

Verilog Simulation & Debugging Tools

Ricky Cheng
EECS Lab

Acknowledgement

- ▶ This slide is adapted from “Verilog Simulation & Debugging Tools”, a teaching slide of Digital Circuit Lab by Po-Chen Wu

Outline

- ▶ Environment Setup
- ▶ NC-Verilog
- ▶ nLint
- ▶ nWave
- ▶ Verdi

Environment Setup

Login to the Linux Server

- ▶ Many EDA tools are provided only for the [Linux](#) OS.
- ▶ So we have to use software like PuTTY/PieTTY/[MobaXterm](#) on our local computer to login to the linux server and use the EDA tools on it.

X Window System

- ▶ X Window System ([X11](#), [X](#), and sometimes informally [X-Windows](#)) is a windowing system for bitmap displays, common on UNIX-like (e.g., [Linux](#)) operating systems.
- ▶ Microsoft Windows is not shipped with support for X, but many third-party implementations exist, as free and open source software such as Cygwin/X, and proprietary products such as [Xming](#).

Introduction to MobaXterm (1/2)

- ▶ MobaXterm is **free software** that can be installed onto your local Windows or Mac computer which provides a **graphical user interface** and a **command line shell** for the server.
- ▶ Official Website <http://mobaxterm.mobatek.net/>

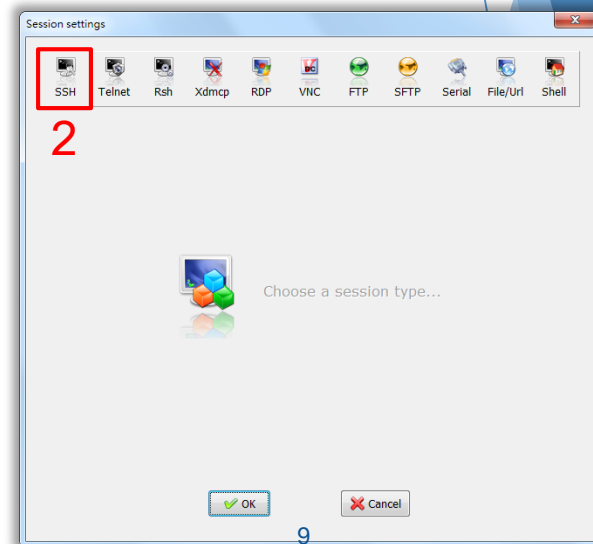
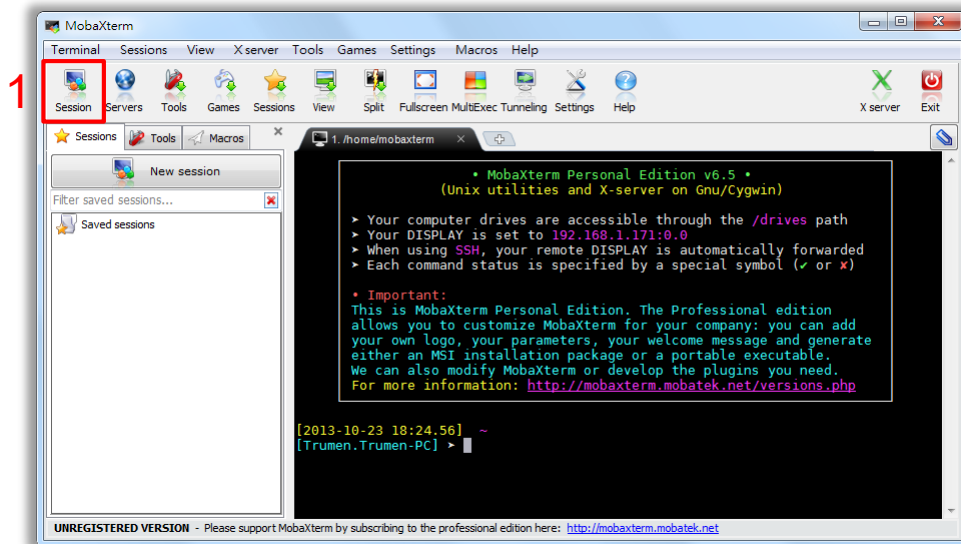


Introduction to MobaXterm (2/2)

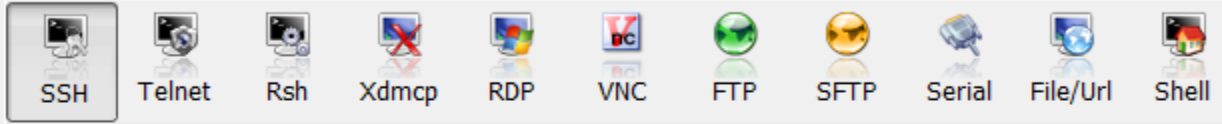
- ▶ MobaXterm provides useful features for developers:
 - ▶ **Multitab terminal** with embedded Unix commands (ls, cd, ...).
 - ▶ Embedded **X11 server** for easily exporting your Linux display.
 - ▶ **Passwords management** for SSH, SFTP, etc (on demand password saving).
 - ▶ ...

Session Settings

- ▶ Click the Session button and specify which session you want. Usually this will be **SSH**. For that click SSH.



Session settings



Basic SSH settings 1

Remote host * cad27.ee.ntu.edu.tw Specify username bXXXXX 2 Port 22

Advanced SSH settings Terminal settings Bookmark settings

X11-Forwarding Compression Remote environment Interactive shell

Execute command Do not exit after command ends

Display SFTP browser Automatically follow current SSH folder path (experimental)

Use private key Extra option

Enable Google 2-step authentication

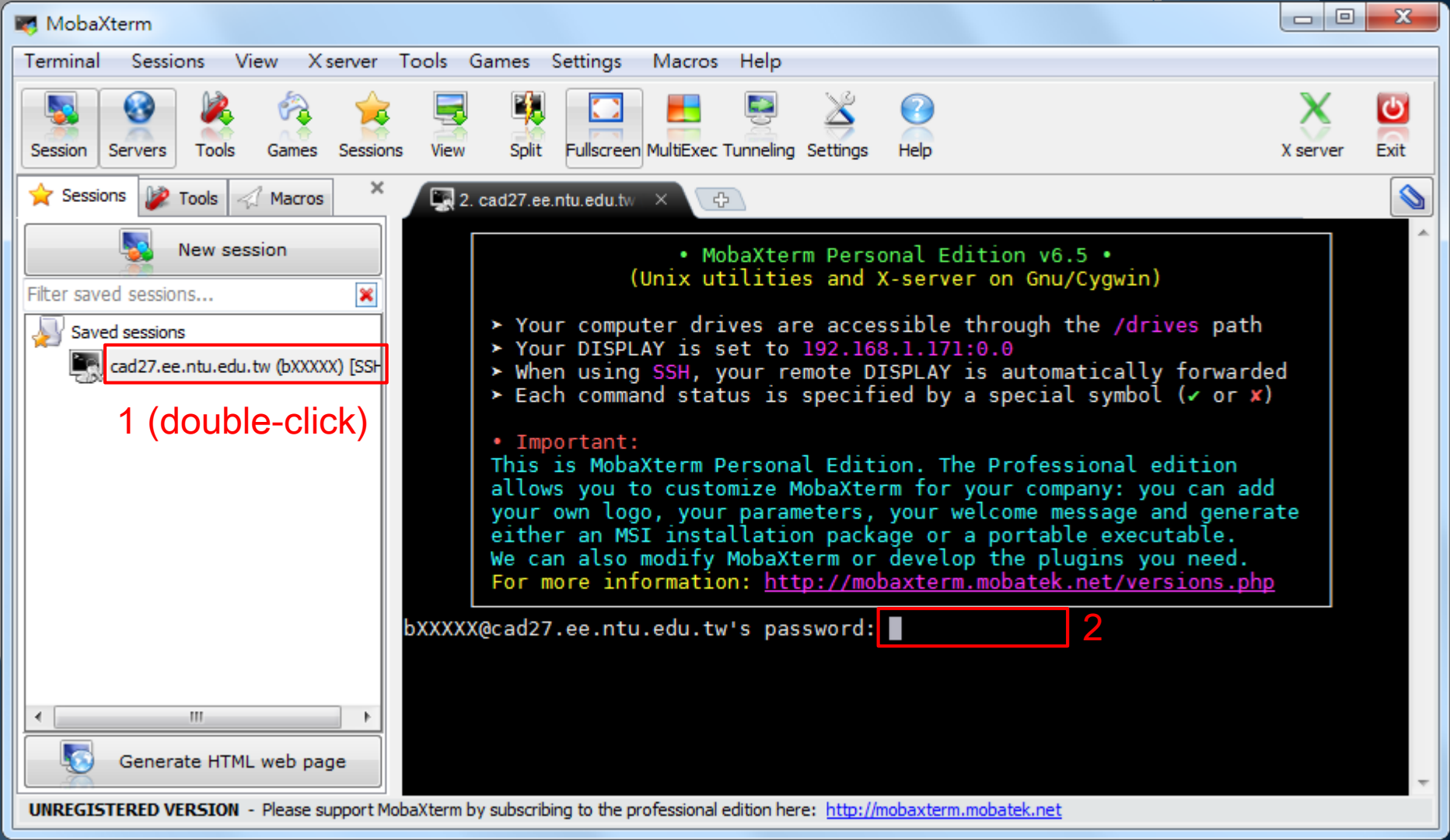
Connect through SSH gateway

Gateway SSH server Port 22 User

Use private key

3

✔ OK ✘ Cancel



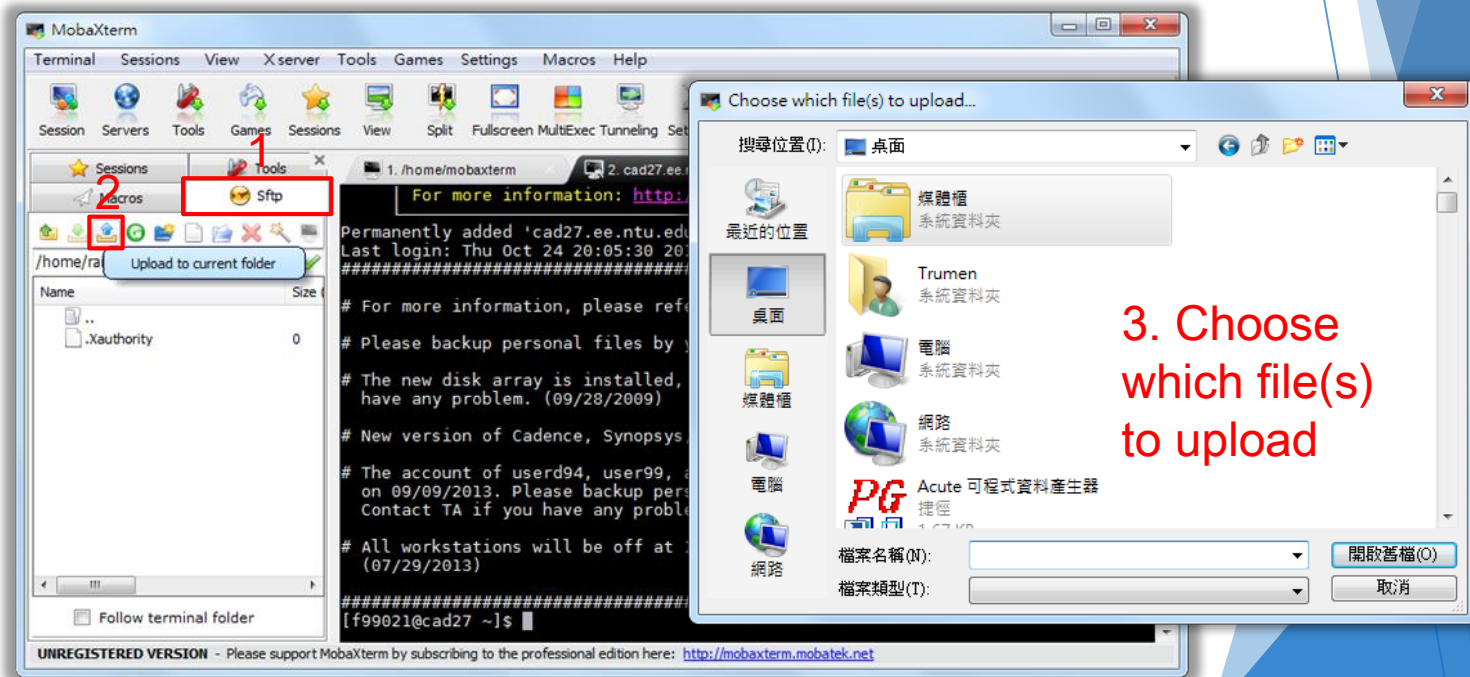
Command Line Shell

- ▶ We can also use the command line shell to login to the server.
 - ▶ `ssh bXXXXXX@cad27.ee.ntu.edu.tw [-p YYYYYY]`
 - ▶ `bXXXXXX`: your user name
 - ▶ `YYYYYY`: port number
 - ▶ here `-p 22` is redundant because 22 is the default port number.

```
[Trumen.Trumen-PC] > ssh bXXXXXX@cad27.ee.ntu.edu.tw  
bXXXXXX@cad27.ee.ntu.edu.tw's password: █
```

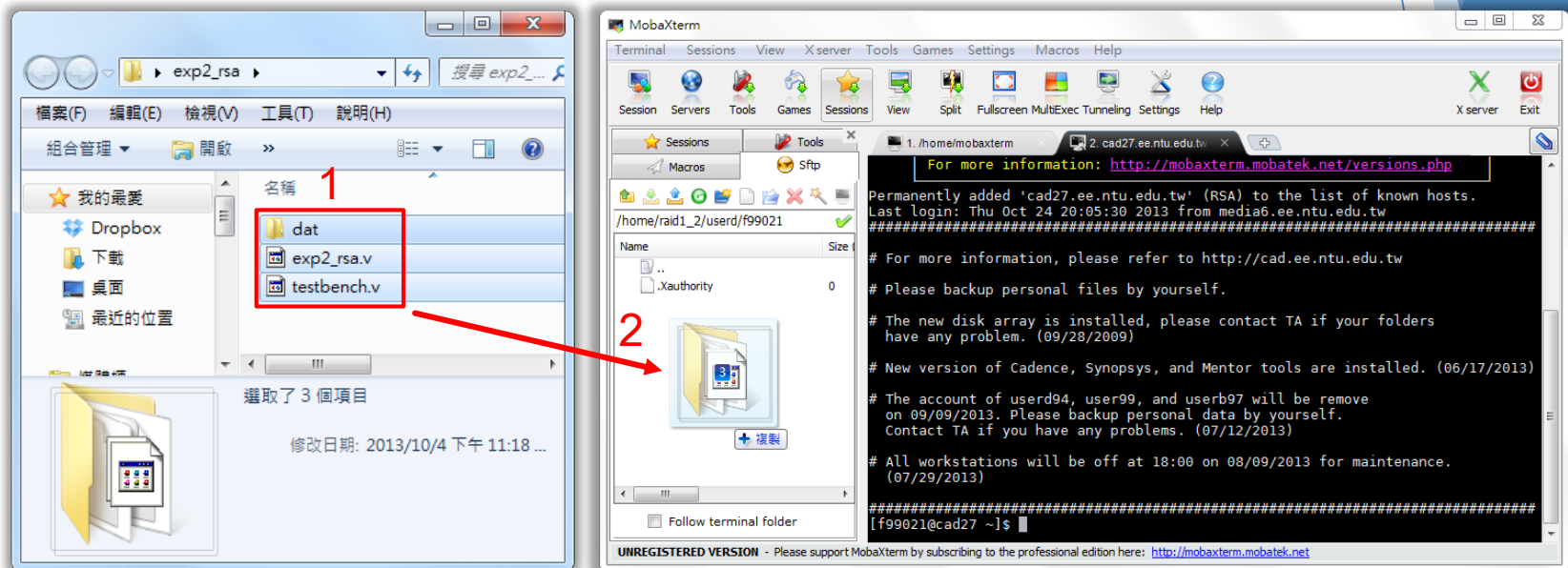
Upload Files (1/2)

- ▶ Uploading files from your local PC to the server.



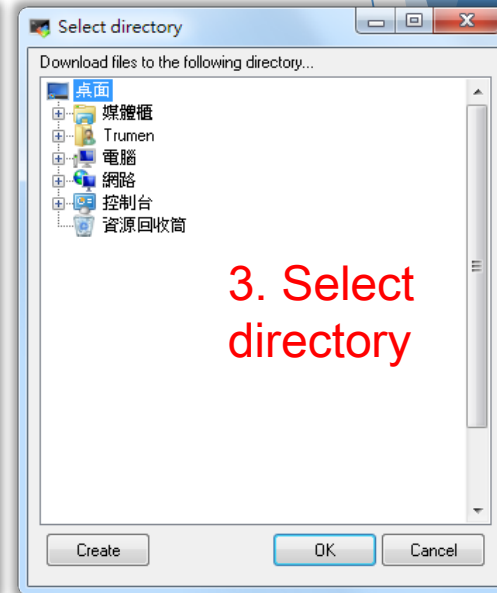
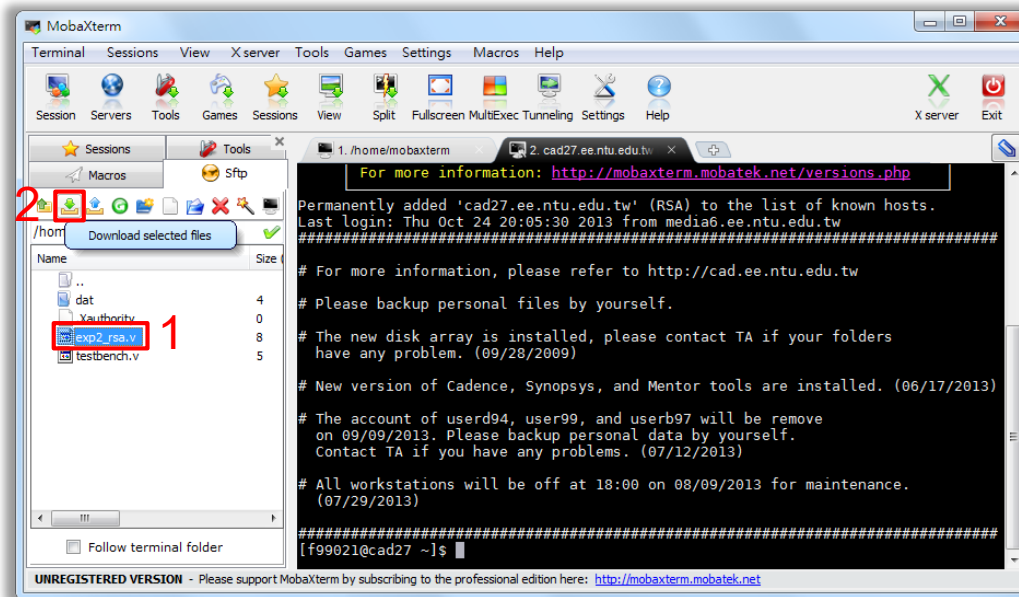
Upload Files (2/2)

- ▶ Moving and copying files by using the drag-and-drop.



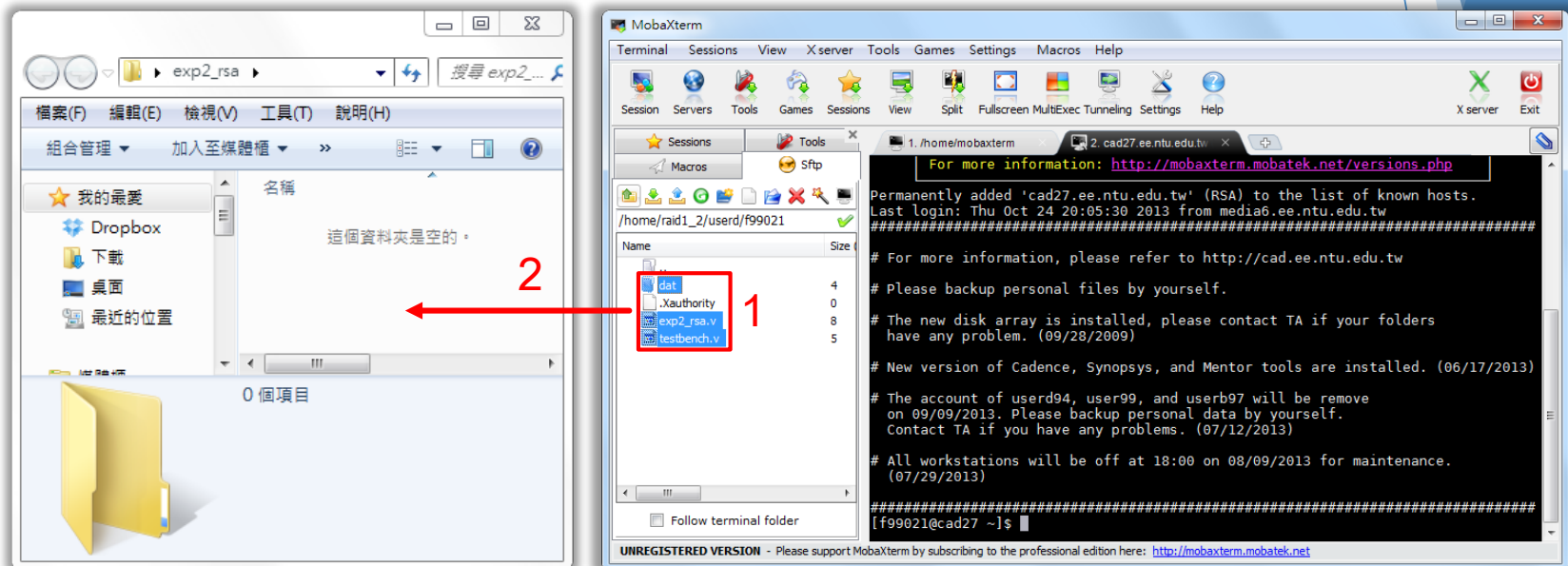
Download Files (1/2)

- ▶ Downloading files from the server to local PC.



Download Files (2/2)

- ▶ Moving and copying files by using the drag-and-drop.



NC-Verilog

Introduction to NC-Verilog

- ▶ The Cadence® NC-Verilog® simulator is a Verilog digital logic simulator.
- ▶ We can use NC-Verilog to
 - ▶ Compiles the Verilog source files.
 - ▶ Elaborates the design and generates a simulation snapshot.
 - ▶ Simulates the snapshot.

Before Using NC-Verilog

- ▶ Source the environment settings of CAD tools.

```
source /usr/cad/cadence/CIC/incisiv.cshrc
```

- ▶ If you try entering the command "`ncverilog`" but it turns out "`command not found`," it means there's something wrong with the "`*.cshrc`" file or the software license is out of date.

Running Verilog (1/2)

- ▶ Run the Verilog simulation:

```
ncverilog testbench.v exp2.rsa.v +access+r
```

- ▶ Another choice of running Verilog simulation:

```
ncverilog -f exp2_rsa.f +access+r
```

In exp2_rsa.f:

```
testbench.v  
exp2_rsa.v  
~  
~
```

Running Verilog (2/2)

- ▶ "+access+r" is added to enable waveform file dumping.

In testbench.v, line 69~72

```
initial begin
    $fsdbDumpfile("exp2_rsa.fsdb");
    $fsdbDumpvars;
end
```

 or

```
initial begin
    $dumpfile("exp2_rsa.vcd");
    $dumpvars;
end
```

- ▶ *.fsdb has smaller file size than *.vcd. But \$fsdbDumpfile cannot work without sourcing verdi.cshrc.

Simulation Results

- ▶ Check the simulation result to see if the Verilog design is finished correctly.

```
ncverilog: 10.20-s114: (c) Copyright 1995-2012 Cadence Design Systems, Inc.
Loading snapshot worklib.testbench.v ..... Done
*Novas* Loading libsscore_ius102.so
ncsim> source /usr/cad/cadence/INCISIV/cur/tools/inca/files/ncsimrc
ncsim> run
Novas FSDB Dumper for IUS, Release 2012.04, Linux, 04/10/2012
Copyright (C) 1996 - 2012 by SpringSoft, Inc.
*Novas* : Create FSDB file 'exp2_rsa.fsdb'
*Novas* : Begin traversing the scopes, layer (0).
*Novas* : End of traversing.
-----

Congratulations! All data have been generated successfully!

-----PASS-----

Simulation complete via $finish(1) at time 100046010 NS + 0
./testbench.v:177      $finish;
ncsim> exit
```

nLint

Introduction to nLint

- ▶ **nLint** is a comprehensive HDL design rule checker fully integrated with the Debussy debugging system (Developed by SpringSoft).
- ▶ We can use nLint to check the **coding style** of our design and if it is **synthesizable**.

