

دانشکدۀ مهندسی کامپیوتر

**گزارش پروژۀ درس طراحی سیستم­های دیجیتال**

استاد:

دکتر فصحتی

بردیا ماندگار | 401170637

**پرسش هشتم**

الف) در این سوال قصد داریم یک ماژول برای کنترل ورودی و خروجی یک پارکینگ طراحی کنیم. این پارکینگ که به طور شبانه­روزی فعال است، در کل 700 ماشین ظرفیت دارد و فضای آن در هر لحظه به نحوی بین دانشجویان و کارمندان دانشگاه و سایر افراد تقسیم می­شود.

این ماژول را ParkingController می­نامیم. ورودی­ها و خروجی­های آن در زیر لیست شده­ است که مفهوم هر یک در صورت سوال ذکر شده است.

    input clk,

    input car\_entered,

    input is\_uni\_car\_entered,

    input car\_exited,

    input is\_uni\_car\_exited,

    output reg [9:0] uni\_parked\_cars,

    output reg [9:0] parked\_cars,

    output reg [9:0] uni\_vacated\_space,

    output reg [9:0] vacated\_space,

    output uni\_is\_vacated\_space,

    output is\_vacated\_space

از آنجایی که در زمان­های متفاوت از روز این تقسیم ظرفیت به گونه­ای متفاوت انجام می­شود، مدار کنترلی باید درکی از زمان داشته باشد. برای این منظور متغیر زیر را تعریف می­کنیم.

integer second;

همچنین یک پارامتر مهم برای ما در طی محاسبات و تولید سیگنال­ها می­تواند ظرفیت سایر افراد باشد. با داشتن این پارامتر، می­توانیم با کم کردن آن از ظرفیت کلی، ظرفیت دانشجویان و کارمندان دانشگاه را به راحتی بدست آوریم. داشتن ظرفیت هر گروه نیز برای این مهم است که تعیین کنیم آیا جای خالی برای ماشین جدید وجود دارد یا خیر. پس این پارامتر کلیدی را نیز به صورت زیر تعریف می­کنیم.

integer total\_others\_space;

می­توانیم فرض کنیم این مدار اولین بار از ساعت 8 صبح فعال می­شود. در آن لحظه ظرفیت دانشجویان و کارمندان دانشگاه 500، و ظرفیت سایرین 200 است. بنابراین پارامترهای آن را در یک بلاک initial مطابق کد زیر مقداردهی می­کنیم.

initial begin

    second  = 8 \* 3600;

    uni\_parked\_cars = 0;

    parked\_cars = 0;

    uni\_vacated\_space = 500;

    vacated\_space = 200;

    total\_others\_space = 200;

end

همچنین خروجی­های uni\_is\_vacated\_space و is\_vacated\_space به سادگی برحسب خروجی­های دیگر بدست می­آیند. بنابراین آنها را از نوع wire تعریف کرده و با دستور assign مقداردهی می­کنیم.

assign uni\_is\_vacated\_space = (uni\_vacated\_space > 0);

assign is\_vacated\_space = (vacated\_space > 0);

حال منطق این ماژول را در یک بلاک always نسبتاً بلند تعیین می­کنیم. این بلاک صرفاً با لبۀ بالاروندۀ clk فعال می­شود. در ابتدای آن بررسی می­کنیم آیا در مرز بین دو ساعت مختلف قرار گرفته­ایم یا خیر. سپس در صورت مثبت بودن پاسخ، ساعت­های خاص را بررسی می­کنیم. ساعت­های خاص، ساعت­هایی هستند که تقسیم­بندی ظرفیت در آن تغییر می­کند که عبارتند از: 13، 14، 15، 16 و 8. اگر در یک ساعت خاص قرار داشتیم، پارامتر total\_others\_space را طبق دستور صورت سوال تغییر می­دهیم و بر اساس آن، تمام خروجی­های دیگر ساخته می­شوند. سپس یک ثانیه زمان را به جلو می­بریم. در مرحلۀ بعدی بررسی می­کنیم که آیا ماشینی قرار است وارد یا خارج شود یا خیر. در هر حالت مجدداً بررسی می­کنیم که آن ماشین مربوط به یک دانشجو یا کارمند دانشگاه است یا خیر و بر اساس آن، تعداد ماشین­های پارک شده و ظرفیت خالی آن گروه به روز می­شود. کد این بلاک به صورت زیر است.

