Verdi

- Verilog Debugging Tool

Introduction to Verdi

- The Verdi Automated Debug System is an advanced open platform for debugging digital designs with powerful technology that helps you:
  1. Comprehend complex and unfamiliar design behavior.
  2. Automate difficult and tedious debug processes.

Basic Function

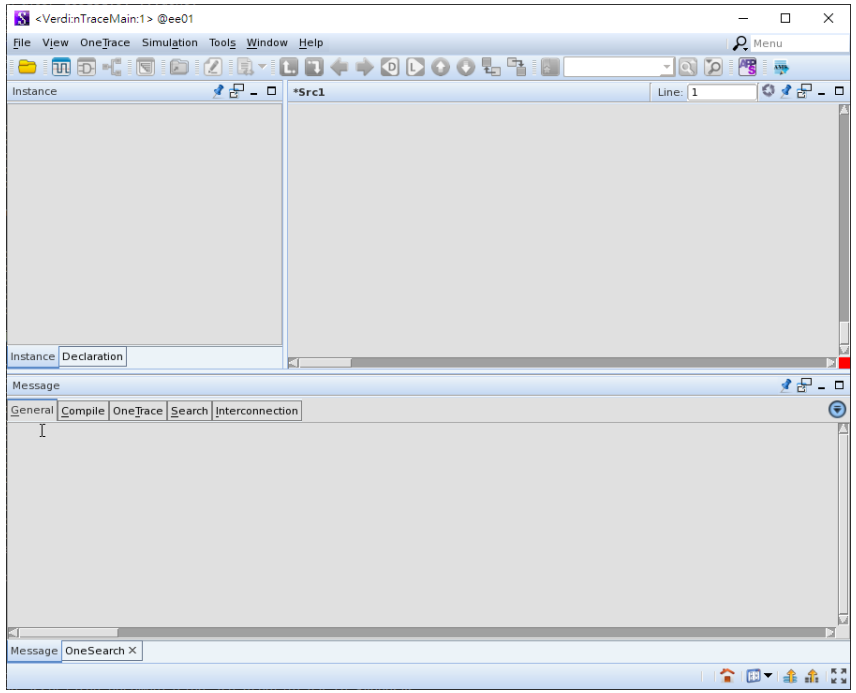
- nTrace
  - A source code viewer and analyzer that operates on the knowledge database to display the design hierarchy and source code for selected design blocks.
  - The main window of Verdi.
- nSchema
  - A schematic viewer and analyzer that generates interactive debug-specific logic diagrams showing the structure of selected portions of a design.
- nWave
  - A state-of-the-art graphical waveform viewer and analyzer that is fully integrated with Verdi's source code, schematic, and flow views.

Start Verdi

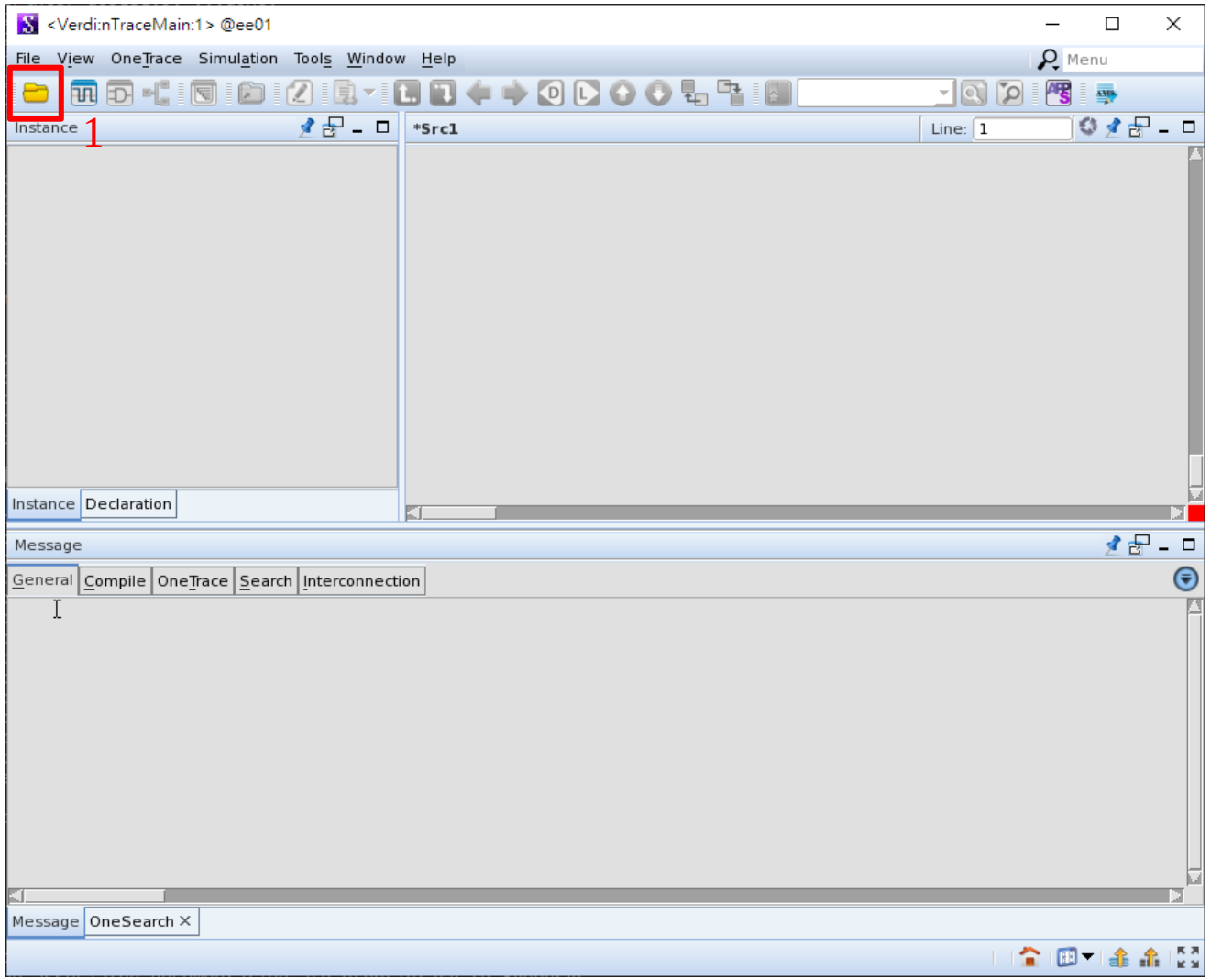
- Type the following command on the terminal:
  - verdi &

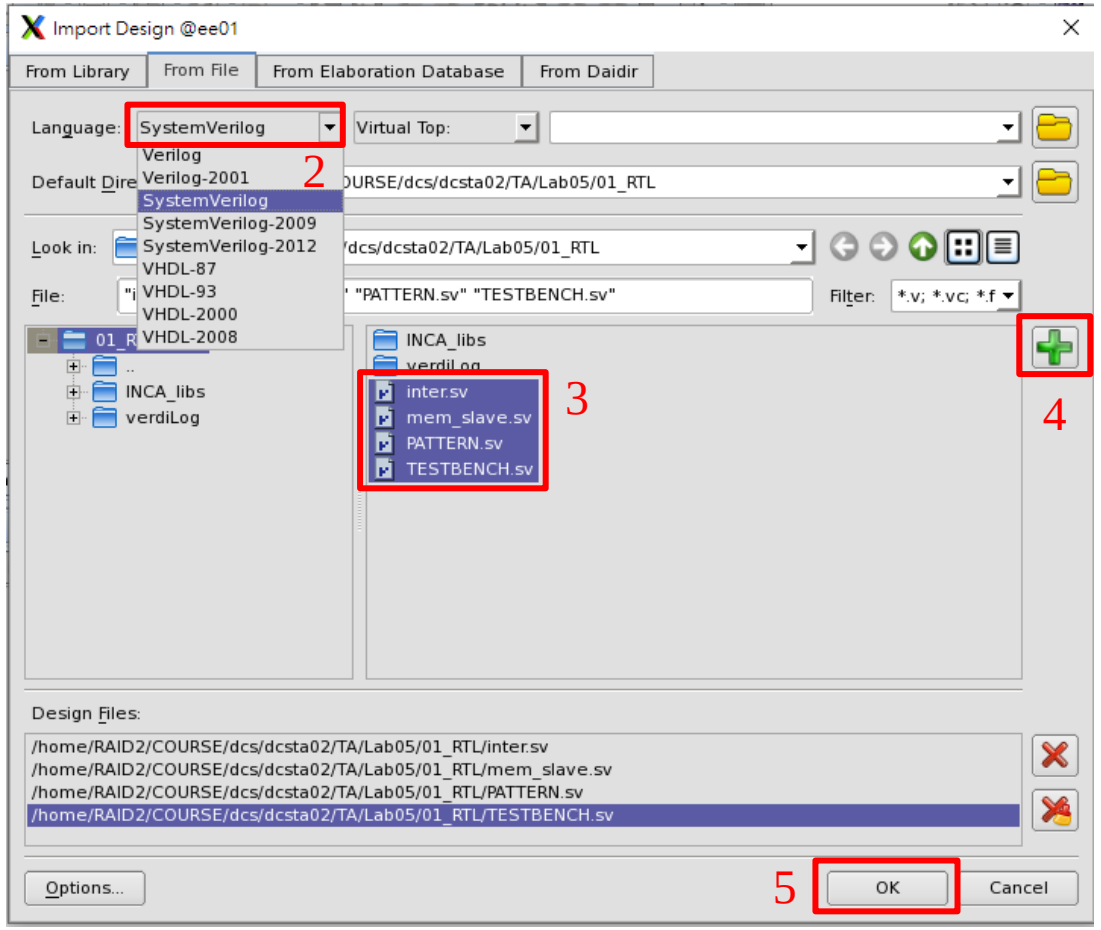
Example: `ee01 [Lab05/01_RTL]% verdi &`

  - Also, the token "&" enable you to use the terminal while Verdi is running in the background.



