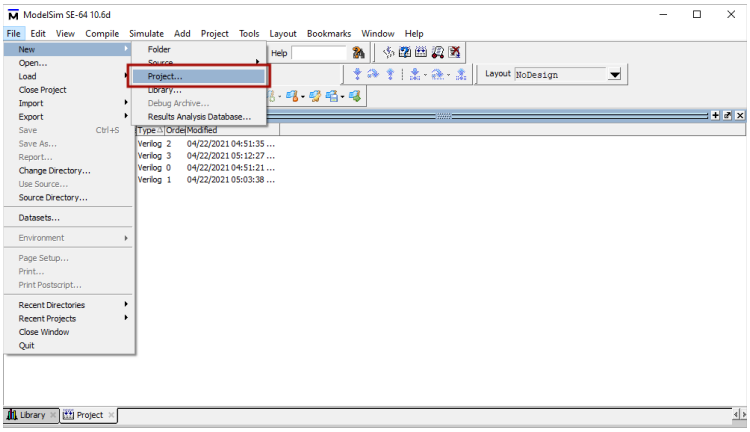
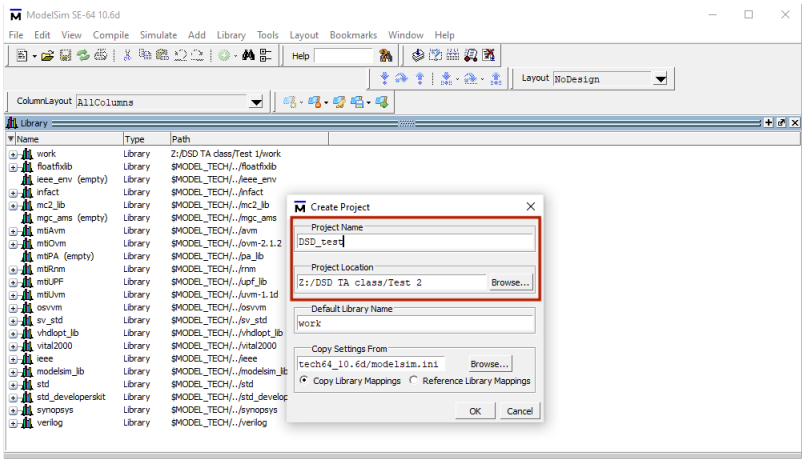
۱ آشنایی با نرم‌افزار ModelSim

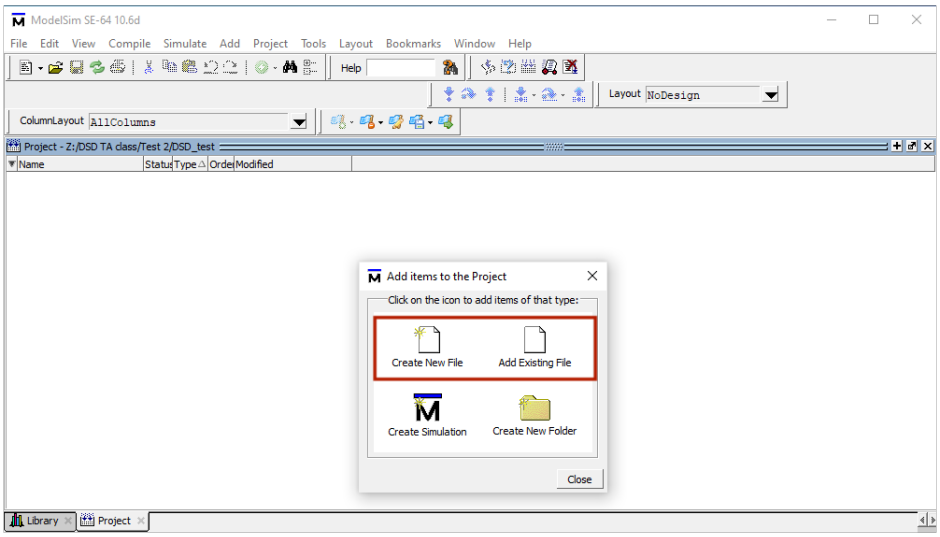
پس از نصب برنامه، طبق مراحل زیر می‌توانید کد verilog خود را در برنامه ModelSim نوشته و اجرا کنید.



