Accessing Code Coverage in Questa

Access the ECE Linux Server and Launch Questa:

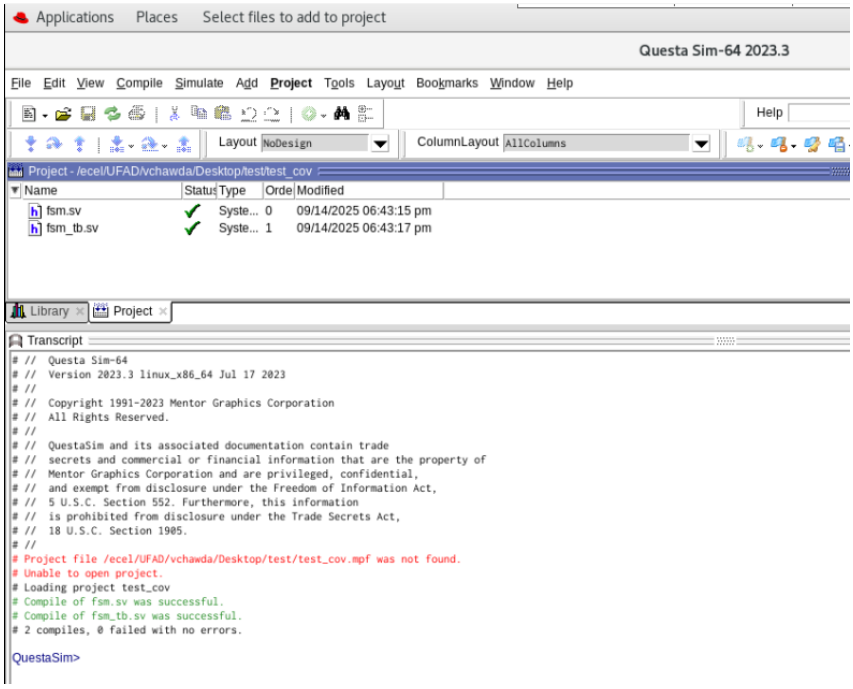
- Connect to the ECE Linux server.
- Once connected start the Questa.

Create a New Project:

- In the Questa GUI, go to **File → New → Project**.
- Enter a project name and select a working directory.
- Add your design and testbench source files to the project.

Compile the Design Files:

- In the **Project** window, right-click on the source files and choose **Compile**.
- Ensure that all files compile successfully and no errors in the **Transcript window**.



Designate Coverage Statistics to Collect:

At the Questa SIM> prompt, type:

