

بسمه تعالی



آزمایش شماره ۶

استاد

دکتر علیرضا اجلائی

سینا مظاهری

متین داغیانی

دانشگاه صنعتی شریف

۱۴۰۰

فهرست

مقدمه و اهداف آزمایش.....	3
تئوری آزمایش.....	Error! Bookmark not defined.
بخش ۱.....	Error! Bookmark not defined.
بخش ۲.....	Error! Bookmark not defined.
کارهای آزمایشگاهی.....	4
بخش ۱.....	Error! Bookmark not defined.
بخش ۲.....	Error! Bookmark not defined.

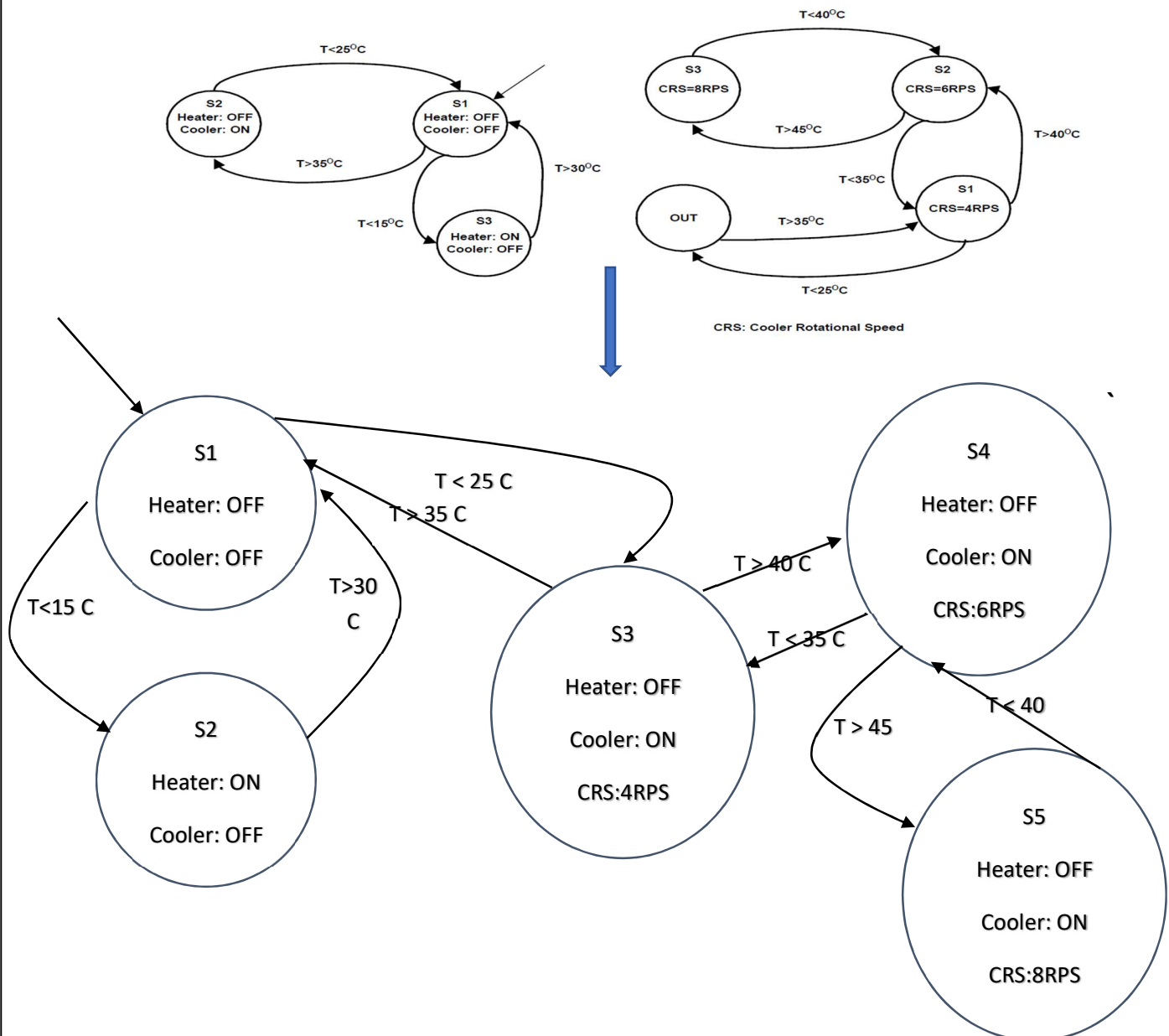
مقدمه و اهداف آزمایش

هدف از انجام این آزمایش طراحی یک انکوباتور با ویژگی های خواسته شده است.

تئوری آزمایش

ما در این قسمت تئوری آزمایش را شرح می دهیم بدین صورت که با کمی تغییر در ماشین حالت متناهی رسم شده در صورت سوال ، آن را به شکل معادلش یعنی ماشین حالت زیر تبدیل می کنیم.

روش کار واحد کنترل دیجیتال توسط دو نمودار حالت زیر توصیف شده است:



حال که توصیف خود را از ماشین حالت متناهی مورد نظرمان را بیان کردیم نوبت آن است که به پیاده سازی کد توصیف سخت افزار آن به شکل زیر می شود.

