Railway Signalling and Blocking System

Introduction

Indian Railways, one of the largest railway networks in the world, relies on a complex signalling and blocking system to manage train operations safely and efficiently. The system plays a critical role in ensuring that trains run without collisions, that tracks are optimally utilized, and that passengers receive timely services. This report presents a simplified simulation of the signalling and blocking system employed by Indian Railways, focusing on stop stations and the integration of loop lines within these stations.

Indian Railway Signalling and Blocking System Overview

The Indian Railway signalling and blocking system is essential for managing the movement of trains across the network. It ensures that no two trains occupy the same section of track at the same time and provides a clear mechanism for train operators to follow. The system includes the following components:

Block Sections: A block section is a defined section of track that allows only one train at a time to occupy it, ensuring safety between trains. Block sections are typically between stations, and they are crucial in preventing train collisions.

Signals: The signalling system primarily uses colour light signals: Red, Yellow, and Green. These signals help train operators understand whether to stop, proceed with caution, or move ahead at full speed.

• **Track Circuiting**: Track circuits are used to detect the presence of trains on specific track sections. When a train occupies a section, the signal turns **Red**. If the track is clear, the signal turns **Green**.

• **Stop Stations and Loop Lines**: At busy stations, **loop lines** are integrated to allow trains to pass or wait, ensuring the main track flow remains uninterrupted. At these stations, trains may either proceed on the main track or be diverted to the loop line for passing.

Track Circuiting and Signal Control: The system detects when a train is present on the track through track circuits. When a train is detected, the signal turns Red to indicate that the train must stop. Once the train moves out of the section, the signal is set to Green, allowing the next train to proceed.

Stop Station and Loop Line Management: At stop stations, trains may need to wait for clearance to pass or may be diverted to the loop line if the station is occupied. The loop line allows trains to pass the stop station without occupying the main track, thereby reducing congestion.

- If a train approaches a stop station, it checks if the station is clear. If the station is occupied, the train can either wait at the main track or be diverted to the loop line if available.
- Once the train exits the stop station or clears the loop line, the signal returns to Green, indicating that the track is free for the next train.
- A Yellow signal is shown when a train is either approaching a stop station or moving into the loop line to manage train traffic.

Project Objective

The main objective of this mini-project was to simulate a modern railway signalling system that handles stop stations with loop line integration. The system models:

The main objective of this mini-project was to simulate a modern railway signalling system that handles stop stations with loop line integration. The system models:

Train detection using track circuits.

Signal control based on track occupancy and the status of stop stations.

Efficient loop line management at stations to allow trains to bypass or wait without blocking the main track.

This system provides a fundamental understanding of railway signalling and loop line operations, and its concepts can be extended to larger systems in real-world railway networks.