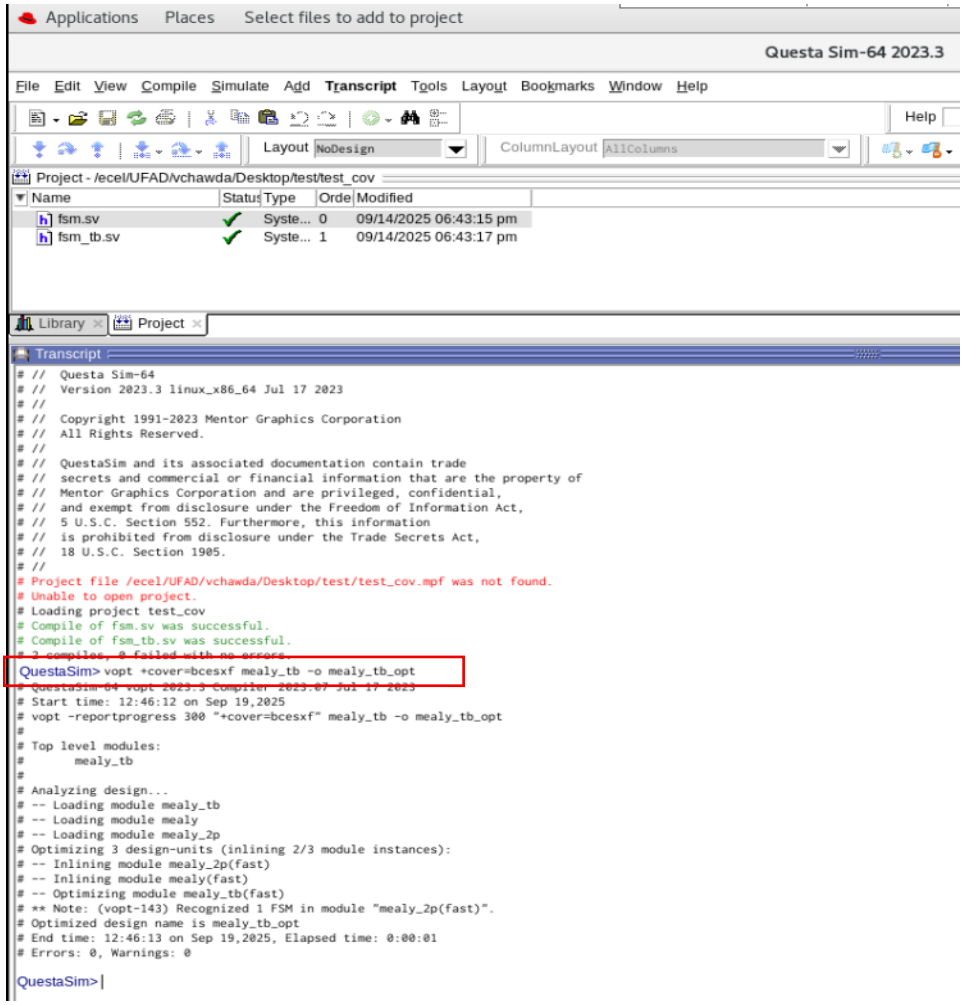
vopt +cover=bcesxf fsm_tb -o fsm_tb _opt and press enter.

// where fsm_tb is the **module** name of your testbench file

In this command:

- **+cover=bcesxf** instructs Questa SIM to collect branch, condition, expression, statement, extended toggle, and FSM coverage statistics.
- **test_sm** is the top-level module (testbench module) you are optimizing for coverage. Replace **test_sm** with the name of your own testbench module.
- **test_sm_opt** is the optimized design that will be used for simulation.

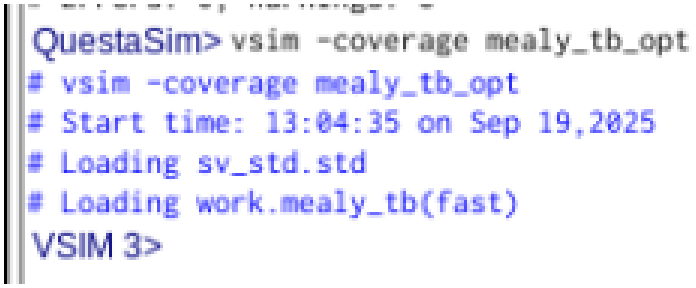
Accessing Code Coverage in Questa



For instance, in the above example the testbench module name is mealy_tb. Therefore, the command becomes: **vopt +cover=bcesxf mealy_tb -o mealy_tb_opt**. When this command is entered, the optimization runs successfully and completes without any errors.

Load the Optimized Design with Coverage Enabled

- At the **Questa SIM> prompt**, type:
  vsim -coverage fsm_tb _opt
- Replace mealy_tb_opt with the name of your own optimized testbench design if different.



Accessing Code Coverage in Questa

Add the Wave Window and select signals to display

- **add wave *** // This will add all the signals in the testbench

Alternatively, if you want to be selective in what signals to add (e.g., from different modules)

- Go to and select **View->Object** and the Object window will appear (if it is not there already)
- Right click on a module (e.g., DUT) and select the signals of interest (e.g., current state and next state)
- Right click on the another module of interest (e.g., fsm.tb) and select the rest of the signals (e.g., clk, rst, inbit, buffull, counten, regld, and outflag).

