Parking Management System

Donya Jafari - 401101524

June 25, 2024

Abstract

This report presents the design and implementation of a parking management system using Verilog HDL. The system tracks the number of parked and vacated cars, differentiating between university and non-university vehicles, and ensures that parking spaces do not exceed their maximum capacities.

1 Details

The parking management system is designed to reset at 8:00 AM each day. The system initially sets the university parking capacity to its maximum at this time. After 1:00 PM, the non-university capacity increases by 50 spaces per hour starting from 200. At 4:00 PM, the non-university capacity is set to its full value of 500 spaces considering the overall capacity is no more than 700 in our scenario.

2 Design and Implementation

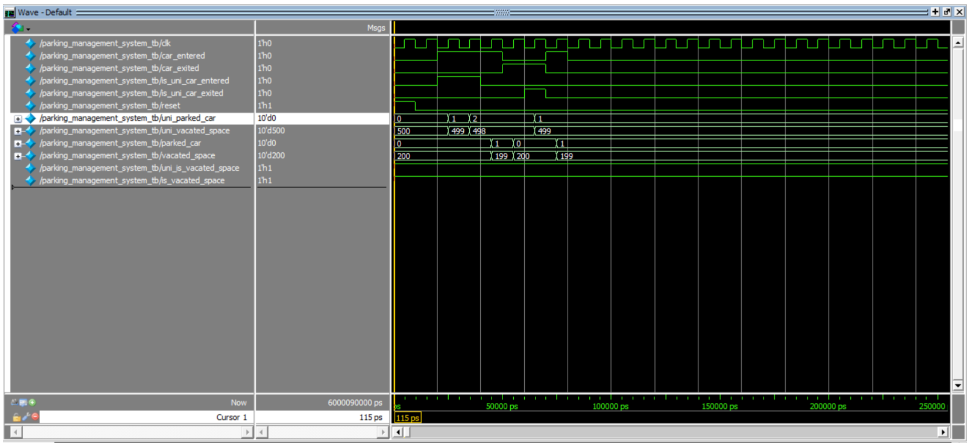
The system is designed using Verilog HDL.

```
    module parking_management_system (
        input wire clk,
        input wire reset,

        input wire car_entered,
        input wire car_exited,
        input wire is_uni_car_entered,
        input wire is_uni_car_exited,

        output reg [9:0] uni_parked_car,
        output reg [9:0] parked_car,
        output reg [9:0] uni_vacated_space,
        output reg [9:0] vacated_space,
        output reg uni_is_vacated_space,
        output reg is_vacated_space
    );

    parameter MAX_PARKING_SPACE = 700;
    parameter MAX_UNI_SPACE = 500;
    parameter CLK_FREQ = 100_000_000;

    reg [31:0] elapsed_time_cycles;
    reg [9:0] NON_UNI_SPACE;
    reg [3:0] time_threshold;

    always @(posedge clk or posedge reset) begin
        if (reset) begin
            elapsed_time_cycles <= 0;
            time_threshold <= 0;
            uni_parked_car <= 0;
            parked_car <= 0;
            uni_vacated_space <= MAX_UNI_SPACE;
            vacated_space <= 200;
            NON_UNI_SPACE <= 200;
            uni_is_vacated_space <= 1;
            is_vacated_space <= 1;
        end else begin
        ## consider all possible cases
```

The complete source code (controller module + test bench) is available in the ’code’ directory of the GitHub repository, which is linked at the end of this document.

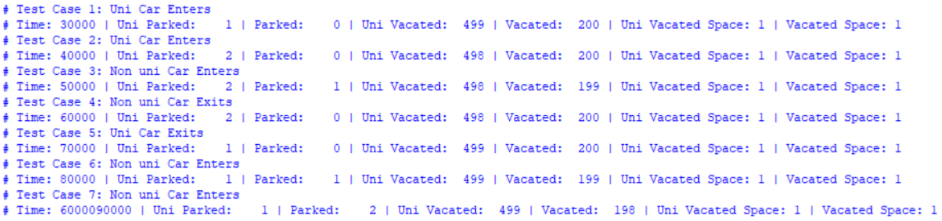
3 Results

The simulation results confirm the correct functionality of the parking management system. The following figures show the waveforms and output of the simulation.



```
Wave - Default
                                                   Msgs
/parking_management_system_tb/clk                  1'h0
/parking_management_system_tb/car_entered          1'h0
/parking_management_system_tb/car_exited           1'h0
/parking_management_system_tb/is_uni_car_entered   1'h0
/parking_management_system_tb/is_uni_car_exited    1'h0
/parking_management_system_tb/reset                1'h1
/parking_management_system_tb/uni_parked_car       10'd0
/parking_management_system_tb/uni_vacated_space    10'd500
/parking_management_system_tb/parked_car           10'd0
/parking_management_system_tb/vacated_space        10'd200
/parking_management_system_tb/uni_is_vacated_space 1'h1
/parking_management_system_tb/is_vacated_space     1'h1
Now                                                6000090000 ps
Cursor 1                                           115 ps
```