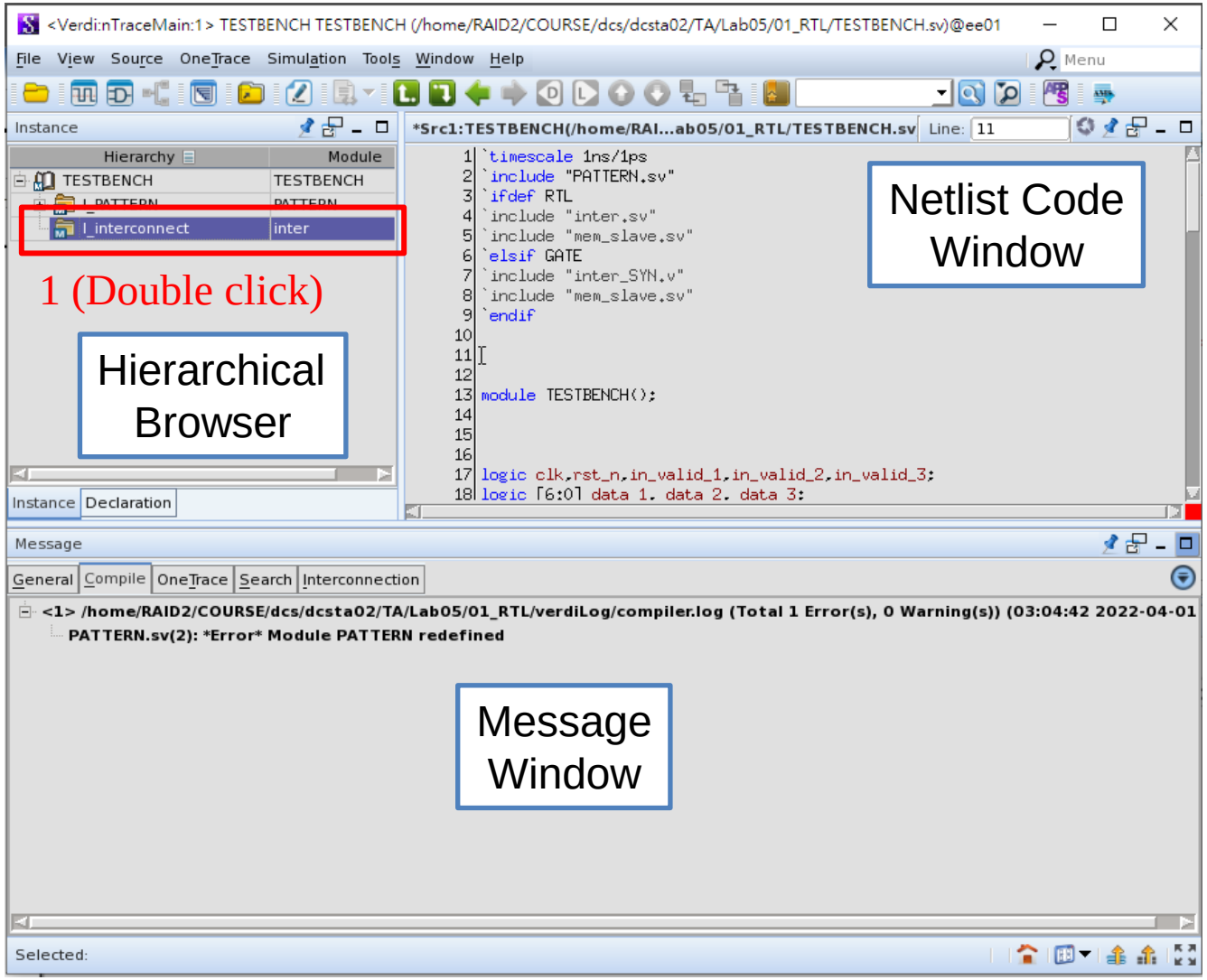
Import Design





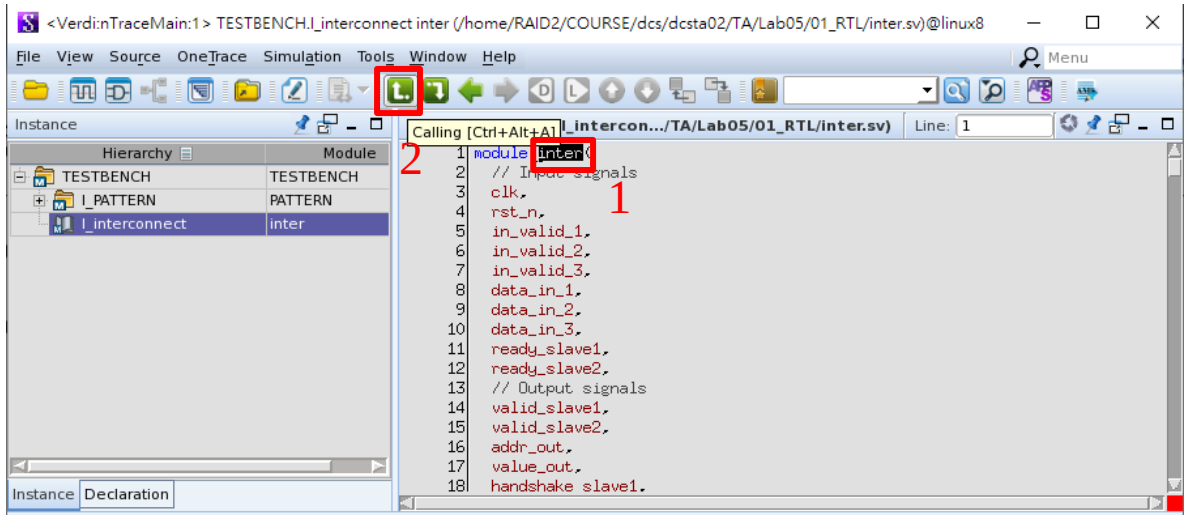
nTrace – Error checking

- After importing all code files, Verdi will compile your code first, and you can check the syntax errors in message window.
  - Because pattern.sv contains non-synthesizable syntax, just ignore the error messages about pattern.
- Through Verdi, you can debug your code before simulation.

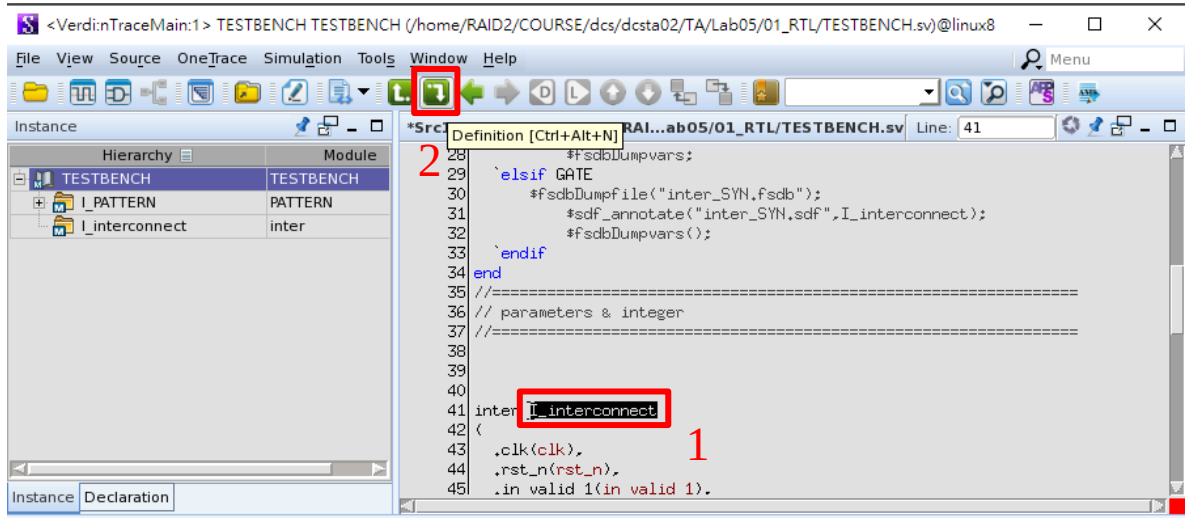


nTrace – Hierarchy tracing

- (1) You can trace which top module calling this submodule.
- (2) You can trace the definition of the called submodule.



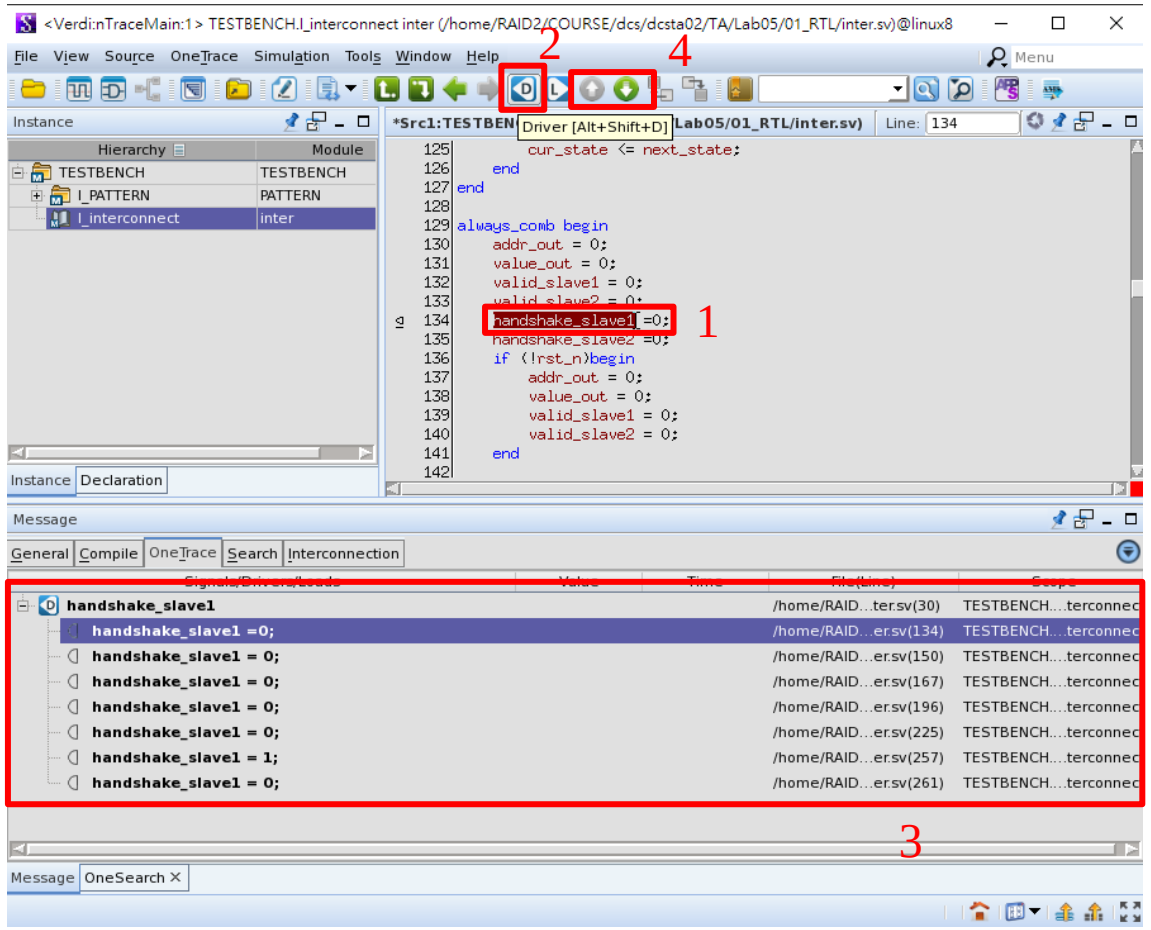
(1) Find which module calling this submodule