always @(posedge clk) begin

    if (second % 3600 == 0)

        if (second == (24 + 8) \* 3600) begin

            second  = 8 \* 3600;

            total\_others\_space = 200;

            uni\_vacated\_space = 700 - total\_others\_space - uni\_parked\_cars;

            parked\_cars = (parked\_cars > total\_others\_space) ? total\_others\_space : parked\_cars;

            vacated\_space = total\_others\_space - parked\_cars;

        end

        else begin

            if (second == 13 \* 3600 || second == 14 \* 3600 || second == 15 \* 3600)

                total\_others\_space = total\_others\_space + 50;

            else if (second == 16 \* 3600)

                total\_others\_space = total\_others\_space + 150;

            uni\_parked\_cars = (uni\_parked\_cars > 700 - total\_others\_space) ? (700 - total\_others\_space) : uni\_parked\_cars;

            uni\_vacated\_space = 700 - total\_others\_space - uni\_parked\_cars;

            vacated\_space = total\_others\_space - parked\_cars;

        end

    second = second + 1;

    if (car\_entered) begin

        if (is\_uni\_car\_entered && uni\_vacated\_space > 0) begin

            uni\_parked\_cars = uni\_parked\_cars + 1;

            uni\_vacated\_space = uni\_vacated\_space - 1;

        end

        else if (~is\_uni\_car\_entered && vacated\_space > 0) begin

            parked\_cars = parked\_cars + 1;

            vacated\_space = vacated\_space - 1;

        end

    end

    if (car\_exited) begin

        if (is\_uni\_car\_exited && uni\_parked\_cars > 0) begin

            uni\_parked\_cars = uni\_parked\_cars - 1;

            uni\_vacated\_space = uni\_vacated\_space + 1;

        end

        else if (~is\_uni\_car\_exited && parked\_cars > 0) begin

            parked\_cars = parked\_cars - 1;

            vacated\_space = vacated\_space + 1;

        end

    end

end

در پایان کد نهایی ماژول به شکل زیر خواهد بود.

module ParkingController (

    input clk,

    input car\_entered,

    input is\_uni\_car\_entered,

    input car\_exited,

    input is\_uni\_car\_exited,

    output reg [9:0] uni\_parked\_cars,

    output reg [9:0] parked\_cars,

    output reg [9:0] uni\_vacated\_space,

    output reg [9:0] vacated\_space,

    output uni\_is\_vacated\_space,

    output is\_vacated\_space

);

integer second;

integer total\_others\_space;

initial begin

    second  = 8 \* 3600;

    uni\_parked\_cars = 0;

    parked\_cars = 0;

    uni\_vacated\_space = 500;

    vacated\_space = 200;

    total\_others\_space = 200;

end

assign uni\_is\_vacated\_space = (uni\_vacated\_space > 0);

assign is\_vacated\_space = (vacated\_space > 0);

always @(posedge clk) begin

    if (second % 3600 == 0)

        if (second == (24 + 8) \* 3600) begin

            second  = 8 \* 3600;

            total\_others\_space = 200;

            uni\_vacated\_space = 700 - total\_others\_space - uni\_parked\_cars;

            parked\_cars = (parked\_cars > total\_others\_space) ? total\_others\_space : parked\_cars;

            vacated\_space = total\_others\_space - parked\_cars;

        end

        else begin

            if (second == 13 \* 3600 || second == 14 \* 3600 || second == 15 \* 3600)

                total\_others\_space = total\_others\_space + 50;

            else if (second == 16 \* 3600)

                total\_others\_space = total\_others\_space + 150;

            uni\_parked\_cars = (uni\_parked\_cars > 700 - total\_others\_space) ? (700 - total\_others\_space) : uni\_parked\_cars;

            uni\_vacated\_space = 700 - total\_others\_space - uni\_parked\_cars;

            vacated\_space = total\_others\_space - parked\_cars;

        end

    second = second + 1;

    if (car\_entered) begin

        if (is\_uni\_car\_entered && uni\_vacated\_space > 0) begin

            uni\_parked\_cars = uni\_parked\_cars + 1;

            uni\_vacated\_space = uni\_vacated\_space - 1;

        end

        else if (~is\_uni\_car\_entered && vacated\_space > 0) begin

            parked\_cars = parked\_cars + 1;

            vacated\_space = vacated\_space - 1;

        end

    end

    if (car\_exited) begin

        if (is\_uni\_car\_exited && uni\_parked\_cars > 0) begin

            uni\_parked\_cars = uni\_parked\_cars - 1;

            uni\_vacated\_space = uni\_vacated\_space + 1;

        end

        else if (~is\_uni\_car\_exited && parked\_cars > 0) begin

            parked\_cars = parked\_cars - 1;

            vacated\_space = vacated\_space + 1;

        end

    end

end

endmodule

سپس جهت بررسی صحت عملکرد مدار، به سراغ طراحی ماژول Testbench می­رویم. ویژگی یک ماژول Testbench خوب، بررسی هرچه دقیق­تر حالت­های مختلف ورودی است. هدف ما نیز طراحی چنین ماژولی است.

