

پروژه مدار مدیریت پارکینگ دانشگاه

طراحی سیستم‌های دیجیتال - بهار ۱۴۰۳

استاد: دکتر فصحتی

سعید فراتی کاشانی - ۴۰۱۱۰۶۲۹۹

مقدمه:

در این پروژه به پیاده‌سازی مداری برای مدیریت پارکینگ دانشگاه پرداختیم. این پارکینگ شامل ویژگی‌های خاصی است که در ادامه آن‌ها را شرح می‌دهیم.

در این پارکینگ، اولویت فضا با اساتید و کارمندان دانشگاه است و این ظرفیت بر اساس آمار حداکثر ۵۰۰ خودرو تعیین گردیده است. با توجه به اینکه فضای کل پارکینگ ۷۰۰ خودرو است از ساعت ۸ تا ۱۳ فقط ۲۰۰ ظرفیت خالی برای ورود آزاد موجود است. در نهایت از ساعت ۱۳ تا ۱۶ به ازای هر ساعت ظرفیت ورود آزاد ۵۰ خودرو افزایش می‌یابد و در ساعت ۱۶، ظرفیت آزاد ورود به ۵۰۰ خودرو می‌رسد.

سیگنال‌های ورودی و خروجی نیز به صورت زیر تعریف می‌شوند:

| خروجی‌ها | |
|----------------------|--|
| uni_parked_car | تعداد خودروهایی متعلق به دانشگاه که در پارکینگ پارک شده‌اند. |
| parked_care | تعداد خودروهای پارک شده در پارکینگ مربوط به ظرفیت آزاد |
| uni_vacated_space | تعداد فضای خالی متعلق به دانشگاه |
| vacated_space | تعداد فضاهای خالی مربوط به ظرفیت آزاد |
| uni_is_vacated_space | آیا فضای خالی برای دانشگاه موجود است؟ |
| is_vacated_space | آیا فضای خالی برای ظرفیت آزاد موجود است؟ |
| ورودی‌ها | |
| car_entered | ورود یک خودرو |
| is_uni_car_entered | آیا خودرو وارد شده متعلق به دانشگاه است؟ |
| car_exited | خروج یک خودرو |
| is_uni_car_exited | آیا خودرو خارج شده متعلق به دانشگاه است؟ |

پیاده‌سازی ماژول:

با قطعه کد زیر، ماژول شبیه‌ساز مدیریت پارکینگ را پیاده‌سازی کردیم.

```
module parking_management (
    input clk,
    input reset,
    input car_entered,
    input is_uni_car_entered,
    input car_exited,
    input is_uni_car_exited,
    input [4:0] current_hour,
    output reg [8:0] uni_parked_car,
    output reg [8:0] parked_car,
    output reg [8:0] uni_vacated_space,
    output reg [8:0] vacated_space,
    output reg uni_is_vacated_space,
    output reg is_vacated_space
);

localparam MAX_UNI_CAPACITY = 500;
localparam MAX_TOTAL_CAPACITY = 700;
localparam BASE_FREE_CAPACITY = 200;

reg [8:0] total_free_capacity;

always @ (posedge clk or posedge reset) begin
    if (reset) begin
        uni_parked_car <= 0;
        parked_car <= 0;
        total_free_capacity <= BASE_FREE_CAPACITY;
    end else begin
        case (current_hour)
            5'd8: total_free_capacity <= BASE_FREE_CAPACITY;
            5'd9: total_free_capacity <= BASE_FREE_CAPACITY;
            5'd10: total_free_capacity <= BASE_FREE_CAPACITY;
            5'd11: total_free_capacity <= BASE_FREE_CAPACITY;
            5'd12: total_free_capacity <= BASE_FREE_CAPACITY;
            5'd13: total_free_capacity <= BASE_FREE_CAPACITY + 50;
            5'd14: total_free_capacity <= BASE_FREE_CAPACITY + 100;
            5'd15: total_free_capacity <= BASE_FREE_CAPACITY + 150;
            5'd16: total_free_capacity <= BASE_FREE_CAPACITY + 300;
            default: total_free_capacity <= BASE_FREE_CAPACITY;
        endcase
    end
end

always @ (posedge clk or posedge reset) begin
    if (reset) begin
        uni_parked_car <= 0;
        parked_car <= 0;
    end else begin
        if (car_entered) begin
            if (is_uni_car_entered) begin
                if (uni_parked_car < MAX_UNI_CAPACITY) begin
                    uni_parked_car <= uni_parked_car + 1;
                end
            end else begin
                if (parked_car < total_free_capacity) begin
                    parked_car <= parked_car + 1;
                end
            end
        end
        if (car_exited) begin
            if (is_uni_car_exited) begin
                if (uni_parked_car > 0) begin
                    uni_parked_car <= uni_parked_car - 1;
                end
            end else begin
                if (parked_car > 0) begin
                    parked_car <= parked_car - 1;
                end
            end
        end
    end
end

always @ (posedge clk or posedge reset) begin
    if (reset) begin
        uni_vacated_space <= MAX_UNI_CAPACITY;
        vacated_space <= total_free_capacity;
    end else begin
        uni_vacated_space <= MAX_UNI_CAPACITY - uni_parked_car;
        vacated_space <= total_free_capacity - parked_car;
        uni_is_vacated_space <= (uni_vacated_space > 0) ? 1'b1 : 1'b0;
        is_vacated_space <= (vacated_space > 0) ? 1'b1 : 1'b0;
    end
end

endmodule
```