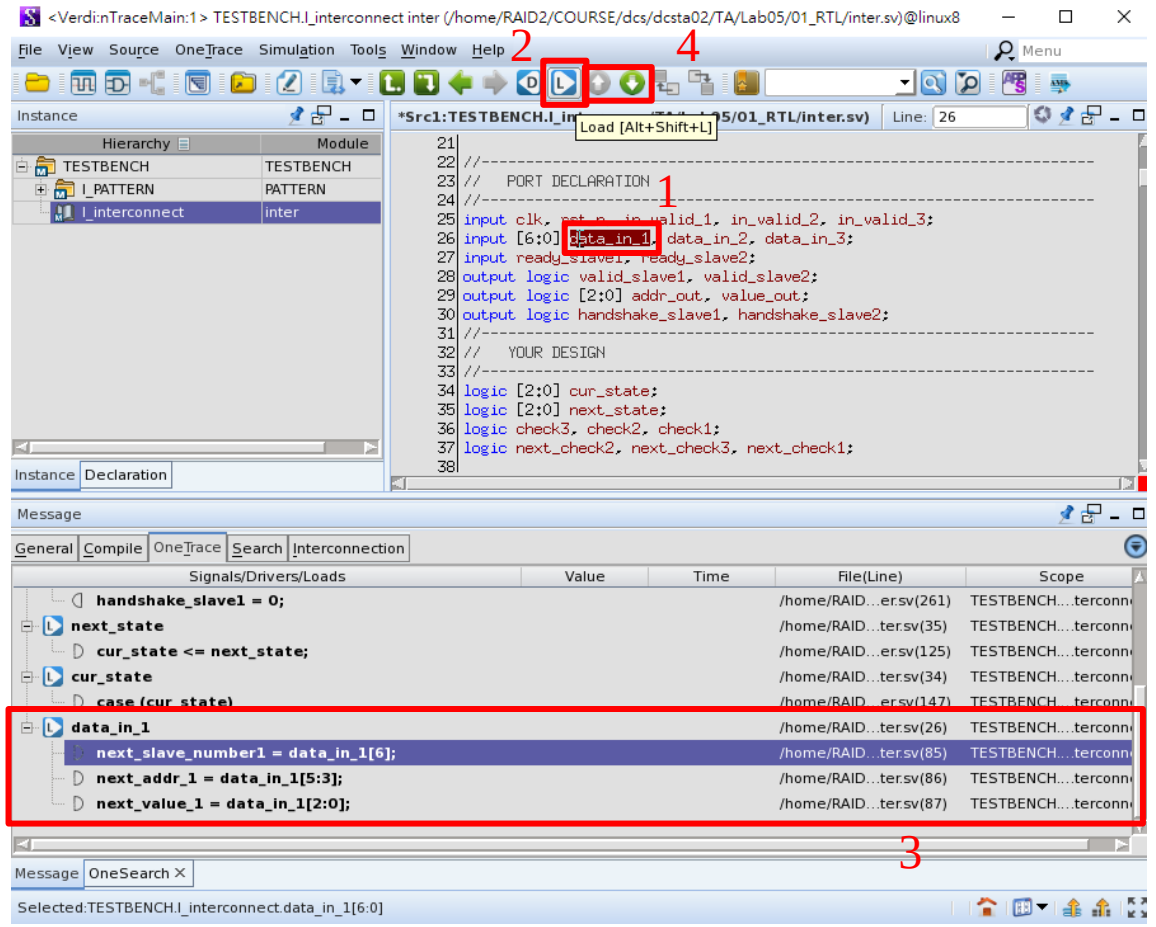


(2) Find the definition of this submodule

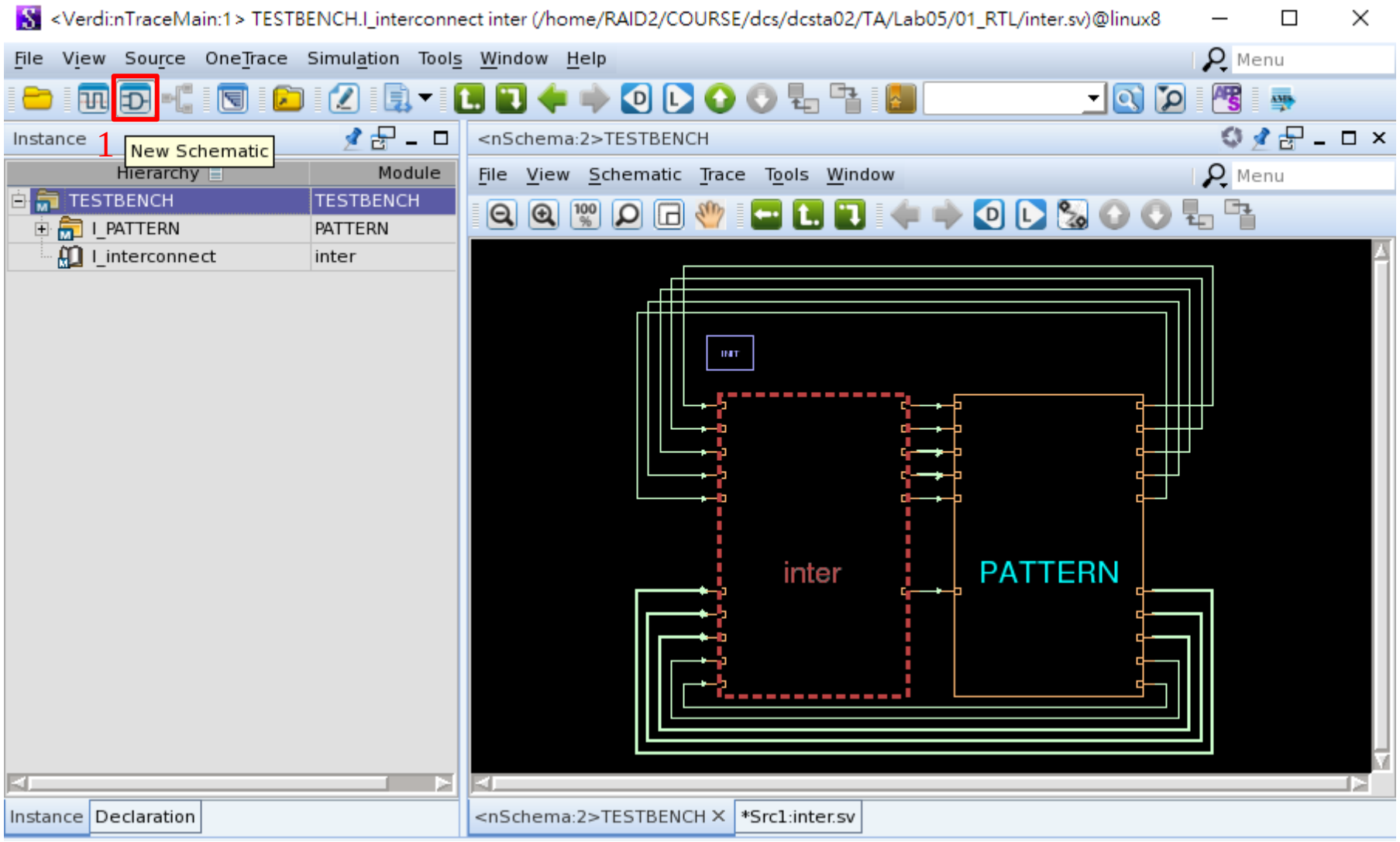
nTrace – Signal tracing

- (1) You can trace this signal is loading to which signals.
- (2) You can trace this signal is driven by which signals.