این ماژول را ParkingController\_tb می­نامیم. حال متغیرهای متناظر با ورودی­ها و خروجی­های ماژول اصلی را به ترتیب از نوع reg و wire تعریف می­کنیم.

  reg clk;

  reg car\_entered;

  reg is\_uni\_car\_entered;

  reg car\_exited;

  reg is\_uni\_car\_exited;

  wire [9:0] uni\_parked\_cars;

  wire [9:0] parked\_cars;

  wire [9:0] uni\_vacated\_space;

  wire [9:0] vacated\_space;

  wire uni\_is\_vacated\_space;

  wire is\_vacated\_space;

  integer i;

سپس یک instance از ماژول ParkingController می­سازیم و متغیرهای تعریف شده را به عنوان argument به آن می­دهیم تا عملکرد آن را مورد آزمون قرار دهیم.

  ParkingController uut (

    .clk(clk),

    .car\_entered(car\_entered),

    .is\_uni\_car\_entered(is\_uni\_car\_entered),

    .car\_exited(car\_exited),

    .is\_uni\_car\_exited(is\_uni\_car\_exited),

    .uni\_parked\_cars(uni\_parked\_cars),

    .parked\_cars(parked\_cars),

    .uni\_vacated\_space(uni\_vacated\_space),

    .vacated\_space(vacated\_space),

    .uni\_is\_vacated\_space(uni\_is\_vacated\_space),

    .is\_vacated\_space(is\_vacated\_space)

  );

در مرحلۀ بعد در یک بلاک initial متغیرها را مقداردهی اولیه می­کنیم و همچنین دوره تناوب clk را نیز 2 واحد زمانی تعیین می­کنیم.

  initial begin

    clk = 1;

    car\_entered = 0;

    is\_uni\_car\_entered = 0;

    car\_exited = 0;

    is\_uni\_car\_exited = 0;

    forever #1 clk = ~clk;

  end

حال در یک بلاک initial دیگر ورودی­های مختلف را می­سازیم. سعی شده است تا در 6 مجموعه تست مختلف عملکرد مدار مورد آزمون قرار گیرد.

* تست 1

این تست مربوط به ساعت 8:00 است. در این زمان می­بایست 500 جایگاه به دانشجویان و کارمندان دانشگاه اختصاص یابد و 200 جایگاه به سایر افراد.

در این تست 600 دانشجو/کارمند دانشگاه و 300 فرد عادی سعی می­کنند تا ماشین خود را در پارکینگ پارک کنند، اما فقط به تعداد ظرفیت مشخص شده این امر قابل انجام است و سایر افراد موفق به پارک کردن در پارکینگ نمی­شوند. سپس 600 درخواست خروج از طرف افراد دانشجو/کارمند دانشگاه و 300 درخواست خروج از طرف افراد دیگر به مدار ارسال می­شود ولی مدار پس از خالی شدن پارکینگ از هر یک از جمعیت­ها، دیگر در را برای خروج باز نمی­کند، چرا که اصلاً ماشینی برای خارج شدن وجود ندارد!

$display("--------------------------------------------------------------Test Part 1--------------------------------------------------------------");

    $display("------------------------------------------------------------Start Time: 08:00----------------------------------------------------------");

    car\_entered = 1;

    is\_uni\_car\_entered = 1;

    for (i = 0; i < 600; i=i+1)

        #2;

    is\_uni\_car\_entered = 0;

    for (i = 0; i < 300; i=i+1)

        #2;

    car\_entered = 0;

    #2;

    car\_exited = 1;

    is\_uni\_car\_exited = 1;

    for (i = 0; i < 600; i=i+1)

        #2;

    is\_uni\_car\_exited = 0;

    for (i = 0; i < 300; i=i+1)

        #2;

    car\_exited = 0;

A screenshot of a computer screen

Description automatically generated

* تست 2

این تست مربوط به ساعت 13:00 است. در این زمان می­بایست 450 جایگاه به دانشجویان و کارمندان دانشگاه اختصاص یابد و 250 جایگاه به سایر افراد.

