

کد مربوط به این سوال از ۳ بخش اصلی تشکیل شده است:

۱- fsm\_parser: در این بخش کلاس FSMParser مسئولیت پارس کردن توصیف FSM داده شده در صورت سوال را بر عهده دارد. این کلاس دقیقاً بر اساس فرمت ورودی، بخش‌های مختلف FSM را به دست می‌آورد و برای مراحل بعد ذخیره می‌کند.

۲- verilog\_generator: در این بخش مسئولیت اصلی را کلاس VerilogFSMGenerator بر عهده دارد که باید از با توجه به FSM به دست آمده در مرحله قبل، کد ورپلاگ مربوط به آن را تولید کند. برای انجام این کار تعدادی کلاس که همگی از کلاس VerilogComponentGenerator ارث بری کرده‌اند برای کامپوننت‌های مختلف زبان ورپلاگ پیاده‌سازی شده‌اند که شامل VerilogInitialGenerator، VerilogAlwaysGenerator، VerilogCaseItemGenerator و VerilogCaseGenerator وجود دارد.

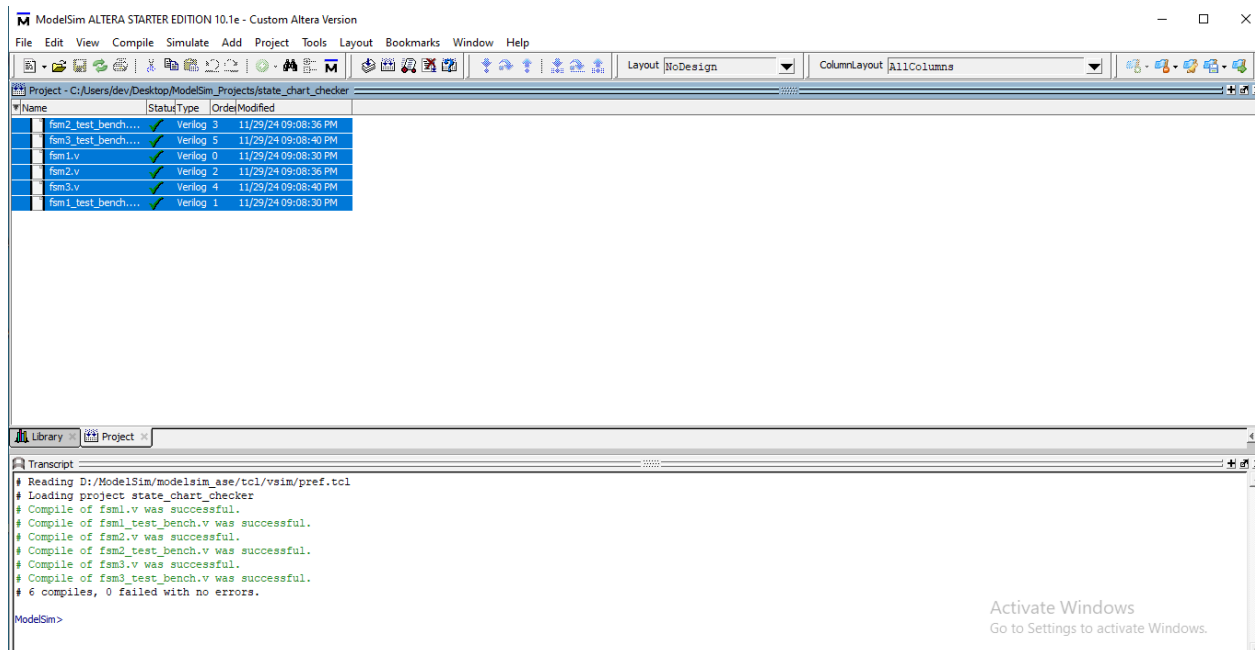
۳- test\_bench\_generator: این بخش از برنامه وظیفه‌ی ایجاد تولید تست بنچ برای fsm به دست آمده در مرحله اول را دارد. برای این که بتوانیم تمامی transition‌های موجود در FSM را تست کنیم و مسیرهای مختلف را تولید کنیم از DFS استفاده شده است. به این صورت که روی استیت آغازین FSM، الگوریتم DFS را اجرا می‌کنیم و با استفاده از mark کردن استیت‌های مختلف، مسیرهای مختلف موجود در FSM را به دست آورده و در نهایت آن‌ها را در قالب کد ورپلاگ تست بنچ تولید می‌کنیم.