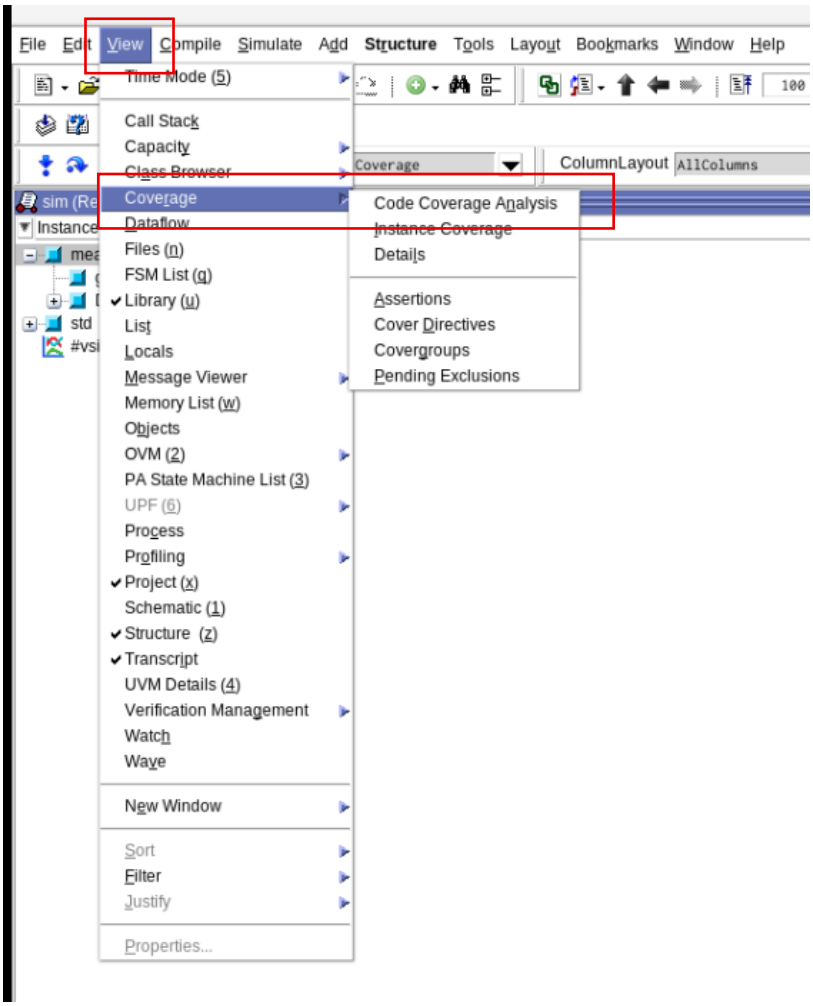
Run the Simulation

- At the **Questa SIM> prompt**, type:
  run -all (in transcript window)

