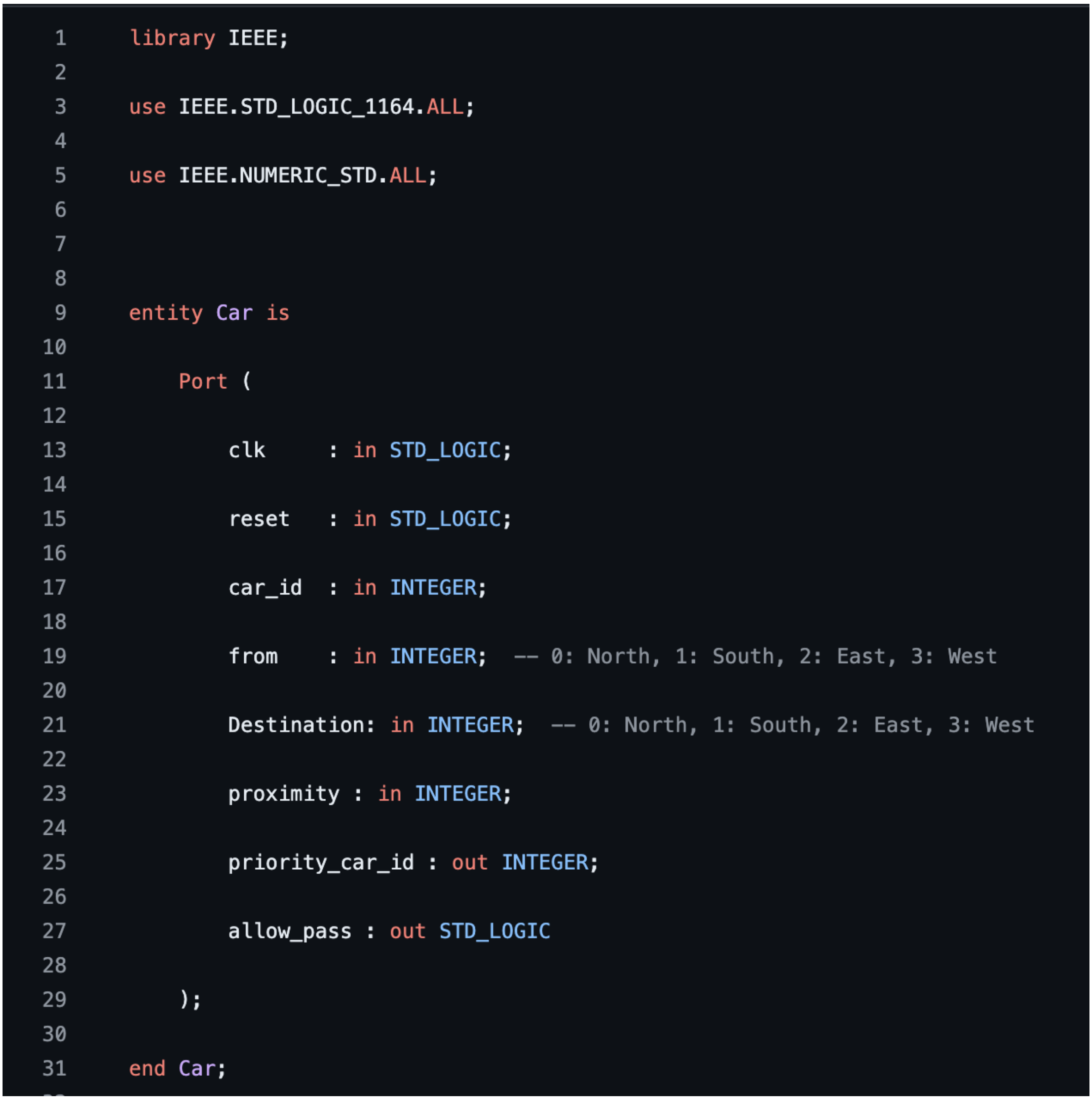
Analysis of VHDL Code, Test Bench and Output Waveform:



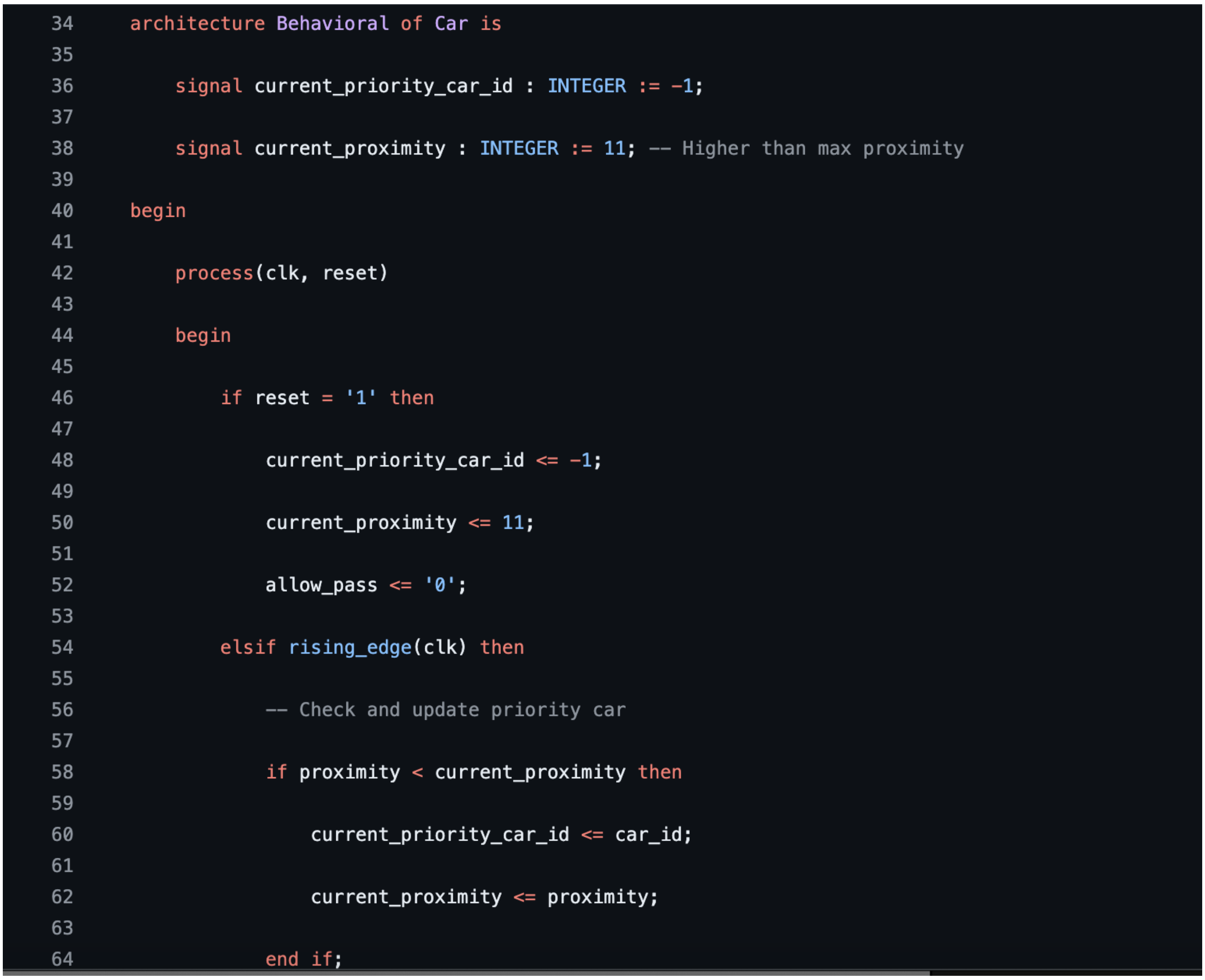
```
library IEEE;

use IEEE.STD_LOGIC_1164.ALL;

use IEEE.NUMERIC_STD.ALL;



entity Car is

    Port (

        clk     : in STD_LOGIC;

        reset   : in STD_LOGIC;

        car_id  : in INTEGER;

        from    : in INTEGER;  -- 0: North, 1: South, 2: East, 3: West

        Destination: in INTEGER;  -- 0: North, 1: South, 2: East, 3: West

        proximity : in INTEGER;

        priority_car_id : out INTEGER;

        allow_pass : out STD_LOGIC

    );

end Car;
```

Fig.2 Snippet VHDL Code

Entity Declaration:

Defines the inputs and outputs for the Car entity.

- clk and reset are input signals for clock and reset.
- car_id, from, destination, proximity, and priority are all the inputs signals related to the car's attributes.
- allow is the output signal indicating whether the car is allowed to proceed or not.