Source Code-

```
module railway_signal_system (
   input train_detected_main,  // Train detected on the main track
   input train_detected_loop,  // Train detected on the loop line
   input station_clear,       // Station is occupied (1 = occupied, 0 = free)
   input express_train,       // Express train priority signal (1 = Express, 0
= Regular)
   input clk,                // Clock signal for counters
   output reg [1:0] signal_main,  // Main track signal: 00 = Red, 01 =
Yellow, 10 = Green
   output reg [1:0] signal_loop   // Loop track signal: 00 = Red, 01 =
Yellow, 10 = Green
);

// Signal Encoding
parameter RED = 2'b00, YELLOW = 2'b01, GREEN = 2'b10;

// Counters to track train duration on tracks
```

```
reg [3:0] main_track_counter = 0;
reg [3:0] loop_track_counter = 0;

always @(posedge clk) begin
   // Train on main track
   if (train_detected_main) begin
      main_track_counter <= main_track_counter + 1;
   end else begin
      main_track_counter <= 0;
   end

   // Train on loop line
   if (train_detected_loop) begin
      loop_track_counter <= loop_track_counter + 1;
   end else begin
      loop_track_counter <= 0;
   end
end

always @(*) begin
   // Default signals
   signal_main = GREEN;
   signal_loop = GREEN;
```

```
    // Express train priority: If an express train is on the main track, it
always gets priority
    if (express_train) begin
        signal_main = GREEN;
        signal_loop = RED;
    end else begin
        // Main Track Signal Logic
        if (train_detected_main) begin
            if (station_clear) begin
                if (main_track_counter > 3) // Delay before turning Yellow
                    signal_main = YELLOW;
                else
                    signal_main = RED; // Train present + Station occupied →
Red
            end else begin
                signal_main = YELLOW; // Train present + Station clear →
Yellow
            end
        end else begin
            signal_main = GREEN; // No train detected → Green
        end

        // Loop Line Signal Logic
        if (train_detected_loop) begin
            if (station_clear) begin
```

```
            if (loop_track_counter > 3) // Delay before turning Yellow
              signal_loop = YELLOW;
            else
              signal_loop = RED; // Train present + Station occupied → Red
          end else begin
            signal_loop = YELLOW; // Train moving to loop + Station clear → Yellow
          end
        end else begin
          signal_loop = GREEN; // No train detected on loop → Green
        end
      end
    end

endmodule
```

Testbench-

```
module railway_signal_tb;
    reg train_detected_main;
    reg train_detected_loop;
    reg station_clear;
    reg express_train;
    reg clk;
    wire [1:0] signal_main;
    wire [1:0] signal_loop;

    // Instantiate the railway signal system module
    railway_signal_system uut (
        .train_detected_main(train_detected_main),
        .train_detected_loop(train_detected_loop),
        .station_clear(station_clear),
        .express_train(express_train),
        .clk(clk),
        .signal_main(signal_main),
        .signal_loop(signal_loop)
    );

    // Clock generation
    always #5 clk = ~clk;
```

```
initial begin
    // Initialize inputs
    clk = 0;
    train_detected_main = 0;
    train_detected_loop = 0;
    station_clear = 0;
    express_train = 0;

    #10; // Wait some time

    // Test Case 1: No train, idle state
    #10;
    $display("Test Case 1: Idle State - signal_main: %b, signal_loop:
%b", signal_main, signal_loop);

    // Test Case 2: Regular train on main track, station occupied
    train_detected_main = 1;
    station_clear = 1;
    #10;
    $display("Test Case 2: Train on Main - signal_main: %b,
signal_loop: %b", signal_main, signal_loop);

    // Test Case 3: Express train on main track, station occupied
    express_train = 1;
    #10;
```

```
        $display("Test Case 3: Express Train - signal_main: %b, signal_loop: %b", signal_main, signal_loop);

        // Test Case 4: Train on loop track, station free
        train_detected_main = 0;
        train_detected_loop = 1;
        station_clear = 0;
        express_train = 0;
        #10;
        $display("Test Case 4: Train on Loop - signal_main: %b, signal_loop: %b", signal_main, signal_loop);

        // Test Case 5: Train exits, signals return to green
        train_detected_main = 0;
        train_detected_loop = 0;
        #10;
        $display("Test Case 5: Train Exits - signal_main: %b, signal_loop: %b", signal_main, signal_loop);

        $stop;
    end
endmodule
```

Conclusion-

This project effectively used Verilog to simulate train detection, signal control, and loop line management in a railway signalling and blocking system. By avoiding collisions and maximizing track utilization, the system successfully guarantees safe and effective train movement.

Track Safety Mechanism: By employing track circuit-based detection, the system keeps two trains from sharing a section at the same time.

Express Train Priority: To cut down on delays, the express train is always given precedence on the main track.

Loop Line Utilization: To maintain efficient railroad operations, regular trains are rerouted to the loop line when the main track is occupied.

Realistic Signal Transitions: By adding counters, the simulation becomes more realistic by enabling delayed signal transitions.