

سوال اول

ماژول زیر را برای این سوال طراحی می‌کنیم:

```
module bidirectional_bus #(
    parameter N = 8
) (
    input [N-1:0] data_in_1,
    input [N-1:0] data_in_2,
    input oe_1,
    input oe_2,
    output [N-1:0] data_out_1,
    output [N-1:0] data_out_2,
    inout [N-1:0] bi_data
);

assign bi_data = (oe_1 ^ oe_2 == 0) ? {N{1'bZ}} : (oe_1 ? data_in_1 : data_in_2);
// if both signals are active or inactive, send high impedance on the wire
assign data_out_1 = oe_2 ? bi_data : {N{1'bZ}};
assign data_out_2 = oe_1 ? bi_data : {N{1'bZ}};

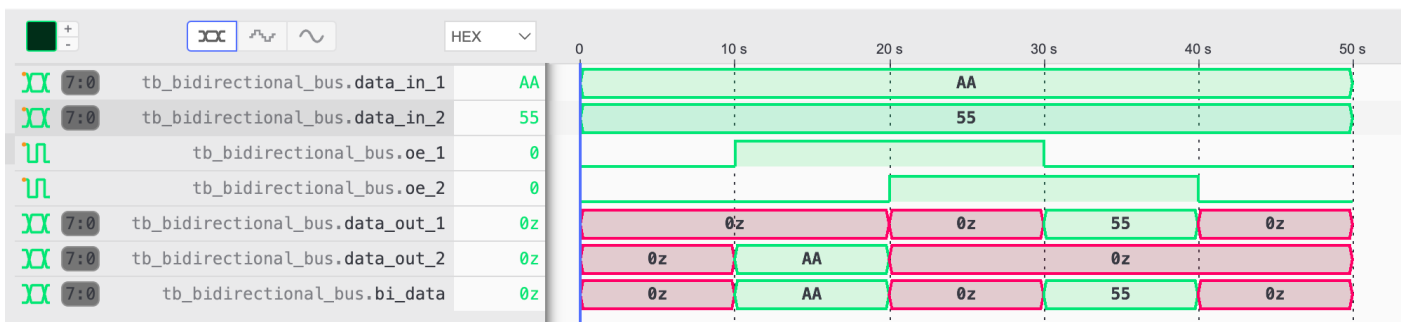
endmodule
```

توضیح این ماژول واضح است. اگر هر دو سیگنال oe فعال یا غیر فعال باشند، مقدار High Impedance وارد bi_data می‌شود. سپس با استفاده از سیگنال‌های oe_1 و oe_2 مشخص می‌شود که هر کدام از خروجی‌ها چه مقدار را بگیرند.

فایل تست‌بنچ طراحی شده است و در پیوست قرار گرفته است. خروجی فایل تست‌بنچ را در زیر مشاهده می‌کنید:

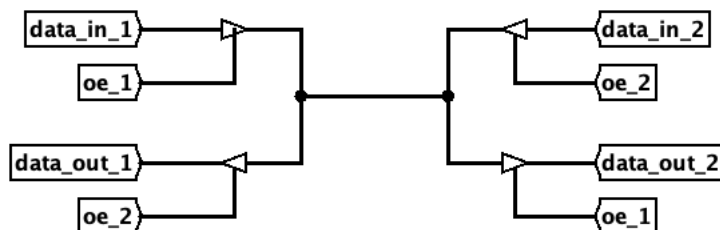
Time	OE1	OE2	BiData	Out1	Out2
10	0	0	zz	zz	zz
20	1	0	aa	zz	aa
30	1	1	zz	zz	zz
40	0	1	55	55	zz
50	0	0	zz	zz	zz

خروجی waveform این تست‌بنچ را در زیر می‌توانید مشاهده کنید.



سوال دوم

طراحی گیت لول این مدار با توجه به تاخیرهای داده شده به این صورت می باشد:



برای این طراحی ماژول زیر را طراحی می کنیم:

