

سوال اول) در این سوال، باید یک حافظه با دو پورت inout طراحی کنیم. بدین منظور، از ماژول طراحی شده در اسلاید هشتم ایده می‌گیریم و ماژول زیر را طراحی می‌کنیم.

```

mem.v
1  module mem (
2      input [8:0] addr_A,
3      input [8:0] addr_B,
4      input rd_en_A,
5      input rd_en_B,
6      input we_A,
7      input we_B,
8      input rst,
9      input clk,
10     inout [15:0] data_A,
11     inout [15:0] data_B
12 );
13
14     reg [15:0] mem [0:511];
15     reg [15:0] read_data_A;
16     reg [15:0] read_data_B;
17     reg [15:0] write_data_A;
18     reg [15:0] write_data_B;
19
20     always @(posedge clk) begin
21         if (rst) begin
22             read_data_A <= 16'b0;
23             read_data_B <= 16'b0;
24         end
25         else begin
26             if (rd_en_A) begin
27                 read_data_A <= mem[addr_A];
28             end
29             if (rd_en_B) begin
30                 read_data_B <= mem[addr_B];
31             end
32
33             if (we_A && ~rd_en_A) begin
34                 mem[addr_A] <= data_A;
35             end
36             if (we_B && ~rd_en_B) begin
37                 mem[addr_B] <= data_B;
38             end
39         end
40     end
41
42     assign data_A = rd_en_A ? read_data_A : 16'bz;
43     assign data_B = rd_en_B ? read_data_B : 16'bz;
44
45 endmodule
46

```

در این ماژول، دو پورت inout با نام‌های data\_A و data\_B داریم. برای هر کدام از این ورودی‌ها، سیگنال کنترلی read enable و write enable داریم و همچنین برای هر کدام، address\_A و address\_B را داریم. با توجه به خواست سوال، عملیات read نسبت به write اولویت دارند. در نتیجه، در بلوک always که در لبه بالا رونده کلاک فعال می‌شود، ابتدا عملیات read را انجام می‌دهیم و سپس در صورت فعال نبودن read enable برای یک ورودی خاص، عملیات write را از آن انجام می‌دهیم.

برای تست این ماژول، از تست‌بنچ mem\_tb.v استفاده می‌کنیم که در پیوست آمده‌است. در این ماژول بستر آزمون، ابتدا داده‌های رندوم را در پورت‌های A و B قرار می‌دهیم تا در آدرس‌های ذیل نوشته شوند. این کار را ۱۰ بار انجام می‌دهیم. سپس به ترتیب عکس، از همان آدرس‌ها مقادیر گفته شده را می‌خوانیم و با توجه به خروجی فایل بستر آزمون، واضح است که این حافظه به درستی کار می‌کند.

برای خواندن و نوشتن از این تکه‌کدها استفاده می‌کنیم:

```
wire [15:0] io_data_A;
wire [15:0] io_data_B;

assign io_data_A = we_A ? data_A : 16'bz;
assign io_data_B = we_B ? data_B : 16'bz;

mem uut (
    .addr_A(addr_A),
    .addr_B(addr_B),
    .rd_en_A(rd_en_A),
    .rd_en_B(rd_en_B),
    .we_A(we_A),
    .we_B(we_B),
    .rst(rst),
    .clk(clk),
    .data_A(io_data_A),
    .data_B(io_data_B)
);
```

```
repeat (10) begin
    data_A = $random;
    data_B = $random;
    #CLK_PERIOD;

    $display("Write: \tAddress = %h, Data = %h", addr_A, io_data_A);
    $display("Write: \tAddress = %h, Data = %h", addr_B, io_data_B);

    addr_A = addr_A + 1;
    addr_B = addr_B + 1;
end
```

```
repeat (10) begin
    #CLK_PERIOD;
    addr_A = addr_A - 1;
    addr_B = addr_B - 1;
    #CLK_PERIOD;
    $display("Read: \tAddress = %h, Data = %h", addr_A, io_data_A);
    $display("Read: \tAddress = %h, Data = %h", addr_B, io_data_B);
end
```