Before Using nLint

- ▶ Source the environment settings of CAD tools.

```
source /usr/cad/synopsys/CIC/verdi.cshrc
```

- ▶ To avoid the warning [*WARN* Failed to check out license.](#) occurs when starting nLint, please type the following command:

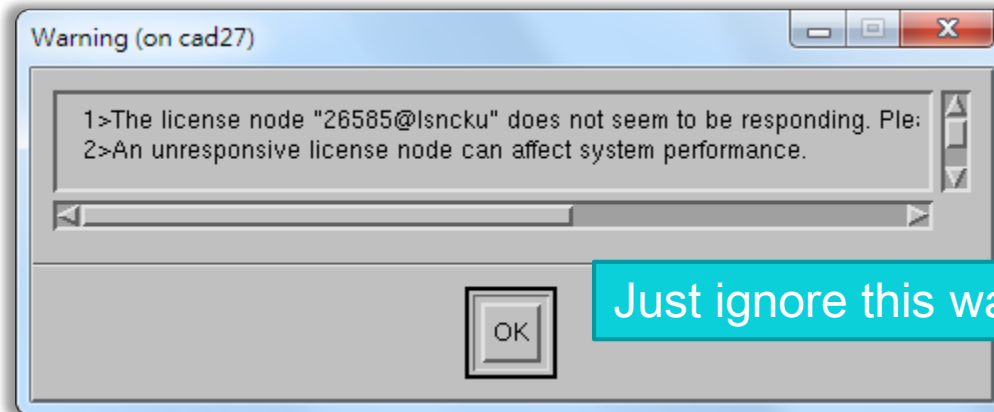
```
setenv LM_LICENSE_FILE '26585@lsntu:26585@lsncku'
```

Start nLint

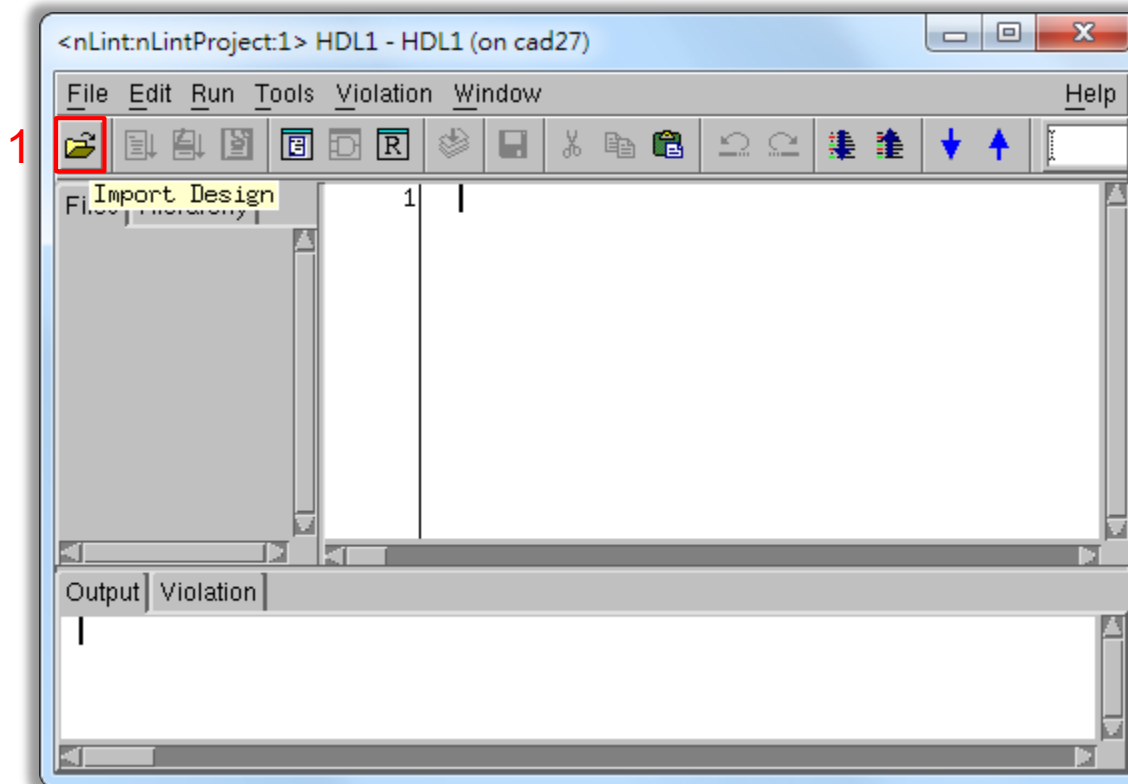
- ▶ Type the following command:

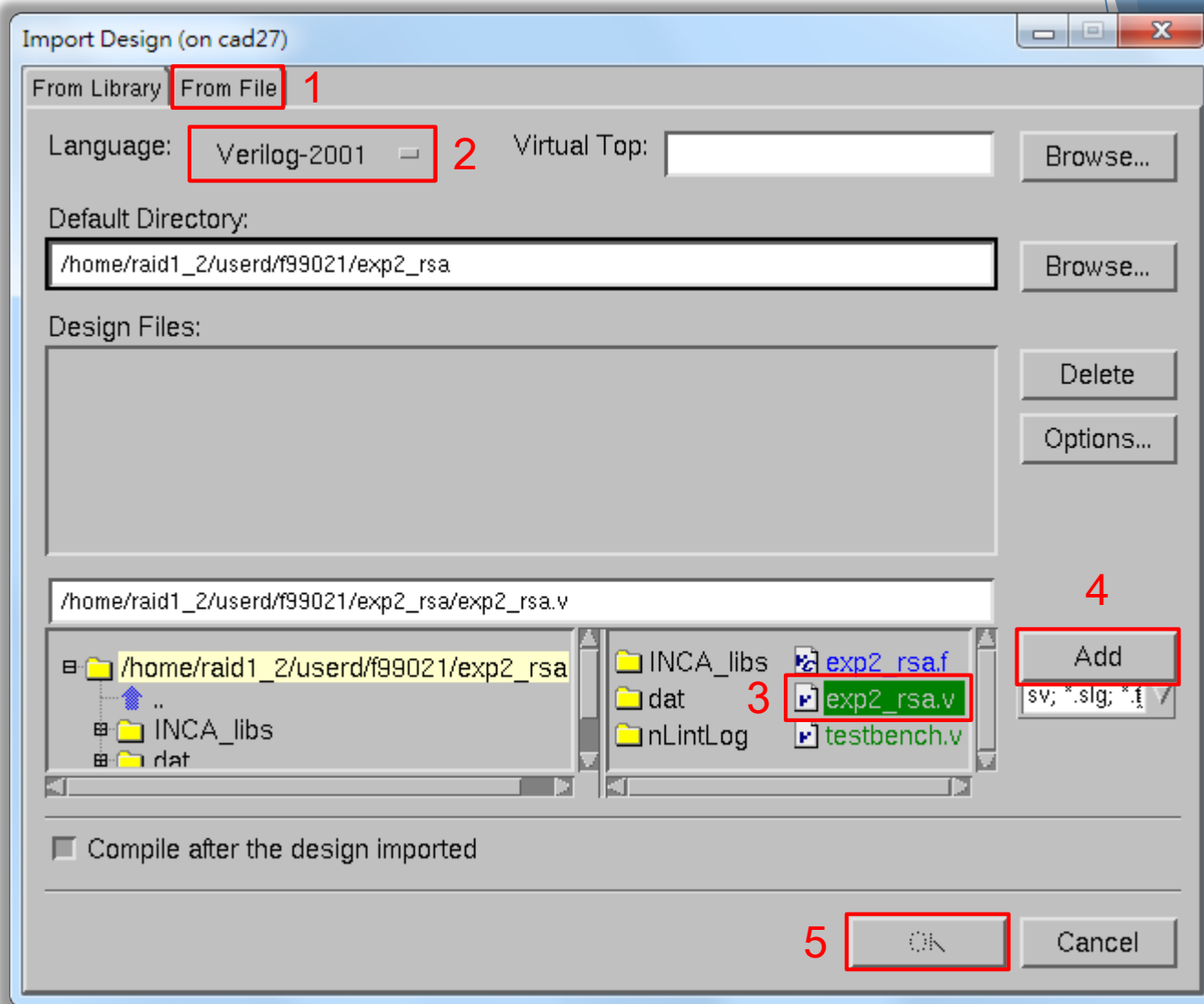
```
nLint -gui &
```

- ▶ The token "&" enable you to use the terminal while nLint is running in the background.

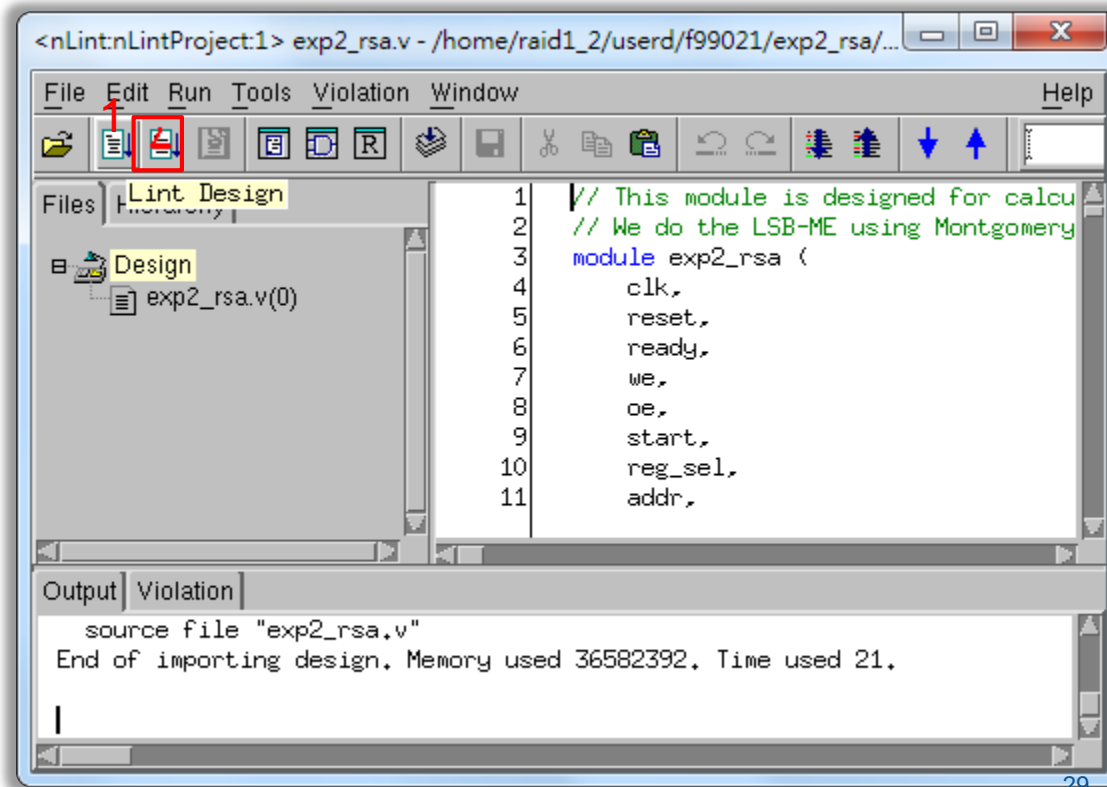


Specify the Design File





Start Checking



nLint:ReportViewer:2> nLintDB* - /home/raid1_2/userd/f99021/exp2_rsa/nLintDB (on cad27)

File View Domain Tools Window Help

Filter...

- [-] Clock Domain Analysis
- [-] Total - 8 Warning(s), 4 Error(s)
- [-] [x] DFT - 1 Warning(s)
- [-] [x] Design Style - 3 Warning(s)
- [-] [x] Language Construct - 3 Warning(s), 4 Error(s)
 - [-] [x] 22209 - Insufficient Index Variable - Language Construct - 1 Warning(s)
 - [-] [x] 22265 - Operand Bit Size Mismatch in Addition or Subtraction - Language Construct - 4 Error(s)
 - [x] exp2_rsa.v(226) : Error : bit width of left-hand-side operand "T2"(257) does not match the right-h
 - [x] exp2_rsa.v(260) : Error : bit width of left-hand-side operand "{B, 1'b0}"(257) does not match the r
 - [x] exp2_rsa.v(266) : Error : bit width of left-hand-side operand "V_i"(257) does not match the right-l
 - [x] exp2_rsa.v(282) : Error : bit width of left-hand-side operand "V_1"(259) does not match the right-
 - [-] [x] 23007 - Case Statement Not Fully Specified - Language Construct - 2 Warning(s)
- [-] [x] Clock - 1 Warning(s)

Not all the warnings or errors are valuable.

nWave

Introduction to nWave

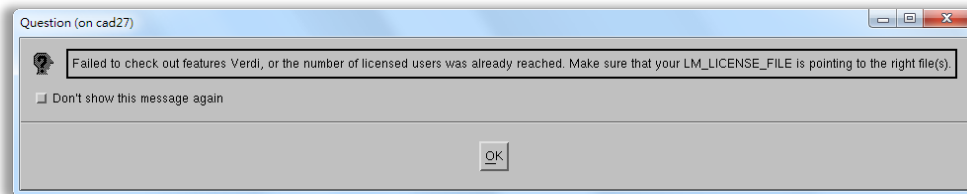
- ▶ **nWave** is one of the best waveform (*.vcd or *.fsdb) viewer.
- ▶ We can debug easily by checking the waveform file dumped during simulation.

Before Using nWave

- ▶ Source the environment settings of CAD tools.

```
source /usr/cad/synopsys/CIC/verdi.cshrc
```

- ▶ To avoid the Verdi warning window occurs,



- ▶ please type the following command:

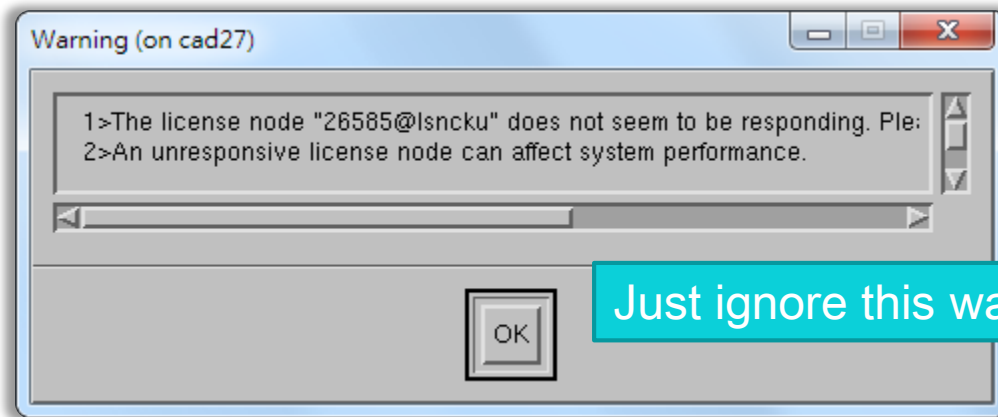
```
setenv LM_LICENSE_FILE '26585@lsntu:26585@lsncku'
```

Start nWave

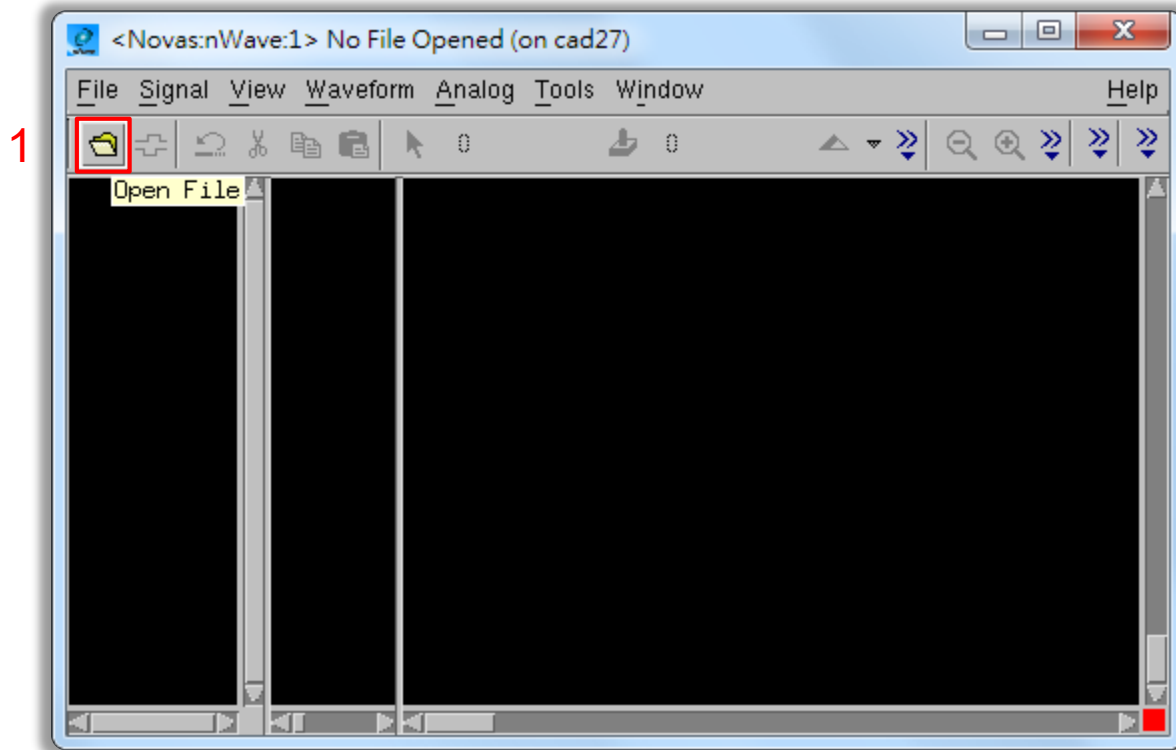
- ▶ Type the following command:

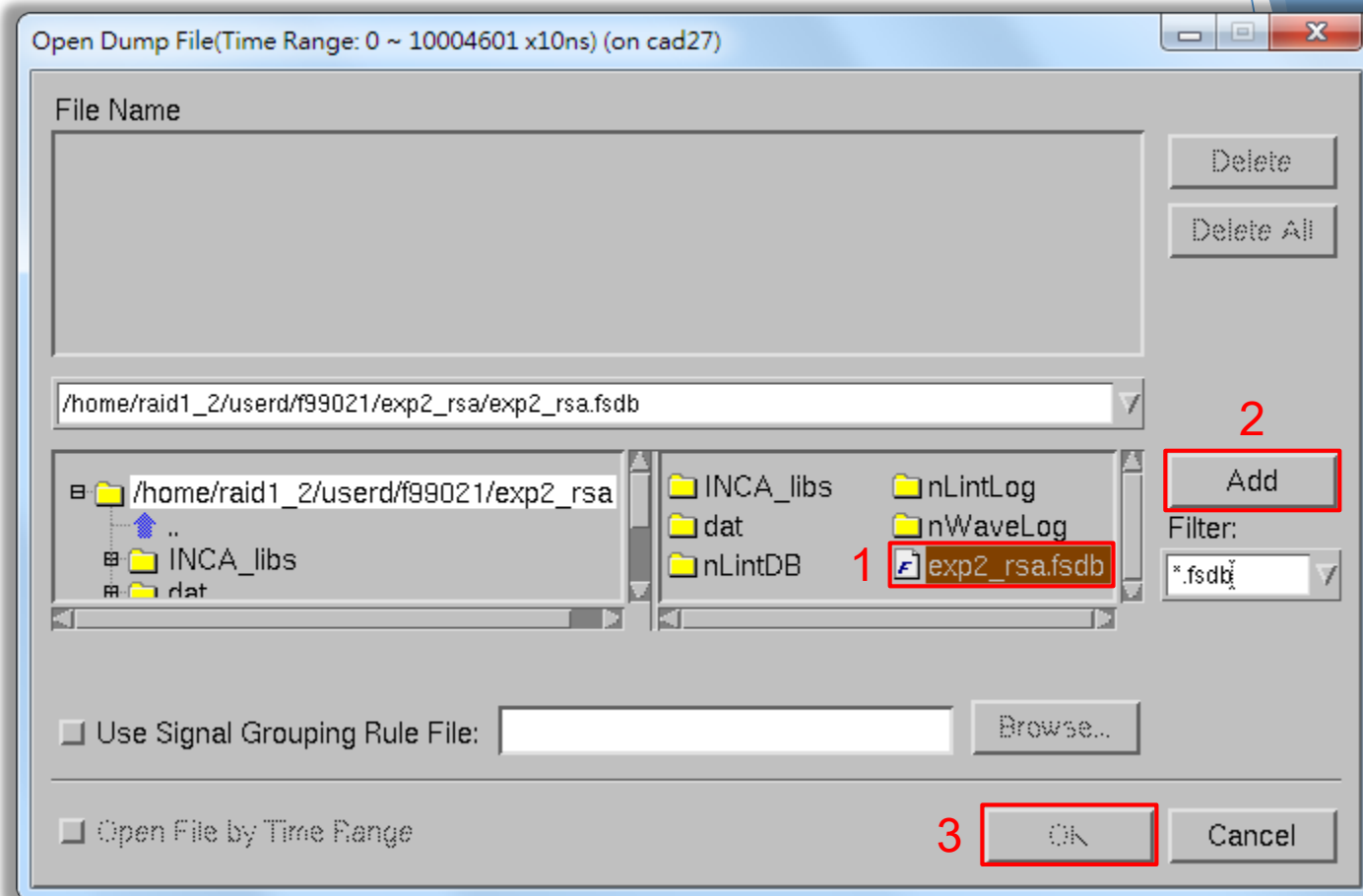
nWave &

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

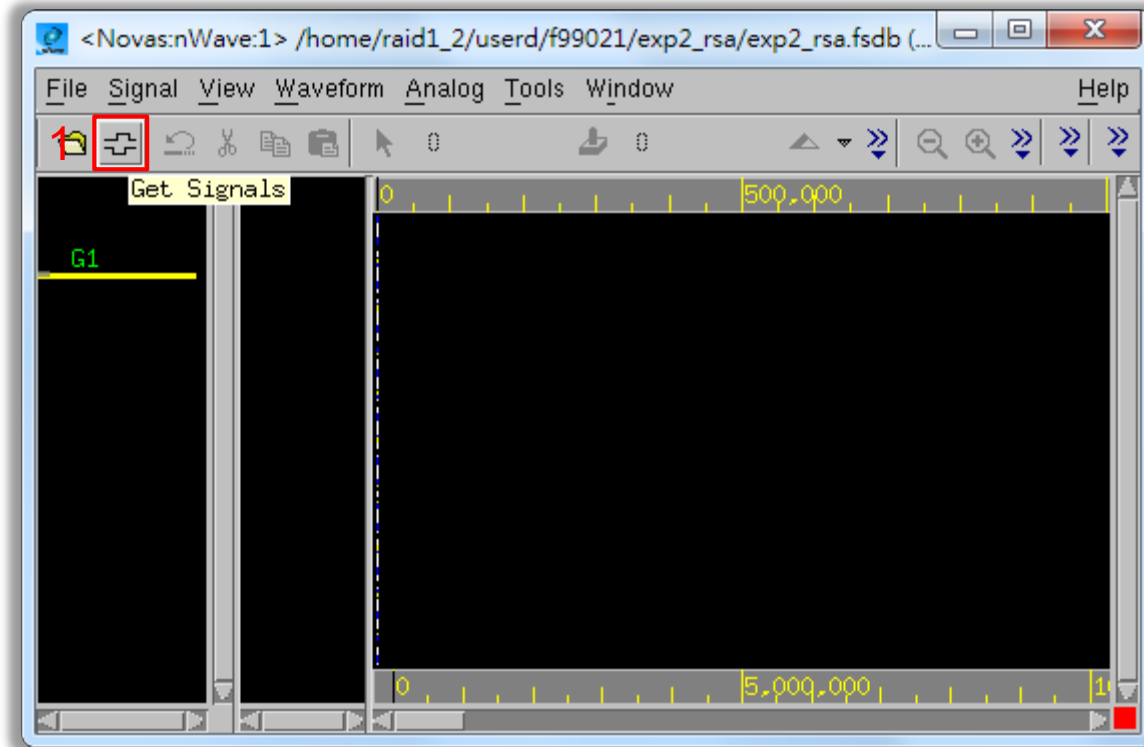


Open the FSDB File





Choose Signals



Get Signals (on cad27)

Scope: /testbench/top

Find Signal: *

testbench(testbench)
↑ top(exp2_rsa)

..