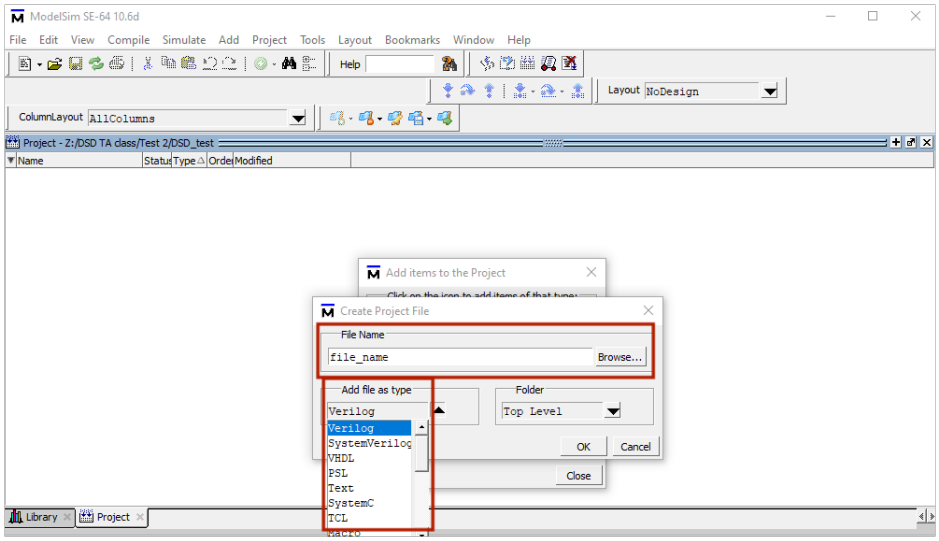
شکل ۱: بعد از باز کردن برنامه، از قسمت File، New و سپس Project را انتخاب کنید.



شکل ۲: نام پروژه و محل ذخیره شدن آن را مشخص کنید.