در این تست 5 دانشجو/کارمند دانشگاه و 5 فرد عادی ماشین خود را در پارکینگ پارک می­کنند و پس از مدتی از آن خارج می­کنند.

 $display("--------------------------------------------------------------Test Part 2--------------------------------------------------------------");

    $display("------------------------------------------------------------Start Time: 13:00----------------------------------------------------------");

    #32398;

    car\_entered = 1;

    is\_uni\_car\_entered = 1;

    for (i = 0; i < 5; i=i+1)

        #2;

    is\_uni\_car\_entered = 0;

    for (i = 0; i < 5; i=i+1)

        #2;

    car\_entered = 0;

    #2;

    car\_exited = 1;

    is\_uni\_car\_exited = 1;

    for (i = 0; i < 5; i=i+1)

        #2;

    is\_uni\_car\_exited = 0;

    for (i = 0; i < 5; i=i+1)

        #2;

    car\_exited = 0;

A table with numbers and letters

Description automatically generated with medium confidence

* تست 3

این تست مربوط به ساعت 14:00 است. در این زمان می­بایست 400 جایگاه به دانشجویان و کارمندان دانشگاه اختصاص یابد و 300 جایگاه به سایر افراد.

در این تست 5 دانشجو/کارمند دانشگاه و 5 فرد عادی ماشین خود را در پارکینگ پارک می­کنند و پس از مدتی از آن خارج می­کنند.

$display("--------------------------------------------------------------Test Part 3--------------------------------------------------------------");

    $display("------------------------------------------------------------Start Time: 14:00----------------------------------------------------------");

    #7158;

    car\_entered = 1;

    is\_uni\_car\_entered = 1;

    for (i = 0; i < 5; i=i+1)

        #2;

    is\_uni\_car\_entered = 0;

    for (i = 0; i < 5; i=i+1)

        #2;

    car\_entered = 0;

    #2;

    car\_exited = 1;

    is\_uni\_car\_exited = 1;

    for (i = 0; i < 5; i=i+1)

        #2;

    is\_uni\_car\_exited = 0;

    for (i = 0; i < 5; i=i+1)

        #2;

    car\_exited = 0;

A blue and white screen shot of a table

Description automatically generated with medium confidence

* تست 4

این تست مربوط به ساعت 15:00 است. در این زمان می­بایست 350 جایگاه به دانشجویان و کارمندان دانشگاه اختصاص یابد و 350 جایگاه به سایر افراد.

در این تست 5 دانشجو/کارمند دانشگاه و 5 فرد عادی ماشین خود را در پارکینگ پارک می­کنند و پس از مدتی از آن خارج می­کنند.

 $display("--------------------------------------------------------------Test Part 4--------------------------------------------------------------");

    $display("------------------------------------------------------------Start Time: 15:00----------------------------------------------------------");

    #7158;

    car\_entered = 1;

    is\_uni\_car\_entered = 1;

    for (i = 0; i < 5; i=i+1)

        #2;

    is\_uni\_car\_entered = 0;

    for (i = 0; i < 5; i=i+1)

        #2;

    car\_entered = 0;

    #2;

    car\_exited = 1;

    is\_uni\_car\_exited = 1;

    for (i = 0; i < 5; i=i+1)

        #2;

    is\_uni\_car\_exited = 0;

    for (i = 0; i < 5; i=i+1)

        #2;

    car\_exited = 0;

A blue and white screen shot of a list

Description automatically generated

* تست 5

این تست مربوط به ساعت 16:00 است. در این زمان می­بایست 200 جایگاه به دانشجویان و کارمندان دانشگاه اختصاص یابد و 500 جایگاه به سایر افراد.

در این تست 5 دانشجو/کارمند دانشگاه و 5 فرد عادی ماشین خود را در پارکینگ پارک می­کنند و پس از مدتی از آن خارج می­کنند.