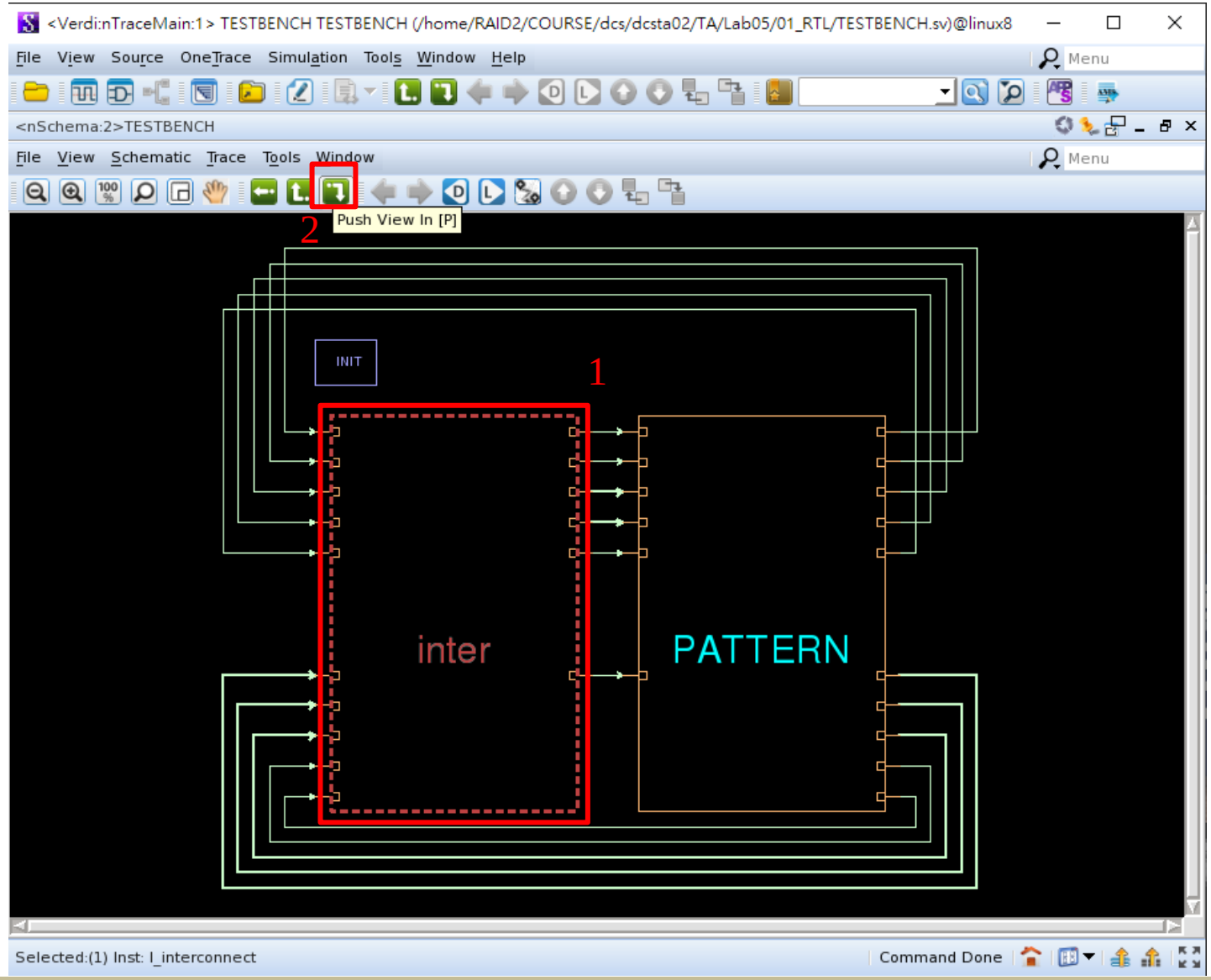




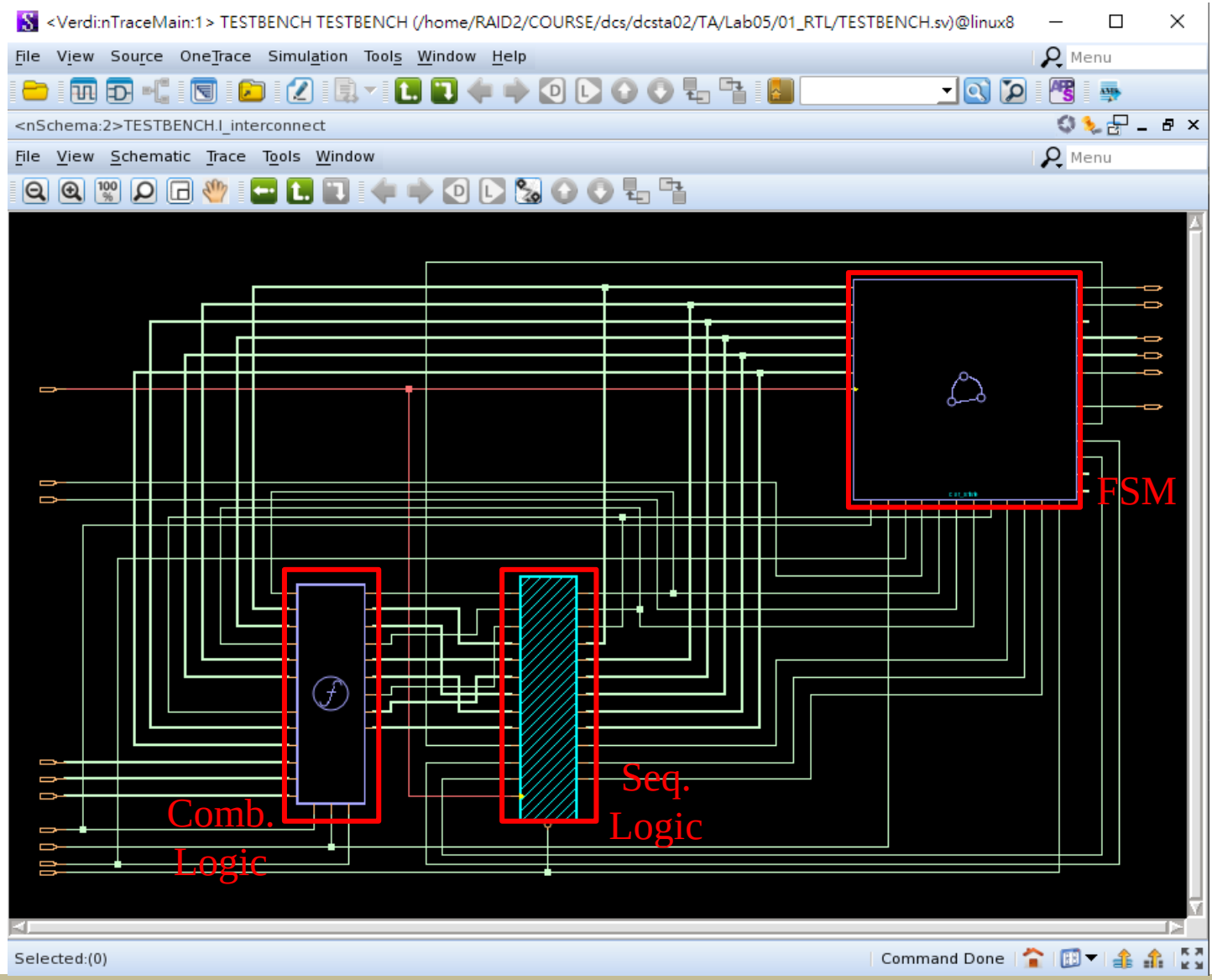
nSchema


<Verdi:nTraceMain:1> TESTBENCH.I_interconnect inter (/home/RAID2/COURSE/dcs/dcsta02/TA/Lab05/01_RTL/inter.sv)@linux8
File View Source OneTrace Simulation Tools Window Help
Menu
Instance
1
New Schematic
Hierarchy
Module
TESTBENCH
TESTBENCH
I_PATTERN
PATTERN
I_interconnect
inter
<nSchema:2>TESTBENCH
File View Schematic Trace Tools Window
Menu
inter
PATTERN
Instance
Declaration
<nSchema:2>TESTBENCH
*Src1:inter.sv

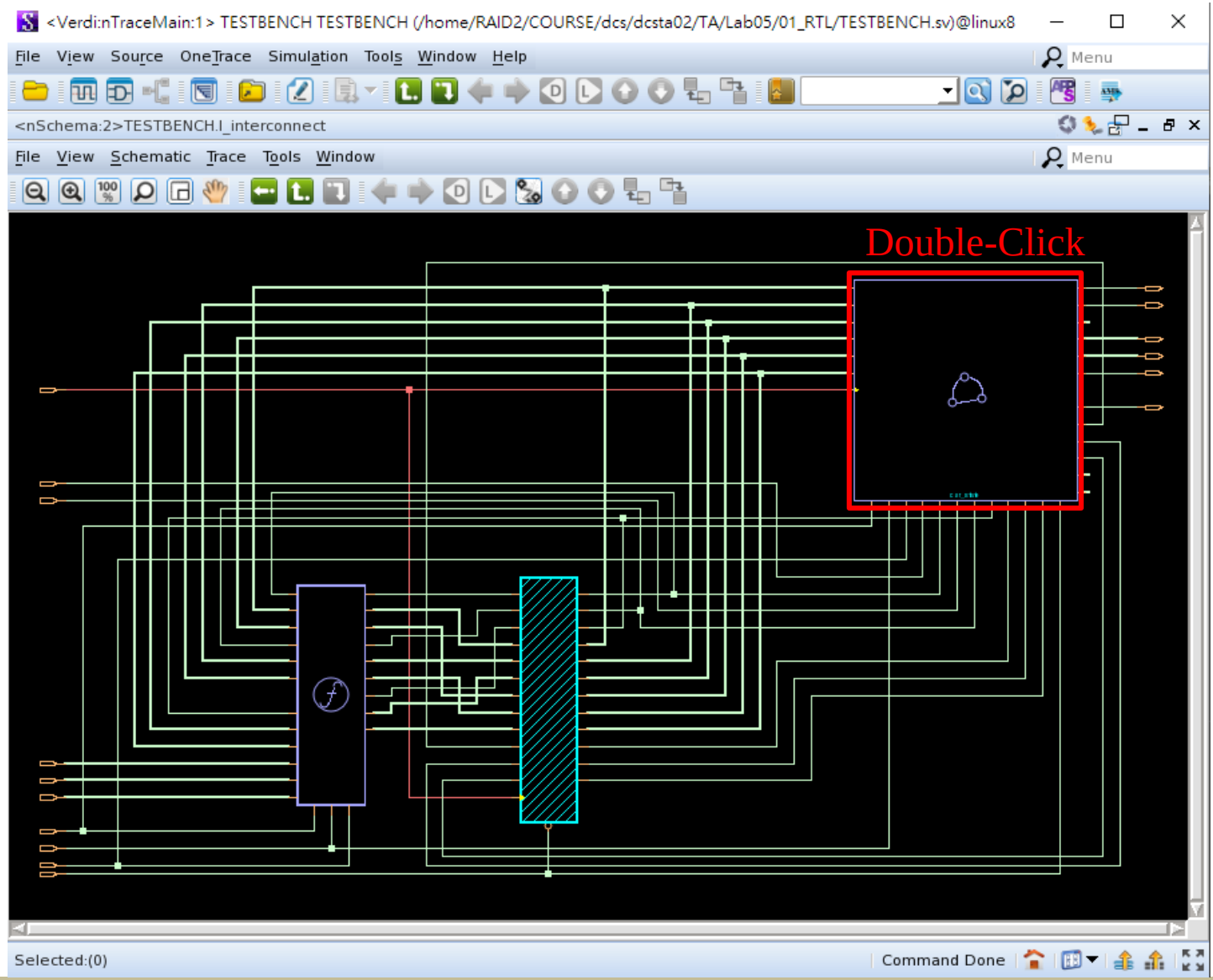
nSchema


<Verdi:nTraceMain:1> TESTBENCH TESTBENCH (/home/RAID2/COURSE/dcs/dcsta02/TA/Lab05/01_RTL/TESTBENCH.sv)@linux8
File View Source OneTrace Simulation Tools Window Help
Menu
<nSchema:2>TESTBENCH
File View Schematic Trace Tools Window
Menu
Push View In [P]
2
1
INIT
inter
PATTERN
Selected:(1) Inst: I_interconnect
Command Done

