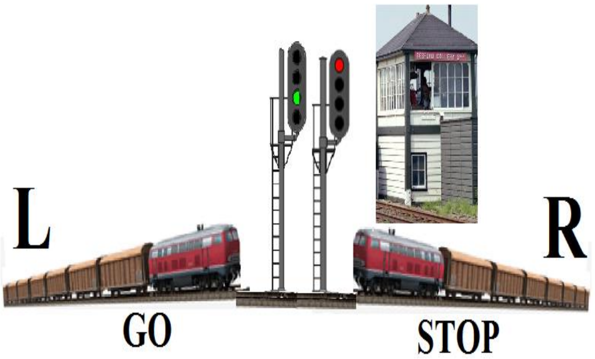

Fig. 1: Example showing two trains with equal priorities

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.

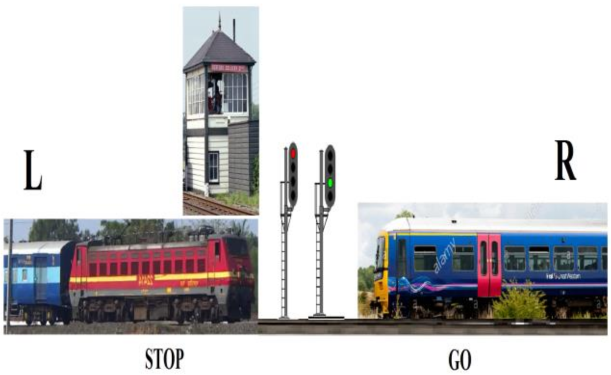

Fig. 2: Example showing two trains with unequal priorities

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).

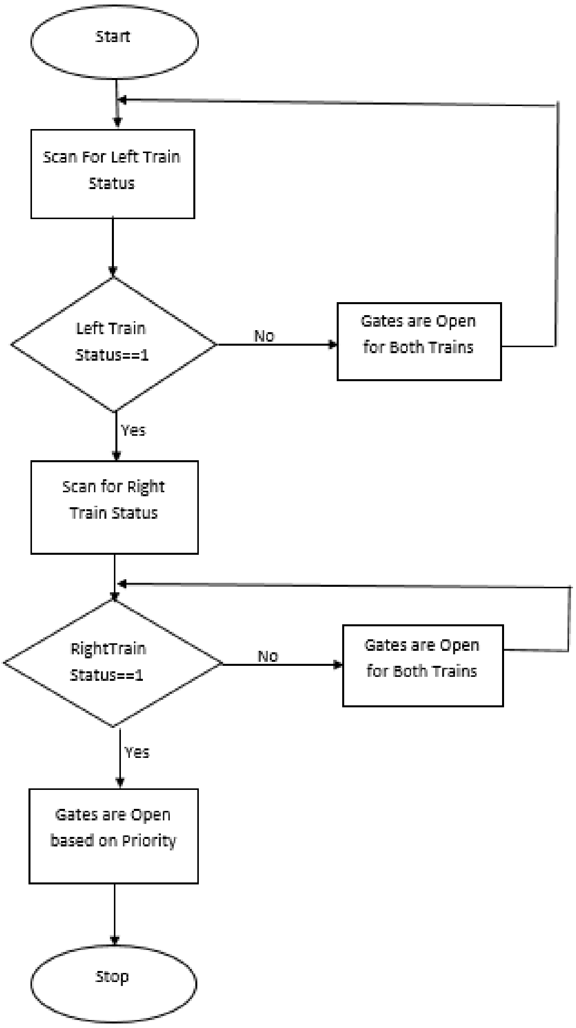

Fig. 3: Flow chart of Train Collision Avoidance System

2.3 Practical Examples

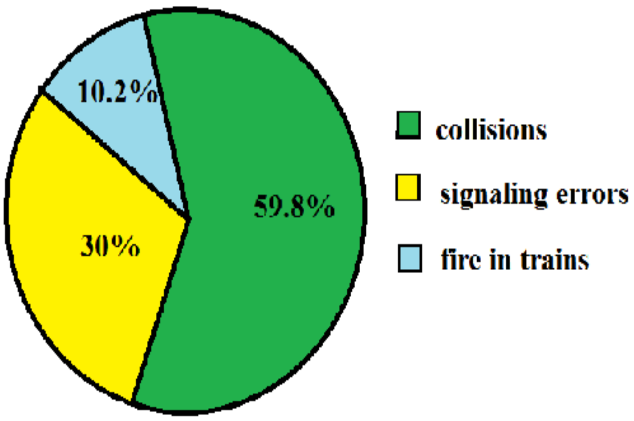

Fig. 4: Pie chart indicating types of accidents

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 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

