

بسمه تعالی



گزارش کار ششم آزمایشگاه طراحی سیستم های دیجیتال

طراحی یک انکوباتور

استاد: دکتر اجلالی

نویسندگان

مریم شیران ۴۰۰۱۰۹۴۴۶

مهدی بهرامیان ۴۰۱۱۷۱۵۹۳

مزدک تیموریان ۴۰۱۱۰۱۴۹۵

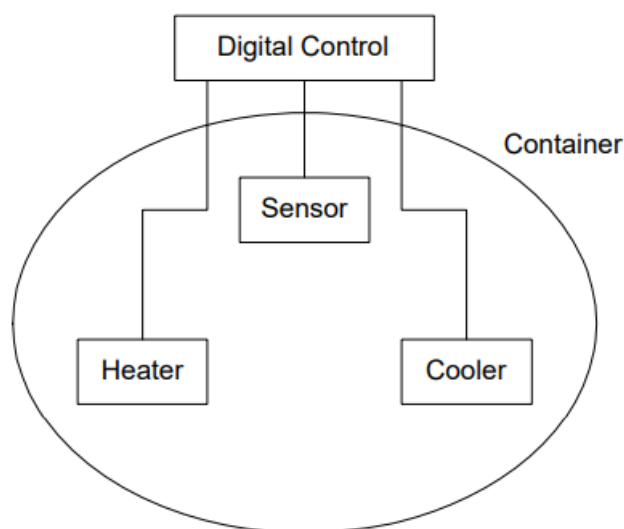
دانشگاه صنعتی شریف

تابستان ۱۴۰۳

فهرست مطالب

۱ ۱
۱-۱ ۱
۱-۲ ۱
۱-۳ ۳
۱-۴ ۸
۱-۵ ۹
۱-۶ ۱۱

۱ آزمایش ششم: طراحی یک انکوباتور



۱-۱ مقدمه

هدف از انجام این آزمایش پیاده سازی واحد کنترل دیجیتال یک سیستم incubator است. این سیستم حاوی یک حسگر دما است که دمای محفظه را که میان -10 تا $+60$ درجه سانتیگراد متغیر است میخواند و در قالب یک عدد ۸ بیتی به سیستم میدهد. دما هر دقیقه یک بار از حسگر دریافت میشود و براساس آن واحد کنترل دیجیتال تصمیم میگیرد که:

- چگونه واحدهای گرم کننده و سرد کنند را فعال و غیر فعال کند.
- چگونه در صورت فعال بودن واحد سرد کننده دور Fan آن را تنظیم کند

۲-۱ طراحی ماژول

۱-۲-۱ ورودی ها و خروجی ها:

ماژول طراحی شده لازم است ورودی و خروجی های زیر را دارا باشد:

ورودی ها:

Temperature: سیگنال ۸ بیتی و نشان دهنده دمای محیط

Clk (سیگنال ساعت): این سیگنال زمانی برای هماهنگی عملکرد عناصر مدار در مدارهای همگام استفاده می شود.

RstN (سیگنال ریست): این سیگنال برای بازنشانی پشته به حالت اولیه‌اش استفاده می‌شود. معمولاً به صورت فعال پایین است، به این معنی که ریست زمانی اتفاق می‌افتد که این سیگنال '۰' باشد.

خروجی‌ها:

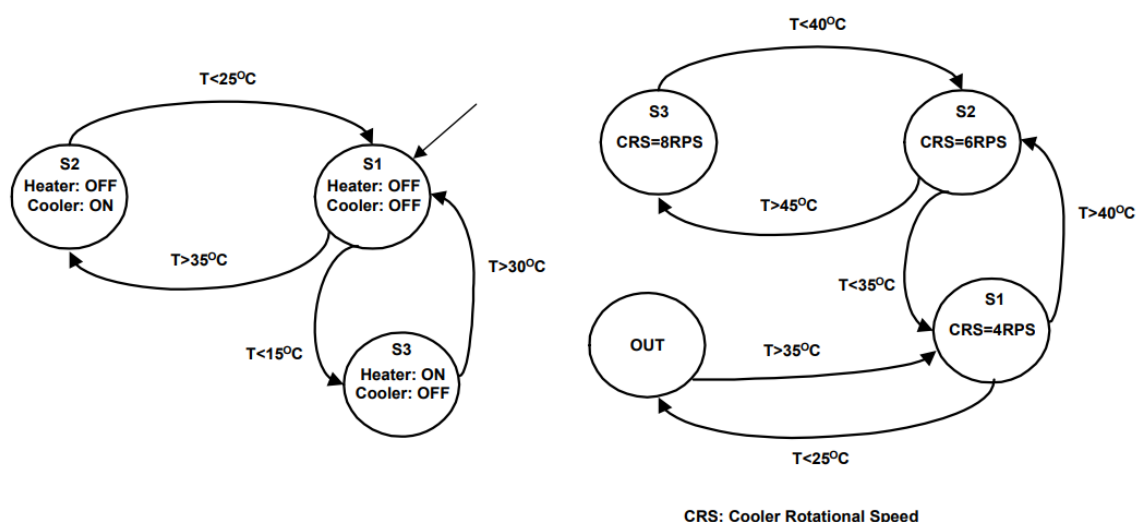
Heater: سیگنال تک بیتی که یک بودن آن به معنی روشن بودن گرم کننده و صفر بودن آن خاموش بودن آن است.

Cooler: سیگنال تک بیتی که یک بودن آن به معنی روشن بودن سرد کننده و صفر بودن آن خاموش بودن آن است.

RPS: سرعت پنکه هنگام روشن بودن سرد کننده را نشان می‌دهد.

۲-۲-۱: بررسی ماشین حالت:

ماشین حالت سیستم فوق به شرح زیر می‌باشد:



طبق دستور کار؛ ماشین حالت سمت راست تنها زمانی فعال می‌شود که در ۲ ماشین حالت سمت چپ باشیم و در غیر این صورت در حالت out آن هستیم.

همچنین طبق ماشین های حالات بالا داریم:

شرایط تغییر درجه فن

اگر درجه فعلی فن ۶ باشد: اگر دمای فعلی کمتر از ۳۵ درجه باشد، درجه فن را به ۴ تغییر می‌دهیم. اگر دمای فعلی بیشتر از ۴۵ درجه باشد، درجه فن را به ۸ تغییر می‌دهیم. در غیر این دو صورت، درجه فن ۶ باقی می‌ماند.

اگر درجه فعلی فن ۸ باشد: اگر دمای فعلی کمتر از ۴۰ درجه باشد، درجه فن را به ۶ تغییر می‌دهیم. در غیر این صورت، درجه فن ۸ باقی می‌ماند.

اگر درجه فعلی فن ۴ باشد: اگر دمای فعلی کمتر از ۲۵ درجه باشد، فن در حالت out قرار گرفته و کولر خاموش می‌شود. اگر دمای فعلی بیشتر از ۴۰ درجه باشد، درجه فن را به ۶ تغییر می‌دهیم. در غیر این صورت، درجه فن ۴ باقی می‌ماند.

شرایط عملکرد کولر و هیتر

اگر کولر و هیتر هر دو خاموش باشند: و دما بیشتر از ۳۵ درجه باشد، کولر روشن و هیتر خاموش شده و فن روی درجه ۴ قرار می‌گیرد. و دما کمتر از ۱۵ درجه باشد، هیتر روشن و کولر خاموش می‌شود. اگر کولر خاموش و هیتر روشن باشد: و دما بیشتر از ۳۰ درجه باشد، هیتر و کولر هر دو خاموش می‌شوند. در غیر این صورت، هیچ تغییری رخ نمی‌دهد.

۳-۱ کدنویسی ماژول اصلی در وریلاگ

با توجه به توضیحات بالا که عملکرد مد نظر را توصیف میکند؛ کد مربوطه در وریلاگ به شرح زیر میشود.