nSchema

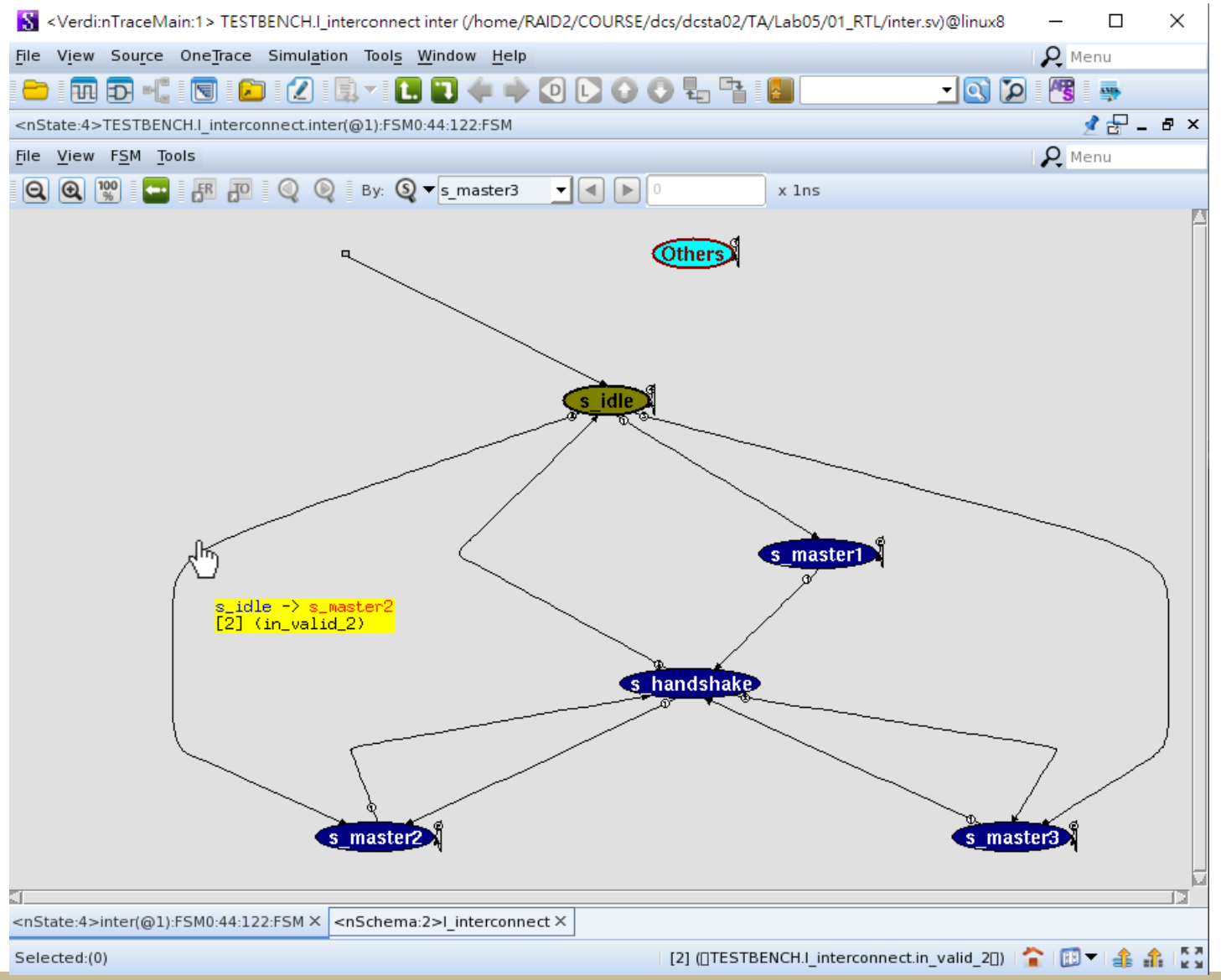

<Verdi:nTraceMain:1> TESTBENCH TESTBENCH (/home/RAID2/COURSE/dcs/dcsta02/TA/Lab05/01_RTL/TESTBENCH.sv)@linux8
File View Source OneTrace Simulation Tools Window Help
Menu
<nSchema:2>TESTBENCH.I_interconnect
File View Schematic Trace Tools Window
Menu
FSM
Comb. Logic
Seq. Logic
Selected:(0)
Command Done

nSchema – FSM

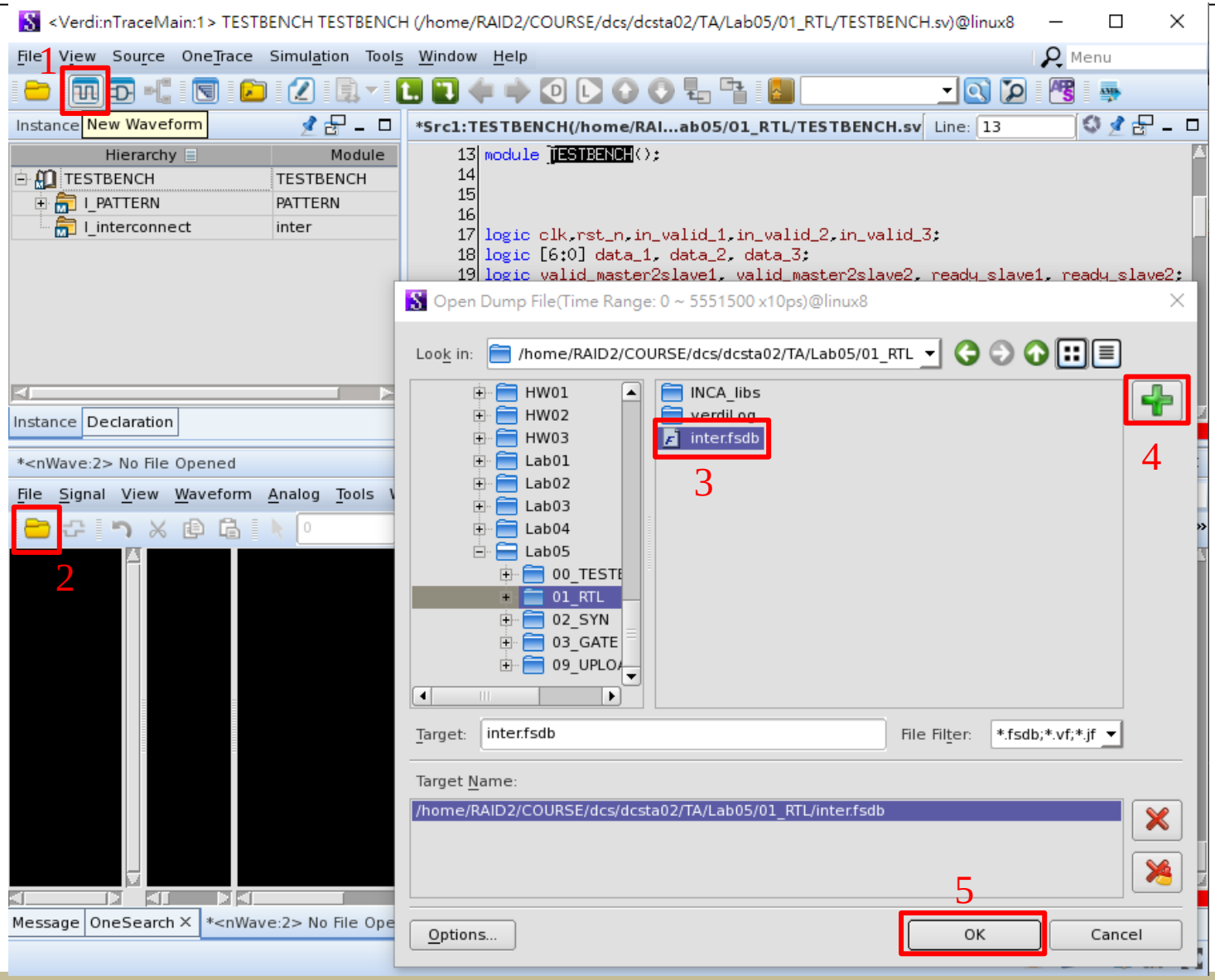

<Verdi:nTraceMain:1> TESTBENCH TESTBENCH (/home/RAID2/COURSE/dcs/dcsta02/TA/Lab05/01_RTL/TESTBENCH.sv)@linux8
File View Source OneTrace Simulation Tools Window Help
Menu
<nSchema:2>TESTBENCH.I_interconnect
File View Schematic Trace Tools Window
Menu
Double-Click
Selected:(0)
Command Done

nSchema – FSM

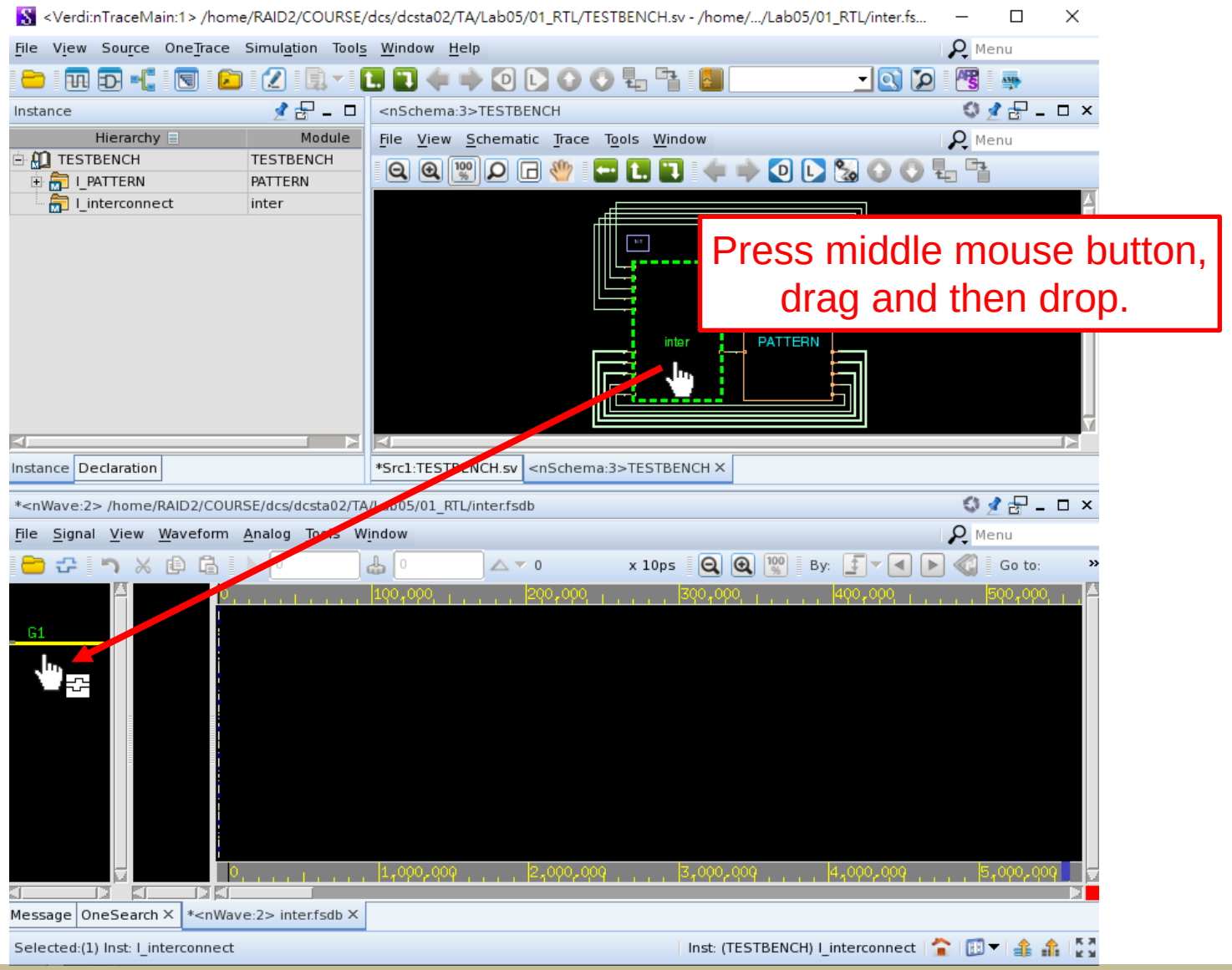
- You can watch the FSM transition diagram here.
- Keep your mouse cursor on the transition arrows, it will show you the state transition condition.