Table 5: Showing Macro Statistics and Utilization of cells

Logic Utilization	Used
LATCHES	1
2 - BIT LATCH	1

3.4 RTL schematic

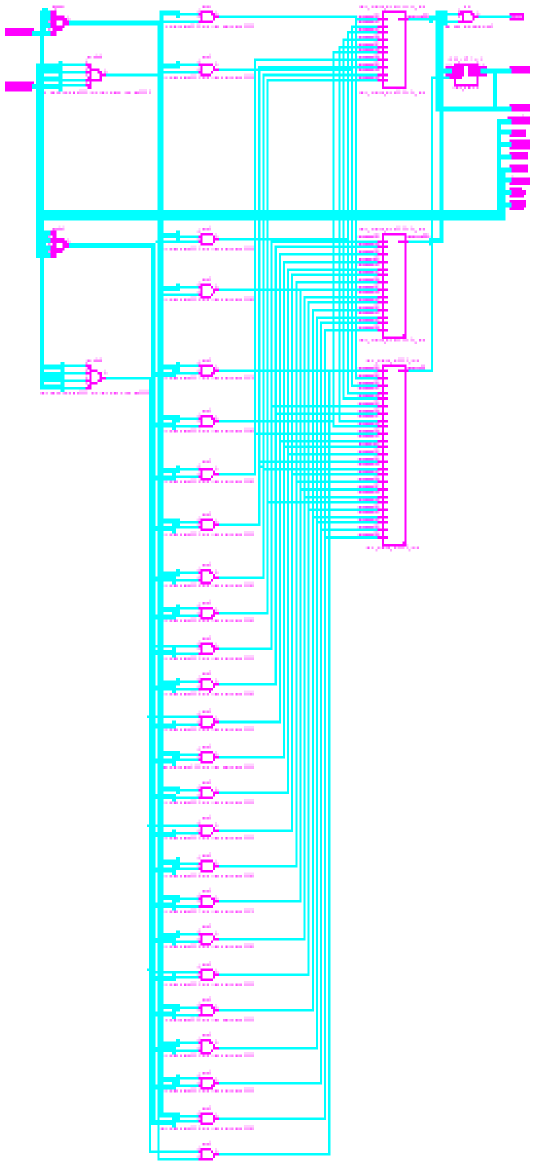

Fig. 6: FPGA implementation

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.

6. 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.

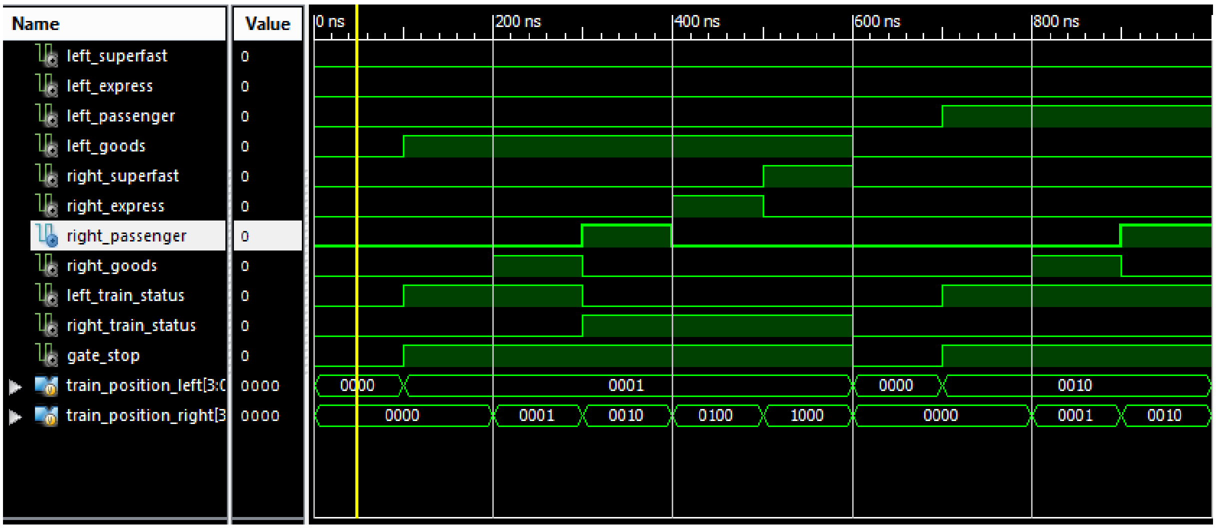

Fig. 7 : Gate stop open indicating that no two trains are arriving

It indicates that no train is arriving on either side of the track so that gate stop should be opened and is indicated by low (0) signal.