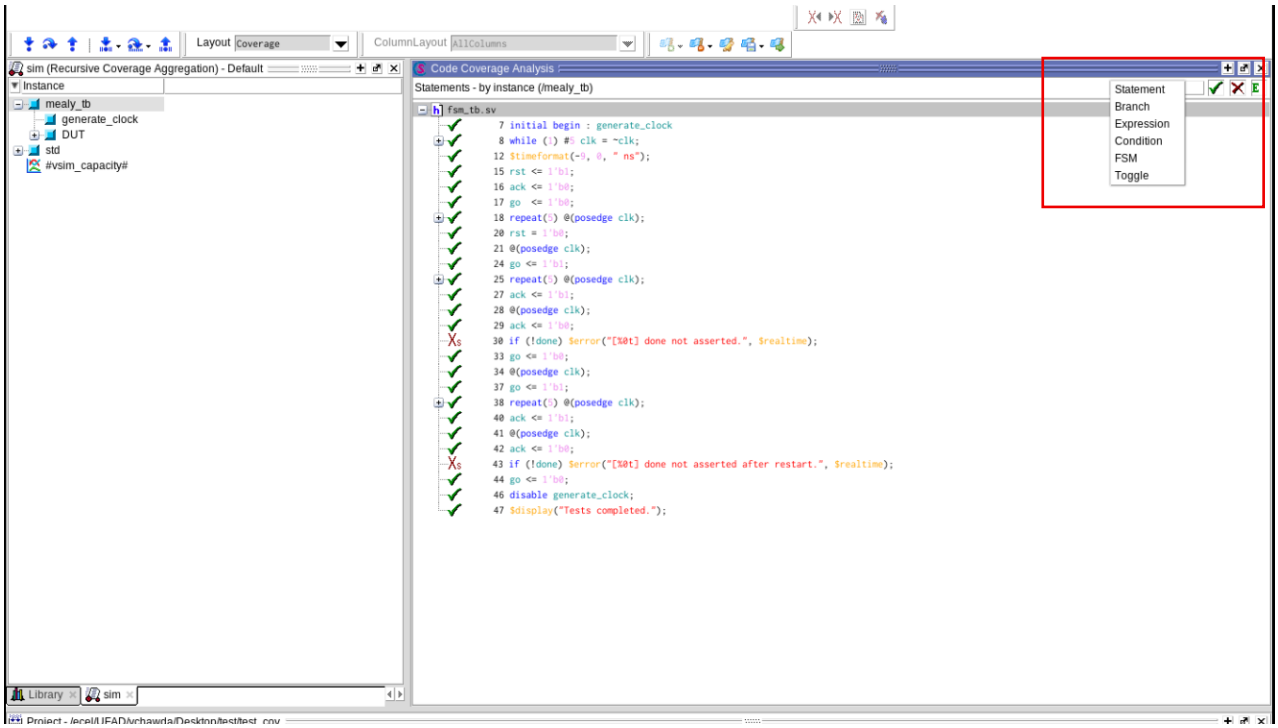
```
VSIM 3> run -all
# Tests completed.
```

View Code Coverage Results:

After the simulation completes, open the coverage window from the menu:
View → Coverage → Code Coverage Analysis



Accessing Code Coverage in Questa



In the **Code Coverage Analysis** window, you can select the type of coverage to view from the drop-down menu in the top-right corner.

How to Generate Coverage Summary in Questa:

- In QuestaSim terminal/transcript after simulation.
- Type the command: **coverage report -summary** and press enter
- The transcript will display the coverage summary (line, branch, expression, toggle, FSM, etc.) as shown below.

```
VSIM 5> coverage report -summary
# Coverage Report Totals BY INSTANCES: Number of Instances 2
#
#     Enabled Coverage              Bins      Hits    Misses    Weight  Coverage
#     ----------------              ----      ----    ------    ------  --------
#     Branches                        11        11         0         1   100.00%
#     FSM States                       4         4         0         1   100.00%
#     FSM Transitions                  6         5         1         1    83.33%
#     Statements                      44        44         0         1   100.00%
#     Toggles                         44        26        18         1    59.09%
# Total coverage (filtered view): 88.48%
#
#
```

Accessing Code Coverage in Questa

Icon	Description/Indication
✓	All statements, branches, conditions, or expressions on a particular line have been executed
X	Multiple kinds of coverage on the line were not executed
X_T	True branch not executed (BC column)
X_F	False branch not executed (BC column)
X_C	Condition not executed (Hits column)
X_E	Expression not executed (Hits column)

Icon	Description/Indication
X_B	Branch not executed (Hits column)
X_S	Statement not executed (Hits column)
E	Indicates a line of code to which active coverage exclusions have been applied. Every item on the line is excluded; none are hit.
Ew	Some excluded items are hit
Ew	Some items are excluded, and all items not excluded are hit
Ex	Some items are excluded, and some items not excluded have missing coverage
Ea	Auto exclusions have been applied to this line. Hover the cursor over the EA and a tool tip balloon appears with the reason for exclusion,

Description of Icons displayed during code coverage

Summary of commands (after successful compilation of design and testbench files)

vopt +cover=bcesxf fsm_tb -o fsm_tb_opt
// where fsm_tb is the **module** name of your testbench file.
// specifies the code coverage statistics to collect and produce an optimized the testbench
vsim -coverage fsm_tb_opt
add wave *
run -all

View → Coverage → Code Coverage Analysis
coverage report -summary