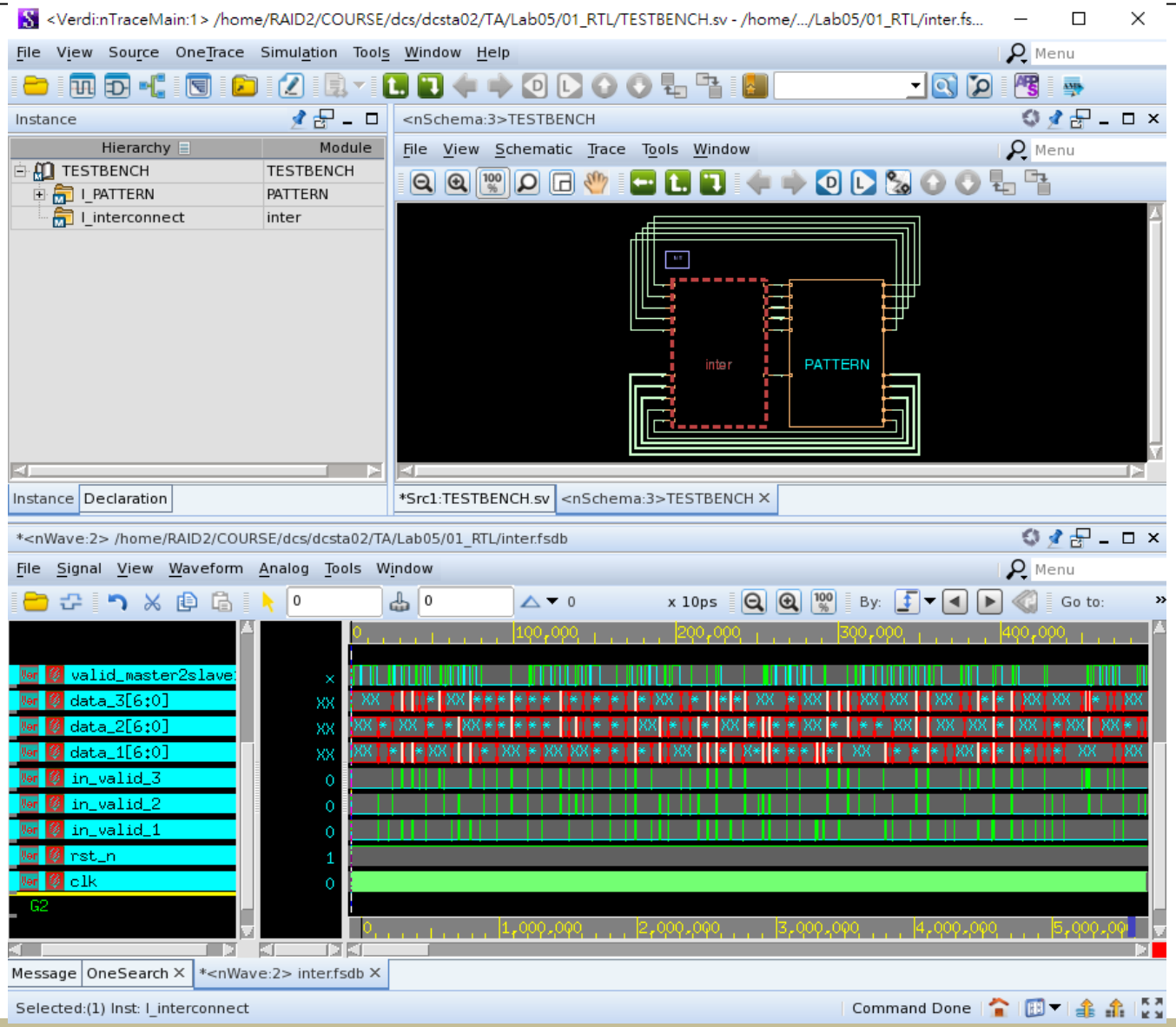
nWave


<Verdi:nTraceMain:1> TESTBENCH TESTBENCH (/home/RAID2/COURSE/dcs/dcsta02/TA/Lab05/01_RTL/TESTBENCH.sv)@linux8
File View Source OneTrace Simulation Tools Window Help
1
New Waveform
Instance
Hierarchy
Module
TESTBENCH
I_PATTERN
I_interconnect
PATTERN
inter
*Src1:TESTBENCH(/home/RAI...ab05/01_RTL/TESTBENCH.sv
Line: 13
module TESTBENCH();
logic clk,rst_n,in_valid_1,in_valid_2,in_valid_3;
logic [6:0] data_1, data_2, data_3;
Instance Declaration
*<nWave:2> No File Opened
File Signal View Waveform Analog Tools
2
Message OneSearch
Open Dump File(Time Range: 0 ~ 5551500 x10ps)@linux8
Look in: /home/RAID2/COURSE/dcs/dcsta02/TA/Lab05/01_RTL
HW01
HW02
HW03
Lab01
Lab02
Lab03
Lab04
Lab05
01_RTL
02_SYN
03_GATE
INCA_libs
inter.fsdb
3
4
Target: inter.fsdb
File Filter: *.fsdb;*.vf;*.jf
Target Name:
/home/RAID2/COURSE/dcs/dcsta02/TA/Lab05/01_RTL/inter.fsdb
5
Options...
OK
Cancel

