طراحی سیستم‌های دیجیتال

پروژه | پوریا عرفانزاده

مقدمه

این پروژه بخشی از درس طراحی سیستم دیجیتال تحت نظارت دکتر فصحتی است. هدف آن طراحی و اجرای یک سیستم قوی مدیریت پارکینگ دانشگاهی است که نظارت بهینه و به صورت زمان واقعی بر فضاهای پارکینگ را فراهم می کند. این سیستم بین وسایل نقلیه مرتبط با دانشگاه و وسایل نقلیه غیر دانشگاهی تفاوت قائل شده و از استفاده بهینه از ظرفیت پارکینگ موجود اطمینان حاصل می کند. این پروژه با استفاده از Verilog توسعه یافته و بر روی یک FPGA پیاده سازی شده است تا از عملکرد و قابلیت اطمینان بالای آن در مدیریت مدارهای دیجیتال بهره ببرد.

طراحی

فاز طراحی

کد وریلاگ ماژول را می‌توانید در زیر مشاهده کنید. توضیحات لازم در قالب کامنت در کد گنجانده شده است.

```
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, // Number of university cars currently parked
    output reg [9:0] parked_car, // Number of non-university cars currently parked
    output reg [9:0] uni_vacated_space, // Number of vacated spaces reserved for university
cars
    output reg [9:0] vacated_space, // Number of vacated spaces for non-university cars
    output reg uni_is_vacated_space, // indicates if there is a vacated space for university
cars
    output reg is_vacated_space //  Indicates if there is a vacated space for non-university
cars
);

    parameter MAX_PARKING_SPACE = 700; // Total maximum parking spaces available
    parameter MAX_UNI_SPACE = 500; // Maximum parking spaces reserved for university cars
    parameter CLK_FREQ = 100_000_000; // Frequency of the clock signal (in Hz)
    parameter NON_UNI_BASE_SPACE = 200; // Initial base space reserved for non-university
cars
```

```
    // these regs are used to adjust the non_uni_space
    reg [31:0] elapsed_time_cycles;
    reg [9:0] non_uni_space;
    reg [3:0] time_threshold;

    task update_space_availability;
        input [9:0] uni_parked_car, parked_car;
        output reg uni_is_vacated_space, is_vacated_space;
    begin
        if (uni_parked_car < MAX_UNI_SPACE && uni_parked_car + parked_car <
MAX_PARKING_SPACE) begin
            uni_is_vacated_space = 1;
        end else begin
            uni_is_vacated_space = 0;
        end

        if (parked_car < non_uni_space && uni_parked_car + parked_car < MAX_PARKING_SPACE)
begin
            is_vacated_space = 1;
        end else begin
            is_vacated_space = 0;
        end
    end
    endtask

    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 <= NON_UNI_BASE_SPACE;
            non_uni_space <= NON_UNI_BASE_SPACE;
            uni_is_vacated_space <= 1;
            is_vacated_space <= 1;
        end else begin
            elapsed_time_cycles <= elapsed_time_cycles + 1;

            // Update time threshold based on elapsed time
            case (elapsed_time_cycles)
                CLK_FREQ * 300 * 60: time_threshold <= 4;
                CLK_FREQ * 240 * 60: time_threshold <= 3;
                CLK_FREQ * 180 * 60: time_threshold <= 2;
                CLK_FREQ * 120 * 60: time_threshold <= 1;
                default: time_threshold <= 0;
            endcase

            // Adjust non-uni space allocation based on time threshold
```

```
            case (time_threshold)
                0: non_uni_space <= NON_UNI_BASE_SPACE;
                1: non_uni_space <= 250;
                2: non_uni_space <= 300;
                3: non_uni_space <= 350;
                4: non_uni_space <= MAX_UNI_SPACE;
            endcase

            // Handle university car entered
            if (car_entered && is_uni_car_entered) begin
                if (uni_parked_car < MAX_UNI_SPACE && uni_parked_car + parked_car <
MAX_PARKING_SPACE) begin
                    uni_parked_car <= uni_parked_car + 1;
                    uni_vacated_space <= uni_vacated_space - 1;
                end
                update_space_availability(uni_parked_car, parked_car, uni_is_vacated_space,
is_vacated_space);
            // Handle university car exited
            end else if (car_exited && is_uni_car_exited) begin
                if (uni_parked_car > 0) begin
                    uni_parked_car <= uni_parked_car - 1;
                    uni_vacated_space <= uni_vacated_space + 1;
                    uni_is_vacated_space <= 1;
                end
            // Handle non-university car entered
            end else if (car_entered && !is_uni_car_entered) begin
                if (parked_car + uni_parked_car < MAX_PARKING_SPACE && parked_car <
non_uni_space) begin
                    parked_car <= parked_car + 1;
                    vacated_space <= vacated_space - 1;
                    update_space_availability(uni_parked_car, parked_car,
uni_is_vacated_space, is_vacated_space);
                end
            // Handle non-university car exited
            end else if (car_exited && !is_uni_car_exited) begin
                if (parked_car > 0) begin
                    parked_car <= parked_car - 1;
                    vacated_space <= vacated_space + 1;
                    is_vacated_space <= 1;
                end
            end
        end
    end
endmodule
```

