آز سیستم دیجیتال

نیم‌سال تابستان ۰۱-۰۲

مدرس: دکتر انصاری

دانشکده مهندسی کامپیوتر

عماد زین‌اوقلی	عطا رحیم‌زاده	سپنتا رحمانی‌زاده
۹۸۱۰۳۲۶۷	۹۸۱۷۰۸۰۵	۹۸۱۱۰۰۴۹

۱ آزمایش چهارم

در این آزمایش بایستی یک مدار پشته ۴ بیتی به اندازه ۸ طراحی کنیم.

۱.۱ مدار اصلی

در ماژول stack پشته‌ای ۴ بیتی به اندازه ۸ طراحی کردیم. در این پشته دو متغیر داریم. متغیر stack یک آرایه ۸ـتایی از ثبات‌های ۴ بیتی است که به عنوان حافظه پشته از آن استفاده می‌شود. متغیر hotbit_address یک ثبات ۹ بیتی است که اولین خانه خالی پشته اشاره می‌کند. بدین ترتیب، زمانی که مقدار آن $9'b0\_0000\_0001$ است، پشته خالی است و زمانی که مقدار آن برابر با $9'b1\_0000\_0000$ است پشته پر است. سیگنال resN به صورت synchronous عمل می‌کند به مقادیر درون stack صفر می‌کند و hotbit_address را برابر با $9'b0\_0000\_0001$ قرار می‌دهد. اگر سیگنال push فعال باشد و پشته پر نباشد (سیگنال full غیر فعال باشد) آنگاه مقدار ورودی data_in درون پشته می‌ریزیم و مقدار hotbit_address به چپ شیفت می‌دهیم. همچنین اگر سیگنال pop فعال باشد و پشته خالی نباشد (سیگنال empty غیر فعال باشد) آنگاه مقدار خروجی data_out برابر با مقدار بالای پشته می‌گذاریم و مقدار hotbit_address به راست شیفت می‌دهیم. دقت کنید که push اولویت دارد. یعنی اگر هر دو سیگنال push و pop فعال شوند آنگاه تنها دستور push انجام می‌شود.

```
module stack (rstN, clk, data_in, push, pop, data_out, full, empty);

        parameter DEPTH = 8;
        parameter WIDTH = 4;

        input wire rstN, push, pop, clk;
        input wire [WIDTH - 1 : 0] data_in;
        output full, empty;
        output reg [WIDTH - 1 : 0] data_out;

        reg [WIDTH - 1 : 0] stack [0 : DEPTH - 1];
        reg [DEPTH : 0] hotbit_address;


        assign  empty = !hotbit_address[0];
        assign  full  = hotbit_address[DEPTH];
        integer i;

        always @(posedge clk) begin
                if (!rstN) begin
                        for (i = 0 ; i < DEPTH ; i = i +1) begin
                                stack[i] = {WIDTH{1'b0}};
```

```
                        end
                        hotbit_address = {{DEPTH-1{1'b0}}, 1'b1};
                end
                else begin
                        if (push &&  !full) begin
                                for ( i = 0 ; i < DEPTH; i = i+ 1) begin
                                        if (hotbit_address[i]) stack[i]
                                           = data_in;
                                end
                                hotbit_address = hotbit_address << 1;
                        end
                        else if (pop && empty) begin
                                for ( i = 1 ; i <= DEPTH ; i = i +1)
                                   begin
                                        if(hotbit_address[i]) data_out =
                                            stack[i-1];
                                end
                                hotbit_address = hotbit_address >> 1;
                        end
                end
        end
endmodule
```

