پایانترم جبرانی طراحی سیستم‌های دیجیتال

دقت کنید که سوال هشتم را انتخاب کردیم.

سوال هشتم

قسمت الف)

در این سوال باید یک مدار برای پارکینگ طراحی کنیم. علاوه بر ورودی های ذکر شده در سوال، یک ورودی جدید به نام hour به مدار اضافه کردیم که نشان‌دهنده ساعت فعلی است. ماژول parking را به صورت زیر طراحی کردیم:

```
module parking #(
  parameter UNI_CAPACITY = 500,
  parameter MISC_CAPACITY = 200,
  parameter STEP = 50
) (
  input wire car_entere
  input wiredis_uni_car_entere
  input wirecar_exited,
  input wire is_uni_car_exited,
  input wire clk,
  input wire [5:0] hour,
  input wire enabled,
  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 wire uni_is_vacated_space,
  output wire is_vacated_space,
  output reg faul
);          t

  initial begin
    uni_parked_car = 0;
    parked_car = 0;
    fault = 0;
  end

  always @(posedge clk) begin
    if (enabled) begin

      if (car_entered) begin
        if (is_uni_car_entered && uni_vacated_space > 0) begin
          uni_parked_car = uni_parked_car + 1;
        end else if (!is_uni_car_entered && vacated_space > 0) begin
          parked_car = parked_car + 1;
        end
      end

      if (car_exited) begin
        if (is_uni_car_exited && uni_parked_car > 0) begin
          uni_parked_car = uni_parked_car - 1;
        end else if (!is_uni_car_exited && parked_car > 0) begin
          parked_car = parked_car - 1;
        end
      end

      if (hour >= 8 && hour < 13) begin
        uni_vacated_space <= UNI_CAPACITY - uni_parked_car;
        vacated_space <= MISC_CAPACITY - parked_car;
      end else if (hour >= 13 && hour < 16) begin
        uni_vacated_space <= UNI_CAPACITY - (hour - 12) * STEP - uni_parked_car;
        vacated_space <= MISC_CAPACITY + (hour - 12) * STEP - parked_car;
      end else if (hour >= 16) begin
        uni_vacated_space <= MISC_CAPACITY - uni_parked_car;
        vacated_space <= UNI_CAPACITY - parked_car;
      end


      if ((hour >= 13 && hour < 16 && UNI_CAPACITY < (hour - 12) * STEP +
 uni_parked_car) ||
          (hour >= 16 && (MISC_CAPACITY < uni_parked_car || UNI_CAPACITY < parked_car)))
begin
        fault <= 1;
      end else begin
        fault <= 0;
      end
    end
  end

  assign uni_is_vacated_space = uni_vacated_space > 0;
  assign is_vacated_space = vacated_space > 0;

endmodule
```

فایل parking.v نیز موجود است در صورتی که عکس واضح نیست، به فایل مربوطه مراجعه کنید.
در این طراحی، ظرفیت پارکینگ برای اساتید (UNI_CAPACITY)، ظرفیت پارکینگ برای عموم (MISC_CAPACITY) و مقداری که در هر ساعت به ظرفیت آزاد افزوده می‌شود (STEP) را به صورت پارامتر برای ماژول در نظر گرفتیم.

علاوه بر تمام ورودی و خروجی های ذکر شده در سوال، ورودی clk، enabled و hour را به مدار اضافه کردیم. ورودی enabled فعال بودن یا نبودن مدار را تعیین می‌کند.
برای این که مدار قابل سنتز باشد، در یک بلوک initial سیگنال ها را مقدار اولیه داده‌ایم.
در یک بلوک always حساس به لبه بالارونده clk، اگر enabled فعال بود، خروجی های مدار را با توجه به ورودی ها مقدار می‌دهیم:

1. در قسمت اول، اگر سیگنال car_entered یک بود، با توجه سیگنال is_uni_car_entered و همچنین وضعیت فعلی پارکینگ، در یک if else تعداد ماشین‌های پارکینگ را مقداردهی می‌کنیم.
2. در قسمت دوم، اگر سیگنال car_exited یک بود، با توجه به سیگنال is_uni_car_exited و همچنین وضعیت فعلی پارکینگ، در یک if else تعداد ماشین‌های پارکینگ را مقداردهی می‌کنیم.
3. در قسمت سوم با توجه hour ظرفیت کل پارکینگ را مقداردهی می‌کنیم.
4. در قسمت چهارم، سیگنال fault را مقداردهی می‌کنیم. برای مثال اگر ظرفیت پارکینگ بعد از کاهش، از تعداد ماشین‌های پارکینگ کمتر شد، این سیگنال یک می‌شود.

در آخر با استفاده از دو assign، سیگنال های uni_is_vacated_space و is_vacated_space را مقداردهی می‌کنیم.

در ماژول تست، به صورت زیر از ماژول parking نمونه گرفتیم:

```
module parking_tb;
  parameter UNI_CAPACITY = 500;
  parameter MISC_CAPACITY = 200;
  parameter STEP = 50;

  reg car_entere
  regdis_uni_car_entere
  regdcar_exited;
  reg is_uni_car_exited;
  reg clk;
  reg [5:0] hour;
  reg enabled;

  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;
  wire fault;

  parking #(
    .UNI_CAPACITY(UNI_CAPACITY),
    .MISC_CAPACITY(MISC_CAPACITY),
    .STEP(STEP)
  ) uut (
    .car_entered(car_entered),
    .is_uni_car_entered(is_uni_car_entered),
    .car_exited(car_exited),
    .is_uni_car_exited(is_uni_car_exited),
    .clk(clk),
    .hour(hour),
    .enabled(enabled),
    .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),
    .fault(fault)
  );

  initial begin
    clk = 0;
    forever #5 clk = ~clk;
  end
```