**خروجی فایل بستر آزمون را در زیر مشاهده می کنید:**

```
Write: Address = 000, Data = 3524
Write: Address = 064, Data = 5e81
Write: Address = 001, Data = d609
Write: Address = 065, Data = 5663
Write: Address = 002, Data = 7b0d
Write: Address = 066, Data = 998d
Write: Address = 003, Data = 8465
Write: Address = 067, Data = 5212
Write: Address = 004, Data = e301
Write: Address = 068, Data = cd0d
Write: Address = 005, Data = f176
Write: Address = 069, Data = cd3d
Write: Address = 006, Data = 57ed
Write: Address = 06a, Data = f78c
Write: Address = 007, Data = e9f9
Write: Address = 06b, Data = 24c6
Write: Address = 008, Data = 84c5
Write: Address = 06c, Data = d2aa
Write: Address = 009, Data = f7e5
Write: Address = 06d, Data = 7277
Read: Address = 009, Data = f7e5
Read: Address = 06d, Data = 7277
Read: Address = 008, Data = 84c5
Read: Address = 06c, Data = d2aa
Read: Address = 007, Data = e9f9
Read: Address = 06b, Data = 24c6
Read: Address = 006, Data = 57ed
Read: Address = 06a, Data = f78c
Read: Address = 005, Data = f176
Read: Address = 069, Data = cd3d
Read: Address = 004, Data = e301
Read: Address = 068, Data = cd0d
Read: Address = 003, Data = 8465
Read: Address = 067, Data = 5212
Read: Address = 002, Data = 7b0d
Read: Address = 066, Data = 998d
Read: Address = 001, Data = d609
Read: Address = 065, Data = 5663
Read: Address = 000, Data = 3524
Read: Address = 064, Data = 5e81
```

سوال دوم) در این سوال، با ورودی  $i = 0$  به حالت بعدی می‌رویم. همچنین اگر ورودی  $i = 1$  باشد، به حالت اولیه یعنی A می‌رویم. باید در نظر داشته باشیم که خروجی output نیز دارای ۲ بیت است. در نهایت واضح است که گزینه درست، گزینه ب می‌باشد.

سوال سوم) در مرحله اول، مقادیر q فلیپ‌فلاپ‌های صفر می‌باشد. در ادامه، ورودی‌های فلیپ‌فلاپ اول برابر  $k=1, z=1$  و ورودی‌های فلیپ‌فلاپ دوم برابر  $k=1, z=0$  می‌شود. در نتیجه، خروجی فلیپ‌فلاپ اول در مرحله دوم برابر  $q[0]=1$  و خروجی فلیپ‌فلاپ دوم در مرحله دوم برابر  $q[1]=0$  می‌شود. در ادامه، ورودی‌های فلیپ‌فلاپ اول برابر  $k=1, z=1$  و ورودی‌های فلیپ‌فلاپ دوم برابر  $k=1, z=0$  می‌شود. در نتیجه، خروجی فلیپ‌فلاپ اول در مرحله دوم برابر  $q[0]=0$  و خروجی فلیپ‌فلاپ دوم در مرحله دوم برابر  $q[1]=1$  می‌شود. در ادامه، ورودی‌های فلیپ‌فلاپ اول برابر  $k=1, z=0$  و ورودی‌های فلیپ‌فلاپ دوم برابر  $k=1, z=0$  می‌شود. در نتیجه  $q[0]=q[1]=0$  می‌شود و ماشین به استتیت اولیه خود باز می‌گردد.

در نهایت، ماشین استتیت‌های زیر را طی می‌کند:

00 -> 01 -> 10 -> 00 -> ...

به بررسی خروجی گزینه‌های مختلف می‌پردازیم:

الف) این یک شمارنده جانسون است که وضوح غلط می‌باشد.

ب) این ماشین دنباله 00, 11, 00, 11, ... را طی می‌کند که نادرست می‌باشد.

ت) این ماشین نیز دنباله عجیبی را طی می‌کند که مطابق توضیح فوق نمی‌باشد.

در نتیجه گزینه پ درست می‌باشد.

سوال چهارم) در این مدار، پس از طی ۱۰ واحد تاخیر، مقدار out مشخص می‌شود. در نتیجه گزینه‌های ب و پ نادرست هستند. همچنین، اگر enable فعال نباشد، خروجی گیت nand برابر 1 خواهد بود که پس از گذر از ۴ گیت not، مقادیر خروجی out برابر 1 می‌شود. در نتیجه گزینه ت درست می‌باشد.

سوال پنجم) در گزینه الف بلاک always حساسیت ندارد پس نادرست می‌باشد. در گزینه ب چون برای wire از assign استفاده نکردیم نادرست است. همچنین در گزینه پ، ورودی مازول از نوع reg است که نادرست است. در نهایت واضح است که گزینه درست، ت می‌باشد.