سر تا سر کد، به منظور خوانایی و توضیح کاملت گذاری شده است.

```
// Define the module 'incubator' with inputs: clk (clock), rst (reset), t
(temperature),
// and outputs: heater (heater control), cooler (cooler control), crs (cooling
rate setting)
module incubator (
    input clk,
    input rst,
    input [7:0] t,
    output reg heater,
```

```
output reg cooler,
output reg [3:0] crs
);

// Internal registers for state machine and states
reg sm;          // State machine selector: 0 or 1
reg [1:0] state0; // State register for state machine 0
reg [1:0] state1; // State register for state machine 1

// Always block triggered on the rising edge of clk or the falling edge of rst
always_ff @(posedge clk, negedge rst) begin
    // If reset is low (active low reset)
    if (!rst) begin
        sm <= 0;      // Initialize state machine selector to 0
        state0 <= 0;   // Initialize state0 to 0
        state1 <= 0;   // Initialize state1 to 0
    end else begin
        // State machine 0 behavior
        if (sm == 0) begin
            // State 0: Idle
            if (state0 == 0) begin
                if (t < 15) begin
                    state0 <= 2; // If temperature is less than 15, transition to state 2
                    (heating)
                end else if (t > 35) begin
                    state0 <= 1; // If temperature is greater than 35, transition to
                    state 1 (cooling)
                end
                sm <= 1; // Switch to state machine 1
            end
        end
    end
end
```

```
        state1 <= 0; // Initialize state1 to 0
    end
    // State 1: Cooling
end else if (state0 == 1) begin
    if (t < 25) begin
        state0 <= 0; // If temperature is less than 25, transition to state 0
(idle)
    end
    // State 2: Heating
end else begin
    if (t > 30) begin
        state0 <= 0; // If temperature is greater than 30, transition to
state 0 (idle)
    end
end
// State machine 1 behavior
end else begin
    // State 0: Idle
    if (state1 == 0) begin
        if (t > 35) begin
            state1 <= state1 + 1; // If temperature is greater than 35,
transition to next state
        end
        // State 1: Cooling Level 1
    end else if (state1 == 1) begin
        if (t > 40) begin
```

```
state1 <= state1 + 1; // If temperature is greater than 40,
transition to next state

end else if (t < 25) begin

state1 <= state1 - 1; // If temperature is less than 25, transition
to previous state

sm <= 0;          // Switch back to state machine 0

end

// State 2: Cooling Level 2

end else if (state1 == 2) begin

if (t > 45) begin

state1 <= state1 + 1; // If temperature is greater than 45,
transition to next state

end else if (t < 35) begin

state1 <= state1 - 1; // If temperature is less than 35, transition
to previous state

end

// State 3: Cooling Level 3

end else begin

if (t < 40) begin

state1 <= state1 - 1; // If temperature is less than 40, transition
to previous state

end

end

end

end

end
```



```
// Combinational logic to control heater, cooler, and crs based on current states
```

```
always_comb begin
```

```
    // Control heater and cooler based on state0
```

```
    if (state0 == 0) begin
```

```
        cooler = 0; // Cooler off
```

```
        heater = 0; // Heater off
```

```
    end else if (state0 == 1) begin
```

```
        cooler = 1; // Cooler on
```

```
        heater = 0; // Heater off
```

```
    end else begin
```

```
        heater = 1; // Heater on
```

```
        cooler = 0; // Cooler off
```

```
    end
```

```
    // Control cooling rate setting (crs) based on state1
```

```
    if (state1 == 0) begin
```

```
        crs = 0; // Cooling rate setting 0
```

```
    end else if (state1 == 1) begin
```

```
        crs = 4; // Cooling rate setting 4
```

```
    end else if (state1 == 2) begin
```

```
        crs = 6; // Cooling rate setting 6
```

```
    end else begin
```

```
        crs = 8; // Cooling rate setting 8
```

```
    end
```

```
end
```

```
endmodule
```