ساختار کلی فولدر سوال نیز به این صورت است که تست‌های داده شده در سوال در فولدر tests، کدها و تست‌بنچ‌های تولید شده به زبان verilog در فولدر verilogs، عکس‌های مربوط به Simulation های انجام شده در نرم‌افزار ModelSim نیز در فولدر simulation\_pics قرار داده شده‌اند.

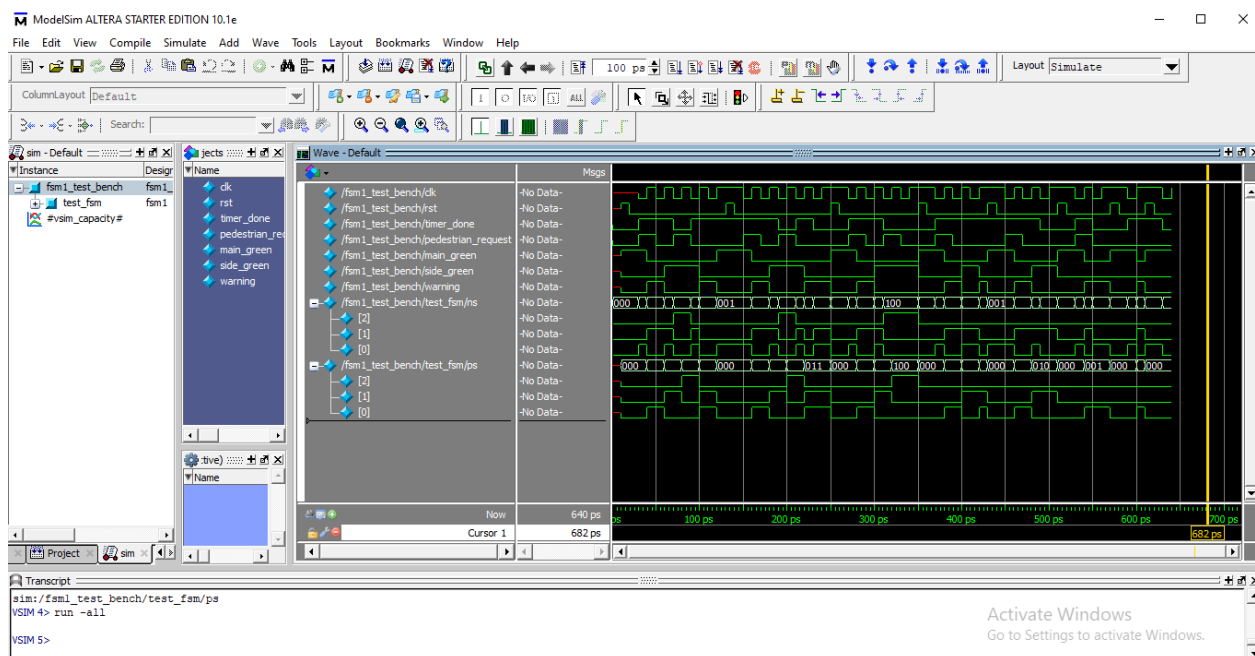
برای اجرای کد و تولید فایل‌های ورپلاگ نیز باید از دستور زیر استفاده شود:

```
python main.py [path/to/fsm_description.txt] output_module_name
```