Figure 1: Simulation Waveform 1



```
# Test Case 1: Uni Car Enters
# Time: 30000 | Uni Parked:    1 | Parked:    0 | Uni Vacated:  499 | Vacated:  200 | Uni Vacated Space: 1 | Vacated Space: 1
# Test Case 2: Uni Car Enters
# Time: 40000 | Uni Parked:    2 | Parked:    0 | Uni Vacated:  498 | Vacated:  200 | Uni Vacated Space: 1 | Vacated Space: 1
# Test Case 3: Non uni Car Enters
# Time: 50000 | Uni Parked:    2 | Parked:    1 | Uni Vacated:  498 | Vacated:  199 | Uni Vacated Space: 1 | Vacated Space: 1
# Test Case 4: Non uni Car Exits
# Time: 60000 | Uni Parked:    2 | Parked:    0 | Uni Vacated:  498 | Vacated:  200 | Uni Vacated Space: 1 | Vacated Space: 1
# Test Case 5: Uni Car Exits
# Time: 70000 | Uni Parked:    1 | Parked:    0 | Uni Vacated:  499 | Vacated:  200 | Uni Vacated Space: 1 | Vacated Space: 1
# Test Case 6: Non uni Car Enters
# Time: 80000 | Uni Parked:    1 | Parked:    1 | Uni Vacated:  499 | Vacated:  199 | Uni Vacated Space: 1 | Vacated Space: 1
# Test Case 7: Non uni Car Enters
# Time: 6000090000 | Uni Parked:    1 | Parked:    2 | Uni Vacated:  499 | Vacated:  198 | Uni Vacated Space: 1 | Vacated Space: 1
```

Figure 2: Simulation output

4 Synthesizing

Here is the synthesized output.