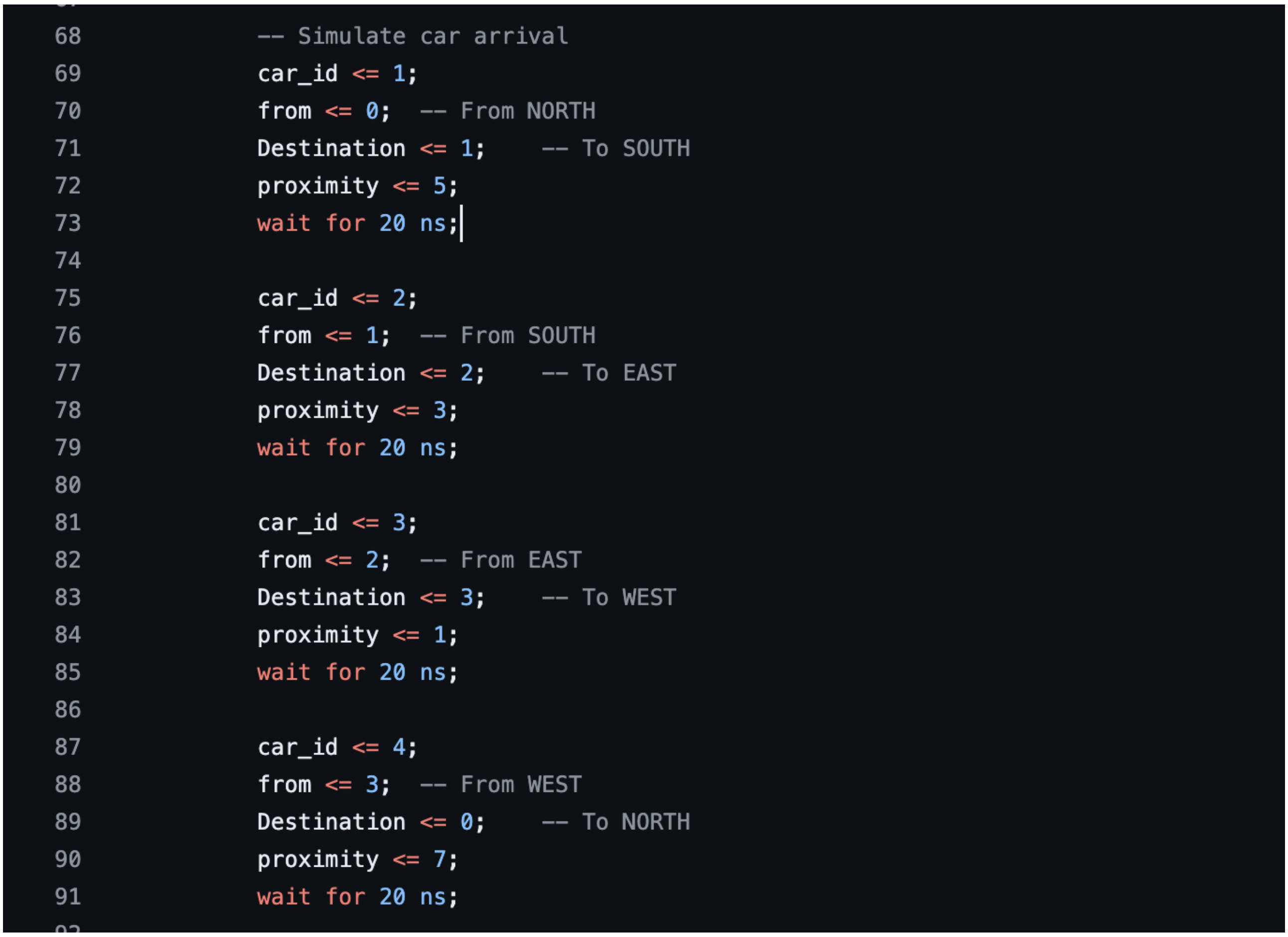
```vhdl
architecture Behavioral of Car is

    signal current_priority_car_id : INTEGER := -1;

    signal current_proximity : INTEGER := 11; -- Higher than max proximity

begin

    process(clk, reset)

    begin

        if reset = '1' then

            current_priority_car_id <= -1;

            current_proximity <= 11;

            allow_pass <= '0';

        elsif rising_edge(clk) then

            -- Check and update priority car

            if proximity < current_proximity then

                current_priority_car_id <= car_id;

                current_proximity <= proximity;

            end if;
```

Fig.3 Snippet VHDL Code

Architecture Behavioral:

Contains the logic for controlling the allow signal based on the priority.