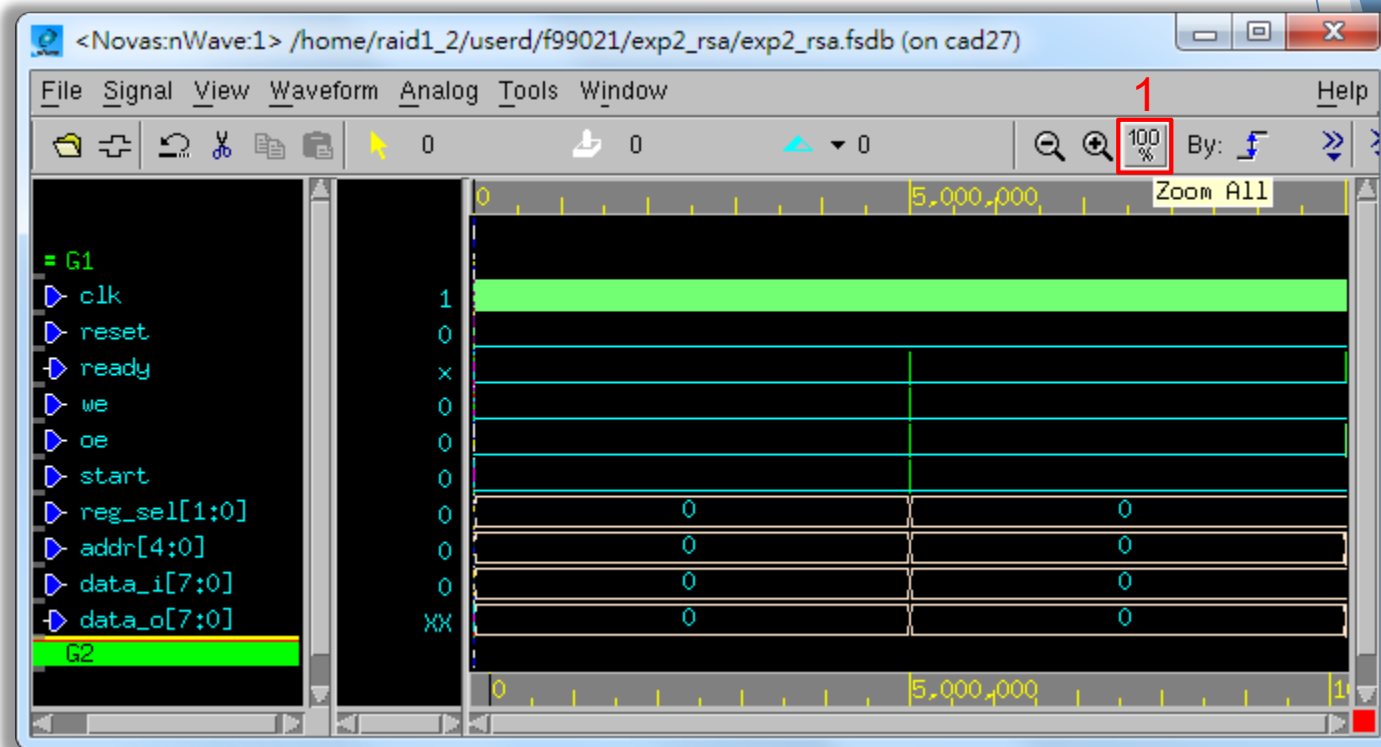
G1

V_i[256:0]	oe
a1[255:0]	oe_o
a2[255:0]	ready
a3[255:0]	ready_o
addr[4:0]	reg_sel[1:0]
addr_o[4:0]	reg_sel_o[1:0]
clk	reset
clk_o	reset_o
counter[7:0]	start
counter_MA[8:0]	start_o
data_i[7:0]	state[2:0]
data_i_o[7:0]	state_MA
data_o[7:0]	we

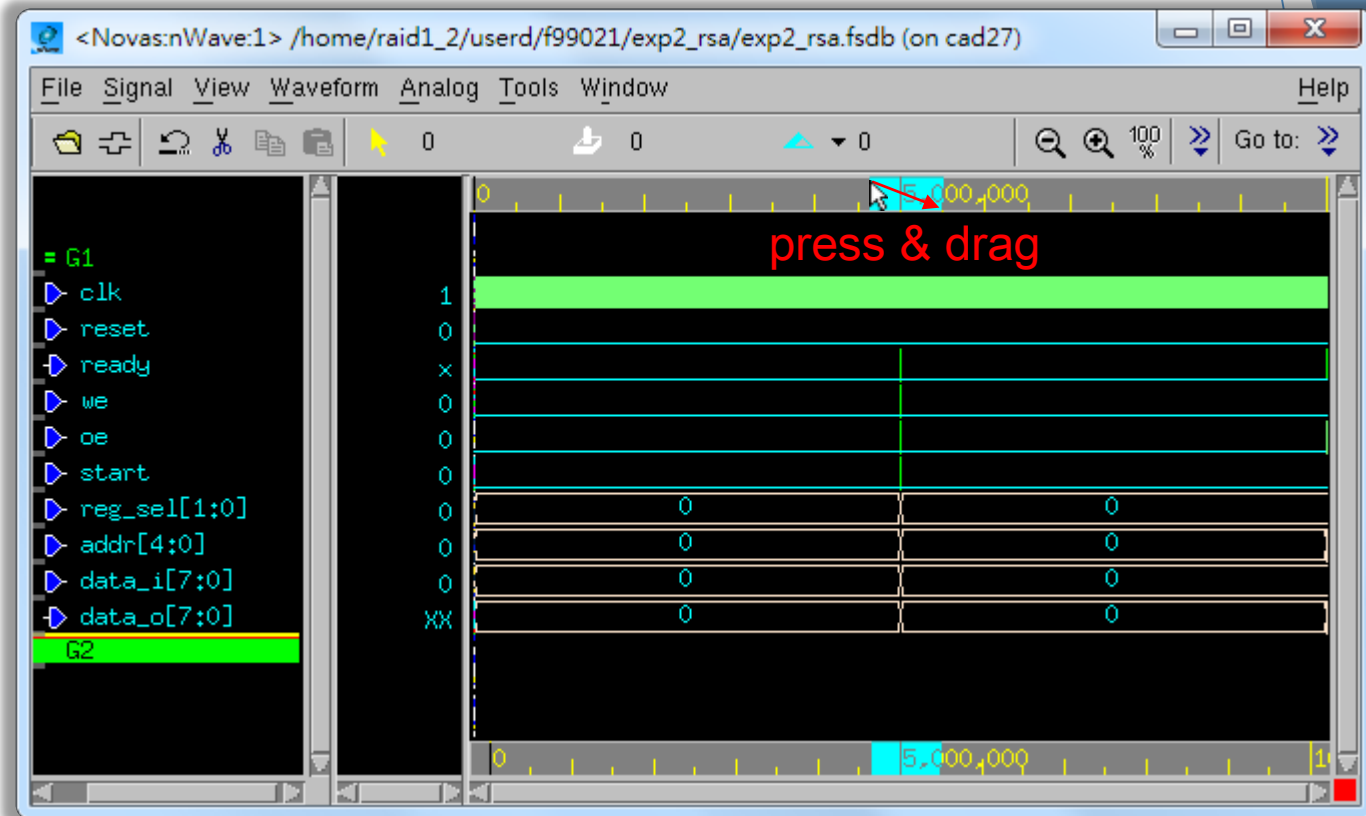
Choose signals we are interested in.

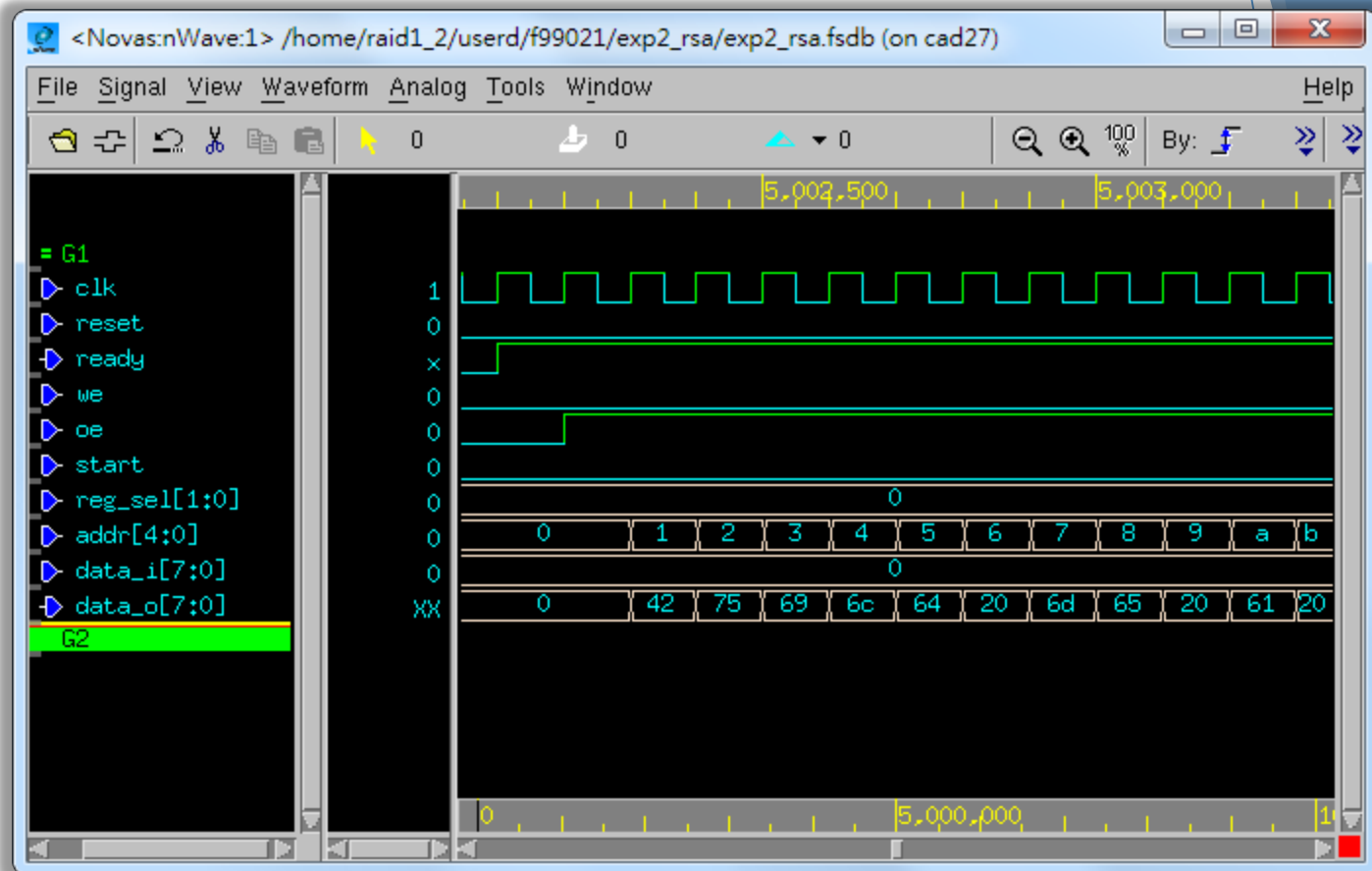
Options... ALL [play] [stop] [NET] [busName] Apply OK Cancel

