**به نام خدا**

**طراحی سیستم‌های دیجیتال – پروژه‌ی جبرانی**

**مهدی شکوفی مقیمان**

**۴۰۱۱۱۰۱۱۵**

با توجه به خواسته‌های سوال، ماژول parking\_management را طراحی می‌کنیم:

ورودی‌ها و خروجی‌های مورد نیاز(همان ورودی و خروجی‌های متن سوال است، با توجه به نیازمان یک سیگنال ورودی current\_hour نیز اضافه کرده‌ایم):

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 [9:0] uni\_parked\_car,

    output reg [9:0] parked\_car,

    output reg [9:0] vacated\_space,

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

    output uni\_is\_vacated\_space,

    output is\_vacated\_space

);

تعریف محدودیت‌های پارکینگ:

//parking constraints

parameter MAX\_TOTAL\_CAPACITY = 700;

reg [9:0] max\_free\_capacity;

wire [9:0] uni\_space;

assign uni\_space = MAX\_TOTAL\_CAPACITY - max\_free\_capacity;

محاسبه‌ی میزان فضای خالی:

//determine free spaces

assign uni\_vacated\_space = uni\_space - uni\_parked\_car;

assign vacated\_space = max\_free\_capacity - parked\_car;

محاسبه‌ی خالی بودن پارکینگ:

//determine does free space exist

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

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

محاسبه‌ی ظرفیت‌ها با توجه به ساعت کنونی:

//determine capacity with current\_hour

always @(posedge clk or posedge reset) begin

    if (reset) begin

        max\_free\_capacity <= 500;

    end else begin

        if (current\_hour < 8)

            max\_free\_capacity <= 500;

        else if (current\_hour >= 8 && current\_hour < 13)

            max\_free\_capacity <= 200;

        else if (current\_hour >= 13 && current\_hour < 16)

            max\_free\_capacity <= 200 + (current\_hour - 12) \* 50;

        else

            max\_free\_capacity <= 500;

    end

end

قسمت اصلی کد(کنترل ورود و خروج ماشین‌ها):

//main

always @(posedge clk or posedge reset) begin

    if (reset) begin

        uni\_parked\_car <= 0;

        parked\_car <= 0;

    end else begin

        //car entered

        if (car\_entered) begin

            //uni car

            if (is\_uni\_car\_entered) begin

                if (uni\_is\_vacated\_space)

                    uni\_parked\_car <= uni\_parked\_car + 1;

            //normal car

            end else begin

                if (is\_vacated\_space)

                    parked\_car <= parked\_car + 1;

            end

        end

        //car exited

        if (car\_exited) begin

            //uni car

            if (is\_uni\_car\_exited) begin

                if (uni\_parked\_car > 0)

                    uni\_parked\_car <= uni\_parked\_car - 1;

            //normal car

            end else begin

                if (parked\_car > 0)

                    parked\_car <= parked\_car - 1;

            end

        end

    end

end

سپس برای این ماژول یک تست بنچ می‌نویسیم که تمام حالات خاص ممکن را پوشش دهد:

module TB;

reg clk;

reg reset;

reg car\_entered;

reg is\_uni\_car\_entered;

reg car\_exited;

reg is\_uni\_car\_exited;

reg [4:0] current\_hour;

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;

// get instant from parking management module

parking\_management PM (

    clk,

    reset,

    car\_entered,

    is\_uni\_car\_entered,

    car\_exited,

    is\_uni\_car\_exited,

    current\_hour,

    uni\_parked\_car,

    parked\_car,

    vacated\_space,

    uni\_vacated\_space,

    uni\_is\_vacated\_space,

    is\_vacated\_space

);

integer i;

//clock

always begin

    for (i = 0; i < 120; i = i + 1) begin

        #5 clk = ~clk;

    end

    current\_hour = current\_hour + 1;

    if (current\_hour > 23)

        current\_hour = 0;

end

initial begin

    //initialize

    clk = 0;

    reset = 1;

    car\_entered = 0;

    is\_uni\_car\_entered = 0;

    car\_exited = 0;

    is\_uni\_car\_exited = 0;

    current\_hour = 0;

    //reset

    #10 reset = 0;

    car\_entered = 1;

    #1200

    car\_entered = 0;

    #3600

    car\_entered = 1;

    is\_uni\_car\_entered = 1;

    #2000

    car\_entered = 0;

    is\_uni\_car\_entered = 0;

    #1000

    car\_exited = 1;

    #3000

    is\_uni\_car\_exited = 1;

    #3600

    is\_uni\_car\_exited = 0;

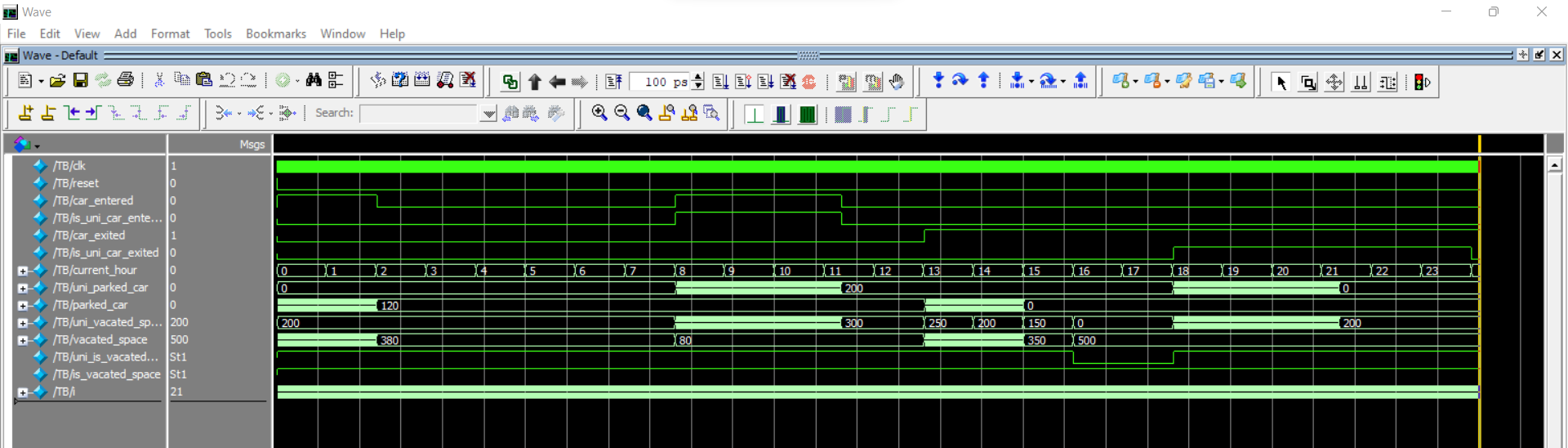
    //stop simulation

    #100 $stop;

end

endmodule

خروجی waveform:



همانطور که مشاهده می‌کنیم، خروجی مطابق انتظار است و طراحی‌مان درست می‌باشد.