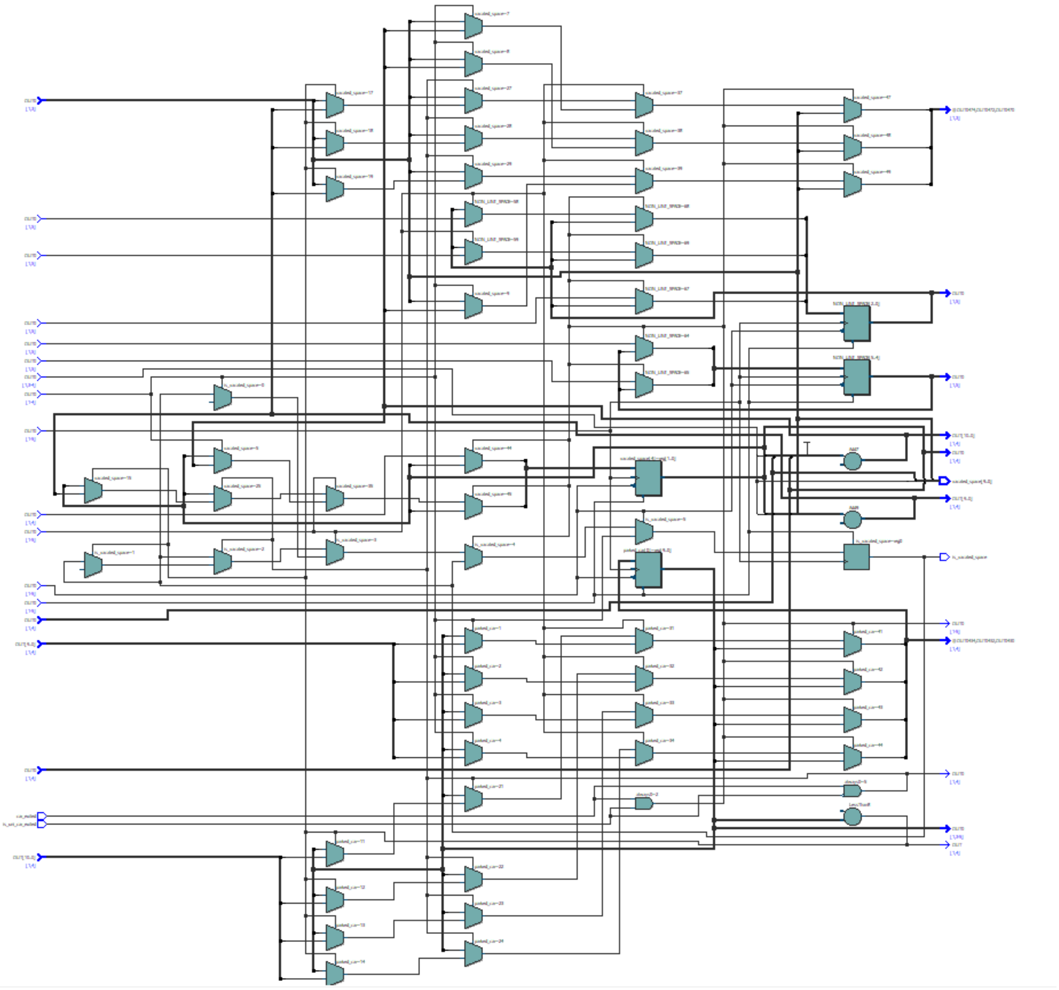
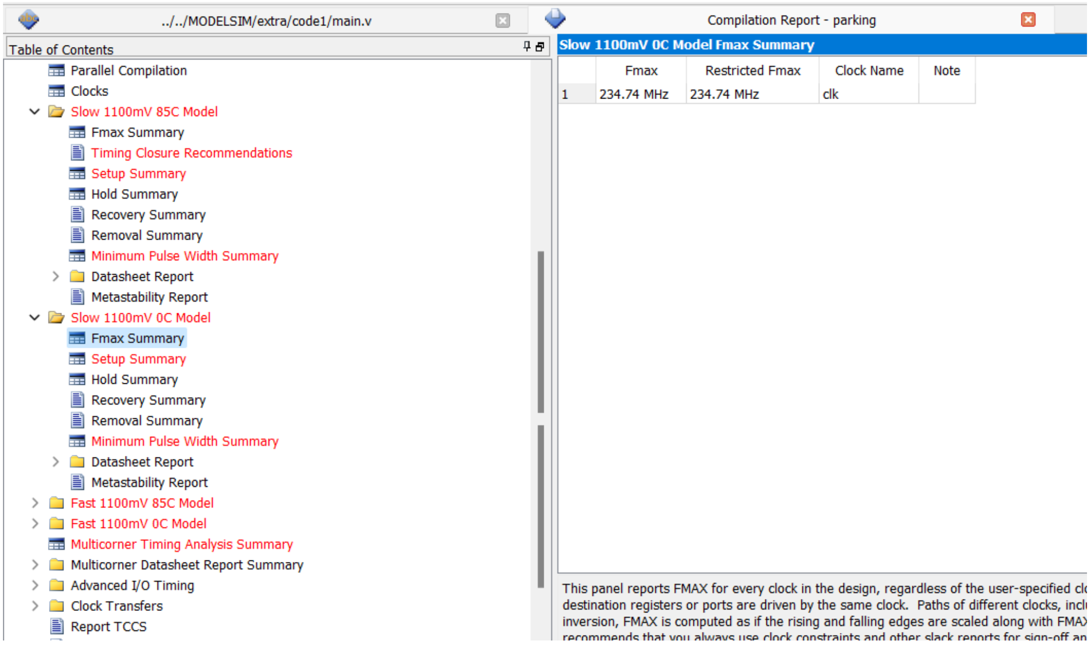


Figure 3: Synthesized circuit

This suggest that the maximum operational clock frequency is 234.74 MHz.



Slow 1100mV 0C Model Fmax Summary

	Fmax	Restricted Fmax	Clock Name	Note
1	234.74 MHz	234.74 MHz	clk	

This panel reports FMAX for every clock in the design, regardless of the user-specified clo
destination registers or ports are driven by the same clock. Paths of different clocks, incl
inversion, FMAX is computed as if the rising and falling edges are scaled along with FMAX
recommends that you always use clock constraints and other slack reports for sign-off an

Figure 4: Frequency

5 References

- *GitHub*: `https://github.com/Donya-Jafari/parking-system-DSDcourse/tree/main`