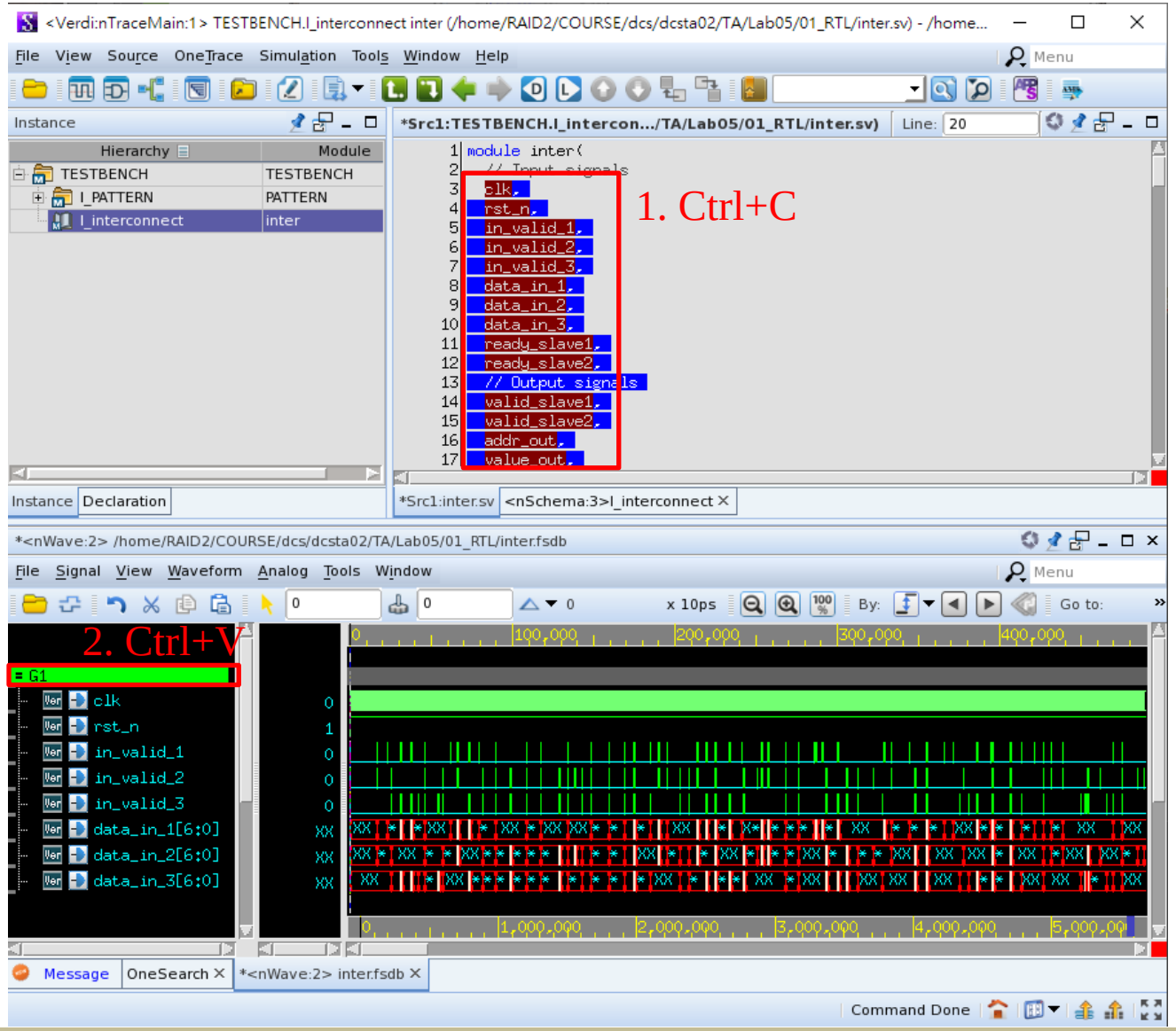
nWave


Press middle mouse button, drag and then drop.
inter
PATTERN
G1

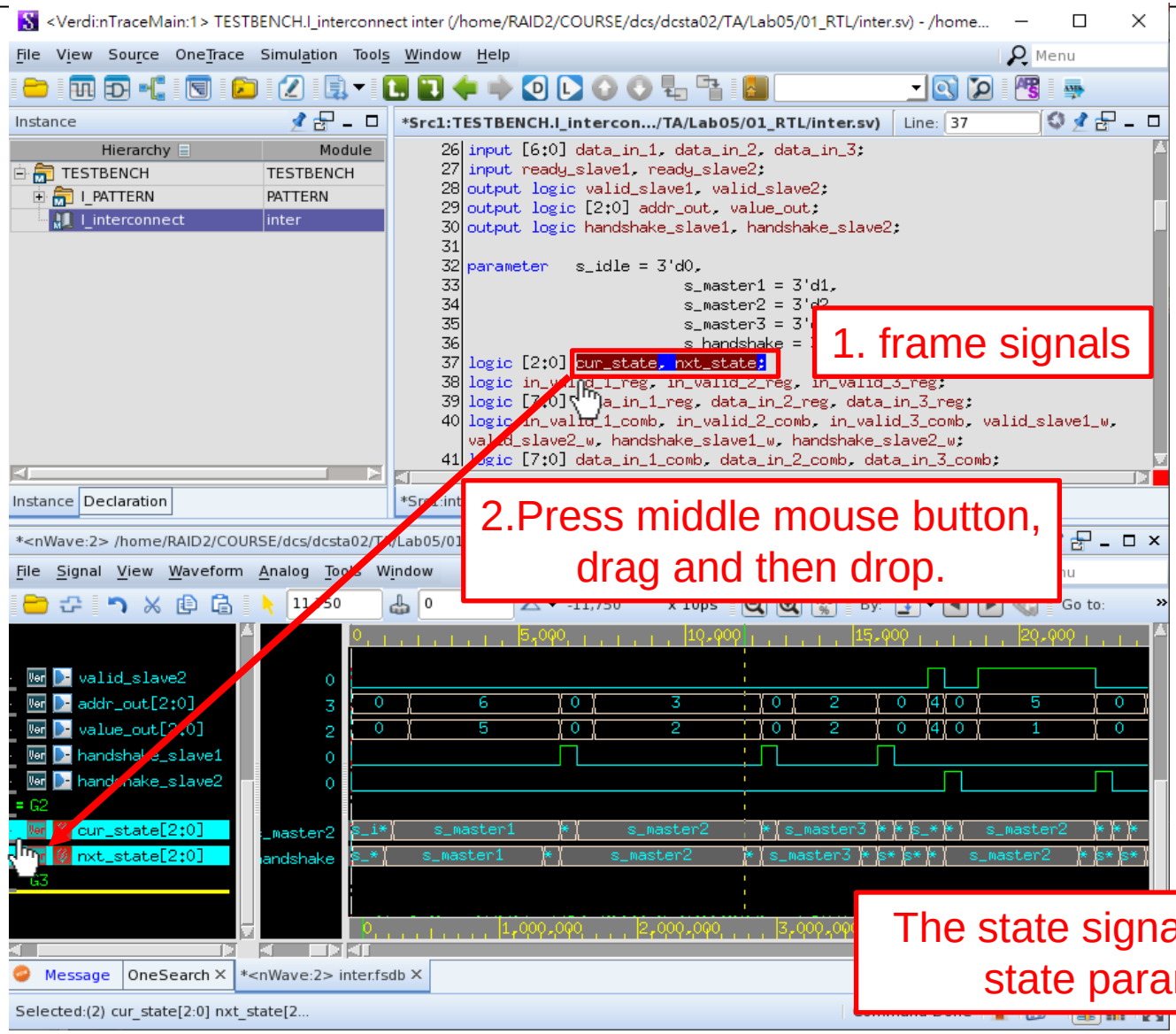
nWave


<Verdi:nTraceMain:1> /home/RAID2/COURSE/dcs/dcsta02/TA/Lab05/01_RTL/TESTBENCH.sv - /home/.../Lab05/01_RTL/inter.fs...
File View Source OneTrace Simulation Tools Window Help
Instance
Hierarchy Module
TESTBENCH TESTBENCH
I_PATTERN PATTERN
I_interconnect inter
Instance Declaration
<nSchema:3>TESTBENCH
File View Schematic Trace Tools Window
inter
PATTERN
*Src1:TESTBENCH.sv
<nSchema:3>TESTBENCH
*<nWave:2> /home/RAID2/COURSE/dcs/dcsta02/TA/Lab05/01_RTL/inter.fsdb
File Signal View Waveform Analog Tools Window
x 10ps
Go to:
valid_master2slave
data_3[6:0]
data_2[6:0]
data_1[6:0]
in_valid_3
in_valid_2
in_valid_1
rst_n
clk
G2
Message OneSearch
*<nWave:2> inter.fsdb
Selected:(1) Inst: I_interconnect
Command Done

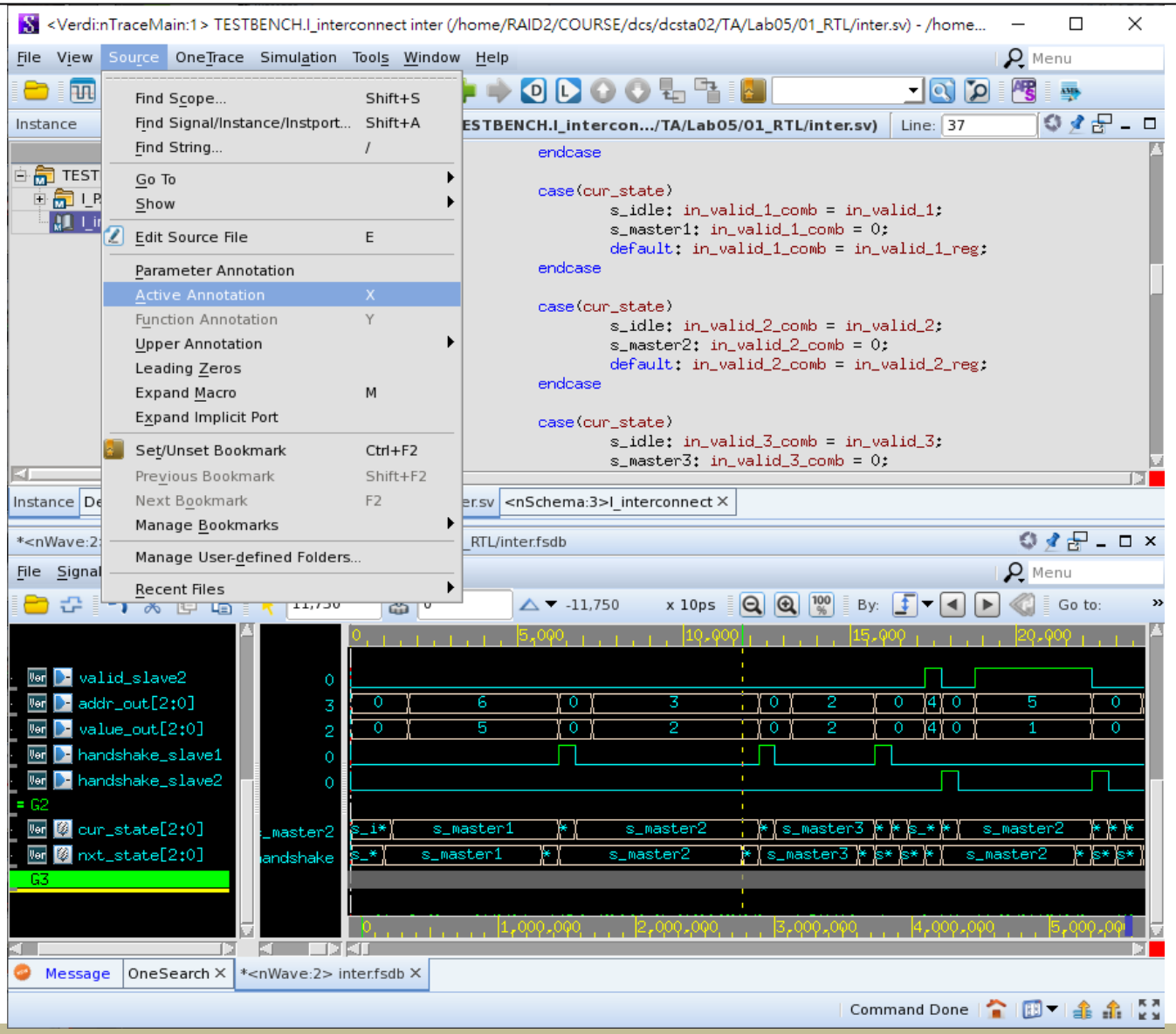
nWave



nWave

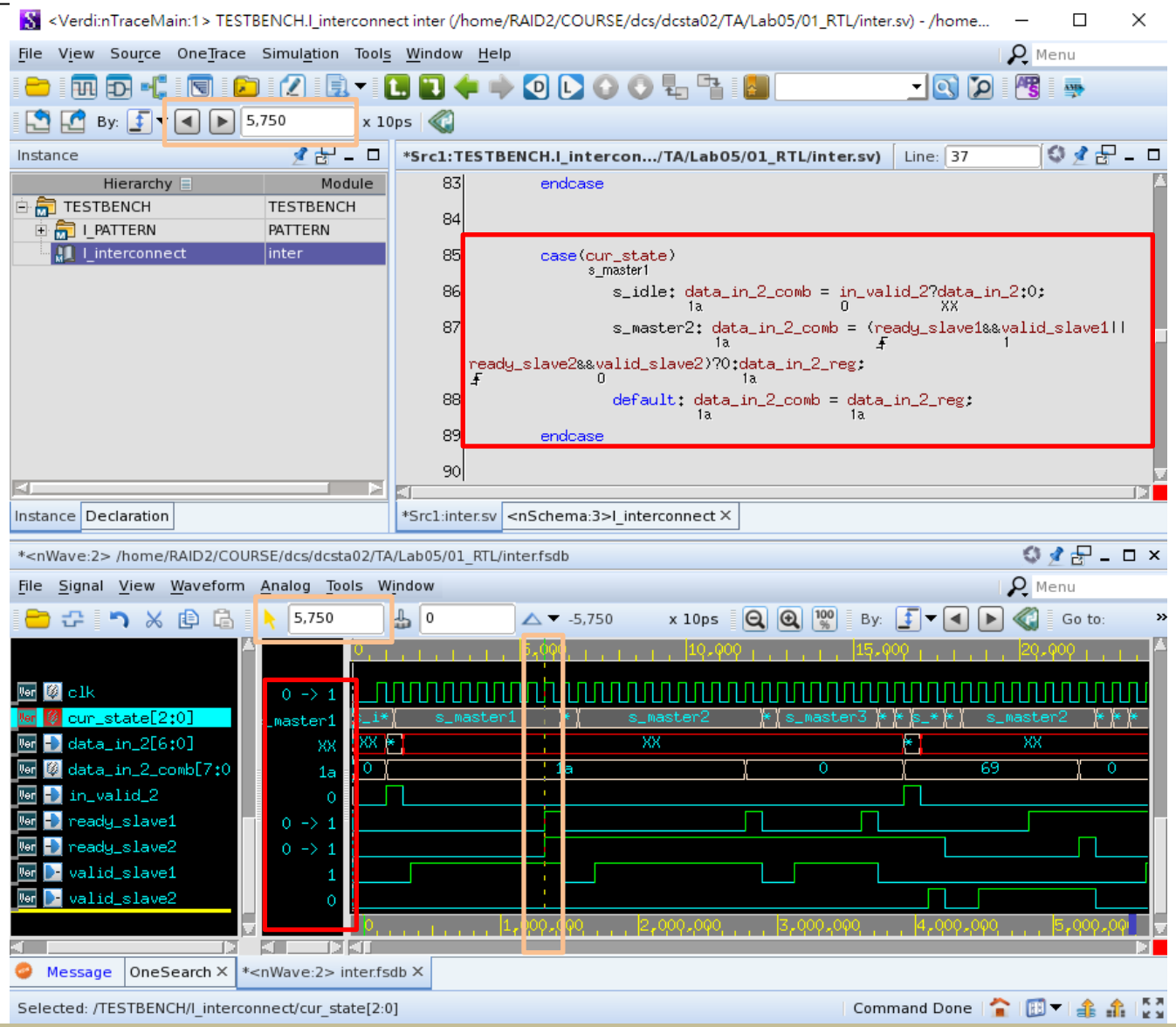

1. frame signals
2.Press middle mouse button, drag and then drop.
The state signals will be shown in state parameter names.