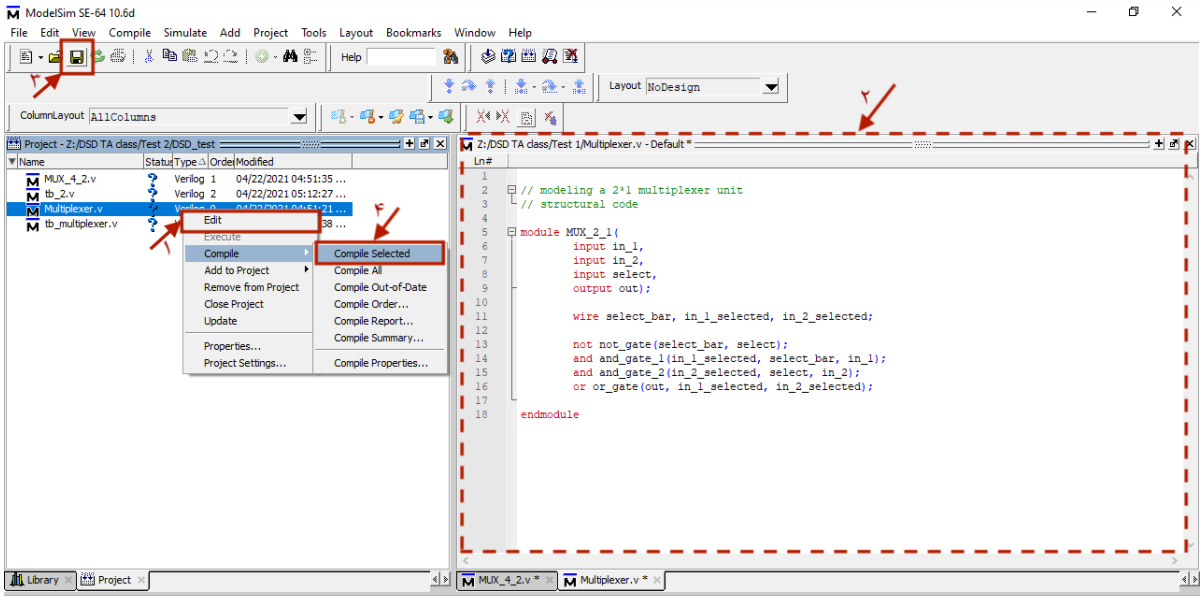


شکل ۳: مشخص کنید که می‌خواهید چه نوع File ای به پروژه خود اضافه کنید. می‌توانید از File های قبلی خود به پروژه اضافه کنید یا File جدید بسازید.

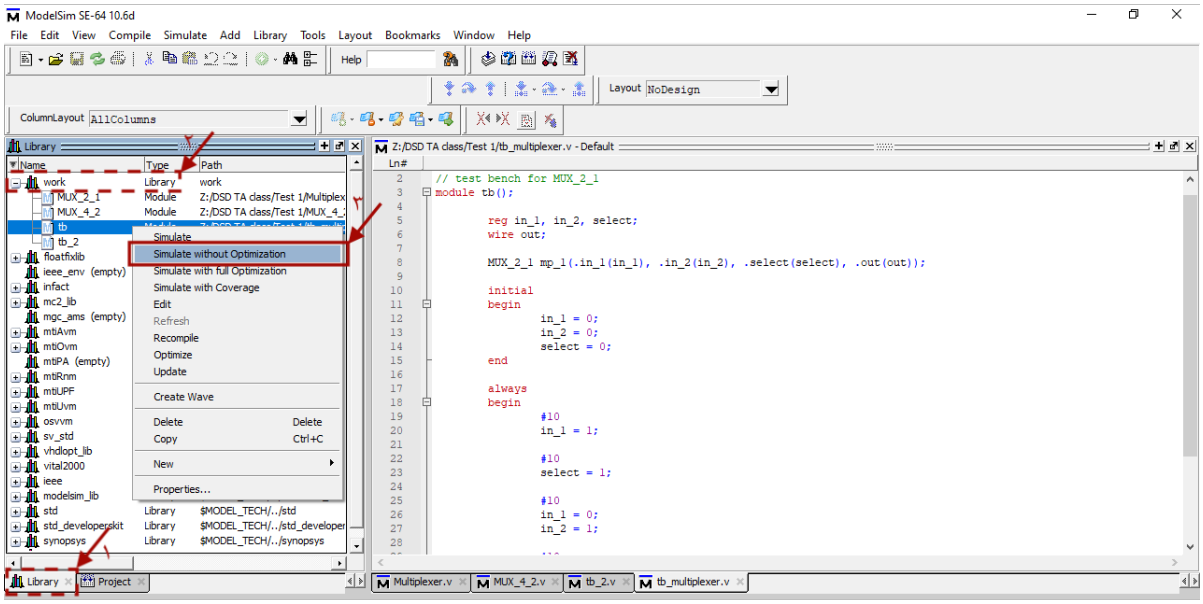


شکل ۴: اگر در قسمت قبل گزینه Create New File را انتخاب کرده باشید، آنگاه در این مرحله باید نام فایل خود را تعیین کنید و type فایل خود را verilog انتخاب کنید.