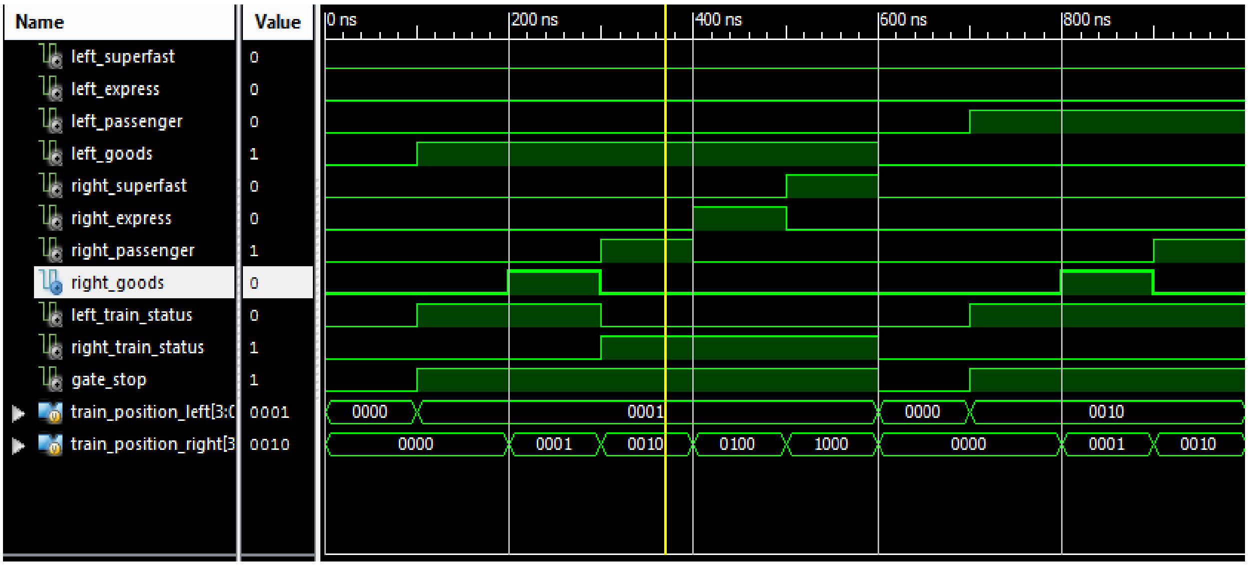

Fig. 8: Gate stop close as right train is having priority

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.

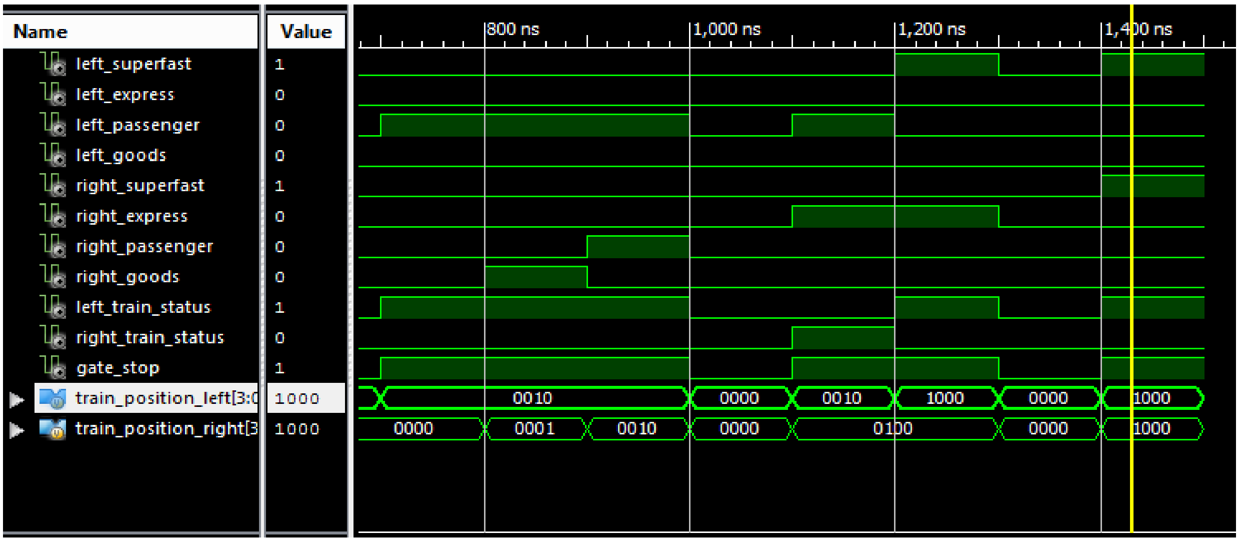

Fig. 9: Gate stop close as left train is having priority

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.