Browse the Whole Waveform

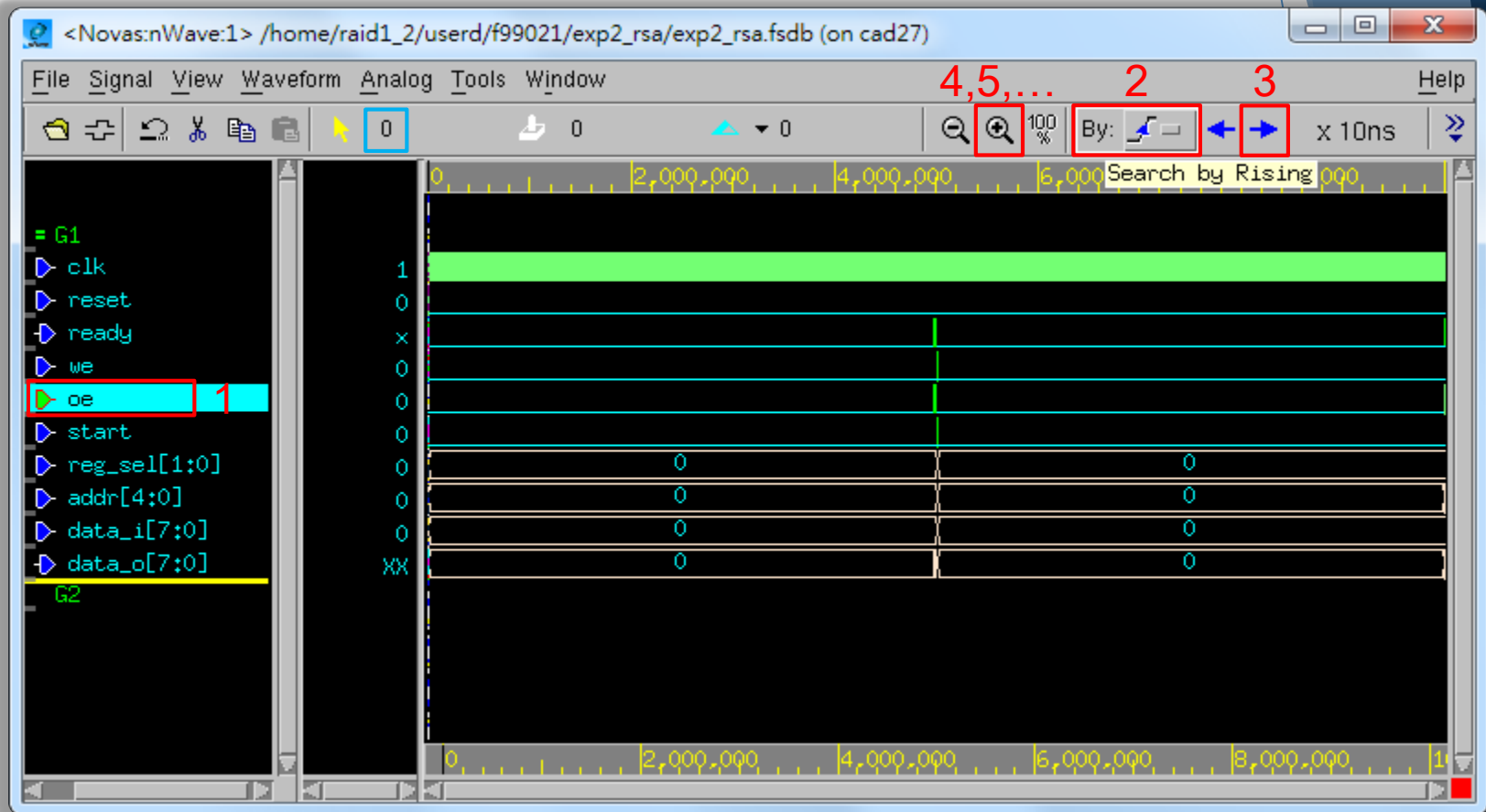


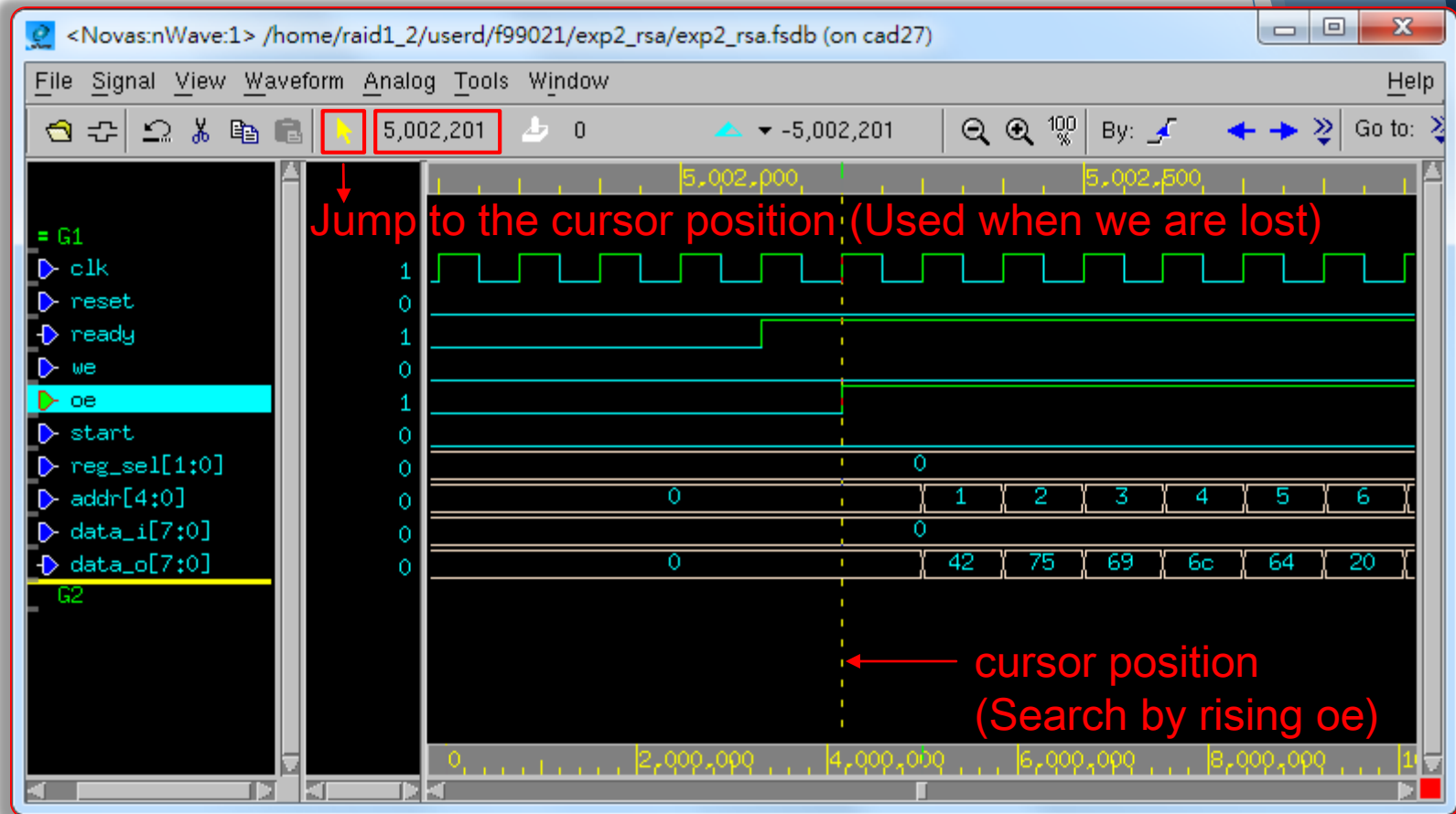
Browse the Specified Interval



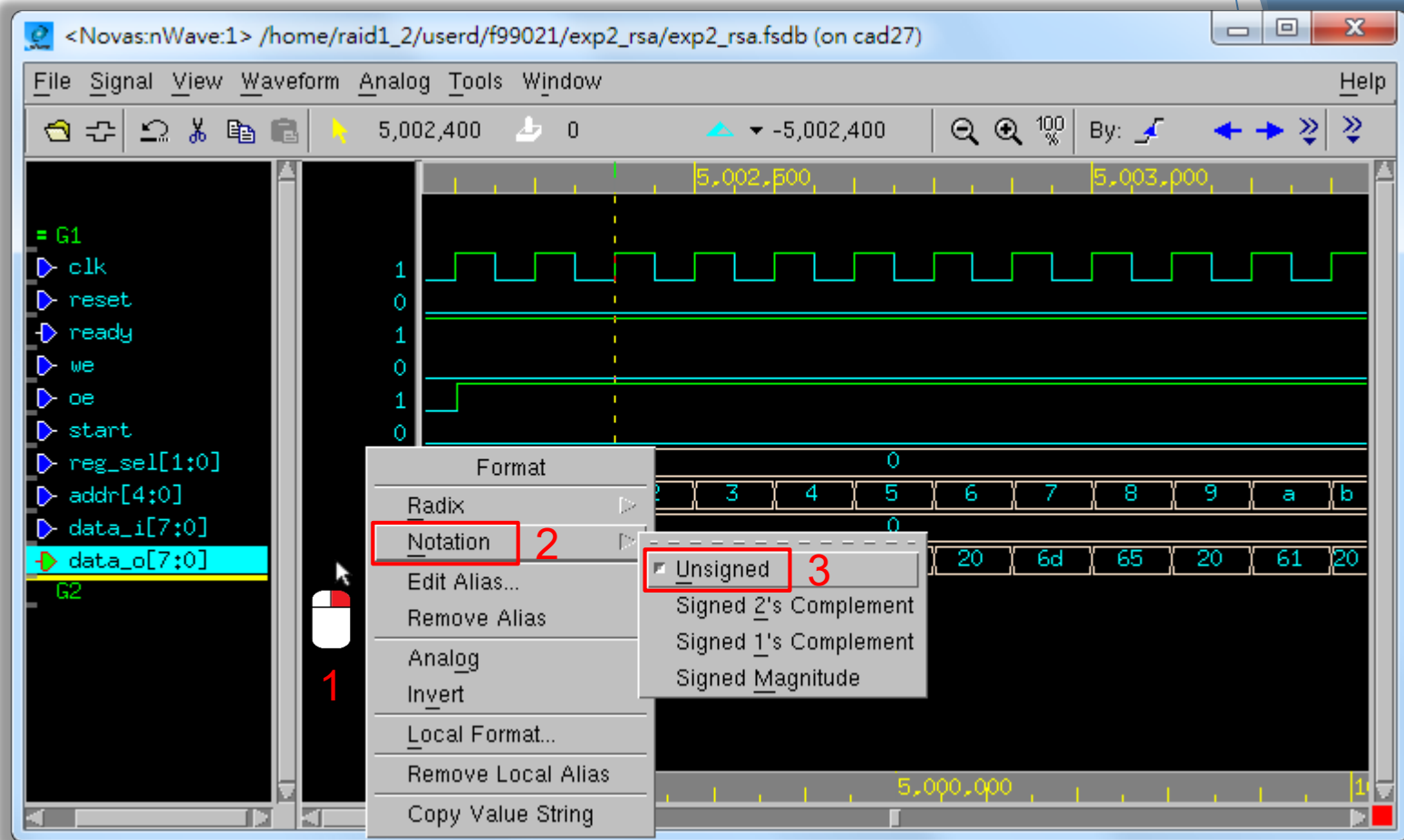


Search for Specified Signal

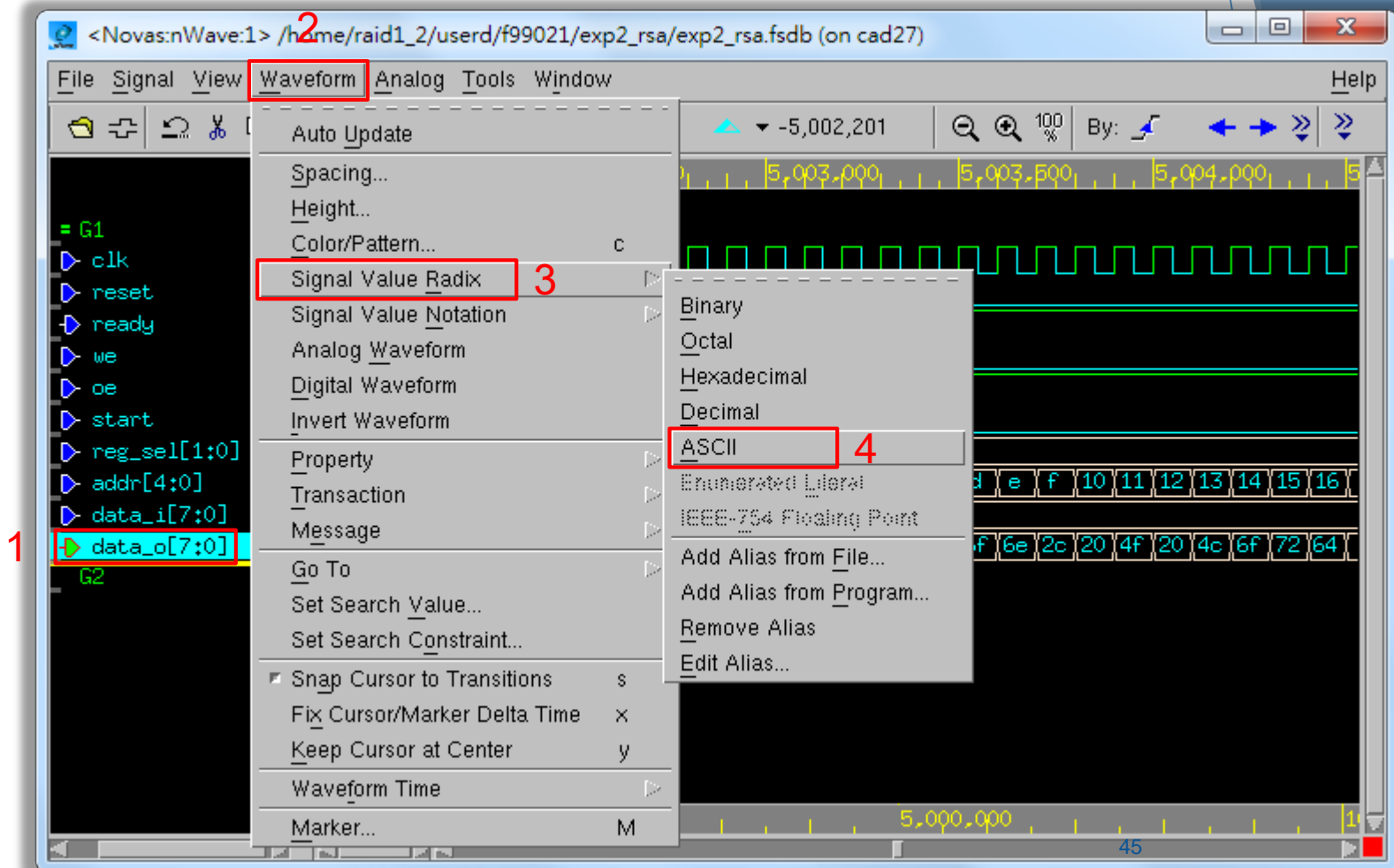


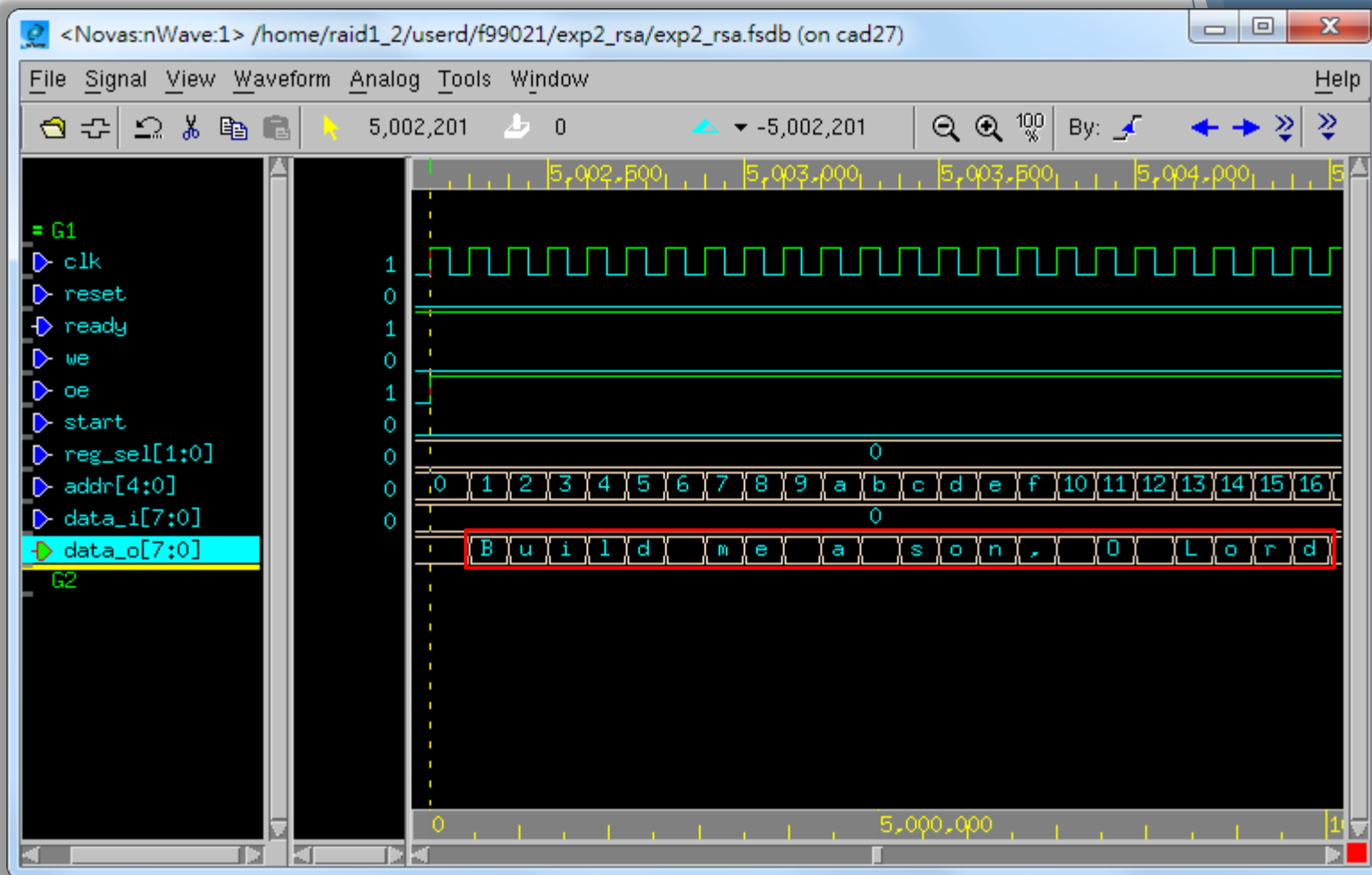


Change Sign Representation

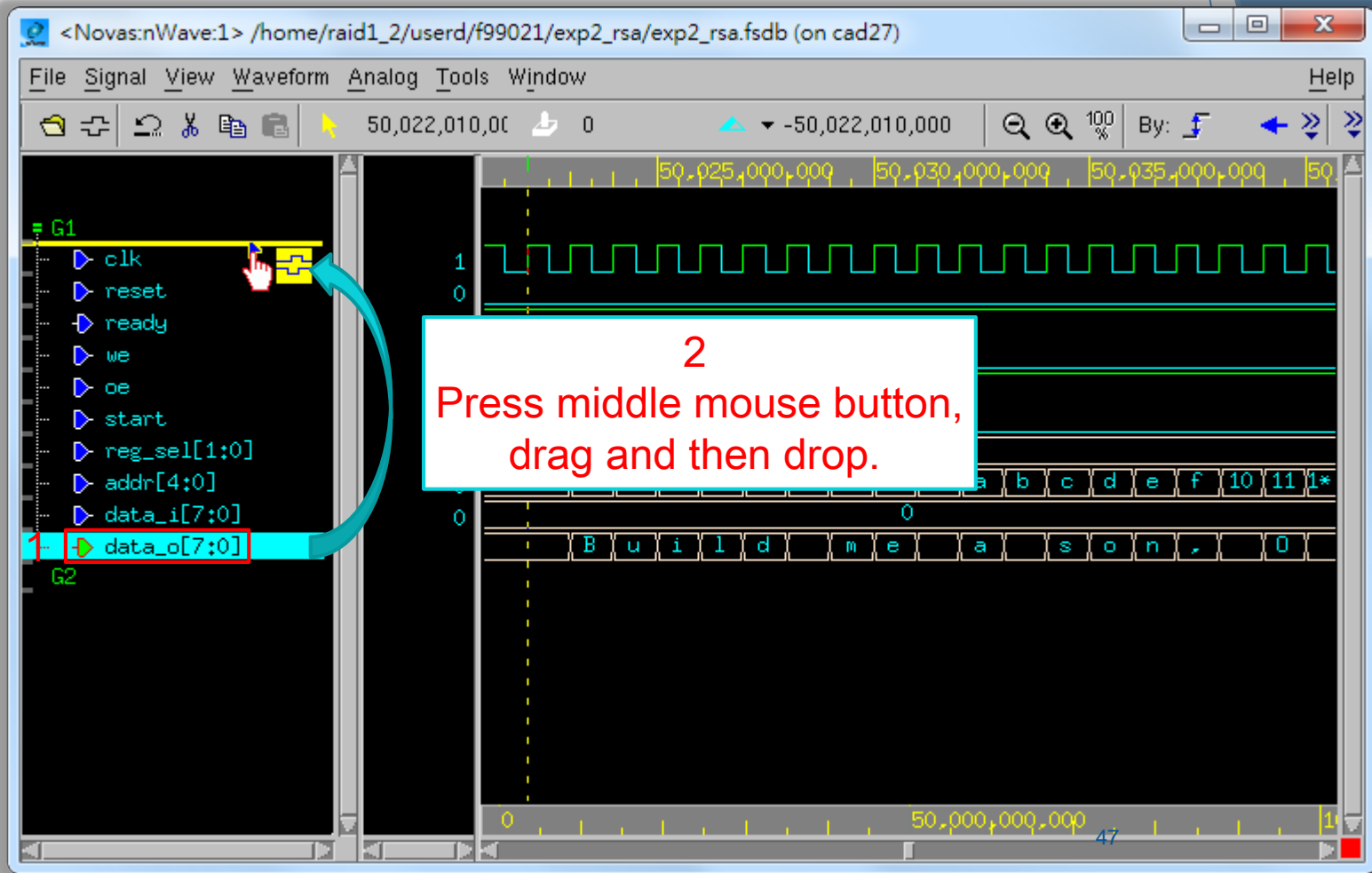


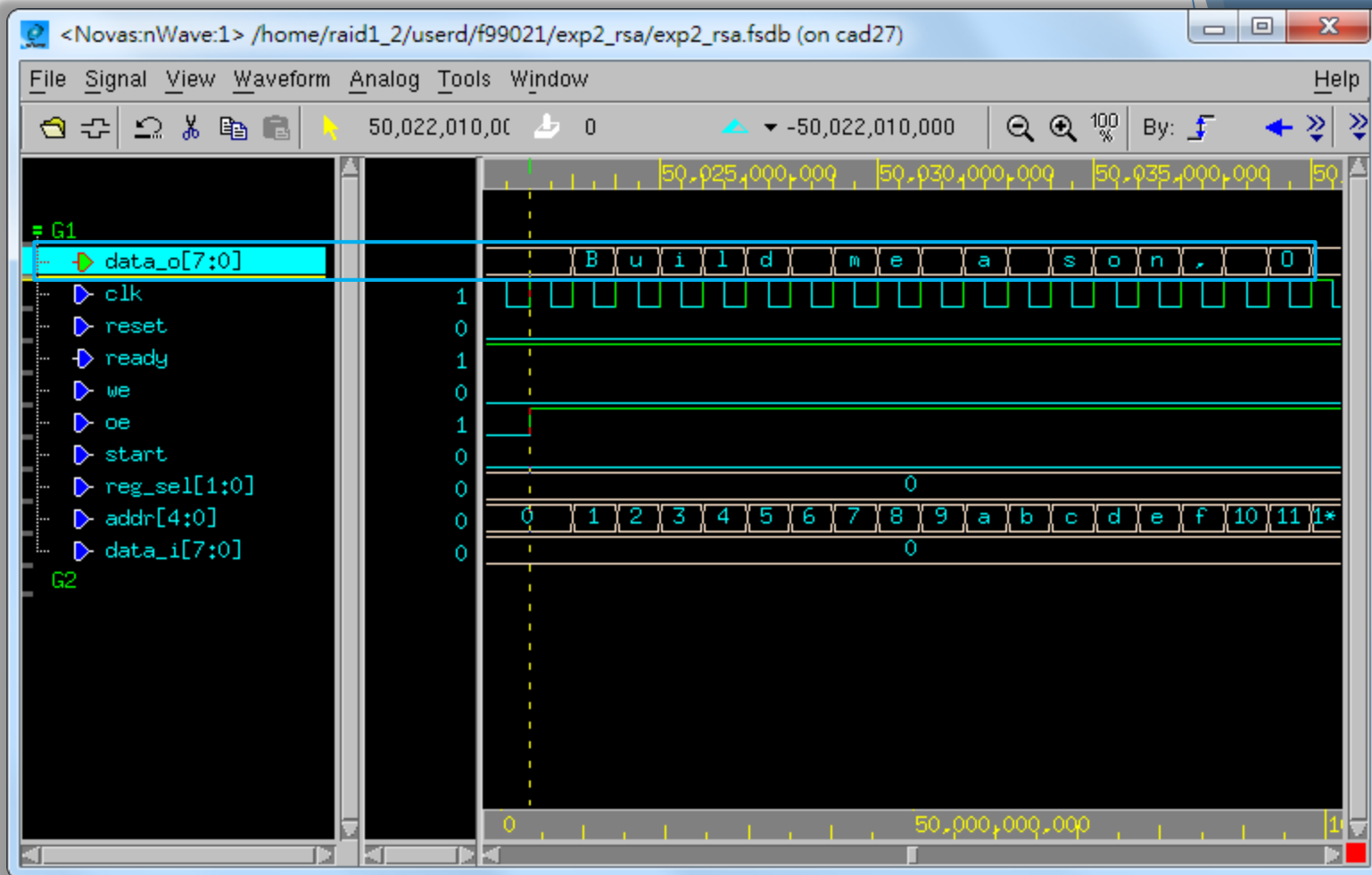
Change Radix Representation



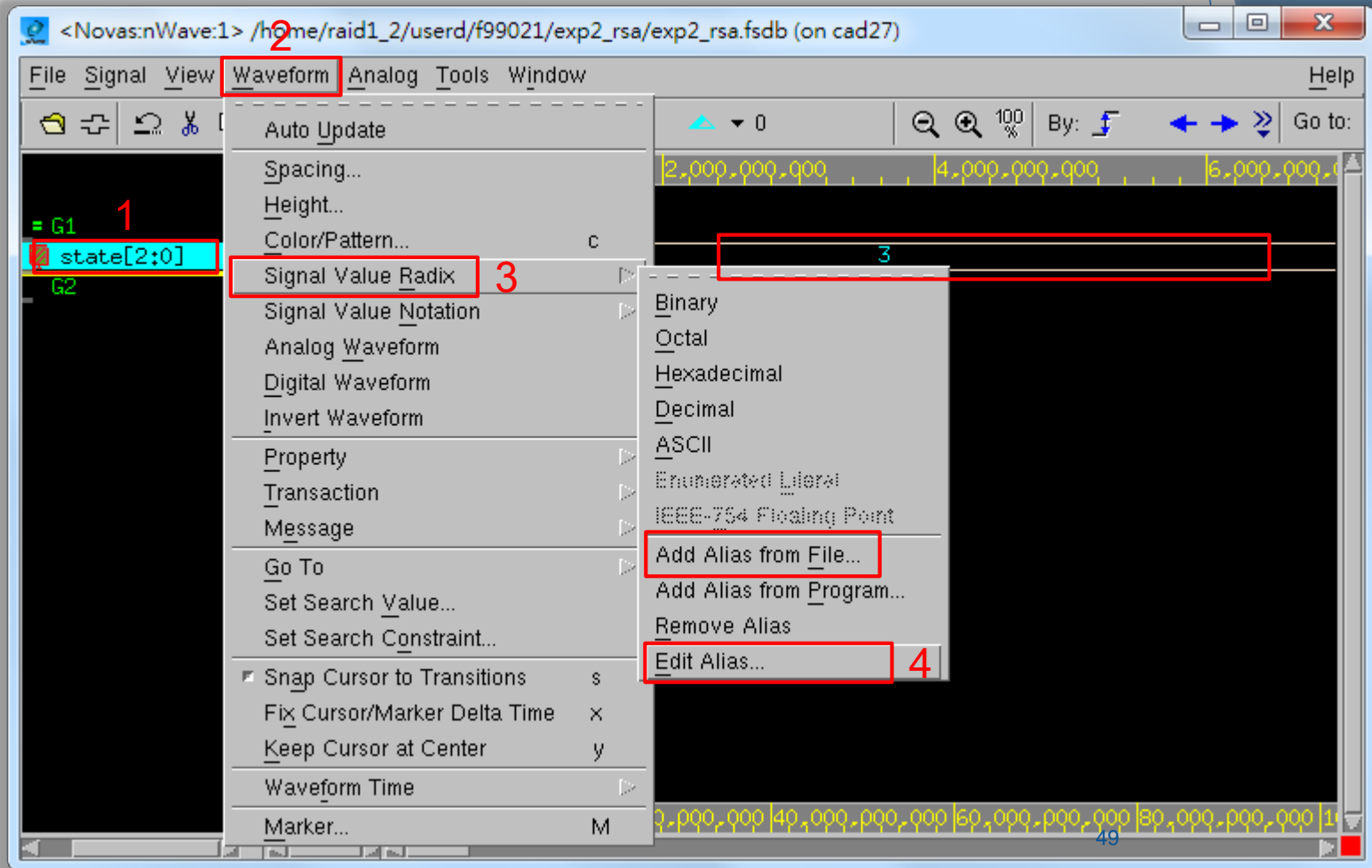


Change Signal Position





Signal Aliasing



Alias Editor (on cad27)

Alias Tables: 1 Slice Tables: 0 Condition Tables: 0

Alias Slice Conditional A

Alias Table: state 1

Alias	Value	Background Color
IDLE	0	NULL
INPUT	1	NULL
OUTPUT	4	NULL
PREMA	2	NULL
MA	3	NULL

Note that signal aliasing is a strict one-to-one correspondence so the value represented in the viewer must exactly represent what format your filter expects. (e.g., binary, hexadecimal)

Reserved Pattern for <value>: Others

Append... Save As... Apply OK Cancel

Save Alias Tables to File (on cad27)

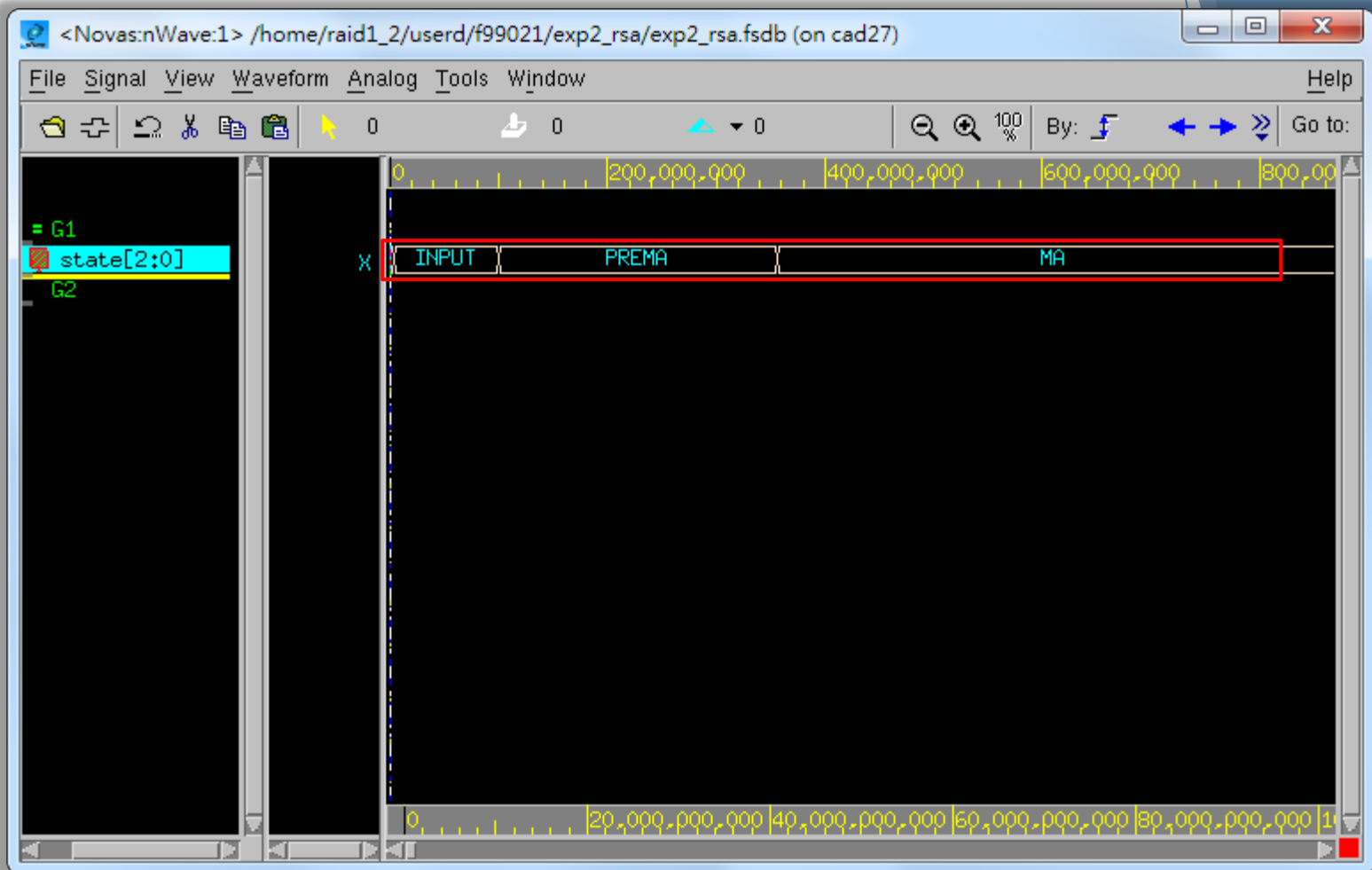
/home/raid1_2/userd/f99021/exp2_rsa/state.alias 4

/home/raid1_2/userd/f99021/exp2_rsa

- INCA_libs
- VerdiLog
- dat
- nWaveLog
- verdiLog

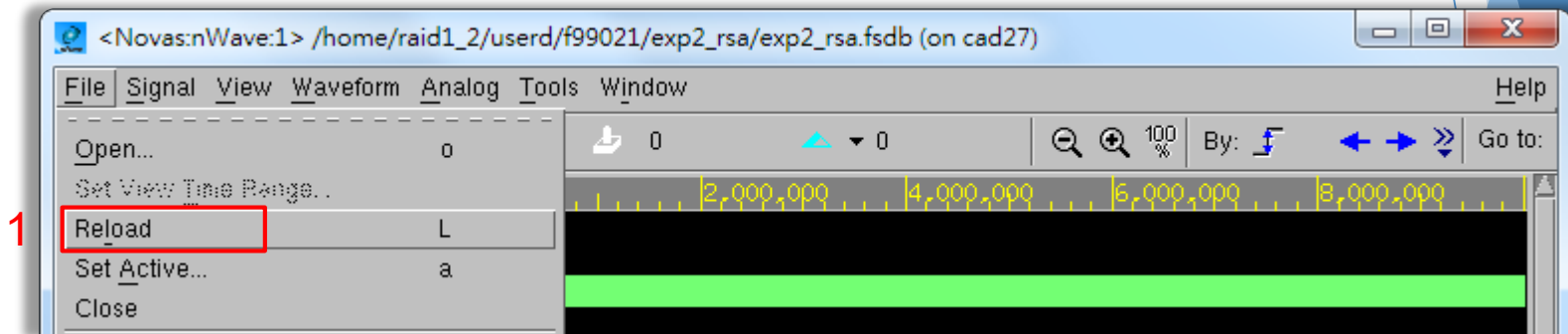
Filter: *.alias

5 OK Cancel



Reload the Waveform

- ▶ Remember to reload the waveform whenever finishing another Verilog simulation.