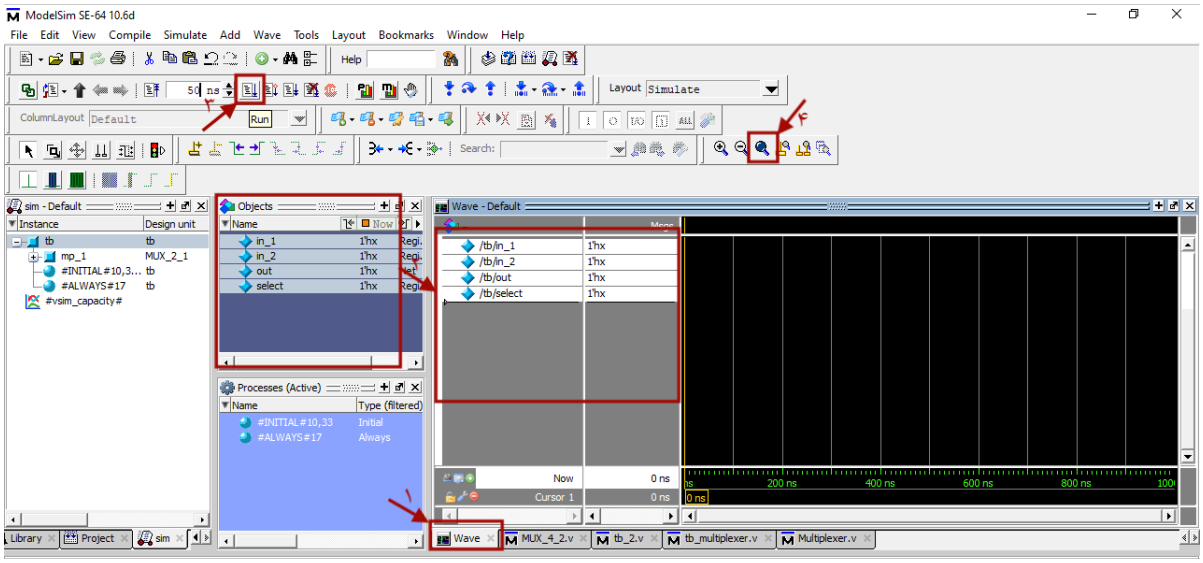
تا کنون نحوه ساخت پروژه و اضافه کردن فایل وریلاگ به آن را نشان داده شده. در شکل‌های بعدی، نحوه نوشتن و کامپایل کردن کد و در نهایت تست پروژه نشان داده می‌شود.



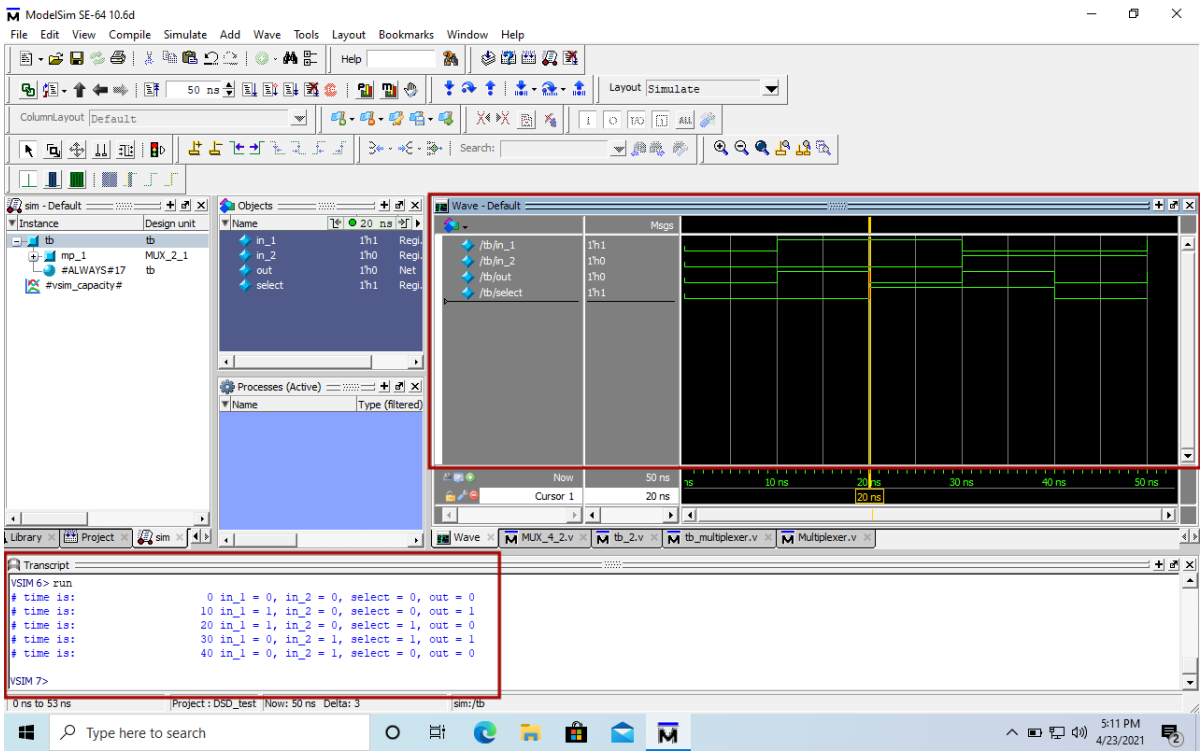
شکل ۵: در این مرحله ابتدا باید روی فایل مورد نظر راست کلیک کرده و گزینه Edit را انتخاب کنید(۱). سپس در پنجره بازه شده می‌توانید کد خود را بزنید(۲). بعد از نوشتن کد باید آن را Save کنید(۳). در نهایت با راست کلیک روی File مورد نظر، آن را کامپایل کنید(۴).



