Test Document

برای تست این ماژول ابتدا instance های مورد نیاز را از دو ماژول تعریف شده در قسمت Design، تعریف کرده و ورودی و خروجی های هر یک تنظیم گردید.

```
module TB;
    wire Process;
    wire Dir;
    wire [4:0] PushedButtons;
    reg clk;
    reg [4:0] InToggle;
    reg [4:0] OutToggle;
    wire [4:0] WhichFloor;
    wire [1:0] FloorInOut [4:0];

    Asansor asansor (
        .Clk(clk),
        .State({Dir, Process}),
        .WhichFloor(WhichFloor),
        .FloorInOut(FloorInOut)
    );

    Handler handler (
        .clk(clk),
        .WhichFloor(WhichFloor),
        .InToggle(InToggle),
        .OutToggle(OutToggle),
        .Dir(Dir),
        .Process(Process),
        .PushedButtons(PushedButtons)
    );
```

پس از آن در بلاک Always، سیگنال کلاک به ازای واحد زمانی ۱۰، مقداردهی شد.

دو task در این ماژول تعریف شده است که یکی برای دریافت فشرده شدن دکمه های درون و دیگری برای دکمه های بیرونی است که پس از گذشت مقدار delay_time دکمه آن طبقه، غیر فعال میشود.

```
    always #10 clk = ~clk;

    task handle_in_toggle(input [4:0] floor, input integer delay_time);
        begin
            InToggle[floor] = 1'b1;
            #delay_time InToggle[floor] = 1'b0;
        end
    endtask

    task handle_out_toggle(input [4:0] floor, input integer delay_time);
```

```
        begin
            OutToggle[floor] = 1'b1;
            #delay_time OutToggle[floor] = 1'b0;
        end
    endtask
```

در یک بلاک initial نیز سناریوی مورد نظر به آسانسور داده شد که در ادامه روند آن توضیح داده خواهد شد.

```
    initial begin
        clk = 0;
        InToggle = 0;
        OutToggle = 0;

        handle_out_toggle(0, 20);
        handle_in_toggle(2, 20);
        #400
        handle_out_toggle(1, 20);
        handle_in_toggle(2, 20);
        handle_out_toggle(3, 20);
        handle_in_toggle(0, 20);
        #400
        handle_out_toggle(2, 20);
        handle_out_toggle(4, 20);
        #400
        handle_in_toggle(0, 20);
        handle_in_toggle(2, 20);
        #1000;
        $stop();
    end
```

در انتها نیز در یک بلاک Always، تغییرات حالات طبقات، حرکت و جهت آسانسور و فشرده شدن دکمه های آسانسور، نمایش داده شد.

```
    reg [1:0] TmpFloor [4:0];
    reg TmpProcess;
    reg TmpDir;
    reg [4:0] TmpButtons;
    integer i;

    always @(posedge clk) begin
        if (FloorInOut !== TmpFloor || Process !== TmpProcess || Dir !== TmpDir ||
PushedButtons !== TmpButtons) begin
            $display("Time: %3d | PushedButtons: %b | Process: %b | Dir: %b | Floors: %s
| %s | %s | %s | %s",
                     $time, PushedButtons, Process, Dir,
                     FloorInOut[0] == 2'b00 ? "STOP" : (FloorInOut[0] == 2'b01 ? "IN" :
(FloorInOut[0] == 2'b10 ? "OUT" : "-")),
                     FloorInOut[1] == 2'b00 ? "STOP" : (FloorInOut[1] == 2'b01 ? "IN" :
(FloorInOut[1] == 2'b10 ? "OUT" : "-")),
```

```
                    FloorInOut[2] == 2'b00 ? "STOP" : (FloorInOut[2] == 2'b01 ? "IN" :
(FloorInOut[2] == 2'b10 ? "OUT" : "-")),
                    FloorInOut[3] == 2'b00 ? "STOP" : (FloorInOut[3] == 2'b01 ? "IN" :
(FloorInOut[3] == 2'b10 ? "OUT" : "-")),
                    FloorInOut[4] == 2'b00 ? "STOP" : (FloorInOut[4] == 2'b01 ? "IN" :
(FloorInOut[4] == 2'b10 ? "OUT" : "-")));
        end
        TmpFloor <= FloorInOut;
        TmpProcess <= Process;
        TmpDir <= Dir;
        TmpButtons <= PushedButtons;
    end
endmodule
```