Verdi

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.
 3. **Unify** diverse and complicated design environments.

Basic Function (1/2)

- ▶ nTrace
 - ▶ A **source code viewer** and analyzer that operates on the knowledge database (**KDB**) to display the **design hierarchy** and **source code** (Verilog, VHDL, SysmVerilog, SystemC, PSL, OVA, mixed) for selected design blocks.
 - ▶ The **main window** of Verdi.

Basic Function (2/2)

- ▶ nWave
 - ▶ A state-of-the-art **graphical waveform viewer** and analyzer that is fully integrated with Verdi's source code, schematic, and flow views.
- ▶ nSchema
 - ▶ A **schematic viewer** and analyzer that generates interactive debug-specific logic diagrams showing the **structure** of selected portions of a design.

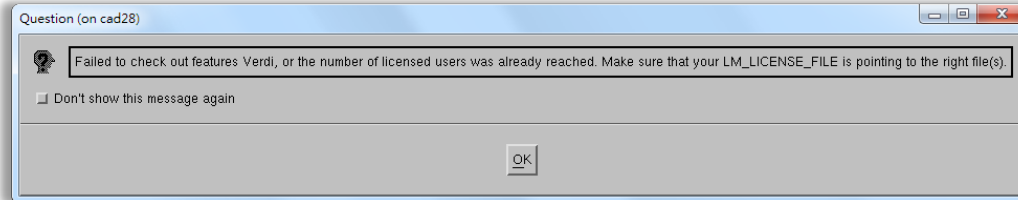
These two tools can be opened through nTrace.

Before Using Verdi

- ▶ Source the environment settings of CAD tools.

```
source /usr/cad/synopsys/CIC/verdi.cshrc
```

- ▶ To avoid the Verdi warning window occurs,



- ▶ please type the following command:

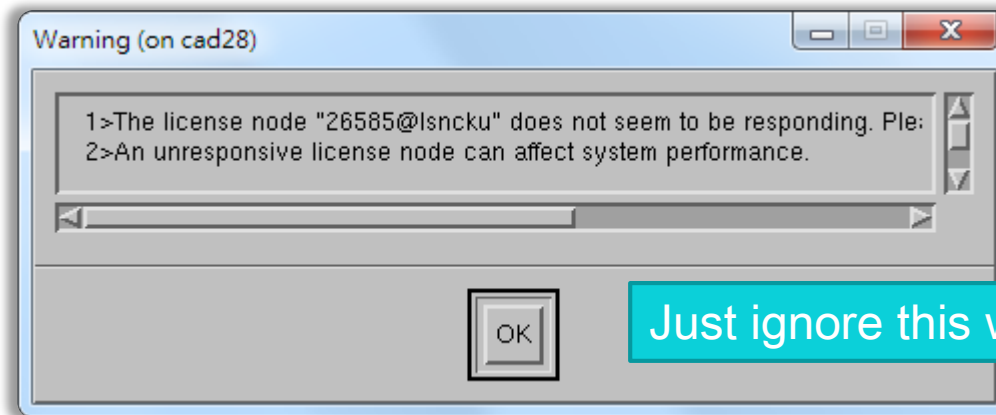
```
setenv LM_LICENSE_FILE '26585@lsntu:26585@lsncku'
```

Start Verdi

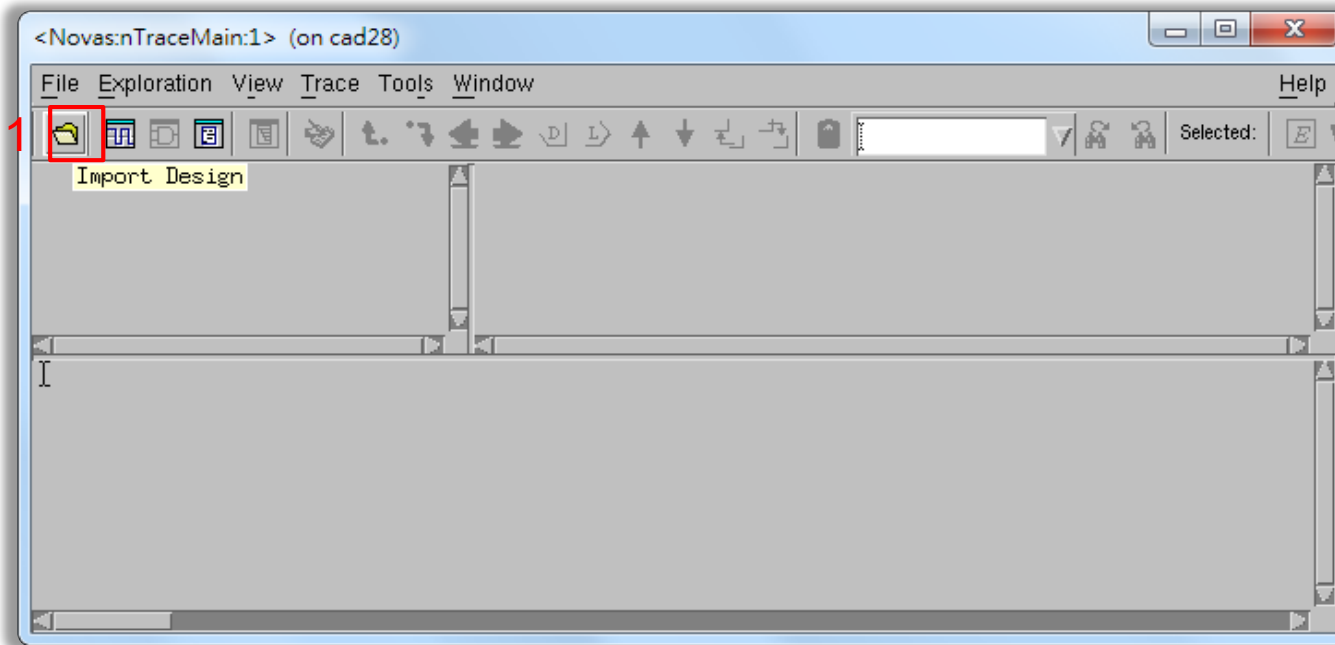
- ▶ Type the following command:

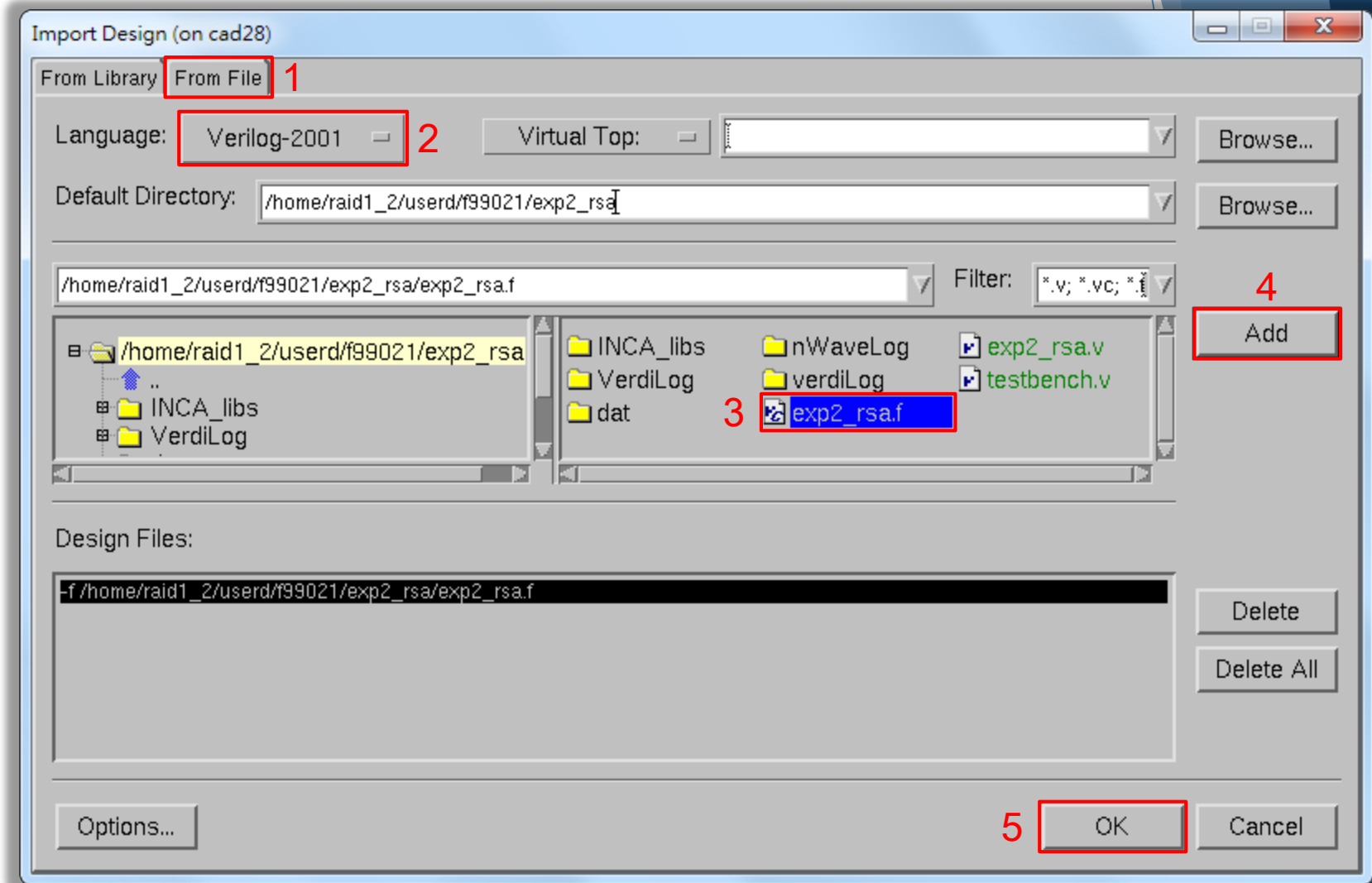
```
verdi &
```

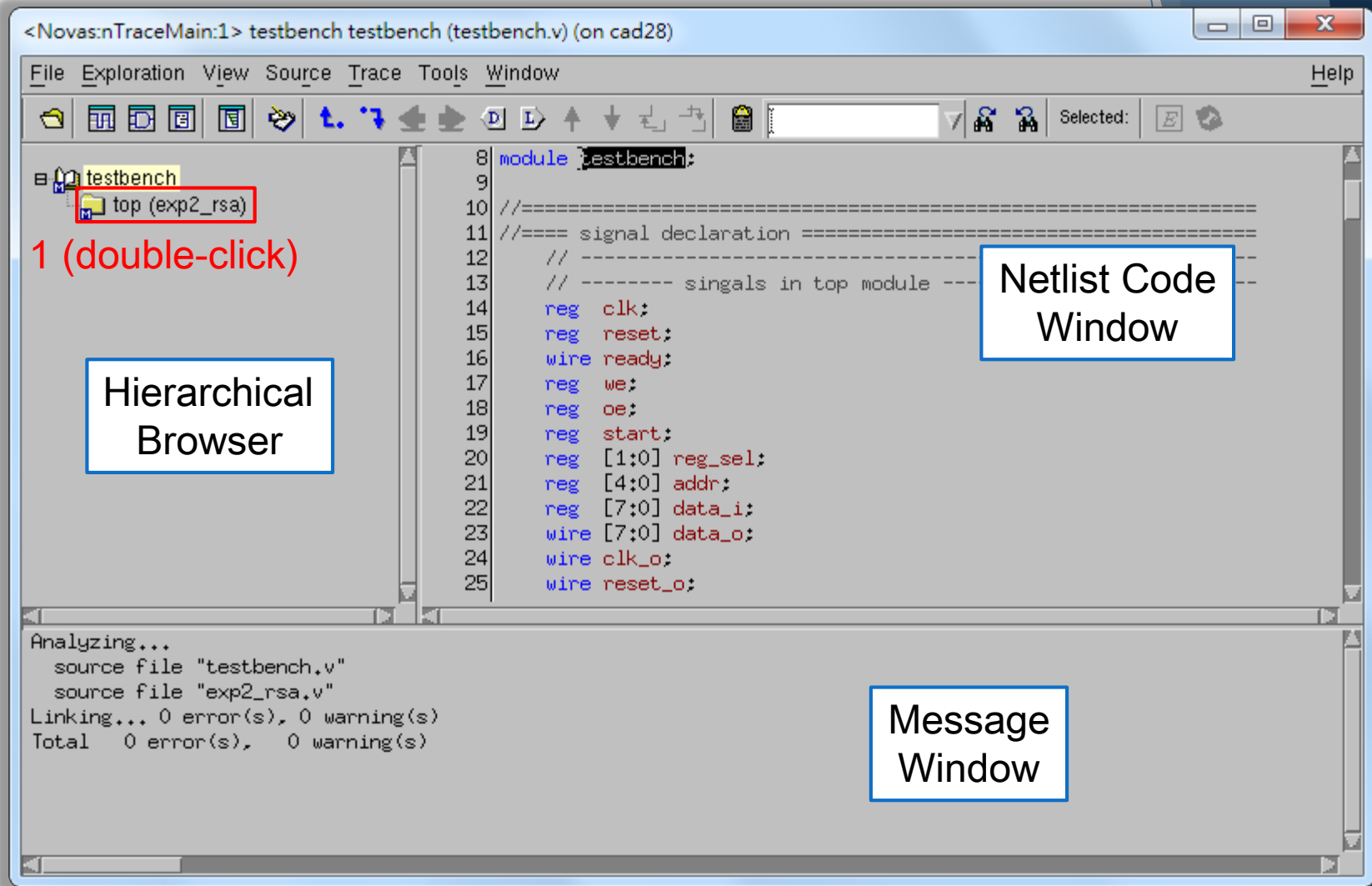
- ▶ Also, the token "&" enable you to use the terminal while Verdi is running in the background.