$display("--------------------------------------------------------------Test Part 5--------------------------------------------------------------");

    $display("------------------------------------------------------------Start Time: 16:00----------------------------------------------------------");

    #7158;

    car\_entered = 1;

    is\_uni\_car\_entered = 1;

    for (i = 0; i < 5; i=i+1)

        #2;

    is\_uni\_car\_entered = 0;

    for (i = 0; i < 5; i=i+1)

        #2;

    car\_entered = 0;

    #2;

    car\_exited = 1;

    is\_uni\_car\_exited = 1;

    for (i = 0; i < 5; i=i+1)

        #2;

    is\_uni\_car\_exited = 0;

    for (i = 0; i < 5; i=i+1)

        #2;

    car\_exited = 0;

A blue and white text on a white background

Description automatically generated

* تست 6

این تست مربوط به ساعت 8:00 روز بعد است. در این زمان می­بایست مجدداً 500 جایگاه به دانشجویان و کارمندان دانشگاه اختصاص یابد و 200 جایگاه به سایر افراد.

در این تست نیز 600 دانشجو/کارمند دانشگاه و 300 فرد عادی سعی می­کنند تا ماشین خود را در پارکینگ پارک کنند، اما فقط به تعداد ظرفیت مشخص شده این امر قابل انجام است و سایر افراد موفق به پارک کردن در پارکینگ نمی­شوند. سپس هیچ یک از این افراد تا ساعت 13:00 ماشین خود را خارج نمی­کند و به ناچار در آن ساعت، 50 ماشین متعلق به دانشجو/کارمند دانشگاه از پارکینگ خارج می­شود تا جا برای افراد دیگر باز شود (مثلا با صاحب آن ماشین­ها تماس گرفته می­شود تا فوراً برای خارج کردن ماشین خود اقدام کنند)! بدین ترتیب 50 ظرفیت خالی برای سایر افراد ایجاد می­شود.

$display("--------------------------------------------------------------Test Part 6--------------------------------------------------------------");

    $display("------------------------------------------------------------Start Time: 08:00----------------------------------------------------------");

    #115170;

    car\_entered = 1;

    is\_uni\_car\_entered = 1;

    for (i = 0; i < 600; i=i+1)

        #2;

    is\_uni\_car\_entered = 0;

    for (i = 0; i < 300; i=i+1)

        #2;

    car\_entered = 0;

    #34200;

A blue and white screen shot of a table

Description automatically generated

لازم به ذکر است خروجی تست­های 1 و 6 به دلیل طولانی بودن به طور کامل قرار نگرفته­اند. لذا توصیه می­شود لطفاً اقدام به run کردن Testbench بفرمایید.

در پایان نیز در یک بلاک initial به مانیتور کردن خروجی­ها پرداختیم.

initial begin

    $monitor("Time: %6d, Uni Parked Cars: %d, Parked Cars: %d, Uni Vacated Space: %d, Vacated Space: %d, Uni Vacancy: %b, Other Vacancy: %b",

             $time, uni\_parked\_cars, parked\_cars, uni\_vacated\_space, vacated\_space, uni\_is\_vacated\_space, is\_vacated\_space);

  end

در پایان کد نهایی ماژول به شکل زیر خواهد بود.

module ParkingController\_tb;

  reg clk;

  reg car\_entered;

  reg is\_uni\_car\_entered;

  reg car\_exited;

  reg is\_uni\_car\_exited;

  wire [9:0] uni\_parked\_cars;

  wire [9:0] parked\_cars;

  wire [9:0] uni\_vacated\_space;

  wire [9:0] vacated\_space;

  wire uni\_is\_vacated\_space;

  wire is\_vacated\_space;

  integer i;

  ParkingController uut (

    .clk(clk),

    .car\_entered(car\_entered),

    .is\_uni\_car\_entered(is\_uni\_car\_entered),

    .car\_exited(car\_exited),

    .is\_uni\_car\_exited(is\_uni\_car\_exited),

    .uni\_parked\_cars(uni\_parked\_cars),

    .parked\_cars(parked\_cars),

    .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 = 1;

    car\_entered = 0;

    is\_uni\_car\_entered = 0;

    car\_exited = 0;

    is\_uni\_car\_exited = 0;

    forever #1 clk = ~clk;

  end

  initial begin

    $display("--------------------------------------------------------------Test Part 1--------------------------------------------------------------");

    $display("------------------------------------------------------------Start Time: 08:00----------------------------------------------------------");

    car\_entered = 1;

    is\_uni\_car\_entered = 1;

    for (i = 0; i < 600; i=i+1)

        #2;

    is\_uni\_car\_entered = 0;

    for (i = 0; i < 300; i=i+1)

        #2;

    car\_entered = 0;

    #2;

    car\_exited = 1;

    is\_uni\_car\_exited = 1;

    for (i = 0; i < 600; i=i+1)

        #2;

    is\_uni\_car\_exited = 0;

    for (i = 0; i < 300; i=i+1)