۴-۱ کدنویسی ماژول تست در وریلاگ

هم اکنون ماژول تست زیر را نوشته و در ادامه آن را اجرا خواهیم کرد تا از صحت کد خود مطمئن شویم.
سر تا سر کد، به منظور خوانایی و توضیح کامنت گذاری شده است.

```
// Testbench module
module test ();

    // Declare registers and wires
    reg [7:0] t;    // 8-bit register to hold temperature
    reg clk, rst;   // Clock and reset signals
    wire heater, cooler; // Heater and cooler control signals
    wire [3:0] crs; // 4-bit wire for control signals

    // Instantiate the incubator module
    incubator_inc (
        .t(t),      // Connect t to the incubator's temperature input
        .clk(clk),   // Connect clk to the incubator's clock input
        .rst(rst),   // Connect rst to the incubator's reset input
        .crs(crs),   // Connect crs to the incubator's control signal output
        .heater(heater), // Connect heater to the incubator's heater control output
        .cooler(cooler) // Connect cooler to the incubator's cooler control output
    );
```

```
// Clock signal generation: toggle clk every 1 time unit
always #1 clk = !clk;

// Initial block to set up the simulation
initial begin

    $dumpfile("test.vcd"); // Specify the name of the dump file for
    waveform

    $dumpvars(0, test); // Specify the scope of variables to dump

    $monitor("state : ", _inc.sm, " ", _inc.state0, " ", _inc.state1, " status : ",
    heater,

        " ", cooler); // Monitor specified signals during simulation

    clk = 0; // Initialize clk to 0
    rst = 0; // Initialize rst to 0
    #2; // Wait for 2 time units
    rst = 1; // Deassert reset signal
    t = 80; // Set temperature to 80
    #20; // Wait for 20 time units
    t = 0; // Set temperature to 0
    #20; // Wait for 20 time units
    $finish(0); // End simulation
end
endmodule
```

۵-۱ شبیه سازی

مراحل زیر در شبیه سازی رخ میدهند.

- تست پنج فایل test.vcd را برای ذخیره داده‌های موج مشخص می‌کند و محدوده

۲. نظارت بر سیگنال‌ها:

(sm, state0, state1, heater, cooler) را هر زمان که تغییر می‌کنند به کنسول

۳. مقداردهی اولیه:

○ سیگنال ریست (rst) به مقدار دهی اولیه می شود.

۴. توالی شبیه‌سازی:

حالت ریست خارج شود.

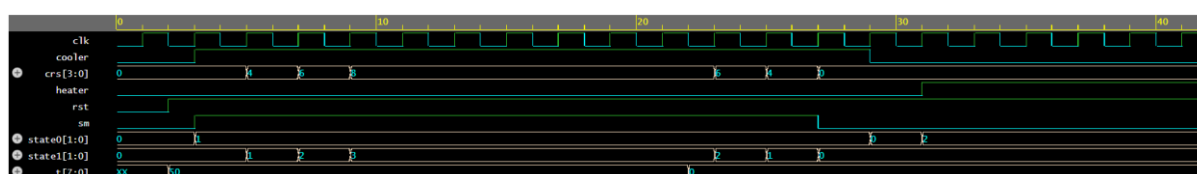
○ ورودی دما (t) به ۸۰ تنظیم می‌شود.

○ پس از ۲۰ واحد زمانی، ورودی دما (t) به • تنظیم می‌شود.

○ پس از ۲۰ واحد زمانی دیگر، شبیه‌سازی با فرمان `$finish(0)` پایان می‌یابد.

در زیر مشاهده میکنیم که خروجی های مربوطه با نتایج مورد انتظار همخوانی دارند.

```
temp : x state : 0 0 0 status : 0 0 0
temp : 80 state : 0 0 0 status : 0 0 0
temp : 80 state : 1 1 0 status : 0 1 0
temp : 80 state : 1 1 1 status : 0 1 4
temp : 80 state : 1 1 2 status : 0 1 6
temp : 80 state : 1 1 3 status : 0 1 8
temp : 0 state : 1 1 3 status : 0 1 8
temp : 0 state : 1 1 2 status : 0 1 6
temp : 0 state : 1 1 1 status : 0 1 4
temp : 0 state : 0 1 0 status : 0 1 0
temp : 0 state : 0 0 0 status : 0 0 0
temp : 0 state : 0 2 0 status : 1 0 0
temp : 80 state : 0 2 0 status : 1 0 0
temp : 80 state : 0 0 0 status : 0 0 0
temp : 80 state : 1 1 0 status : 0 1 0
temp : 80 state : 1 1 1 status : 0 1 4
temp : 80 state : 1 1 2 status : 0 1 6
temp : 80 state : 1 1 3 status : 0 1 8
```



۶-۱ منابع و مراجع

- www.altera.com/literature/manual/intro_to_quartus.pdf
- ModelSim User's Manual, www.actel.com/documents/modelsim_ug.pdf
- S. Palnitkar, Verilog® HDL: A Guide to Digital Design and Synthesis, Second Edition, Prentice Hall, ۲۰۰۳
- وبسایت گیکز فور گیکز
- وبسایت ویکی پدیا