پیادهسازی Test Bench:

حال برای بررسی صحت عملکرد آن، از قطعه کد Test Bench زیر استفاده کردیم. توضیحات مربوط به هر بخش از تست بنچ، به صورت کامنت در کد نوشته شده است.

```

module parking_management_tb;

reg clk, reset, car_entered, is_uni_car_entered, car_exited, is_uni_car_exited;
reg [4:0] current_hour;
wire uni_is_vacated_space, is_vacated_space;
wire [8:0] uni_parked_car, parked_car, uni_vacated_space, vacated_space;

parking_management uut (
    .clk(clk),
    .reset(reset),
    .car_entered(car_entered),
    .is_uni_car_entered(is_uni_car_entered),
    .car_exited(car_exited),
    .is_uni_car_exited(is_uni_car_exited),
    .current_hour(current_hour),
    .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)
);

initial begin

    clk = 0;
    reset = 1;
    car_entered = 0;
    is_uni_car_entered = 0;
    car_exited = 0;
    is_uni_car_exited = 0;
    current_hour = 5'd8; // initialize start time at 8 AM

    $display("Time\tCurrent Hour\tCar Entered\tUni Car Entered\tCar Exited\tUni Car Exited\tUni Parked\tFree Parked\tUni Vacant\tFree Vacant\tUni Vacant Space\tFree Vacant Space");

    $monitor("%d\t%d\t\t%d\t\t%d\t\t%d\t\t%d\t\t%d\t\t%d\t\t%d\t\t%d\t\t",
        $time, current_hour, car_entered, is_uni_car_entered, car_exited, is_uni_car_exited,
        uni_parked_car, parked_car, uni_vacated_space, vacated_space,
        uni_is_vacated_space, is_vacated_space);

    #10;
    reset = 0;

    // Enter a university car at 8 AM
    car_entered = 1;
    is_uni_car_entered = 1;
    #10;
    car_entered = 0;

    // Enter a free capacity car at 8 AM
    car_entered = 1;
    is_uni_car_entered = 0;
    #10;
    car_entered = 0;

    // Exit a university car at 8 AM
    car_exited = 1;
    is_uni_car_exited = 1;
    #10;
    car_exited = 0;

    // Exit a free capacity car at 8 AM
    car_exited = 1;
    is_uni_car_exited = 0;
    #10;
    car_exited = 0;

    // Set time to 13 PM
    current_hour = 5'd13;
    #10;

    // Enter a free capacity car at 13 PM
    car_entered = 1;
    is_uni_car_entered = 0;
    #10;
    car_entered = 0;

    // Set time to 16 PM
    current_hour = 5'd16;
    #10;

    // Enter a free capacity car at 16 PM
    car_entered = 1;
    is_uni_car_entered = 0;
    #10;
    car_entered = 0;

    #100;
    $finish;

end

always #5 clk = ~clk;

endmodule

```

سنتز و آنالیز:

در نهایت به سنتز مدارمان برای یک FPGA دلخواه پرداختیم. گزارش آن را می‌توانید در ادامه مشاهده کنید:

| | |
|--------------------|--|
| Quartus II Version | Version 13.1.0 Build 162 10/23/2013 SJ Web Edition |
| Revision Name | parking |
| Device Family | MAX V |
| Device Name | 5M160ZT100C4 |
| Timing Models | Final |
| Delay Model | Slow Model |
| Rise/Fall Delays | Unavailable |

در این حالت، Fmax ما به صورت زیر محاسبه شد:

| Fmax Summary | | | | |
|--------------|-----------|-----------------|------------|------|
| | Fmax | Restricted Fmax | Clock Name | Note |
| 1 | 79.54 MHz | 79.54 MHz | clk | |

در تصویر زیر، مسیرهایی که بیش‌ترین تاخیر را دارند مشخص شده‌اند. همان‌طور که مشاهده می‌کنید، بیشترین Data Delay برابر با ۱۲/۳۱۳ است که اگر آن را معکوس کنیم به فرکانس تقریبی ۸۱/۲ مگاهرتز می‌رسیم که تقریباً برابر با عدد نمایش داده شده در Fmax Summary است.

| index | Slack | From Node | To Node | Launch clock | Latch Clock | Relationship | Clock Skew | Data Delay |
|-------|---------|------------------------|--------------------|--------------|-------------|--------------|------------|------------|
| 1 | -11.573 | total_free_capacity[8] | parked_car[2]~reg0 | clk | clk | 1 | 0 | 12.313 |
| 2 | -11.573 | total_free_capacity[8] | parked_car[1]~reg0 | clk | clk | 1 | 0 | 12.313 |
| 3 | -11.559 | total_free_capacity[8] | parked_car[5]~reg0 | clk | clk | 1 | 0 | 12.299 |
| 4 | -11.503 | parked_car[7]~reg0 | parked_car[8]~reg0 | clk | clk | 1 | 0 | 12.243 |
| 5 | -11.464 | parked_car[7]~reg0 | parked_car[8]~reg0 | clk | clk | 1 | 0 | 12.204 |
| 6 | -11.389 | parked_car[1]~reg0 | parked_car[2]~reg0 | clk | clk | 1 | 0 | 12.129 |
| 7 | -11.389 | parked_car[1]~reg0 | parked_car[1]~reg0 | clk | clk | 1 | 0 | 12.129 |
| 8 | -11.378 | parked_car[7]~reg0 | parked_car[8]~reg0 | clk | clk | 1 | 0 | 12.118 |
| 9 | -11.375 | parked_car[1]~reg0 | parked_car[5]~reg0 | clk | clk | 1 | 0 | 12.115 |
| 10 | -11.351 | parked_car[7]~reg0 | parked_car[8]~reg0 | clk | clk | 1 | 0 | 12.091 |
| 11 | -11.284 | parked_car[8]~reg0 | parked_car[8]~reg0 | clk | clk | 1 | 0 | 12.024 |
| 12 | -11.275 | parked_car[7]~reg0 | parked_car[7]~reg0 | clk | clk | 1 | 0 | 12.015 |
| 13 | -11.259 | parked_car[7]~reg0 | parked_car[8]~reg0 | clk | clk | 1 | 0 | 11.999 |
| 14 | -11.252 | parked_car[7]~reg0 | parked_car[8]~reg0 | clk | clk | 1 | 0 | 11.992 |
| 15 | -11.245 | parked_car[8]~reg0 | parked_car[8]~reg0 | clk | clk | 1 | 0 | 11.985 |
| 16 | -11.237 | parked_car[7]~reg0 | parked_car[8]~reg0 | clk | clk | 1 | 0 | 11.977 |
| 17 | -11.236 | parked_car[7]~reg0 | parked_car[7]~reg0 | clk | clk | 1 | 0 | 11.976 |
| 18 | -11.159 | parked_car[8]~reg0 | parked_car[8]~reg0 | clk | clk | 1 | 0 | 11.899 |
| 19 | -11.15 | parked_car[7]~reg0 | parked_car[7]~reg0 | clk | clk | 1 | 0 | 11.89 |
| 20 | -11.132 | parked_car[8]~reg0 | parked_car[8]~reg0 | clk | clk | 1 | 0 | 11.872 |