پس از simulate **کردن تست طراحی شده خروجی زیر مشاهده شد**

```
# Time:   15 | PushedButtons: 00001 | Move: 0 | Direction: 0 | Floors: STOP |    - |    - |    - |    -
# Time:   25 | PushedButtons: 00100 | Move: 0 | Direction: 0 | Floors: STOP |    - |    - |    - |    -
# Time:   35 | PushedButtons: 00100 | Move: 1 | Direction: 1 | Floors: STOP |    - |    - |    - |    -
# Time:  105 | PushedButtons: 00100 | Move: 1 | Direction: 1 | Floors:  OUT |   IN |    - |    - |    -
# Time:  115 | PushedButtons: 00100 | Move: 1 | Direction: 1 | Floors:    - |  OUT |   IN |    - |    -
# Time:  125 | PushedButtons: 00000 | Move: 0 | Direction: 0 | Floors:    - |    - | STOP |    - |    -
# Time:  435 | PushedButtons: 00010 | Move: 0 | Direction: 0 | Floors:    - |    - | STOP |    - |    -
# Time:  445 | PushedButtons: 00110 | Move: 1 | Direction: 0 | Floors:    - |    - | STOP |    - |    -
# Time:  455 | PushedButtons: 01010 | Move: 0 | Direction: 0 | Floors:    - |    - | STOP |    - |    -
# Time:  465 | PushedButtons: 01011 | Move: 1 | Direction: 0 | Floors:    - |    - | STOP |    - |    -
# Time:  525 | PushedButtons: 01011 | Move: 1 | Direction: 0 | Floors:    - |   IN |  OUT |    - |    -
# Time:  535 | PushedButtons: 01001 | Move: 0 | Direction: 0 | Floors:    - | STOP |    - |    - |    -
# Time:  545 | PushedButtons: 01001 | Move: 1 | Direction: 1 | Floors:    - | STOP |    - |    - |    -
# Time:  635 | PushedButtons: 01001 | Move: 1 | Direction: 1 | Floors:    - |  OUT |   IN |    - |    -
# Time:  645 | PushedButtons: 01001 | Move: 0 | Direction: 0 | Floors:    - |    - | STOP |    - |    -
# Time:  655 | PushedButtons: 01001 | Move: 1 | Direction: 1 | Floors:    - |    - | STOP |    - |    -
# Time:  745 | PushedButtons: 01001 | Move: 1 | Direction: 1 | Floors:    - |    - |  OUT |   IN |    -
# Time:  755 | PushedButtons: 00001 | Move: 0 | Direction: 0 | Floors:    - |    - |    - | STOP |    -
# Time:  765 | PushedButtons: 00001 | Move: 1 | Direction: 0 | Floors:    - |    - |    - | STOP |    -
# Time:  855 | PushedButtons: 00001 | Move: 1 | Direction: 0 | Floors:    - |    - |   IN |  OUT |    -
# Time:  865 | PushedButtons: 00001 | Move: 1 | Direction: 0 | Floors:    - |   IN |  OUT |    - |    -
# Time:  875 | PushedButtons: 00101 | Move: 1 | Direction: 0 | Floors:   IN |  OUT |    - |    - |    -
# Time:  885 | PushedButtons: 10101 | Move: 0 | Direction: 0 | Floors: STOP |    - |    - |    - |    -
# Time:  895 | PushedButtons: 10100 | Move: 0 | Direction: 0 | Floors: STOP |    - |    - |    - |    -
# Time:  905 | PushedButtons: 10100 | Move: 1 | Direction: 1 | Floors: STOP |    - |    - |    - |    -
# Time:  985 | PushedButtons: 10100 | Move: 1 | Direction: 1 | Floors:  OUT |   IN |    - |    - |    -
# Time:  995 | PushedButtons: 10100 | Move: 1 | Direction: 1 | Floors:    - |  OUT |   IN |    - |    -
# Time: 1005 | PushedButtons: 10000 | Move: 0 | Direction: 0 | Floors:    - |    - | STOP |    - |    -
# Time: 1015 | PushedButtons: 10000 | Move: 1 | Direction: 1 | Floors:    - |    - | STOP |    - |    -
# Time: 1105 | PushedButtons: 10000 | Move: 1 | Direction: 1 | Floors:    - |    - |  OUT |   IN |    -
# Time: 1115 | PushedButtons: 10000 | Move: 1 | Direction: 1 | Floors:    - |    - |    - |  OUT |   IN
# Time: 1125 | PushedButtons: 00000 | Move: 0 | Direction: 0 | Floors:    - |    - |    - |    - | STOP
# Time: 1295 | PushedButtons: 00001 | Move: 0 | Direction: 0 | Floors:    - |    - |    - |    - | STOP
# Time: 1305 | PushedButtons: 00101 | Move: 1 | Direction: 0 | Floors:    - |    - |    - |    - | STOP
# Time: 1325 | PushedButtons: 00101 | Move: 1 | Direction: 0 | Floors:    - |    - |    - |   IN |  OUT
# Time: 1335 | PushedButtons: 00101 | Move: 1 | Direction: 0 | Floors:    - |    - |   IN |  OUT |    -
# Time: 1345 | PushedButtons: 00001 | Move: 0 | Direction: 0 | Floors:    - |    - | STOP |    - |    -
# Time: 1355 | PushedButtons: 00001 | Move: 1 | Direction: 0 | Floors:    - |    - | STOP |    - |    -
# Time: 1445 | PushedButtons: 00001 | Move: 1 | Direction: 0 | Floors:    - |   IN |  OUT |    - |    -
# Time: 1455 | PushedButtons: 00001 | Move: 1 | Direction: 0 | Floors:   IN |  OUT |    - |    - |    -
# Time: 1465 | PushedButtons: 00001 | Move: 0 | Direction: 0 | Floors: STOP |    - |    - |    - |    -
```