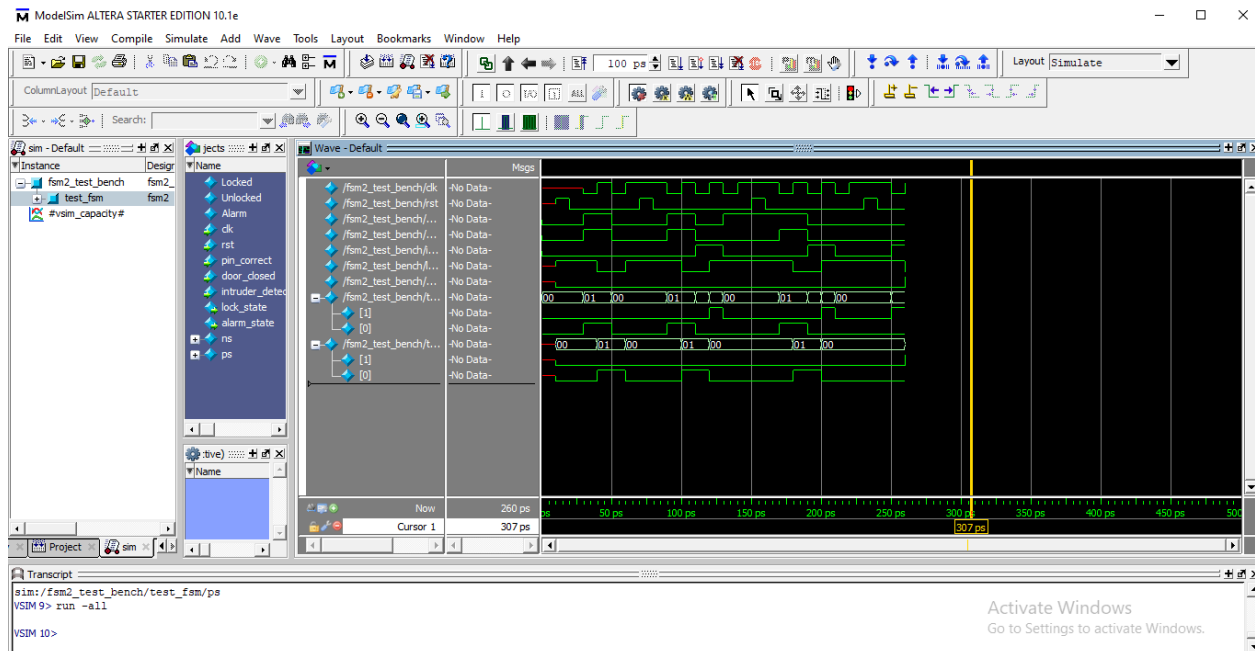
همچنین عکس‌های به دست از شبیه‌سازی کدهای به دست آمده برای هر ۳ تست صورت سوال در نرم‌افزار ModelSim در ادامه قرار داده شده است:



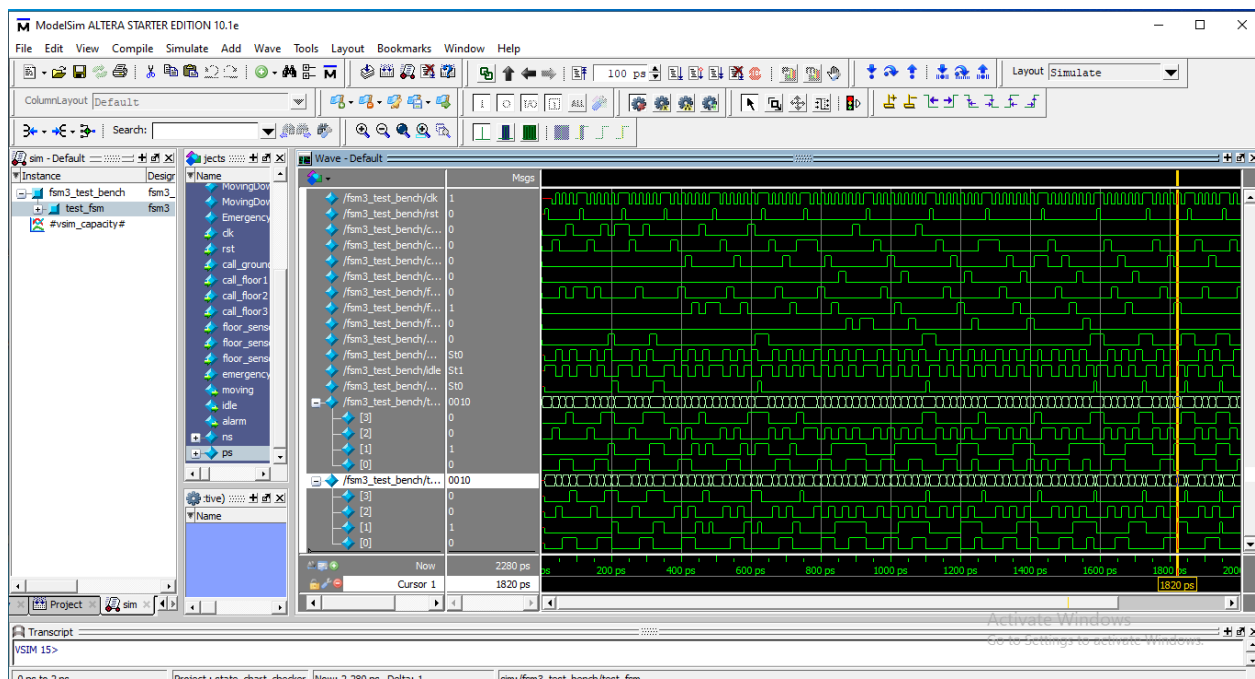
تصویر مربوط به کامپایل شدن تمامی کدهای تولید شده



تصویر مربوط به شبیه سازی تست اول



تصویر مربوط به شبیه‌سازی تست دوم



تصویر مربوط به شبیه‌سازی تست سوم