Verdi – Active Annotation



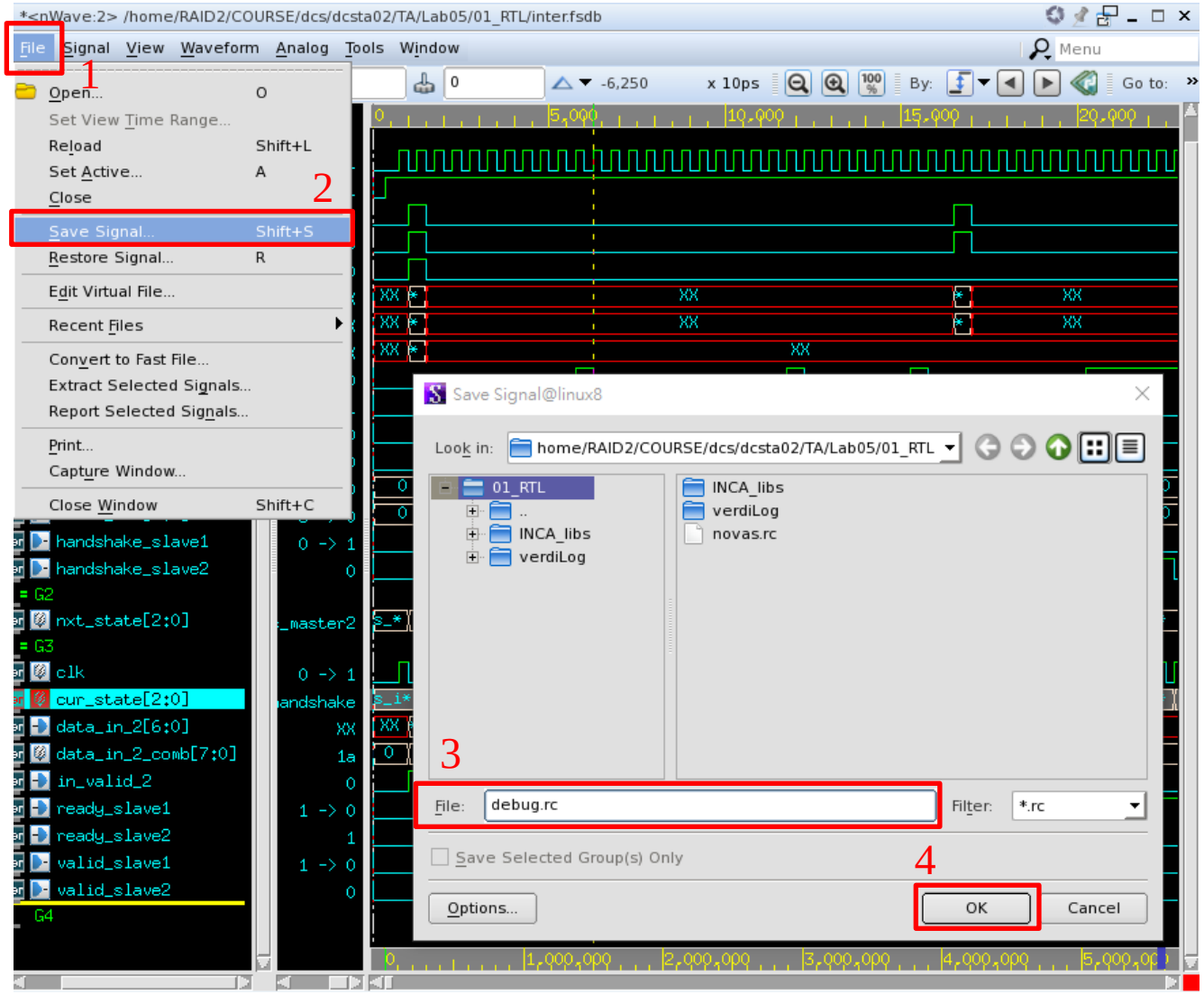
Verdi – Active Annotation

- The values stored in variables will be shown as same as the waveform where marker points to.



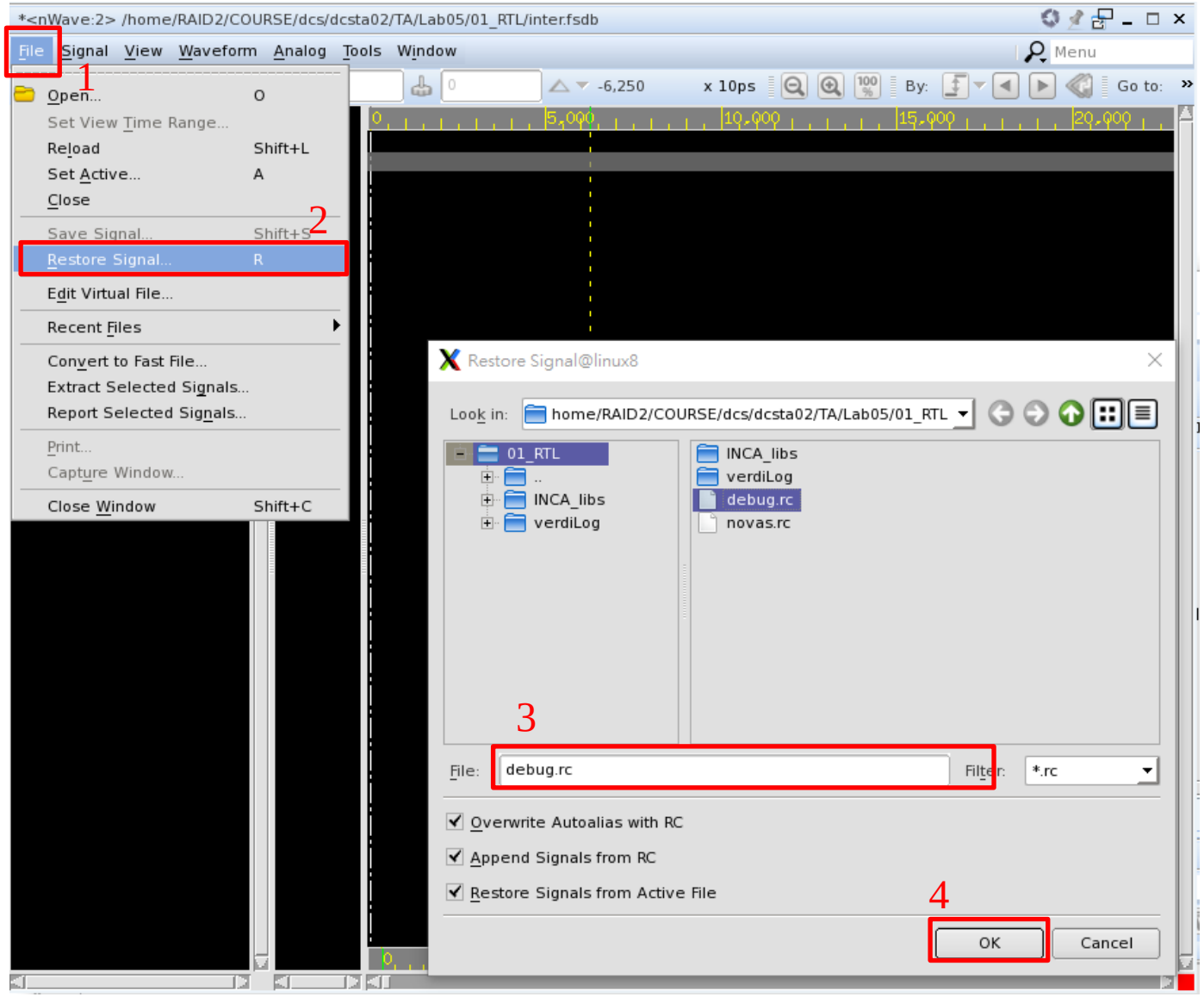
nWave - Saving waveform

- You can save the signal order for next time using nWave.
  - Naming it as “debug.rc”.

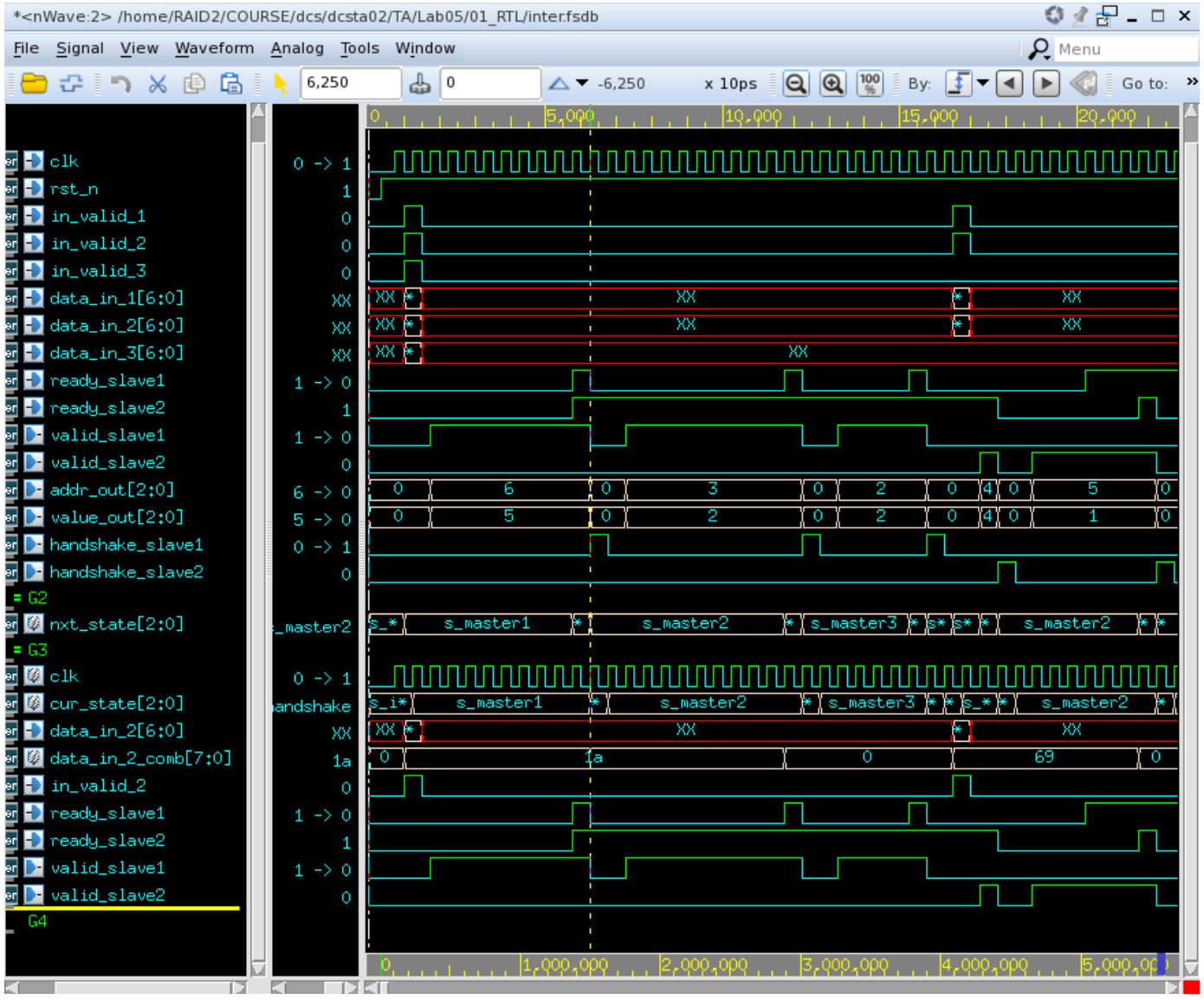


nWave - Saving waveform

- Next time you using nWave, you can simply restore the signals instead of choosing signal again and again.
  - Don't forget you have to import .fsdb first.



nWave - Saving waveform



Reference

1. "Introduction to Verdi" by Abel Hu
2. "Verdi^3 datasheet" by Synopsys
3. Verilog Simulation & Debugging Tools by NTU