شکل ۶: بعد از تکمیل فایل‌ها و نوشتن test benchای‌برای تست برنامه، ابتدا وارد قسمت Library شوید(۱). work را انتخاب کنید(۲). روی test bench خود راست کلیک کنید و گزینه Simulate Without Optimization را انتخاب کنید(۳).



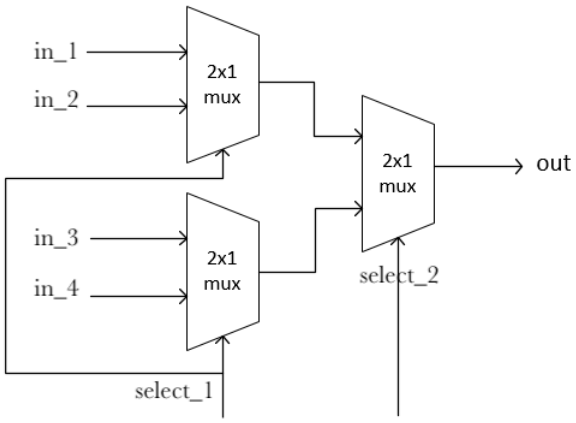
شکل ۷: برای دیدن Wave Form روی پنجره Wave کلیک کنید(۱). رجیسترها و سیم‌هایی که می‌خواهید تغییرات آن‌ها را مشاهده کنید به پنجره Wave منتقل کنید(۲). روی run کلیک کنید و برنامه را اجرا کنید(۳). برای اینکه بهتر بتوانید Wave را مشاهده کنید، روی Zoom کلیک کنید(۴).



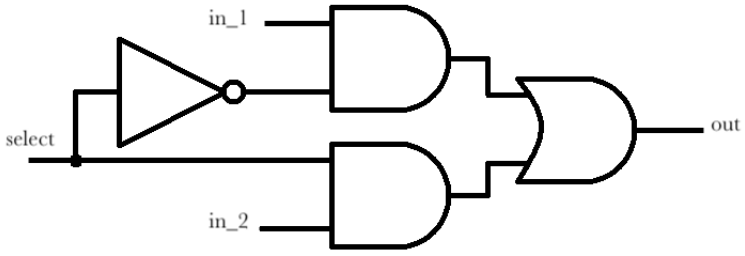
شکل ۸: می‌توانید Wave Form تغییرات رجیسترها و سیم‌های خود را در پنجره Wave ببینید. همینطور خروجی‌های برنامه (به عنوان مثال خروجی‌های monitor یا display) در قسمت Transcript قابل مشاهده است.

۲ برنامه‌نویسی verilog

در این قسمت یک مولتی‌پلکسر با ۴ ورودی و ۲ سلکت را به کمک کد ساختاری verilog می‌سازیم و تست می‌کنیم. همانطور که در شکل زیر مشخص شده می‌توان یک مولتی‌پلکسر 4 to 1 را به کمک سه مولتی‌پلکسر 2 to 1 زد.