۲.۱ آزمون

برای آزمایش طراحی بالا، آزمون زیر را طراحی کردیم. در ابتدا پشته reset می‌شود. سپس اعداد ۱ تا ۸ درون پشته می‌ریزیم. پشته پر می‌شود اما سعی می‌کنیم که عدد ۹ را درون پشته بریزیم. سپس پشته را خالی می‌کنیم و سعی می‌کنیم پس از خالی شدن باز هم pop کنیم. در نهایت، رفتار پشته را زمانی که دو سیگنال push و pop روشن باشد بررسی می‌کنیم.

```
`include "stack.v"
`timescale 1ns/1ns
module stack_tb;

        parameter cycle = 10;
        reg     rstN, push, pop, clk;
        reg     [3 : 0] data_in;
        wire full, empty;
        wire [3 : 0] data_out;
        integer i;

        stack #(.DEPTH(8),.WIDTH(4)) stack0 (
                .rstN(rstN),
                .clk(clk),
                .data_in(data_in),
                .push(push),
                .pop(pop),
                .data_out(data_out),
                .full(full),
                .empty(empty));

```

```
22          initial begin
23                  clk     = 0;
24                  forever #(cycle/2) clk = ~clk;
25          end
26
27          initial begin
28                  $monitor($time,"\tpush=%b, pop=%b, data_in=%d, empty=%b,
                       full=%b, data_out=%d", push, pop, data_in, empty,
                       full, data_out);
29                  $dumpfile("stack.vcd");
30                  $dumpvars(0,stack0);
31                  rstN = 0;
32                  push = 0;
33                  pop = 0;
34                  data_in = 0;
35                  #cycle rstN = 1;
36
37                  for (i = 1; i <= 8; i = i +1) begin
38                  #cycle push = 1;
39                  pop = 0;
40                  data_in = i;
41                  end
42
43                  #cycle data_in = 9;
44
45                  #cycle push = 0;
46                  pop = 1;
47                  for (i = 0; i <= 8; i = i + 1) begin
48                          #cycle ;
49                  end
50
51                  #cycle push = 1;
52                  data_in = 1;
53                  #cycle data_in = 2;
54                  #cycle push = 0;
55                  $finish;
56          end
57  endmodule
```

خروجی به صورت زیر است.

```
   VCD info: dumpfile stack.vcd opened for output.
0 push=0, pop=0, data_in= 0, empty=x, full=x, data_out= x
5 push=0, pop=0, data_in= 0, empty=0, full=0, data_out= x
20 push=1, pop=0, data_in= 1, empty=0, full=0, data_out= x
25 push=1, pop=0, data_in= 1, empty=1, full=0, data_out= x
30 push=1, pop=0, data_in= 2, empty=1, full=0, data_out= x
50 push=1, pop=0, data_in= 4, empty=1, full=0, data_out= x
60 push=1, pop=0, data_in= 5, empty=1, full=0, data_out= x
70 push=1, pop=0, data_in= 6, empty=1, full=0, data_out= x
80 push=1, pop=0, data_in= 7, empty=1, full=0, data_out= x
90 push=1, pop=0, data_in= 8, empty=1, full=0, data_out= x
```

```
95 push=1, pop=0, data_in= 8, empty=1, full=1, data_out= x
100 push=1, pop=0, data_in= 9, empty=1, full=1, data_out= x
110 push=0, pop=1, data_in= 9, empty=1, full=1, data_out= x
115 push=0, pop=1, data_in= 9, empty=1, full=0, data_out= 8
125 push=0, pop=1, data_in= 9, empty=1, full=0, data_out= 7
135 push=0, pop=1, data_in= 9, empty=1, full=0, data_out= 6
145 push=0, pop=1, data_in= 9, empty=1, full=0, data_out= 5
155 push=0, pop=1, data_in= 9, empty=1, full=0, data_out= 4
165 push=0, pop=1, data_in= 9, empty=1, full=0, data_out= 3
175 push=0, pop=1, data_in= 9, empty=1, full=0, data_out= 2
185 push=0, pop=1, data_in= 9, empty=0, full=0, data_out= 1
210 push=1, pop=1, data_in= 1, empty=0, full=0, data_out= 1
215 push=1, pop=1, data_in= 1, empty=1, full=0, data_out= 1
220 push=1, pop=1, data_in= 2, empty=1, full=0, data_out= 1
stack_tb.v:55: $finish called at 230 (1ns)
230 push=0, pop=1, data_in= 2, empty=1, full=0, data_out= 1
```