فاز تست

می‌توانید کد ماژول تست بنچ را در زیر مشاهده کنید:

```
module TB_parking_management_system;
```

```
    reg clk;
    reg reset;
    reg car_entered;
    reg car_exited;
    reg is_uni_car_entered;
    reg is_uni_car_exited;
    wire [9:0] uni_parked_car;
    wire [9:0] parked_car;
    wire [9:0] uni_vacated_space;
    wire [9:0] vacated_space;
    wire uni_is_vacated_space;
    wire is_vacated_space;

    integer outfile;

    parking_management_system uut (
        .clk(clk),
        .reset(reset),
        .car_entered(car_entered),
        .car_exited(car_exited),
        .is_uni_car_entered(is_uni_car_entered),
        .is_uni_car_exited(is_uni_car_exited),
        .uni_parked_car(uni_parked_car),
        .parked_car(parked_car),
        .uni_vacated_space(uni_vacated_space),
        .vacated_space(vacated_space),
        .uni_is_vacated_space(uni_is_vacated_space),
        .is_vacated_space(is_vacated_space)
    );

    parameter CLK_PERIOD = 10;
    parameter DELAY = 10;

    always begin
        # (CLK_PERIOD / 2) clk = ~clk;
    end

    task display_state;
        begin
            $fwrite(outfile, "Time: %0t\nNumber of university cars currently parked:
%d\nNumber of non-university cars currently parked: %d\nNumber of vacated spaces reserved for
university cars: %d\nNumber of vacated spaces for non-university cars: %d\nIs there a vacated
space for university cars? %s\nIs there a vacated space for non-university cars? %s\n",
                    $time, uni_parked_car, parked_car, uni_vacated_space, vacated_space,
                    uni_is_vacated_space ? "yes" : "no",
                    is_vacated_space ? "yes" : "no");
        end
    endtask

    task car_enter(
```

```
        input is_uni
    );
        begin
            car_entered = 1;
            is_uni_car_entered = is_uni;
            #DELAY;
            car_entered = 0;
            is_uni_car_entered = 0;
            display_state();
        end
    endtask

    task car_exit(
        input is_uni
    );
        begin
            car_exited = 1;
            is_uni_car_exited = is_uni;
            #DELAY;
            car_exited = 0;
            is_uni_car_exited = 0;
            display_state();
        end
    endtask

    initial begin
        outfile = $fopen("output_results.txt", "w");

        clk = 0;
        reset = 0;
        car_entered = 0;
        car_exited = 0;
        is_uni_car_entered = 0;
        is_uni_car_exited = 0;

        $fwrite(outfile, "Resetting the system...\n");
        reset = 1;
        #DELAY;
        reset = 0;
        #DELAY;

        // Test normal operations
        $fwrite(outfile, "Event 1: Uni Car Enters");
        car_enter(1);

        $fwrite(outfile, "Event 2: Another Uni Car Enters");
        car_enter(1);

        $fwrite(outfile, "Event 3: Non-Uni Car Enters");
        car_enter(0);
```