کد توصیف سخت افزار

کد توصیف سخت افزار ماشین حالت متناهی فوق به شکل زیر در عکس آمده است.

```
22 |
23 | module incubator(
24 |     input wire [7:0] T,
25 |     input wire clk,
26 |     input wire reset,
27 |     output reg Cooler,
28 |     output reg Heater,
29 |     output reg [7:0] CRS
30 | );
31 |
32 |     parameter S1 = 3'b000,
33 |                S2 = 3'b001,
34 |                S3 = 3'b010,
35 |                S4 = 3'b011,
36 |                S5 = 3'b100;
37 |
38 |     reg [2:0] current_state, next_state;
39 |     //
40 |     always @(posedge clk or negedge reset)
41 |     begin
42 |         if (!reset)
43 |             current_state <= S1;
44 |         else
45 |             current_state <= next_state;
46 |     end
47 |     //
48 |
```

در عبارت بالا ماژول انکوباتور، ۳ ورودی که شامل دما (۸ بیتی)، کلاک و ریست آسنکرون می باشد و در خروجی نیز دو سیگنال فعال ساز Heater و Cooler و یک ثبات ۸ بیتی که نشان دهنده تعداد دور های فن کولر هستند، آمده است. در عبارت داخل ماژول ۵ حالت به عنوان به پارامتر تعریف شده اند. دو متغیر که شامل حالت فعلی و حالت بعدی نیز می باشد، برای گذر بین حالات نیز، تعریف شده اند. در بلوک **always** اول شرایط مدار را در گذر بین حالات نشان می دهد. اگر ریست غیر فعال بود. مدار به حالت اولیه می رود.

```

always @(T or current_state)
begin
    next_state = current_state;
    case (current_state)
        S1:
            begin
                if (T < 15)
                    next_state = S2;
                else if (T > 35)
                    next_state = S3;
                else
                    next_state = S1;
            end
        S2:
            begin
                if (T > 30)
                    next_state = S1;
                else
                    next_state = S2;
            end
        S3:
            begin
                if (T > 40)
                    next_state = S4;
                else if (T < 25)
                    next_state = S1;
                else
                    next_state = S3;
            end
        S4:
            begin
                if (T > 45)
                    next_state = S5;
                else if (T < 35)
                    next_state = S3;
                else
                    next_state = S4;
            end
        S5:
            begin
                if (T < 40)
                    next_state = S4;
                else
                    next_state = S5;
            end
    endcase
end

```

در بلوک **always** دوم ، شرایط ورودی فلیپ فلاپ های حالت تعریف شده اند، یعنی با توجه به ورودی و حالت فعلی ما به چه حالت بعدی گذر خواهیم کرد.

```

always @(T or current_state)
begin
    Cooler = 0;
    Heater = 0;
    CRS = 0;
    case (current_state)
        S1:
            begin
                Heater = 0;
                Cooler = 0;
                CRS = 0;
            end
        S2:
            begin
                Heater = 1;
                Cooler = 0;
                CRS = 0;
            end
        S3:
            begin
                Heater = 0;
                Cooler = 1;
                CRS = 4;
            end
        S4:
            begin
                Heater = 0;
                Cooler = 1;
                CRS = 6;
            end
        S5:
            begin
                Heater = 0;
                Cooler = 1;
                CRS = 8;
            end
    endcase
end

endmodule

```

در بلوک **always** سوم ، شرایط حالت بررسی شده و با توجه به آن و از آن جا که ماشین از نوع مور می باشد، خروجی متناظر اعمال می شود تا مدار به درستی کار کند.

کارهای آزمایشگاهی

شبیه سازی

برای بررسی صحت عملکرد سیستم دیجیتال طراحی شده نیاز داریم تا آن را تست و شبیه سازی کنیم. بدین منظور مراحل زیر برای مشاهده صحت خروجی ها و گذار بین حالات در نظر گرفته شده اند.

۱. ریست مدار و مقداردهی اولیه کلاک: پیش از هر چیز نیاز است تا مدار را در وضعیت اولیه (S1) قرار دهیم.
۲. دما را به ۱۰ درجه تغییر می دهیم. مطابق دستور کار به وضعیت S2 می رویم. در این حالت Heater=1 و Cooler=0 است.

۳. دما را به ۶۰ درجه می‌رسانیم. مطابق دستور کار ابتدا به وضعیت S1 باز می‌گردیم.
۴. پس از یک کلاک وارد وضعیت S3 می‌شویم که در آن Cooler=1 و CRS=4 است.
۵. چون دما بالاتر از ۳۵ درجه است به وضعیت S4 می‌رویم تا CRS=6 شود.
۶. مجدداً به دلیل بالاتر بودن دما از حد مجاز وارد وضعیت S5 می‌شویم تا CRS=8 شود.
۷. حال دما را به ۱۸ درجه می‌رسانیم. بدین ترتیب در اولین کلاک به S4 می‌رویم تا دور موتور به ۶ کاهش یابد.
۸. سپس به وضعیت S3 باز می‌گردیم.
۹. در نهایت به دلیل کمتر بودن دما از حد مشخص شده، کولر خاموش می‌شود و به وضعیت اولیه S1 باز می‌گردیم.

```

module tb();
    reg clk, reset;
    reg [7:0] T;
    wire Cooler, Heater;
    wire [7:0] CRS;

    incubator incubator_test(
        .clk(clk),
        .reset(reset),
        .T(T),
        .Cooler(Cooler),
        .Heater(Heater),
        .CRS(CRS)
    );