سناریوی تعریف شده برای آسانسور بدین صورت است که در ابتدا یک شخص دکمه طبقه همکف را میفشارد و پس از ورود به آسانسور، مقصد طبقه دوم را برای خود انتخاب میکند. پس از آن در لحظاتی بعد فردی در طبقه اول دکمه آسانسور را فشرده و مقصد طبقه دوم را برای خود انتخاب میکند و به طور همزمان فرد دیگری در طبقه سوم دکمه آسانسور را میفشارد که آسانسور ابتدا در طبقه اول نفر اول را سوار میکند . سپس او را در طبقه دوم پیاده کرده و پس از آن به طبقه سوم رفته و نفر بعدی را سوار کرده و به طبقه همکف میرود تا او نیز پیاده شود. پس از گذشت یک بازه زمانی، فرد دیگری در طبقه دوم دکمه آسانسور را فشرده و سوار میشود و نفر دیگری در طبقه چهارم نیز دکمه آسانسور را میفشارد که در ابتدا فرد در طبقه دوم و پس از آن فرد در طبقه چهارم سوار آسانسور میشوند. پس از گذشت یک بازه زمانی یکی از آنها ابتدا دکمه طبقه همکف را میفشارد و دیگری با تاخیر، دکمه طبقه دوم را میفشارد که آسانسور ابتدا در طبقه دوم توقف میکند و سپس به طبقه همکف میرود تا نفر دوم نیز پیاده شود.

در سناریوی طراحی شده سعی شد تا حرکات در جهات مختلف برای آسانسور در نظر گرفته شود تا تمامی موارد پوشش داده شوند. همچنین در آخرین بخش نیز مسئله بیان شده در صورت سوال که اگر در میانه راه آسانسور، طبقه دیگری فشرده شود، باید آنجا توقف کند نیز آورده شده است که در خروجی تست، صحت عملکرد آسانسور به ازای این ورودی مشهود است.