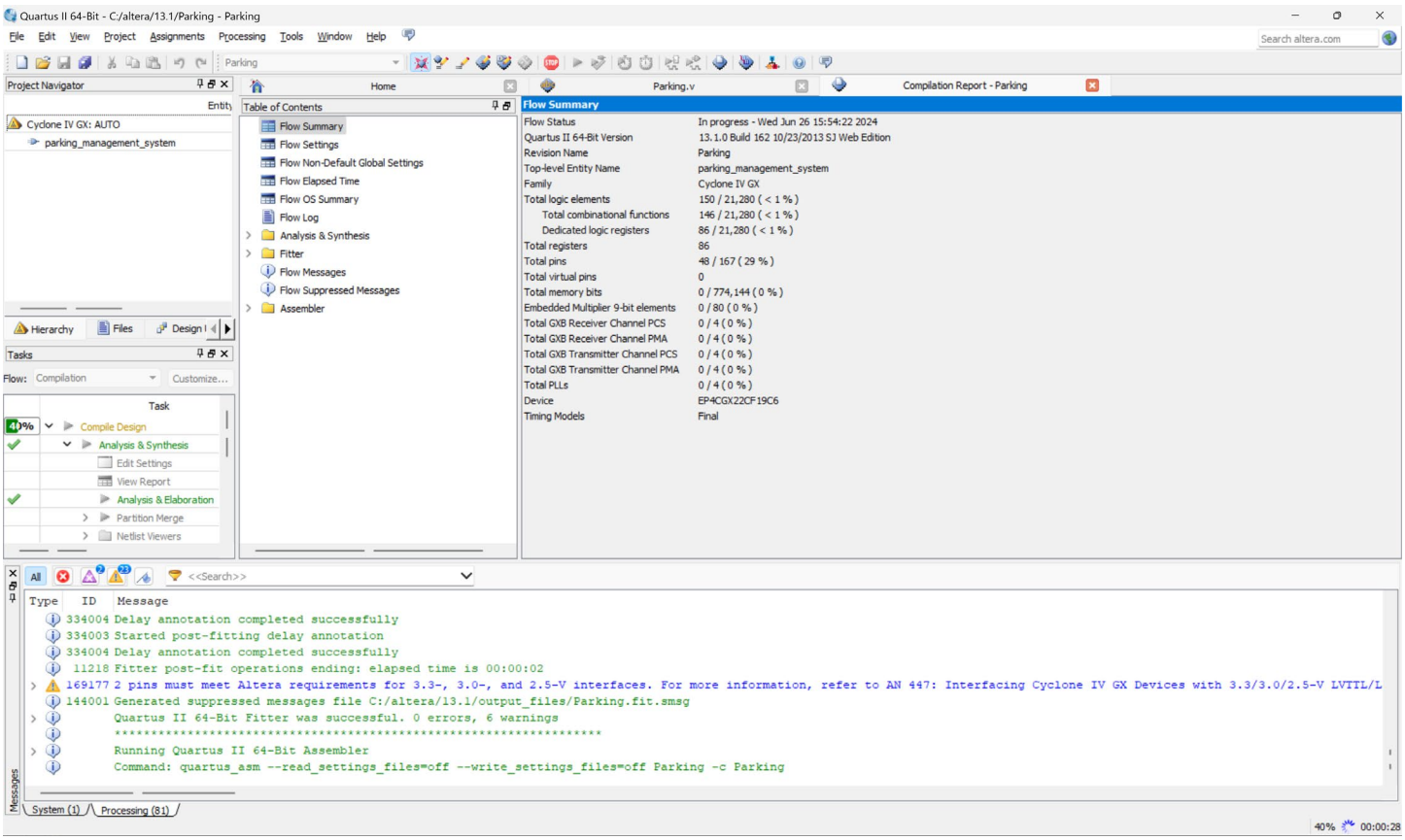
```
        $fwrite(outfile, "Event 4: Non-Uni Car Exits");
        car_exit(0);

        $fwrite(outfile, "Event 5: Uni Car Exits");
        car_exit(1);

        $fwrite(outfile, "Event 6: Non-Uni Car Enters Again");
        car_enter(0);

        #6000000;
        $fwrite(outfile, "Event 7: Non-Uni Car Enters After Long Delay");
        car_enter(0);

        // Test edge cases
        $fwrite(outfile, "Event 8: Fill up university parking spaces");
        repeat (500) begin
            car_enter(1);
        end

        $fwrite(outfile, "Event 9: Fill up non-university parking spaces");
        repeat (200) begin
            car_enter(0);
        end

        $fwrite(outfile, "Event 10: Attempt to park another university car (should fail)");
        car_enter(1);

        $fwrite(outfile, "Event 11: Attempt to park another non-university car (should
fail)");
        car_enter(0);

        $fwrite(outfile, "Event 12: Exit a university car");
        car_exit(1);

        $fwrite(outfile, "Event 13: Exit a non-university car");
        car_exit(0);

        $fwrite(outfile, "Event 14: Park a university car after space is vacated");
        car_enter(1);

        $fwrite(outfile, "Event 15: Park a non-university car after space is vacated");
        car_enter(0);

        $fclose(outfile);

        $finish;
    end

endmodule
```