nTrace



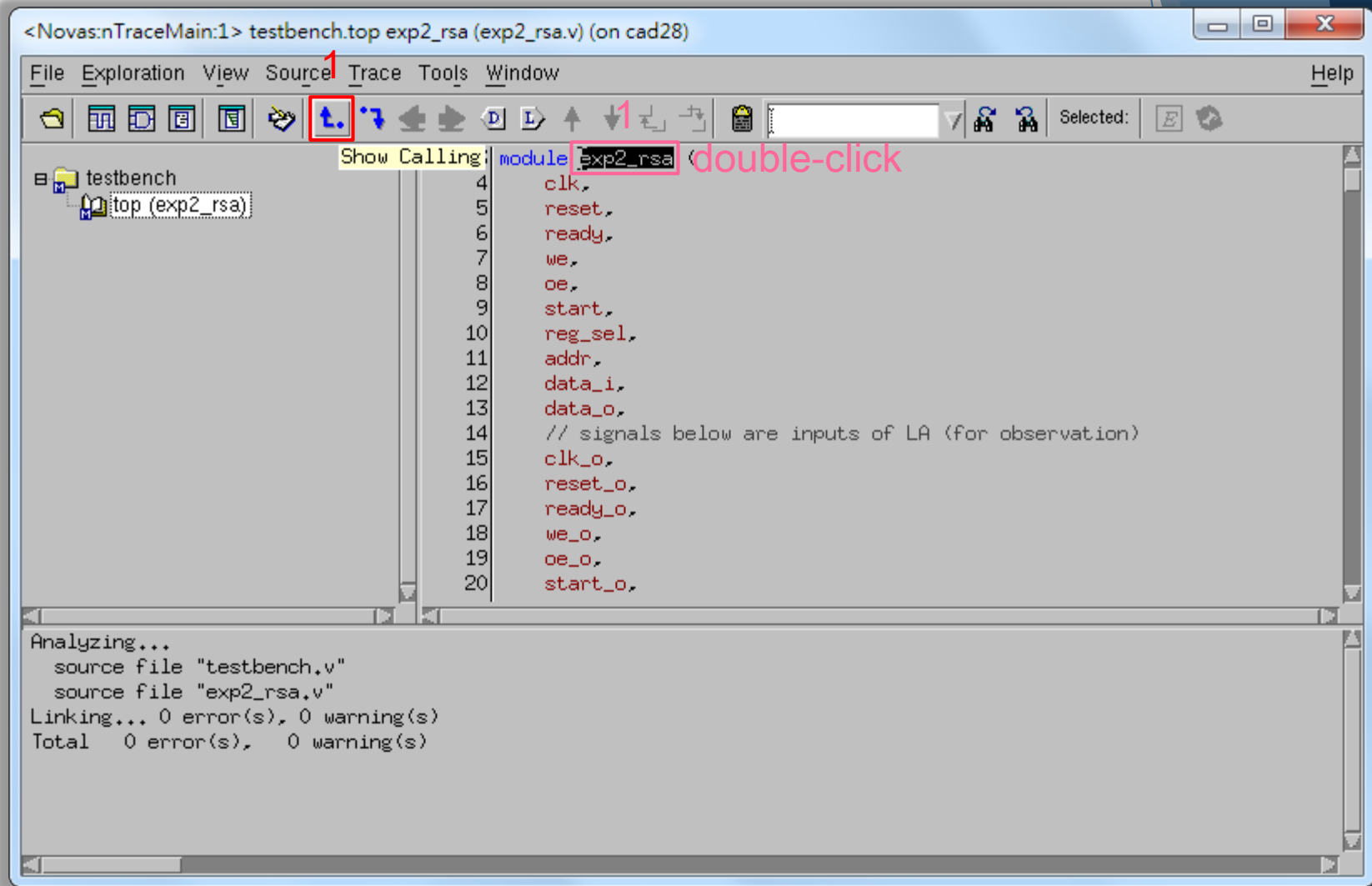


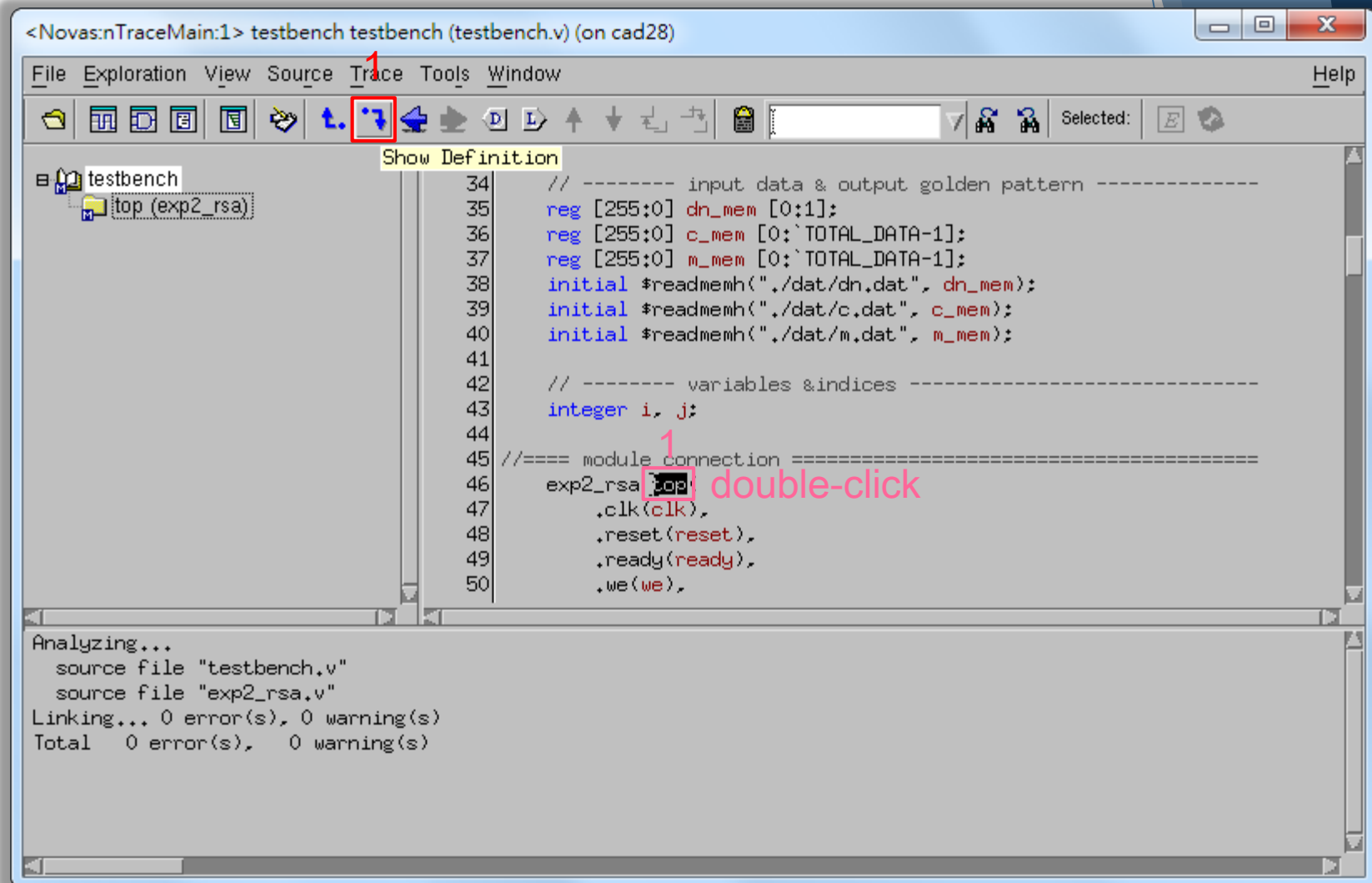


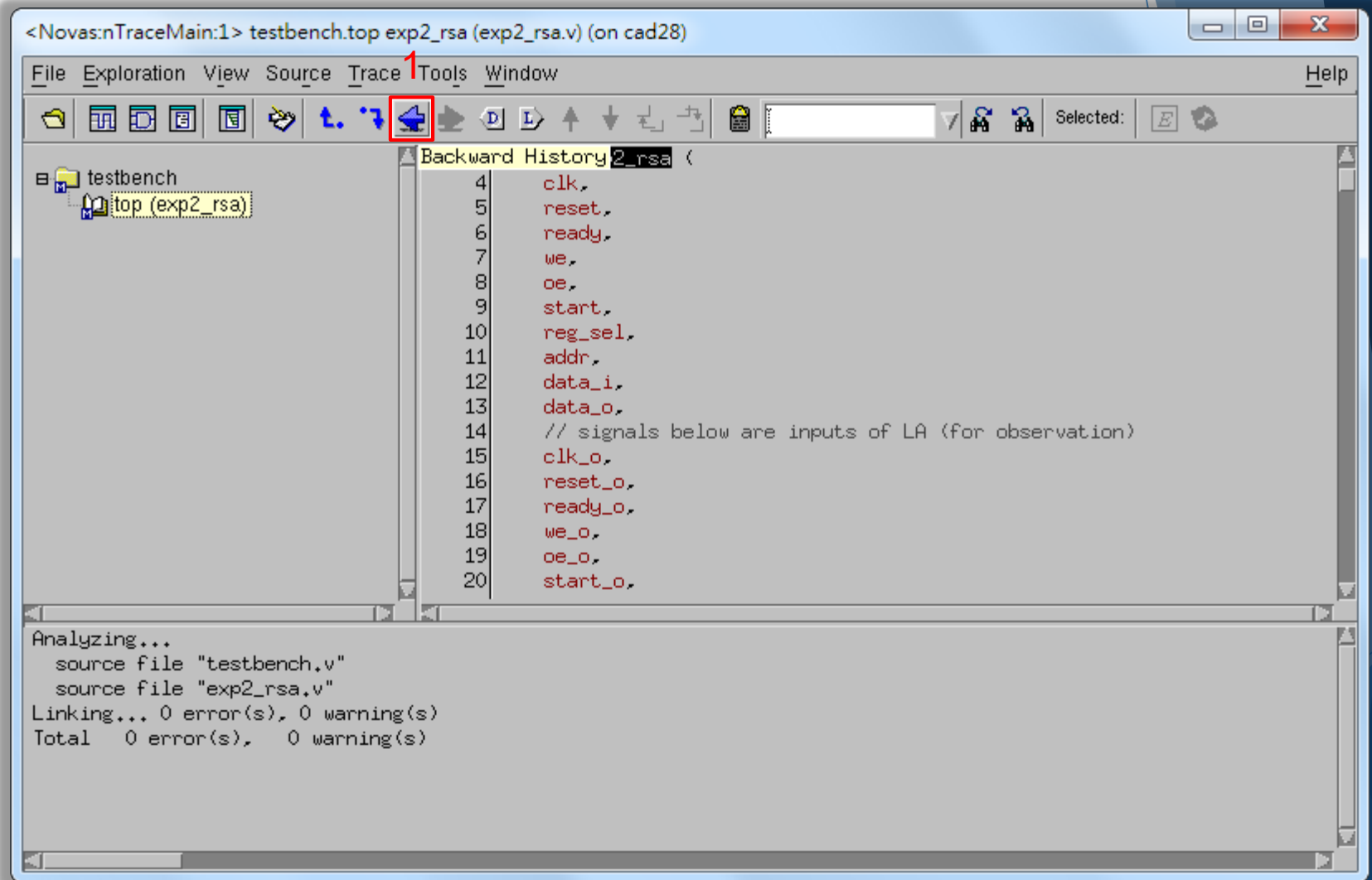
Hierarchical
Browser

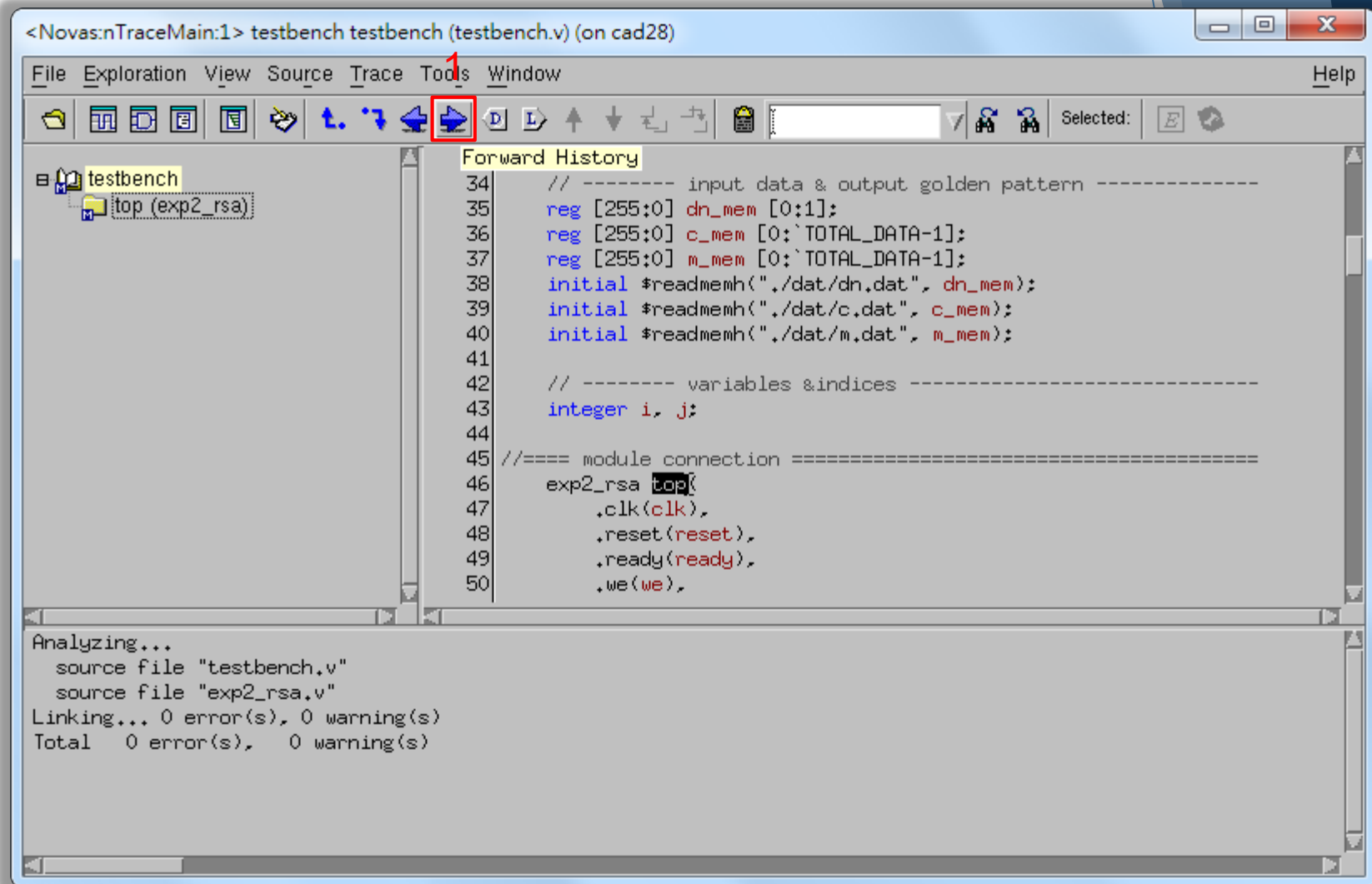
Netlist Code
Window

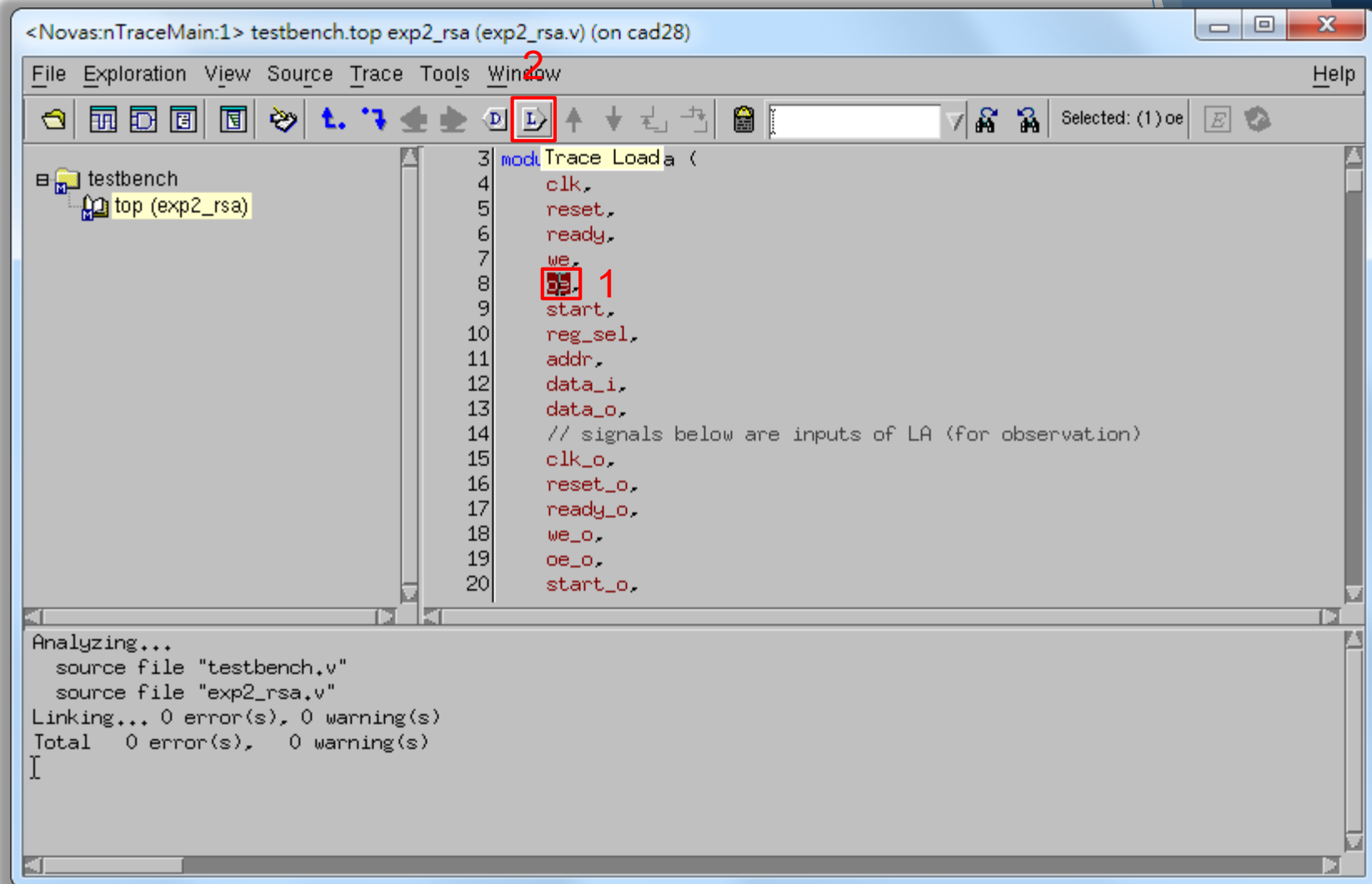
Message
Window

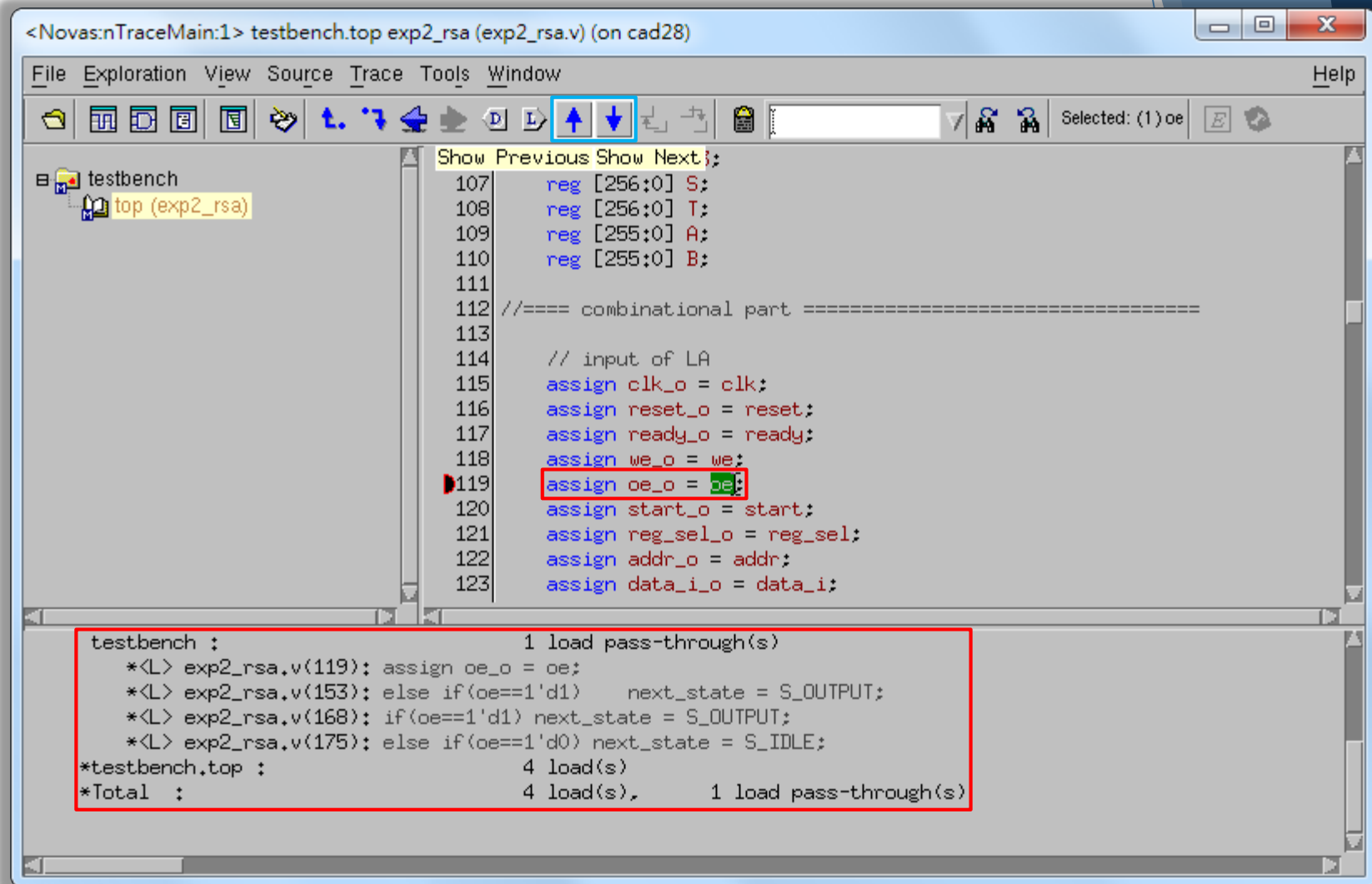


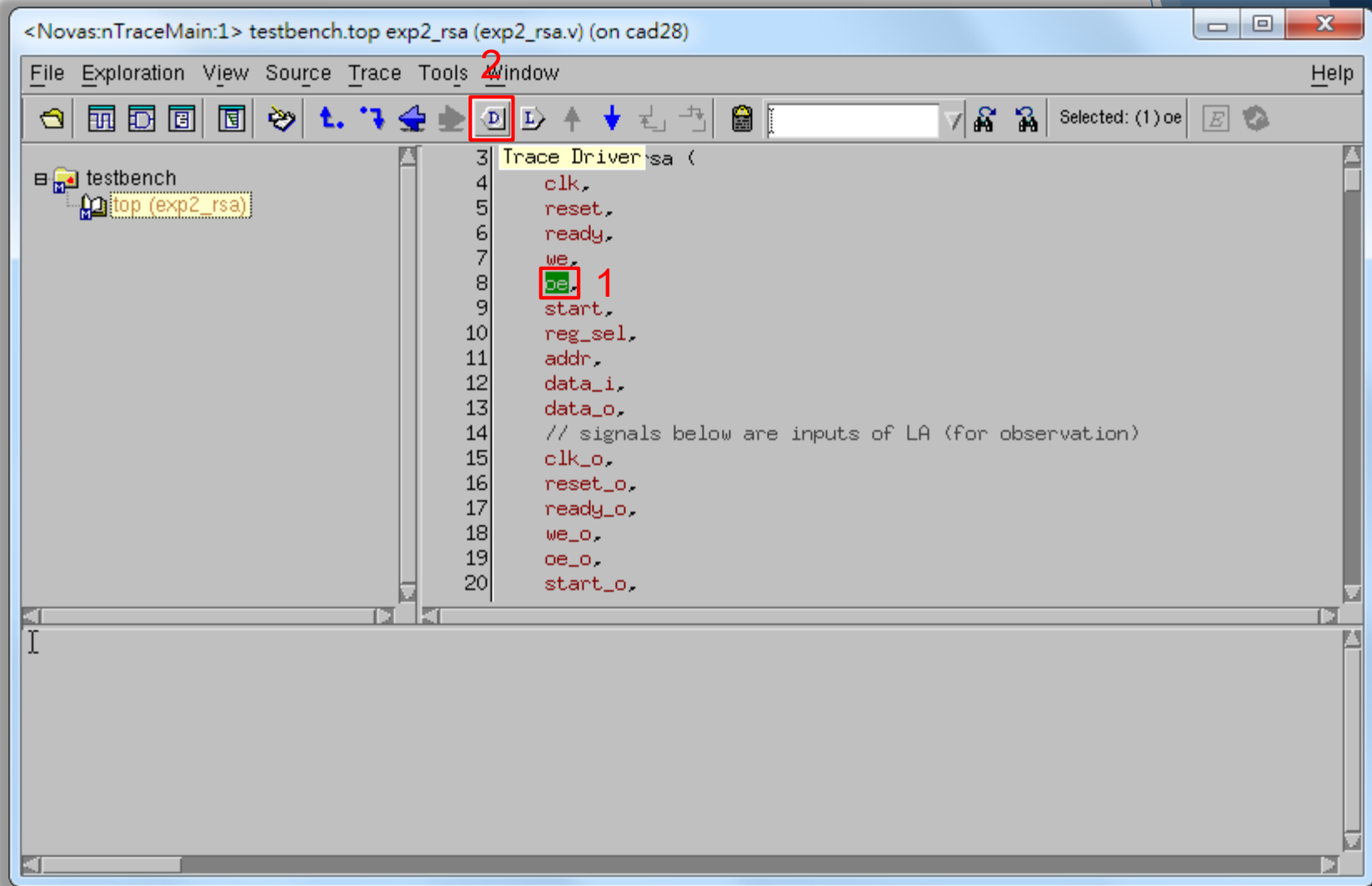












<Novas:nTraceMain:1> testbench testbench (testbench.v) (on cad28)

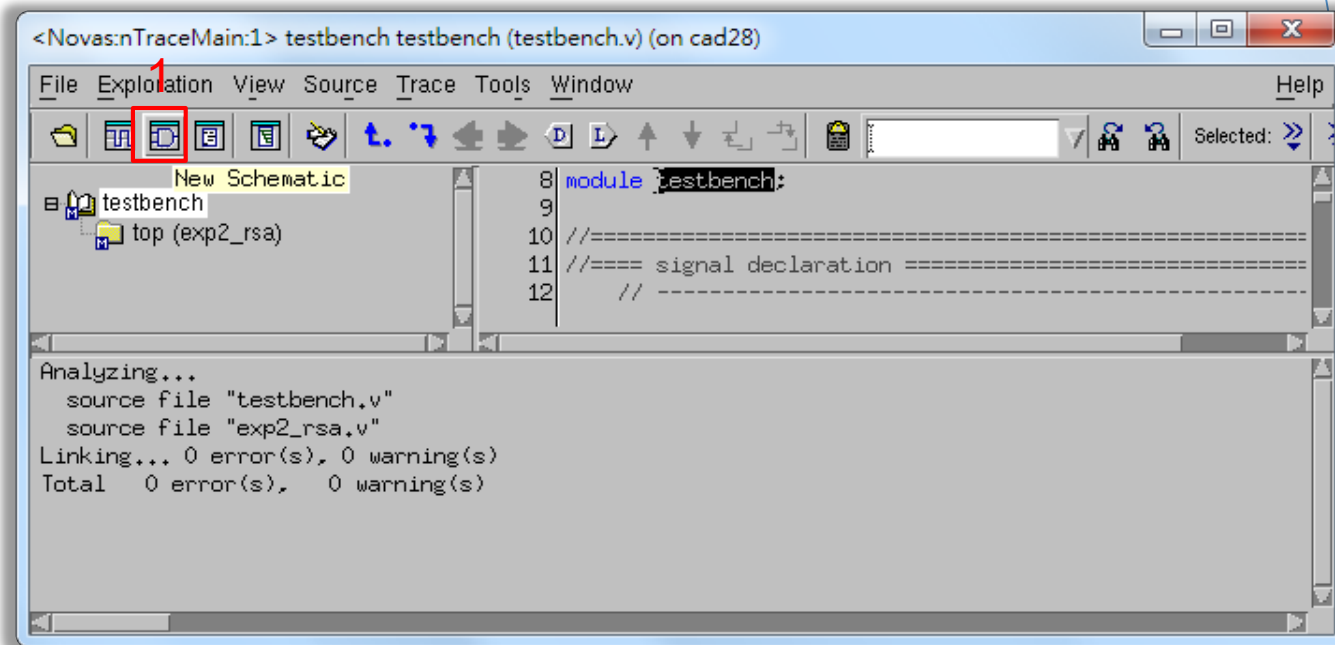
File Exploration View Source Trace Tools Window Help

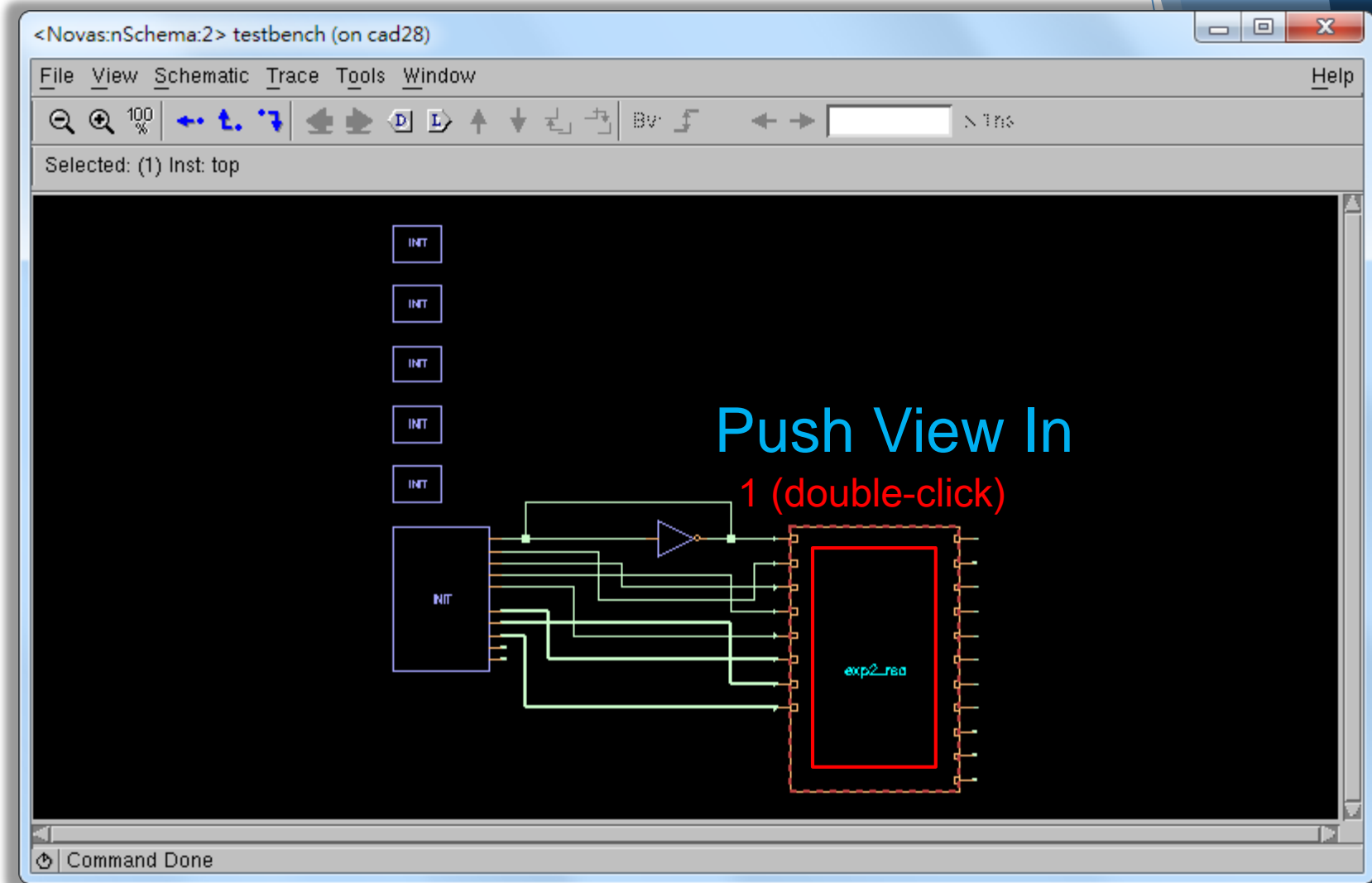
testbench
 top (exp2_rsa)