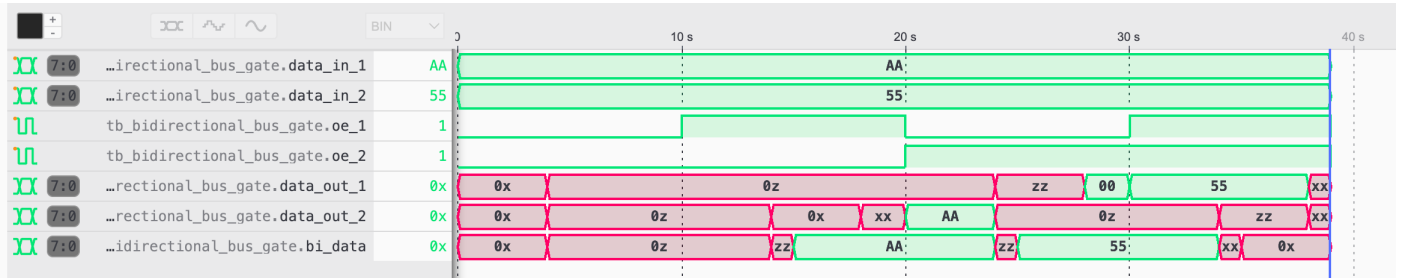
```
module bidirectional_bus_gate #(
    parameter N = 8
) (
    input [N-1:0] data_in_1,
    input [N-1:0] data_in_2,
    input oe_1,
    input oe_2,
    output [N-1:0] data_out_1,
    output [N-1:0] data_out_2,
    inout [N-1:0] bi_data
);

integer i;
for (i = 0; i < N; i = i + 1) {
    bufif1 #(5:6:7, 4:5:6, 4:5:6) tri1 (bi_data[i], data_in_1[i], oe_1);
    bufif1 #(5:6:7, 4:5:6, 4:5:6) tri2 (bi_data[i], data_in_2[i], oe_2);
    bufif1 #(5:6:7, 4:5:6, 4:5:6) tri3 (data_out_1[i], bi_data[i], oe_2);
    bufif1 #(5:6:7, 4:5:6, 4:5:6) tri4 (data_out_2[i], bi_data[i], oe_1);
}

endmodule
```

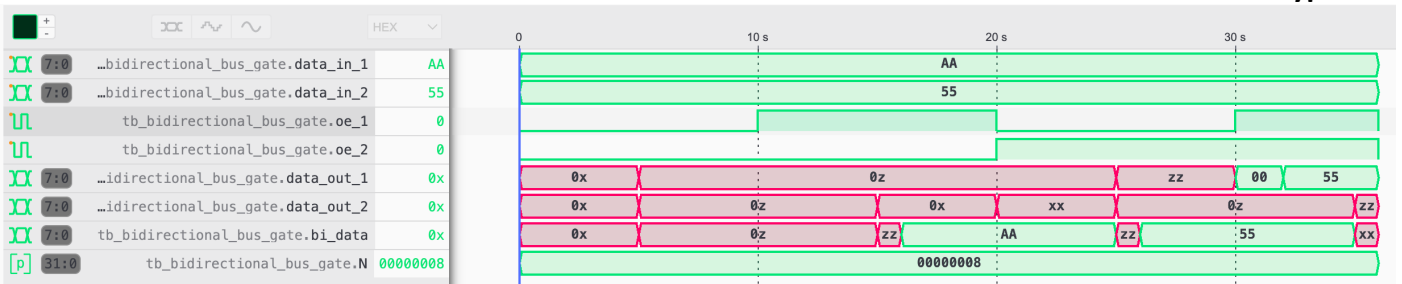
ابتدا با استفاده از فلگ -T، به کامپایلر iverilog می‌فهمانیم که ماژول را در چه حالتی min/typ/max ران بگیرد.

حالت min:



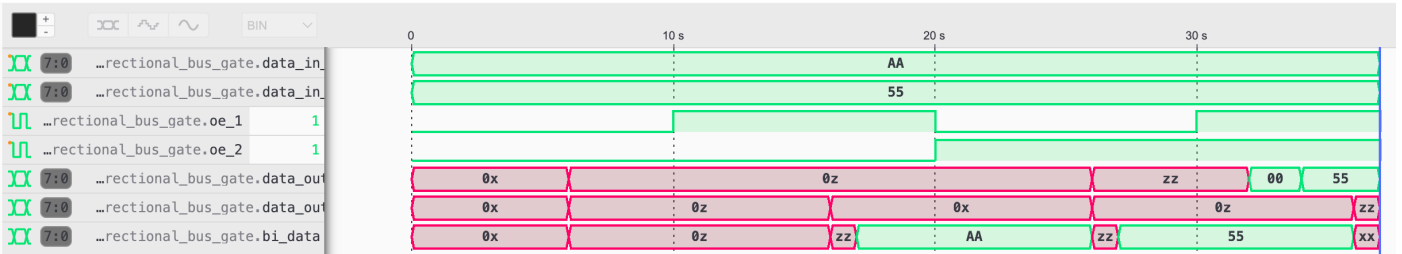
همانطور که مشاهده می‌شود، تاخیر مسیر بحرانی برای حالت min برابر 10 ثانیه است. با هر بار تغییر ورودی، ۱۰ ثانیه طول می‌کشد تا خروجی پایدار بشود.

حالت typ:



همانطور که مشاهده می‌شود، تاخیر مسیر بحرانی برای حالت typ برابر 12 ثانیه است. با هر بار تغییر ورودی، ۱۲ ثانیه طول می‌کشد تا خروجی پایدار بشود.

حالت max:



همانطور که مشاهده می‌شود، تاخیر مسیر بحرانی برای حالت max برابر 12 ثانیه است. با هر بار تغییر ورودی، ۱۴ ثانیه طول می‌کشد تا خروجی پایدار بشود.