        #2;

    car\_exited = 0;

    $display("--------------------------------------------------------------Test Part 2--------------------------------------------------------------");

    $display("------------------------------------------------------------Start Time: 13:00----------------------------------------------------------");

    #32398;

    car\_entered = 1;

    is\_uni\_car\_entered = 1;

    for (i = 0; i < 5; i=i+1)

        #2;

    is\_uni\_car\_entered = 0;

    for (i = 0; i < 5; i=i+1)

        #2;

    car\_entered = 0;

    #2;

    car\_exited = 1;

    is\_uni\_car\_exited = 1;

    for (i = 0; i < 5; i=i+1)

        #2;

    is\_uni\_car\_exited = 0;

    for (i = 0; i < 5; i=i+1)

        #2;

    car\_exited = 0;

    $display("--------------------------------------------------------------Test Part 3--------------------------------------------------------------");

    $display("------------------------------------------------------------Start Time: 14:00----------------------------------------------------------");

    #7158;

    car\_entered = 1;

    is\_uni\_car\_entered = 1;

    for (i = 0; i < 5; i=i+1)

        #2;

    is\_uni\_car\_entered = 0;

    for (i = 0; i < 5; i=i+1)

        #2;

    car\_entered = 0;

    #2;

    car\_exited = 1;

    is\_uni\_car\_exited = 1;

    for (i = 0; i < 5; i=i+1)

        #2;

    is\_uni\_car\_exited = 0;

    for (i = 0; i < 5; i=i+1)

        #2;

    car\_exited = 0;

    $display("--------------------------------------------------------------Test Part 4--------------------------------------------------------------");

    $display("------------------------------------------------------------Start Time: 15:00----------------------------------------------------------");

    #7158;

    car\_entered = 1;

    is\_uni\_car\_entered = 1;

    for (i = 0; i < 5; i=i+1)

        #2;

    is\_uni\_car\_entered = 0;

    for (i = 0; i < 5; i=i+1)

        #2;

    car\_entered = 0;

    #2;

    car\_exited = 1;

    is\_uni\_car\_exited = 1;

    for (i = 0; i < 5; i=i+1)

        #2;

    is\_uni\_car\_exited = 0;

    for (i = 0; i < 5; i=i+1)

        #2;

    car\_exited = 0;

    $display("--------------------------------------------------------------Test Part 5--------------------------------------------------------------");

    $display("------------------------------------------------------------Start Time: 16:00----------------------------------------------------------");

    #7158;

    car\_entered = 1;

    is\_uni\_car\_entered = 1;

    for (i = 0; i < 5; i=i+1)

        #2;

    is\_uni\_car\_entered = 0;

    for (i = 0; i < 5; i=i+1)

        #2;

    car\_entered = 0;

    #2;

    car\_exited = 1;

    is\_uni\_car\_exited = 1;

    for (i = 0; i < 5; i=i+1)

        #2;

    is\_uni\_car\_exited = 0;

    for (i = 0; i < 5; i=i+1)

        #2;

    car\_exited = 0;

    $display("--------------------------------------------------------------Test Part 6--------------------------------------------------------------");

    $display("------------------------------------------------------------Start Time: 08:00----------------------------------------------------------");

    #115170;

    car\_entered = 1;

    is\_uni\_car\_entered = 1;

    for (i = 0; i < 600; i=i+1)

        #2;

    is\_uni\_car\_entered = 0;

    for (i = 0; i < 300; i=i+1)

        #2;

    car\_entered = 0;

    #34200;

    $stop;

  end

  initial begin

    $monitor("Time: %6d, Uni Parked Cars: %d, Parked Cars: %d, Uni Vacated Space: %d, Vacated Space: %d, Uni Vacancy: %b, Other Vacancy: %b",

             $time, uni\_parked\_cars, parked\_cars, uni\_vacated\_space, vacated\_space, uni\_is\_vacated\_space, is\_vacated\_space);

  end

endmodule

ب) در این بخش ابتدا یک پروژۀ جدید روی نرم­افزار Quartus ایجاد کردیم که شامل فایل verilog ماژول ParkingController بود. سپس آن را روی ماشین Max2 کامپایل کردیم و سنتز آن با موفقیت انجام شد. با رفتن به قسمت TimeQuest Timing Analyzer و گزینۀ Fmax Summary توانستیم حداکثر فرکانس clk این مدار را مشاهده کنیم. تصویر زیر گویای مراحل این بخش است.

A screenshot of a computer

Description automatically generated