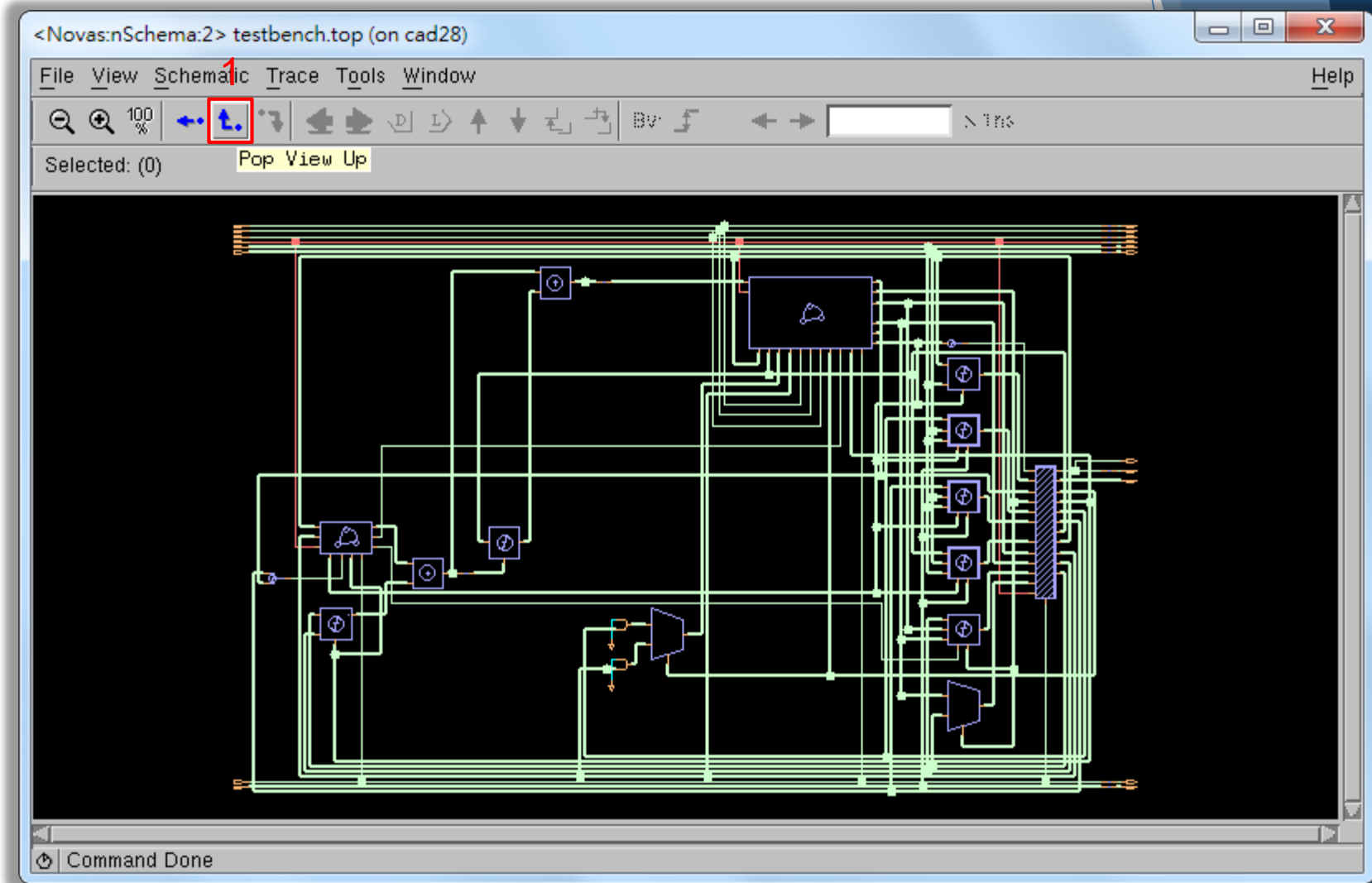
```
81      #0; // t = 0
82      clk      = 1'b1;
83      reset    = 1'b0;
84      we       = 1'b0;
85      oe       = 1'b0;
86      start    = 1'b0;
87      reg_sel  = 2'd0;
88      addr     = 4'd0;
89      data_i   = 8'd0;
90
91      #(`CYCLE) reset = 1'b1; // t = 1
92      #(`CYCLE) reset = 1'b0; // t = 2
93
94      #(`CYCLE*0.01);
95      // a3 & a2
96      i = 0;
97      while(i<64) begin
98          #(`CYCLE);
```

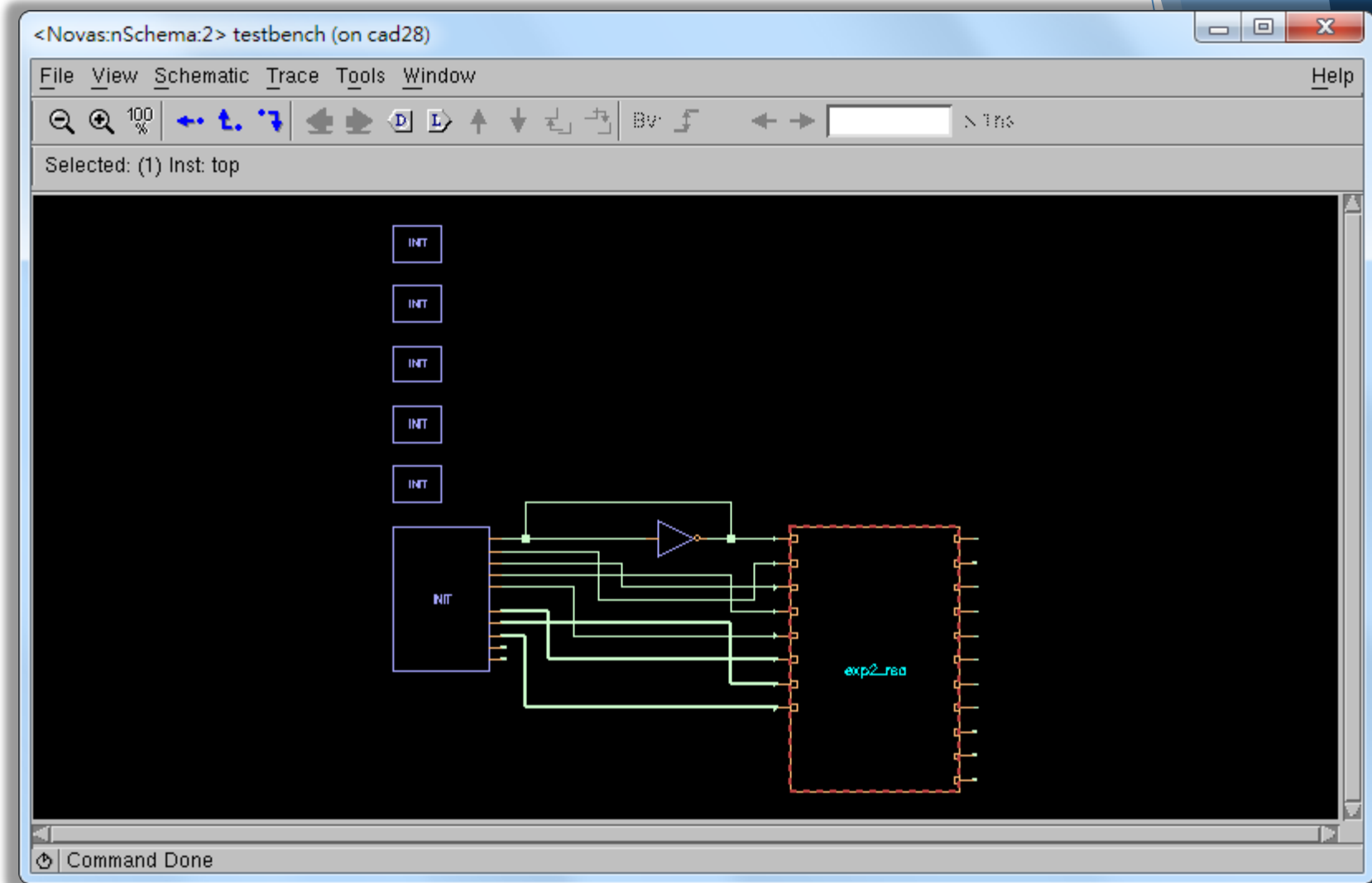
```
< 3> testbench.top.oe /* results of trace driver */
*
```