سپس در ادامه ماژول، در یک بلوک inital:

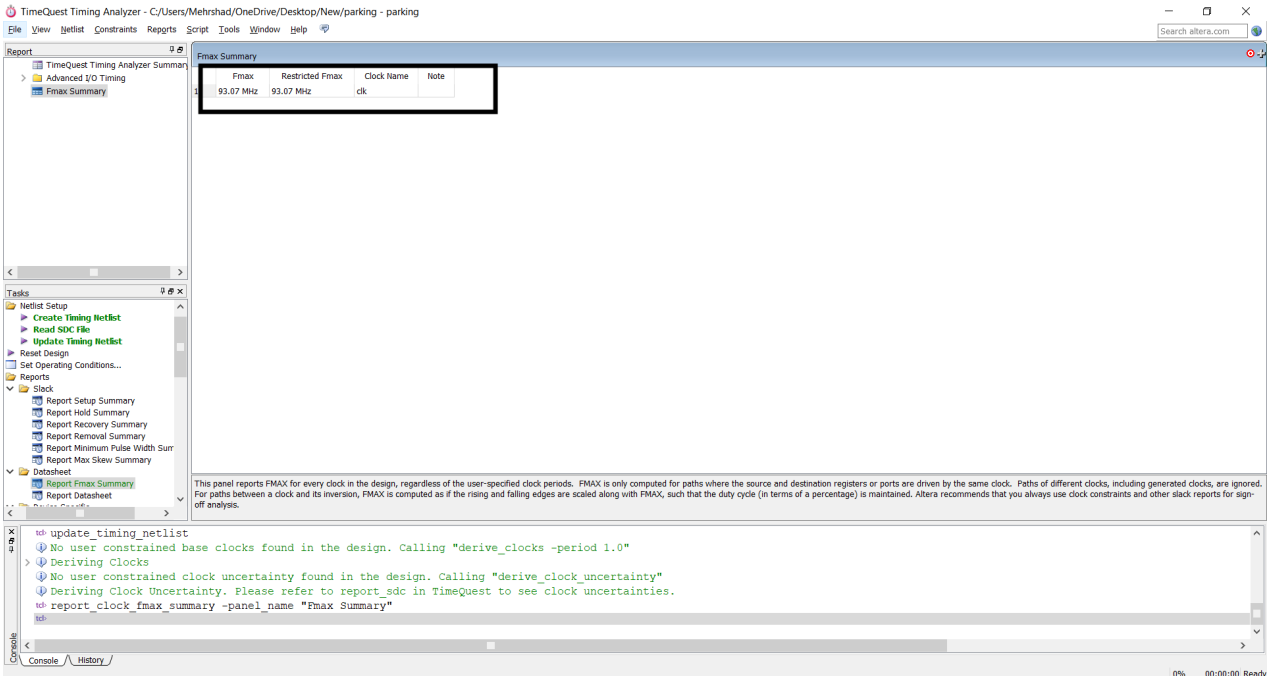
- ابتدا در ساعت ۸، ۵۰ بار به صورت تصادفی ورودی‌ها را مقدار دادیم.
- سپس ساعت را برابر ۱۳ قرار دادیم و مجددا ۵۰ بار ورودی ها را به صورت تصادفی ورودی‌ها را مقدار دادیم.
- سپس ساعت را برابر ۱۵ قرار دادیم و ۱۰ بار ورودی ها را به صورت تصادفی ورودی‌ها را مقدار دادیم.
- سپس ساعت را برابر ۱۷ قرار دادیم و ۲۰۰ بار ماشینی در قسمت university وارد کردیم تا این قسمت پر شود.

```
initial begin
    enabled = 1;
    hour = 8;
    car_entered = 0;
    is_uni_car_entered = 0;
    car_exited = 0;
    is_uni_car_exited = 0;

    $dumpfile("parking_tb.vcd");
    $dumpvars(0, parking_tb);



    // hour = 8:00
    repeat (50) begin
      #10 car_entered = $random % 2;
      is_uni_car_entered = $random % 2;
      car_exited = $random % 2;
      is_uni_car_exited = $random % 2;
    end

    // advance to 13:00
    hour = 13;
    repeat (50) begin
      #10 car_entered = $random % 2;
      is_uni_car_entered = $random % 2;
      car_exited = $random % 2;
      is_uni_car_exited = $random % 2;
    end

    // advance to 15:00
    hour = 15;
    repeat (10) begin
      #10 car_entered = $random % 2;
      is_uni_car_entered = $random % 2;
      car_exited = $random % 2;
      is_uni_car_exited = $random % 2;
    end

    // advance to 17:00
    hour = 17;
    // fill up the university section of the parking
    repeat (200) begin
      #10 car_entered = 1;
      is_uni_car_entered = 1;
      car_exited = 0;
      is_uni_car_exited = 0;
    end

    // Finish simulation
    #20 $finish;
  end
endmodule
```

خروجی به دلیل طولانی بودن، در فایل out.txt قرار داده شده است. همچنین فایل vcd و فایل اجرایی نیز وجود دارد.
فایل تولید شده توسط yosys بعد از سنتز نیز در فولدر synthesis with yosys موجود است.

قسمت ب)

برای بدست آوردن ماکزیمم فرکانسی که مدار می‌تواند کار کند، ابتدا در quartus ماژول را کامپایل می‌کنیم. سپس با استفاده از ابزار time quest analyzer این مقدار را مشاهده می‌کنیم:



همانطور که مشخص است، این مقدار 93 MHz است.