سوال سوم

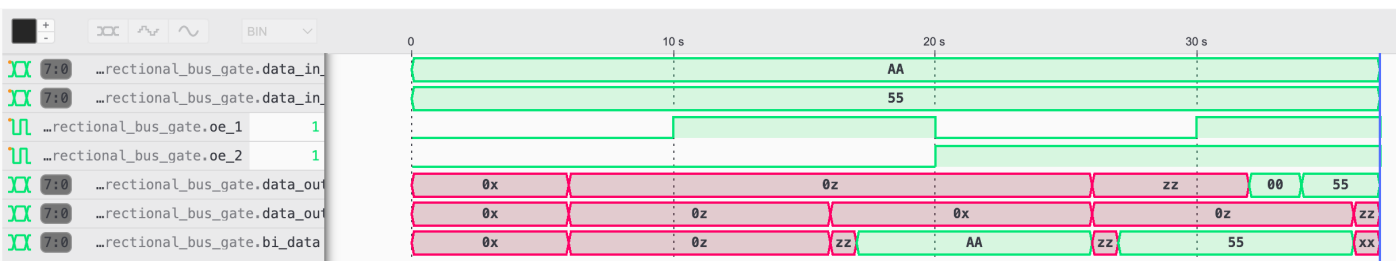
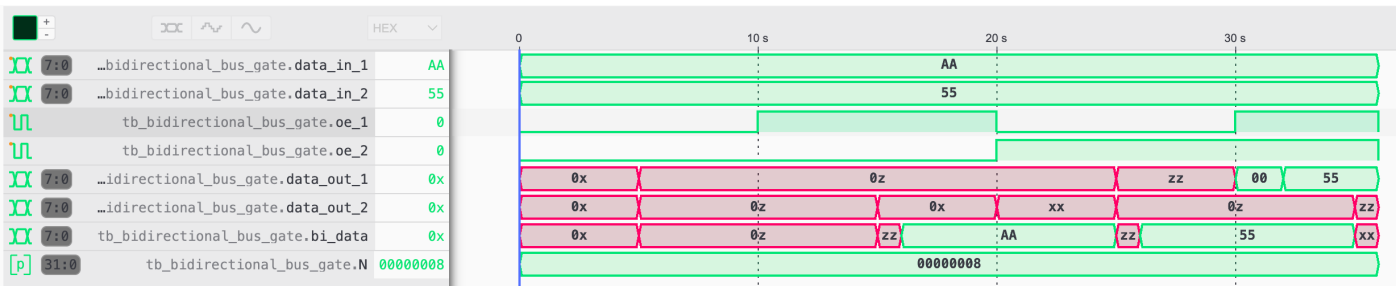
با توجه به اینکه برای طراحی درست، در ماژول سوال اول از گیت xor استفاده کردیم و تاخیر آن در صورت سوال نیامده است، از همان تاخیرهای داده شده استفاده می‌کنیم. ماژول به حالت زیر تغییر می‌کند:

```
module bidirectional_bus #(
    parameter N = 8
) (
    input [N-1:0] data_in_1,
    input [N-1:0] data_in_2,
    input oe_1,
    input oe_2,
    output [N-1:0] data_out_1,
    output [N-1:0] data_out_2,
    inout [N-1:0] bi_data
);

assign #(4:5:6) bi_data = (oe_1 ^ oe_2 == 0) ? {N{1'bZ}} : (oe_1 ? data_in_1 : data_in_2);
// if both signals are active or inactive, send high impedance on the wire
assign #(4:5:6) data_out_1 = oe_2 ? bi_data : {N{1'bZ}};
assign #(4:5:6) data_out_2 = oe_1 ? bi_data : {N{1'bZ}};

endmodule
```

سپس از ماژول تست‌های زیر را به ترتیب برای حالات min، typ و max می‌گیریم.



سوال چهارم)

در مدل سازی سطح گیت (Gate-Level Modeling) ، به دلیل استفاده مستقیم از گیت ها و اتصالات، دقت بیشتری در طراحی حاصل می شود. به ویژه زمانی که نیاز به تحلیل دقیق تر زمان بندی و تاخیر وجود دارد، این روش گزینه بهتری است. در مدل سازی سطح گیت، می توانیم تاخیرهای مختلفی مانند Rise ، Fall ، و Turn-off را تعریف کنیم.

در حالی که در مدل سازی جریان داده (Data Flow Modeling) معمولاً فقط یک نوع تاخیر برای دستور assign در نظر گرفته می شود.

اما مزیت مدل سازی جریان داده در سادگی آن است. طراحی با استفاده از این روش نسبت به مدل سازی سطح گیت آسان تر است و هنگامی که با طراحی های بزرگ سروکار داریم، پیچیدگی مدل سازی سطح گیت می تواند چالش برانگیز باشد. این سادگی در مدل سازی جریان داده به معنای آن است که پیدا کردن مشکلات نیز در این روش ساده تر از مدل سازی سطح گیت خواهد بود.

بنابراین، اگرچه مدل سازی سطح گیت دقت و کنترل بیشتری در تحلیل زمان بندی فراهم می کند، اما مدل سازی جریان داده می تواند برای طراحی های پیچیده تر یا زمانی که نیاز به ساده سازی طراحی داریم، گزینه بهتری باشد.