همانطور که می‌بینید ۱۵ حالت مختلف در این تست بنچ بررسی شده‌است. می‌توانید نتیجه‌ها را در فایل output_results.txt ببینید. نتیجه ۴۹۹۳ خط شده است به همین دلیل دیگه در این داکیومنت آورده نشده است.

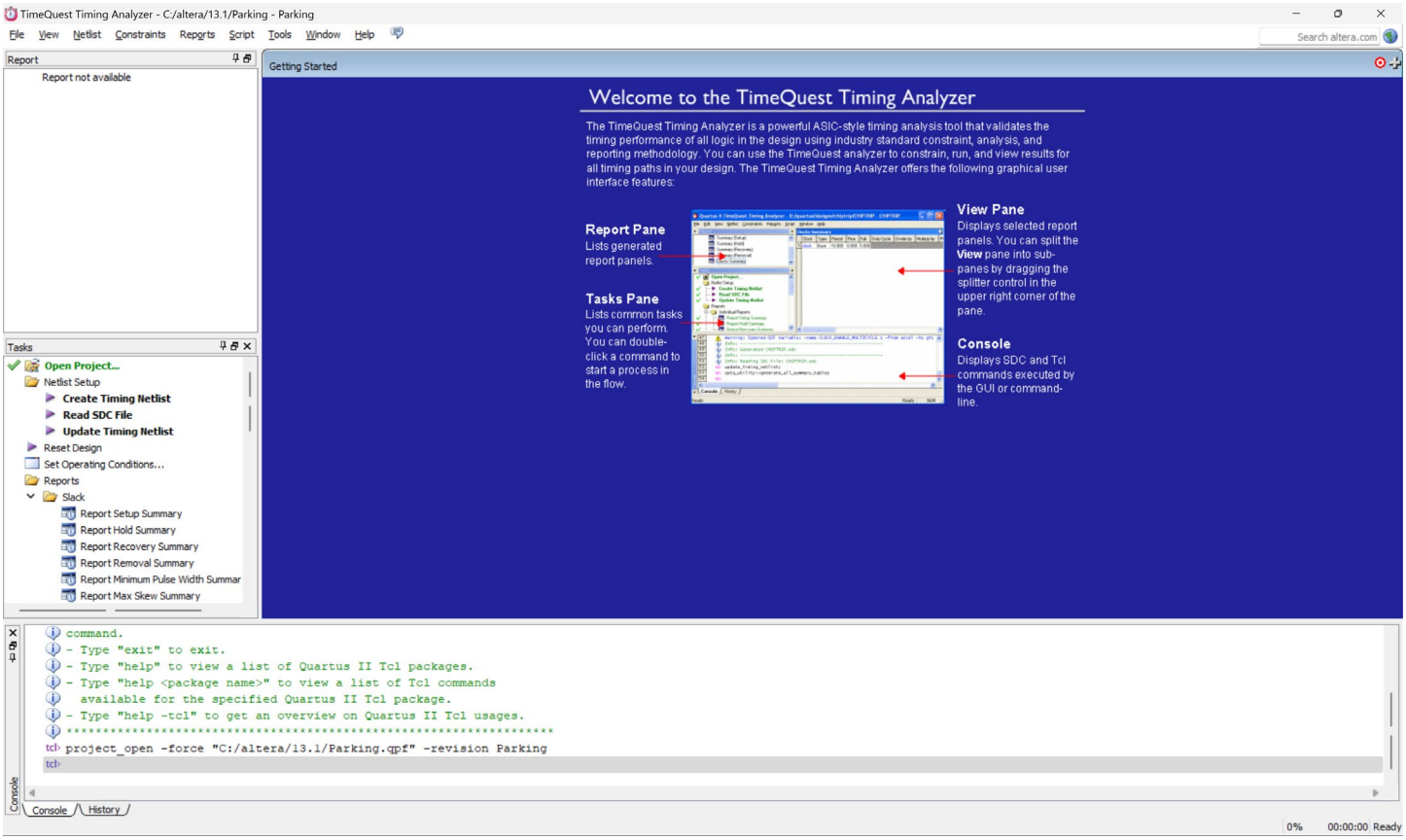
سنتز

فایل وریلاگ را در کوارتوس باز می‌کنیم. از لیست دیوایس‌ها Cydone IV GX را انتخاب کرده‌ایم.

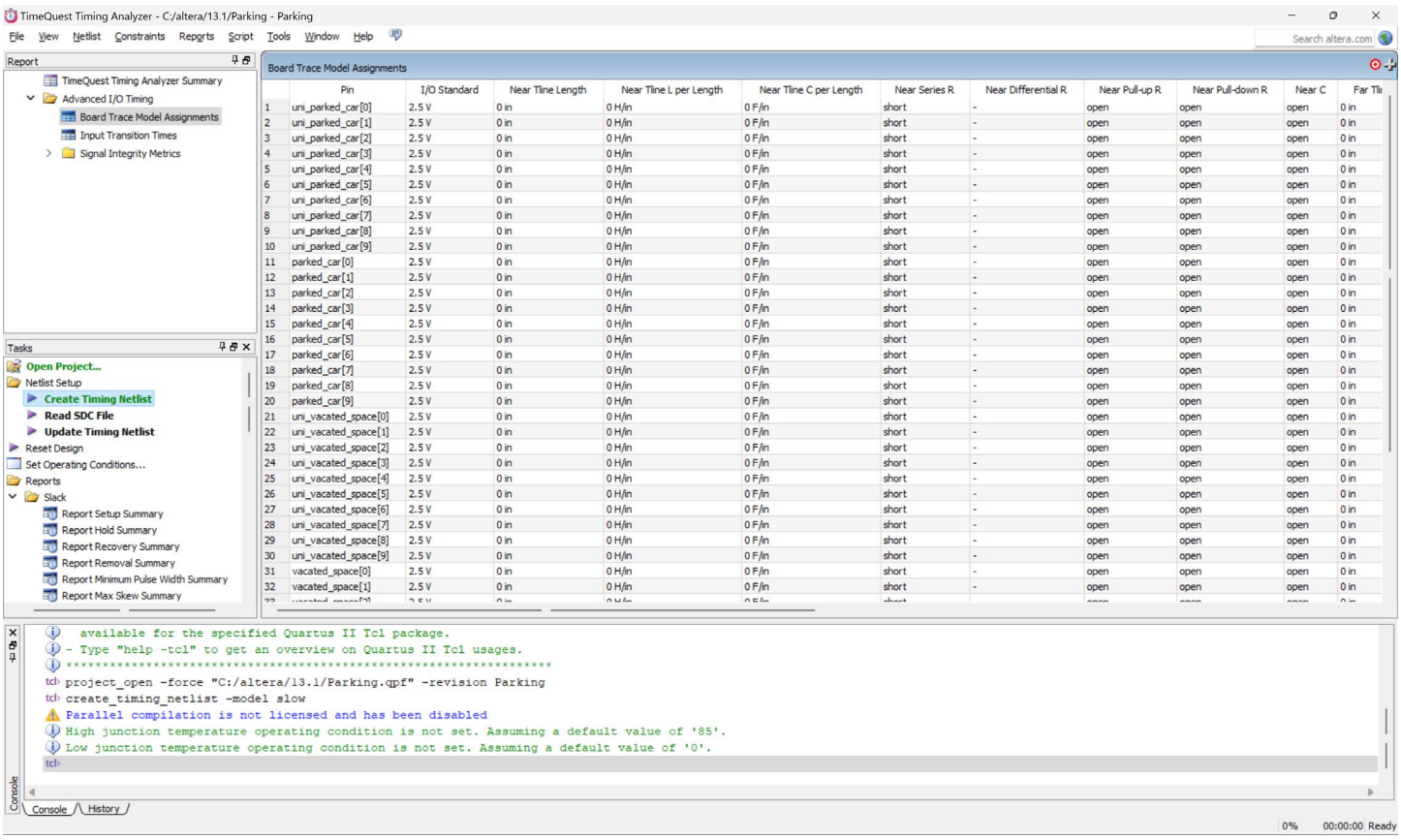
کامپایل می‌کنیم.