مولتی‌پلکسرهای 2 t0 1 از ۲ گیت AND، ۱ گیت NOT و ۱ گیت OR تشکیل می‌شوند. در ادامه اتصالات این واحد و کد verilog مربوط به آن آمده است.




```
1
2  // modeling a 2*1 multiplexer unit
3  // structural code
4
5  module MUX_2_1(
6          input in_1,
7          input in_2,
8          input select,
9          output out);
10
11         wire select_bar, in_1_selected, in_2_selected;
12
13         not not_gate(select_bar, select);
14         and and_gate_1(in_1_selected, select_bar, in_1);
15         and and_gate_2(in_2_selected, select, in_2);
16         or or_gate(out, in_1_selected, in_2_selected);
17
18 endmodule
```

می‌توان برای تست و شبیه‌سازی این ماژول، از test bench زیر استفاده کرد.

```
1
2  // test-bench for MUX_2_1
3
4  module tb();
5
6          reg in_1, in_2, select;
7          wire out;
8
9          MUX_2_1 mp_1(.in_1(in_1), .in_2(in_2), .select(select), .out(out));
10
11         initial
12         begin
13                 in_1 = 0;
14                 in_2 = 0;
15                 select = 0;
16         end
17
18         always
19         begin
20                 #10
21                 in_1 = 1;
22
23                 #10
24                 select = 1;
25
26                 #10
27                 in_1 = 0;
28                 in_2 = 1;
29
30                 #10
31                 select = 0;
32         end
33
34         initial
35                 $monitor("time is: ", $time, " in_1 = %b, in_2 = %b, select = %b, out = %b"
36                 , in_1, in_2, select, out);
37
38 endmodule
```

برای ساخت یک مولتی‌پلکسر 4 to 1 به کمک ماژول ساخته شده، از کد زیر استفاده می‌کنیم.

```
1
2  // modeling a 4 2 multiplexer
3
4  module MUX_4_2(
5          input in_1, in_2, in_3, in_4,
6          input select_1, select_2,
7          output out);
8
9          wire mux_1_out, mux_2_out;
10
11         MUX_2_1 mux_1(.in_1(in_1), .in_2(in_2), .select(select_1), .out(mux_1_out));
12         MUX_2_1 mux_2(.in_1(in_3), .in_2(in_4), .select(select_1), .out(mux_2_out));
13         MUX_2_1 mux_3(.in_1(mux_1_out), .in_2(mux_2_out), .select(select_2), .out(out));
14
15 endmodule
```

در نهایت می‌توان با test bench زیر مولتی‌پلکسر 4 to 1 را طبق مراحل ذکر شده شبیه‌سازی کرد.

```
1
2  // test-bench for MUX_4_2
3  module tb_2();
4          reg in_1, in_2, in_3, in_4, select_1, select_2;
5          wire out;
6
7          MUX_4_2 mp_1(.in_1(in_1), .in_2(in_2),
8          .in_3(in_3), .in_4(in_4), .select_1(select_1), .select_2(select_2), .out(out));
9
10         initial
11         begin
12                 in_1 = 0;
13                 in_2 = 0;
14                 in_3 = 0;
15                 in_4 = 0;
16                 select_1 = 0;
17                 select_2 = 0;
18         end
19         always
20         begin
21                 #5
22                 in_1 = 1;
23
24                 #5
25                 select_1 = 1;
26                 select_2 = 0;
27                 // choose in_2
28
29                 #5
30                 in_1 = 0;
31                 in_2 = 1;
32
33                 #5
34                 select_1 = 0;
35                 select_2 = 1;
36                 // choose in_3
37
38                 #5
39                 in_2 = 0;
40                 in_3 = 1;
41
42                 #5
43                 select_1 = 1;
44                 select_2 = 1;
45                 // choose in_4
46
47                 #5
48                 in_3 = 0;
49                 in_4 = 1;
50         end
51
52         initial
53                 $monitor("time is: ", $time, " in_1 = %b, in_2 = %b, in_3 = %b, in_4 = %b,
54                 select_1 = %b, select_2 = %b, out = %b",
55                 in_1, in_2, in_3, in_4, select_1, select_2, out);
56 endmodule
```