```

ورودی ها و خروجی ها

انکوباتور

```
initial
```

```
begin
```

```
    reset = 0;
```

```
    clk = 0;
```

```
    forever #10 clk = ~clk;
```

```
end
```

مقداردهی اولیه کلاک

ورست سیستم

```
initial
```

```
begin
```

```
    #15 reset = 1; T = 10;    S1->S2
```

```
    #35 T = 60;    S2->S1->S3->S4->S5
```

```
    #100 T = 18;    S5->S4->S3->S2->S1
```

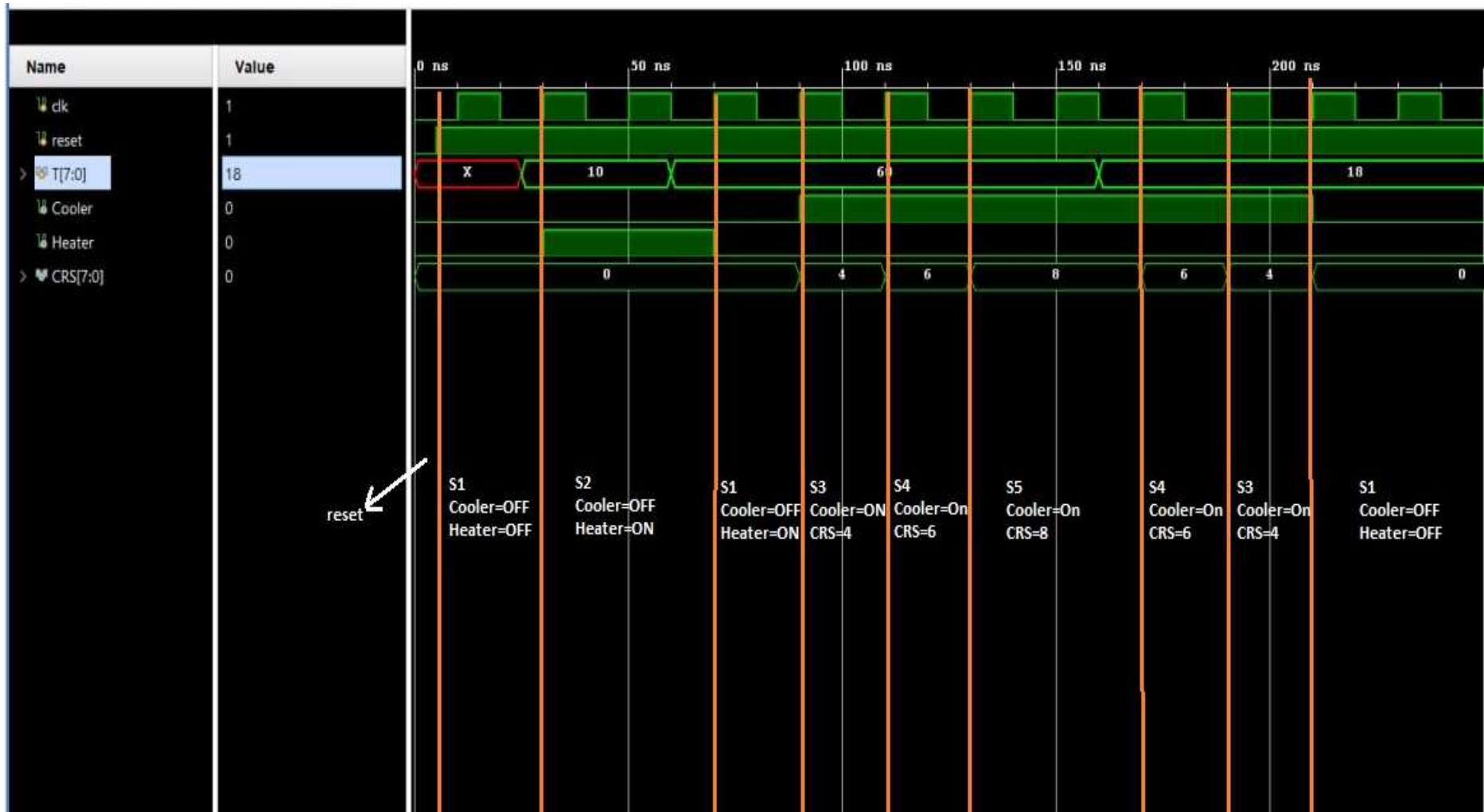
```
    #120
```

```
    $finish;
```

```
end
```

```
initial
```

```
    $monitor("time=%d T=%d Cooler=%d Heater=%d CRS=%d ", $time, T, Cooler, Heater, CRS);
```

سنتز