- If reset is active ('1'), allow is set to '0'.
- On the rising edge of the clock, if priority is greater than 5, allow is set to '1'; otherwise, it remains ‘0'.

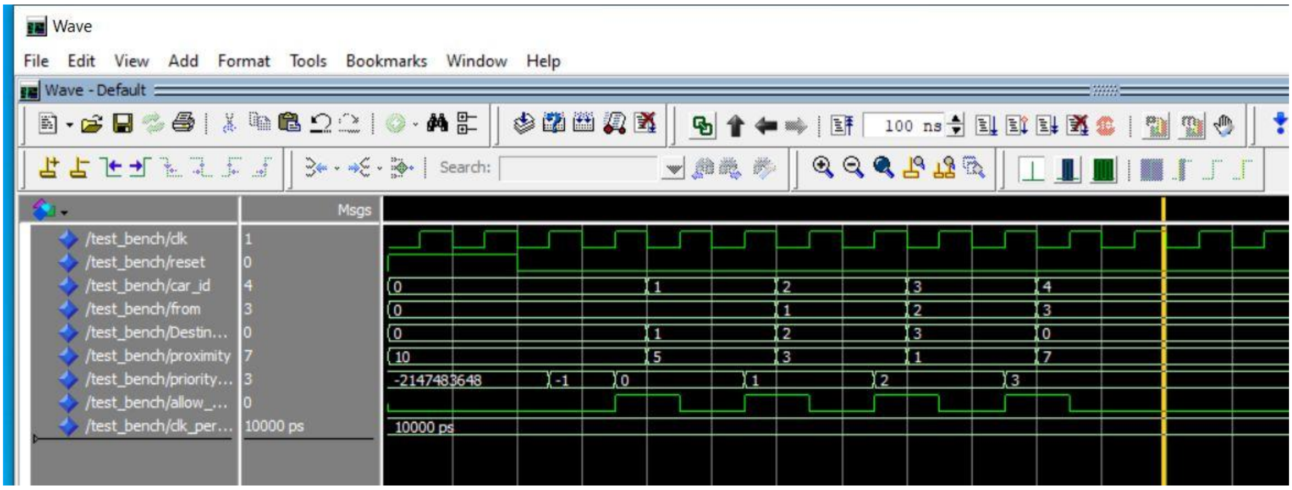


```
            -- Simulate car arrival
            car_id <= 1;
            from <= 0;  -- From NORTH
            Destination <= 1;    -- To SOUTH
            proximity <= 5;
            wait for 20 ns;

            car_id <= 2;
            from <= 1;  -- From SOUTH
            Destination <= 2;    -- To EAST
            proximity <= 3;
            wait for 20 ns;

            car_id <= 3;
            from <= 2;  -- From EAST
            Destination <= 3;    -- To WEST
            proximity <= 1;
            wait for 20 ns;

            car_id <= 4;
            from <= 3;  -- From WEST
            Destination <= 0;    -- To NORTH
            proximity <= 7;
            wait for 20 ns;
```

Fig.4 Snippet Test Bench Code

this part of the test bench code is describing how we simulating each car arriving to the intersection from different directions, so when each car is arriving from specific origin and heading to specific destination, with a predefined proximity value, the simulation involving setting the ‘car_id ’and ‘from ’and ‘destination ’and ‘proximity ’signals for every car and then waiting for 20 ns before moving to the next car, and this way will ensure us that the system is accurately processes cars arrival from different directions to different destinations and proximity values, and the wait periods is simulating the time intervals between the arrivals of successive car, and that is allowing for a thorough evaluation of the car system’s response to dynamic car traffic, and car can cross the intersection with very less possibility of accidents.

Output Wave Form Analysis:



	Msgs
/test_bench/clk	1
/test_bench/reset	0
/test_bench/car_id	4
/test_bench/from	3
/test_bench/Destin...	0
/test_bench/proximity	7
/test_bench/priority...	3
/test_bench/allow_...	0
/test_bench/clk_per...	10000 ps

Fig.5 Output Waveform

The figure is describing the output waveform that shows the simulation results of the Car entity in VHDL and Test Bench.

- So as it is shown in the waveform it shows that the allow signal is correctly determined by the priority value of each car.
- The reset mechanism works as expected, setting allow to '0' initially.
- The CLK signal is responsible to drives the synchronous behaviour, that ensuring the changes will happen at the correct times.