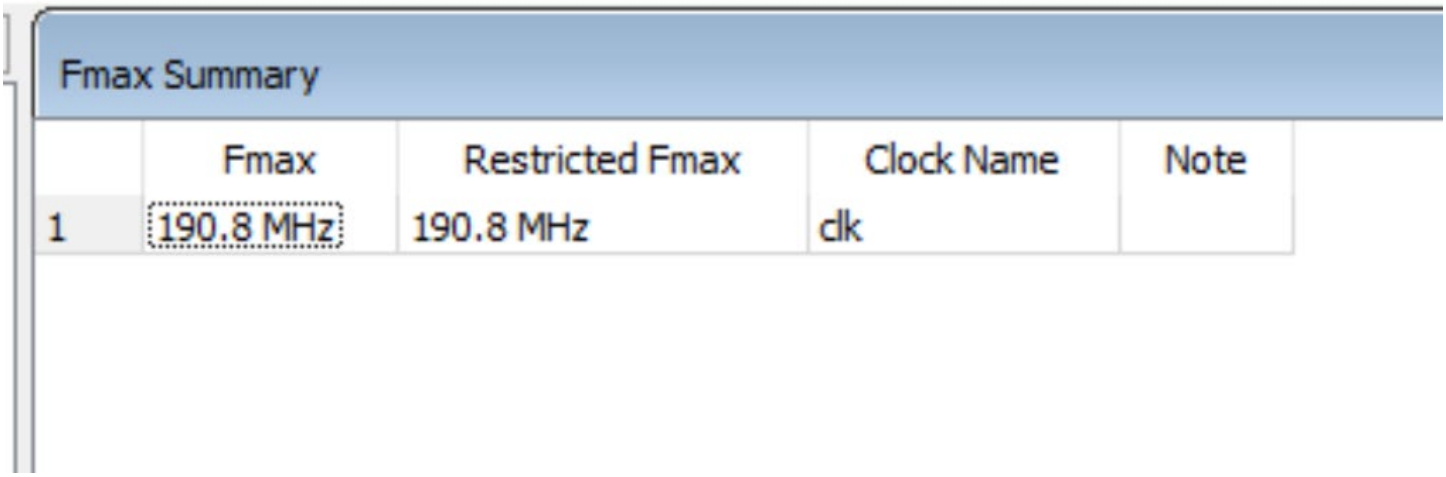
حال ابزار Time Quest Analyzer را باز میکنیم.



سپس در مرحله بعدی Create Timing Netlist:



و سپس Read STC File و بعد از آن Update Timing Netlist را انتخاب می‌کنیم. سپس از بین گزارش‌های آماده شده Report Fmax Summary را انتخاب می‌کنیم. حال فرکانس ماکس را در اینجا می‌توانیم مشاهده کنیم:



Fmax Summary

	Fmax	Restricted Fmax	Clock Name	Note
1	190.8 MHz	190.8 MHz	clk	