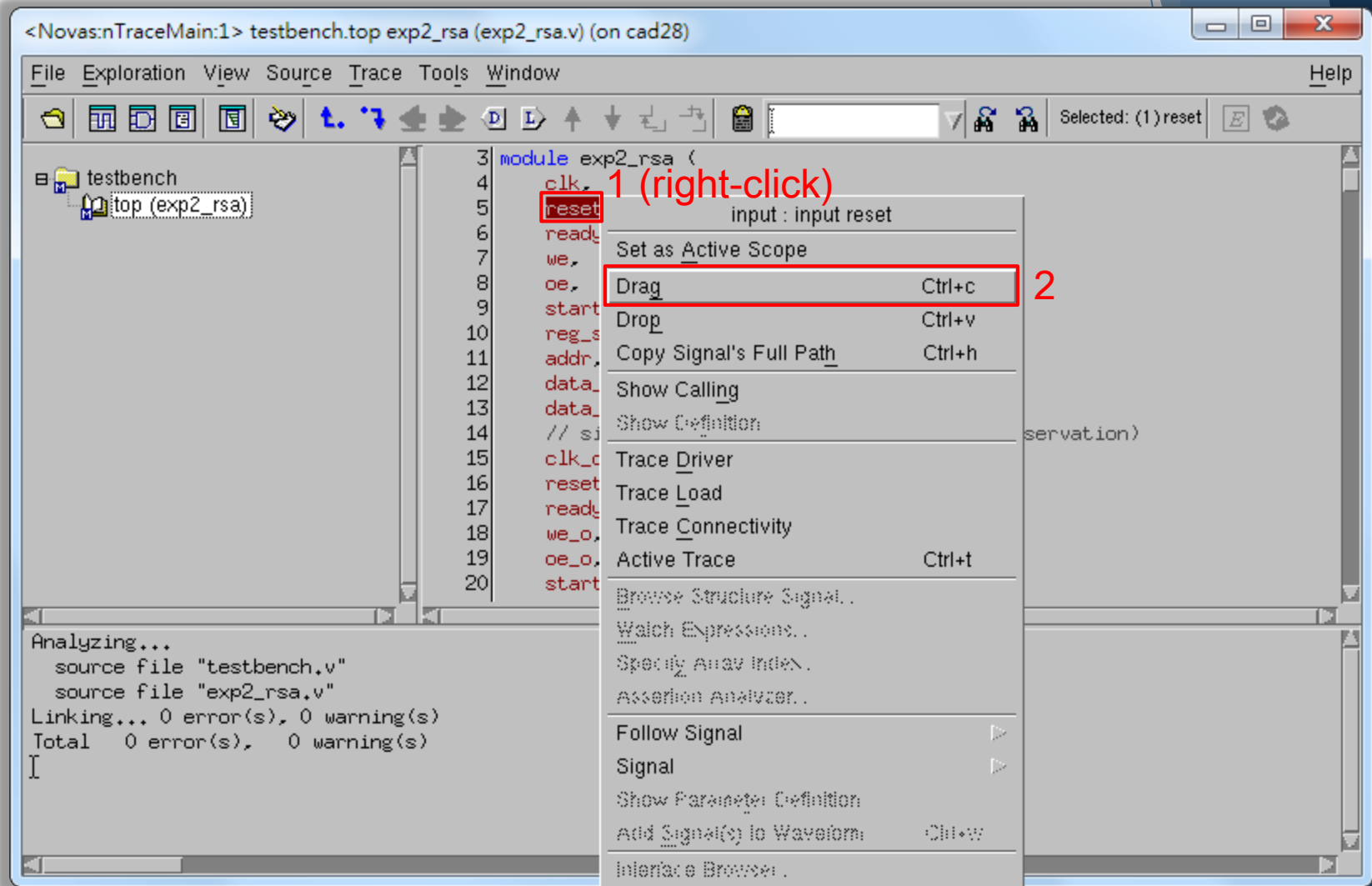
nSchema

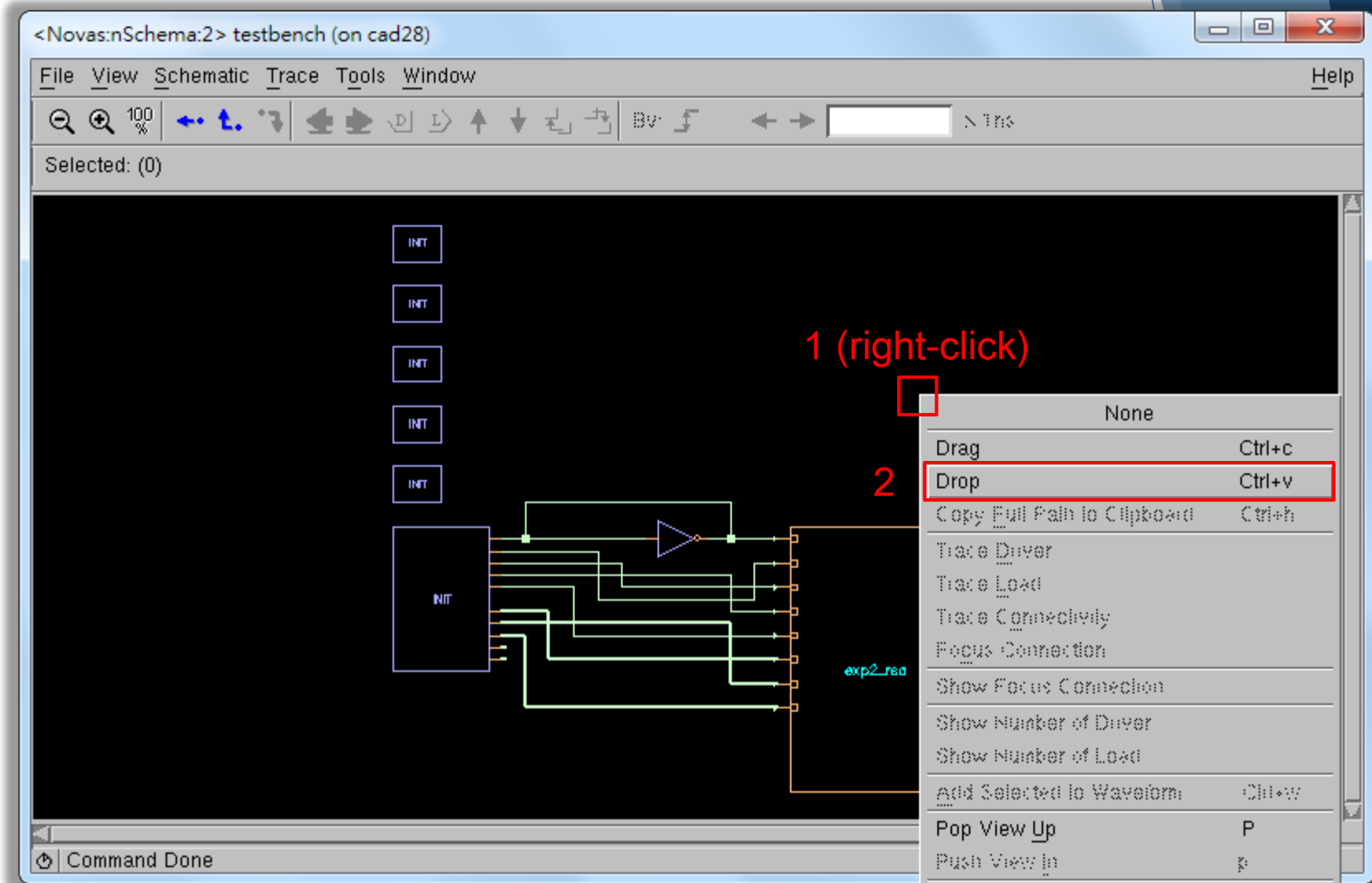


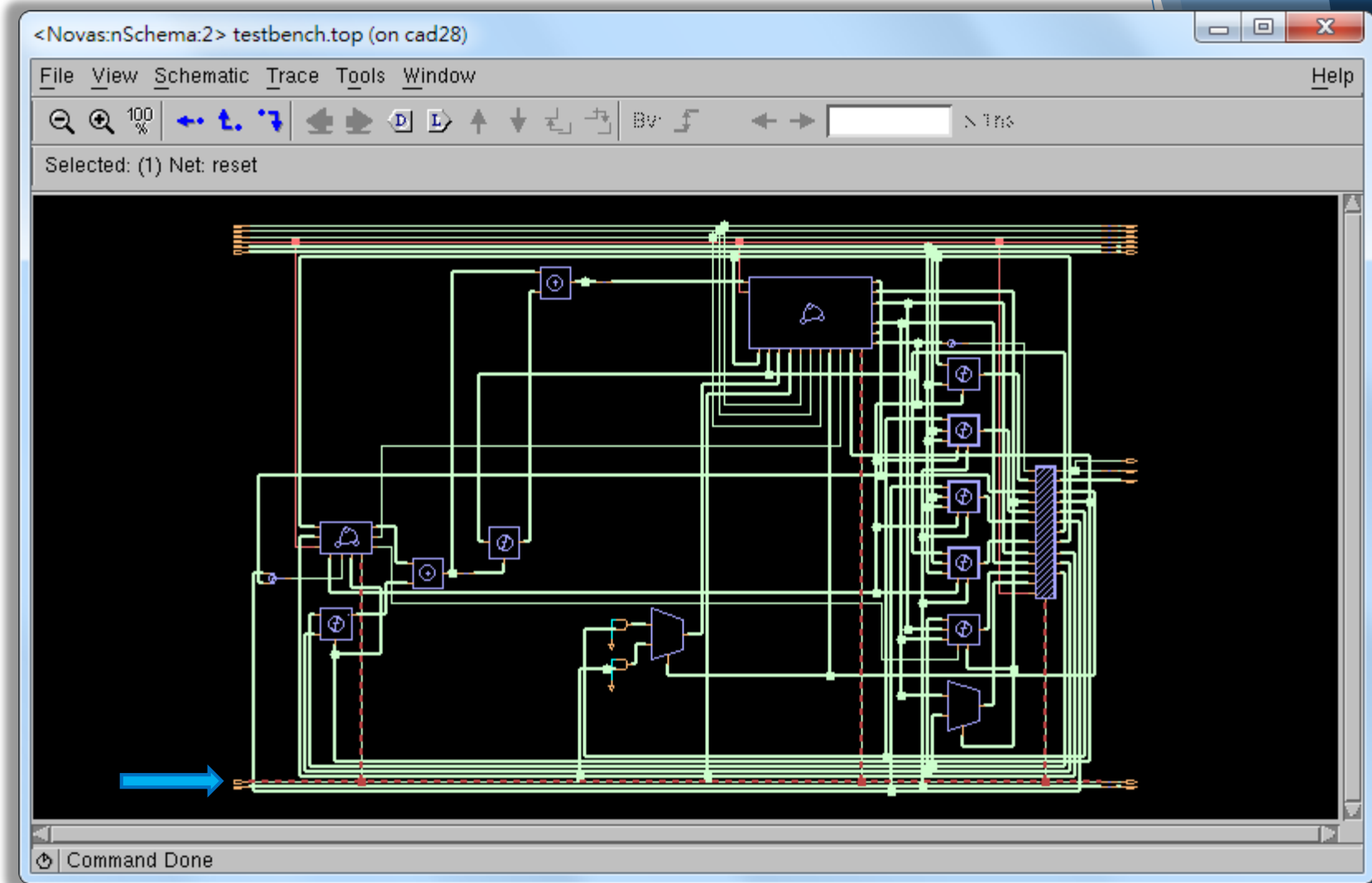




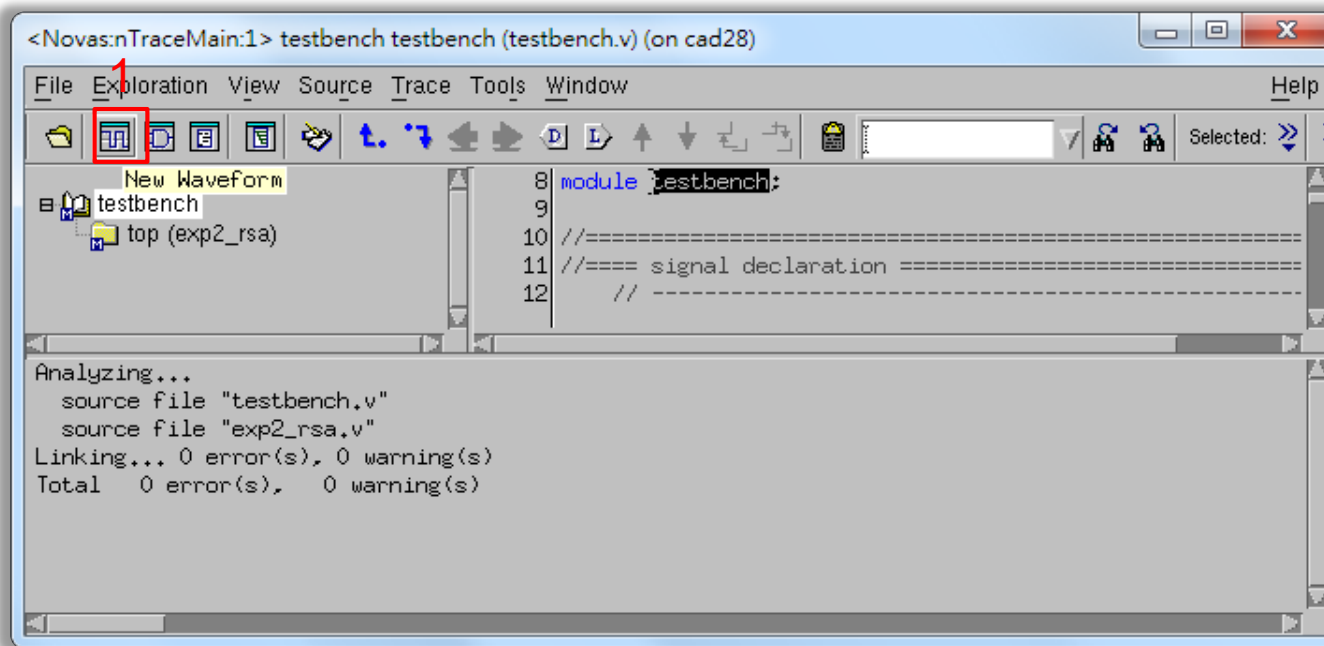


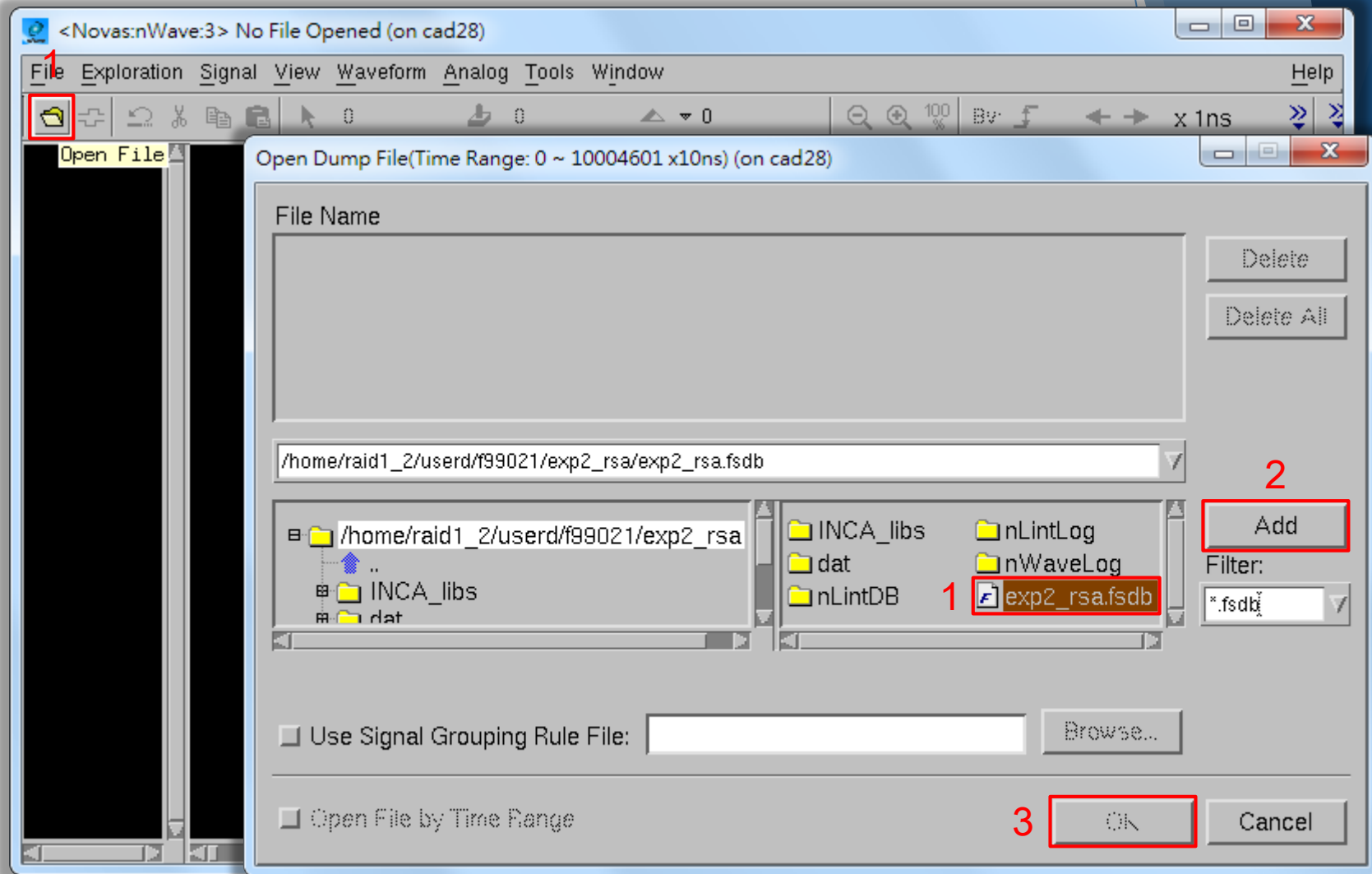


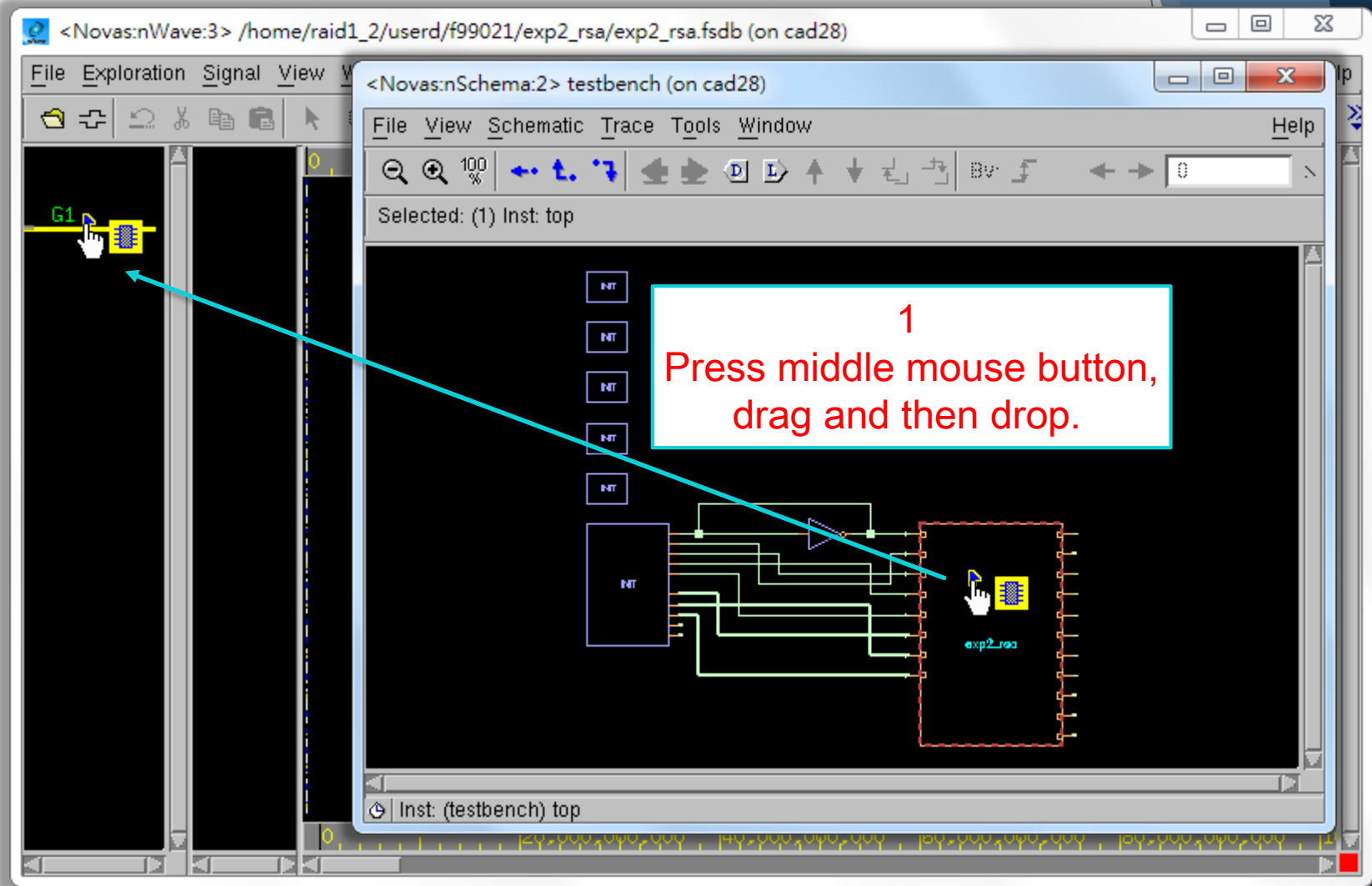


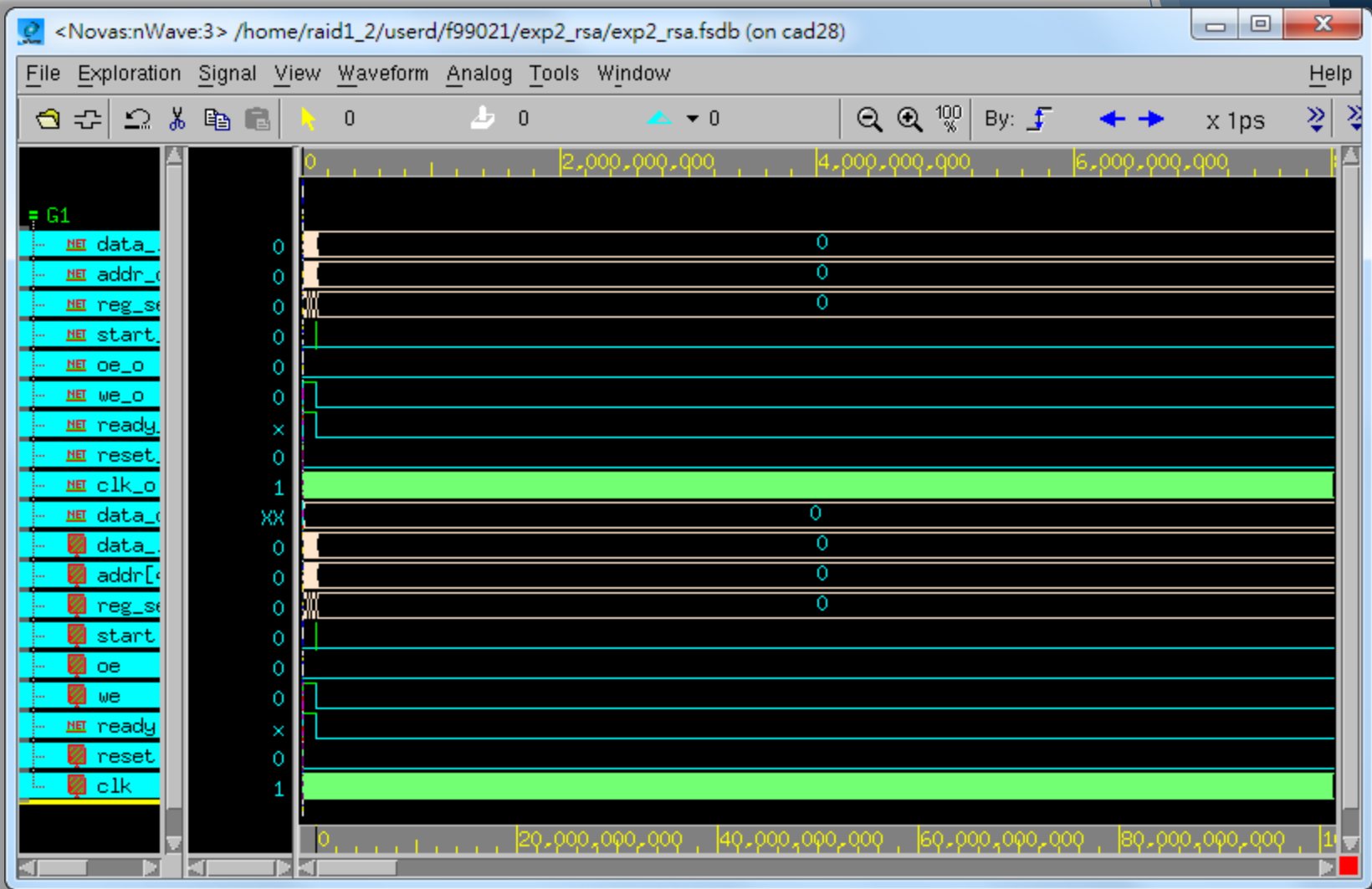


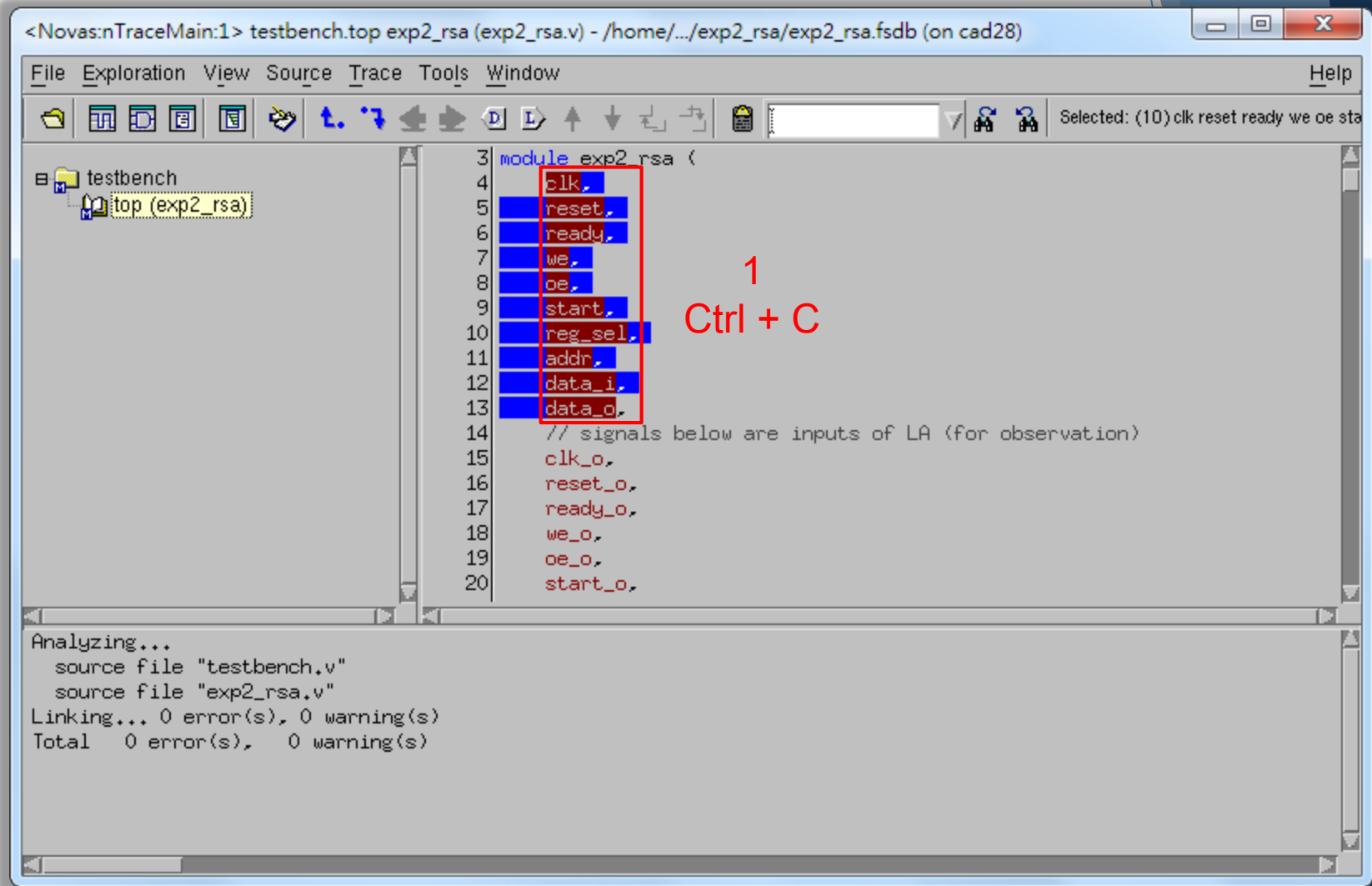
nWave

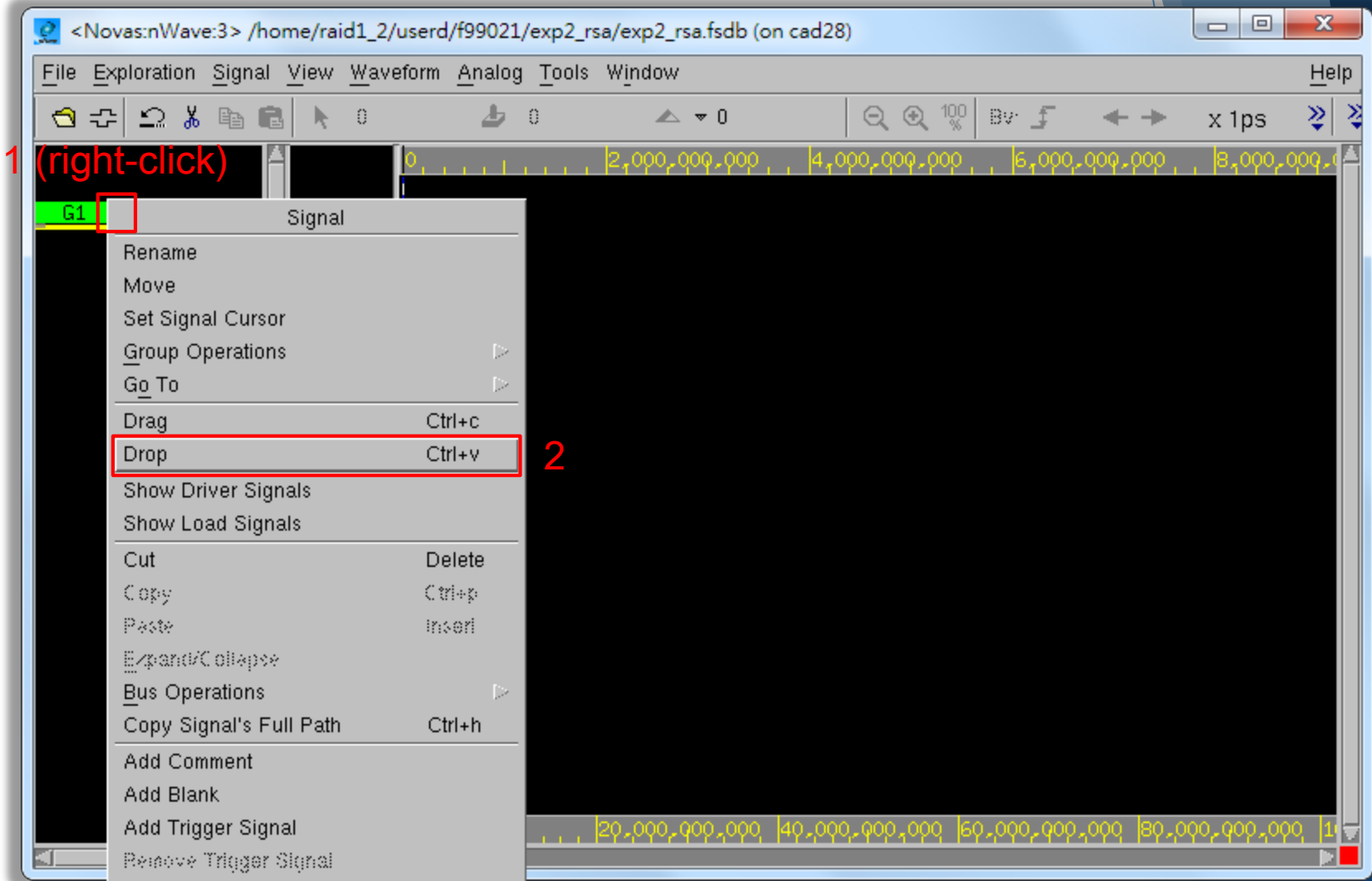


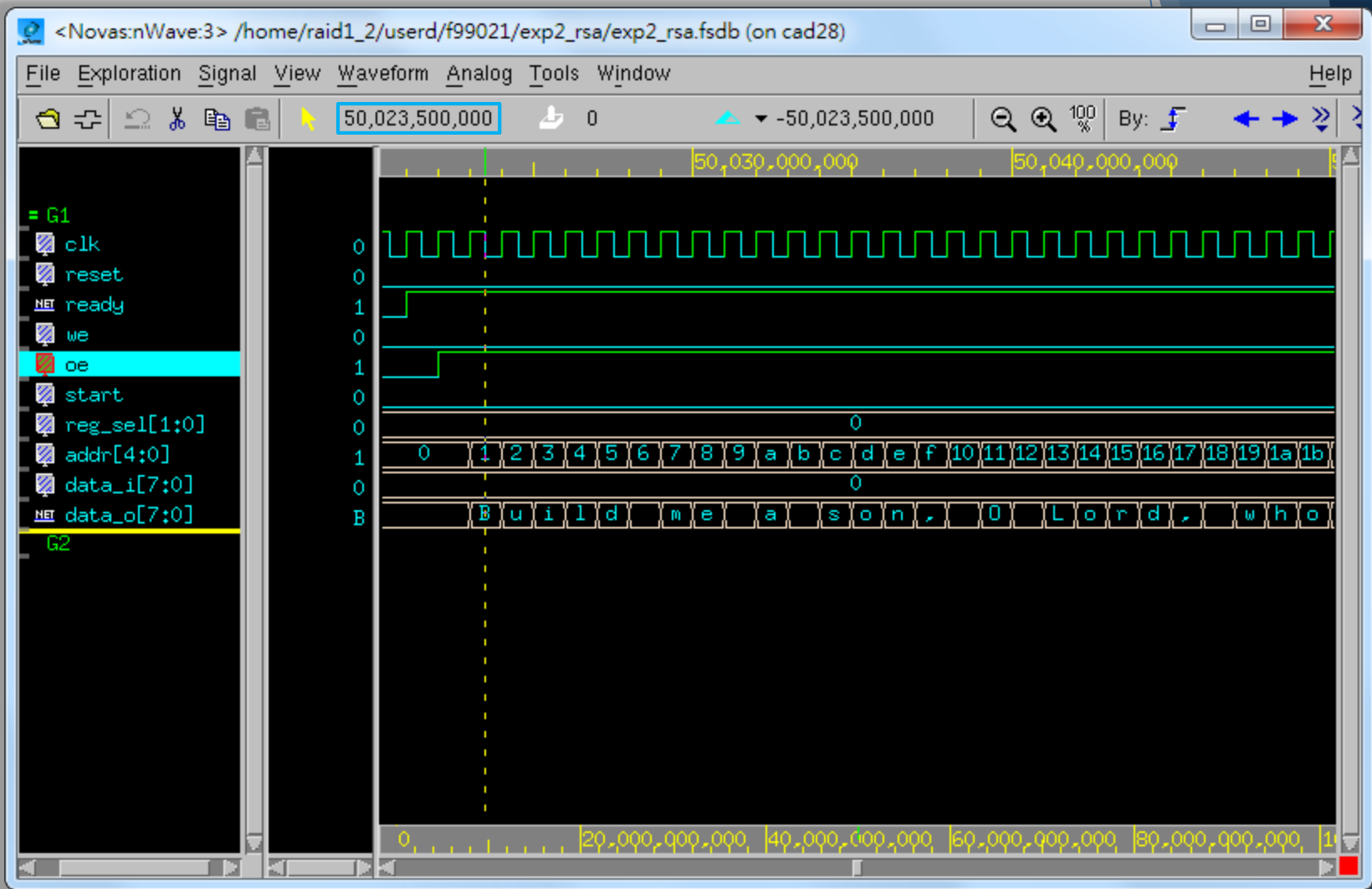


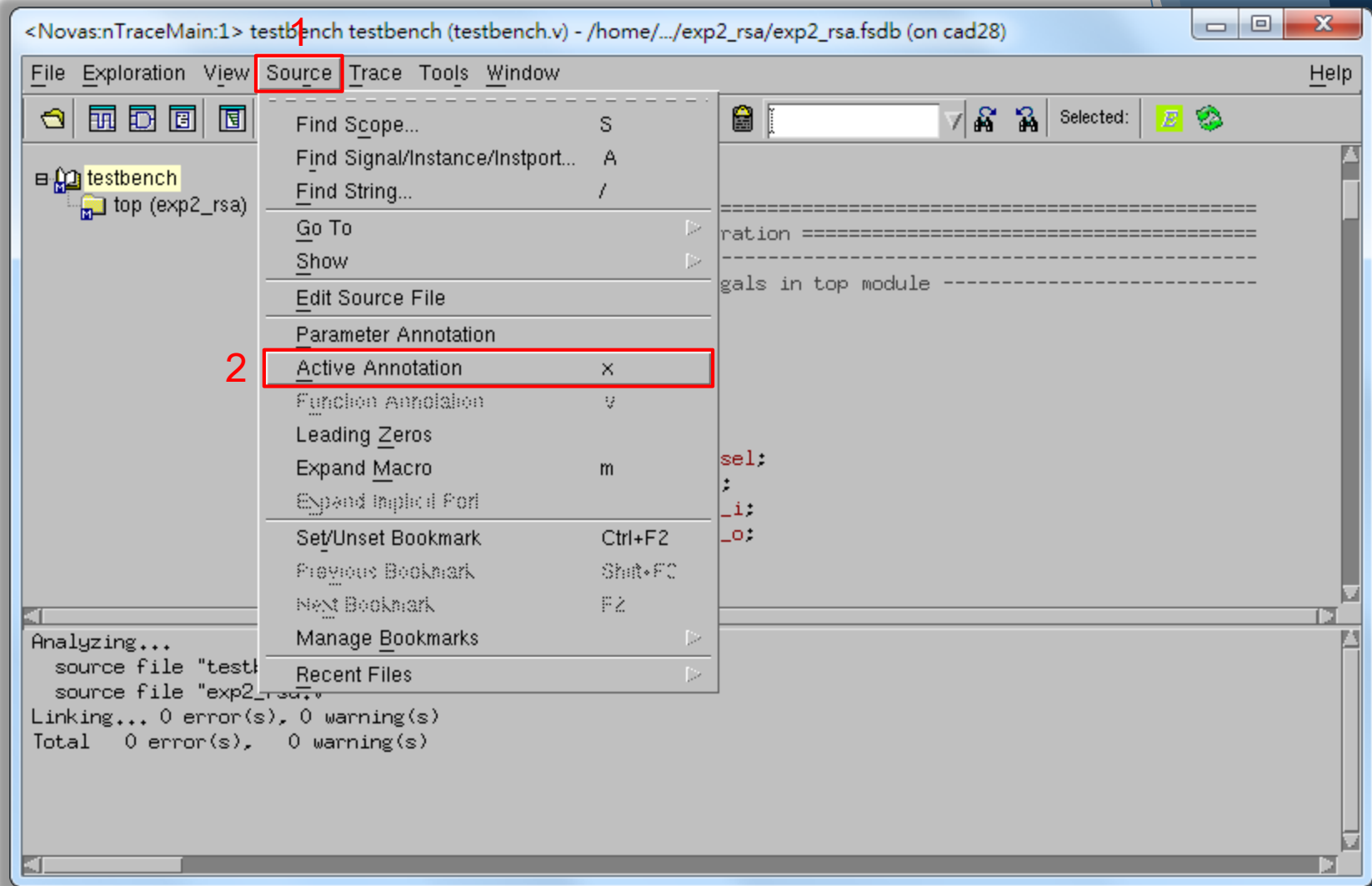


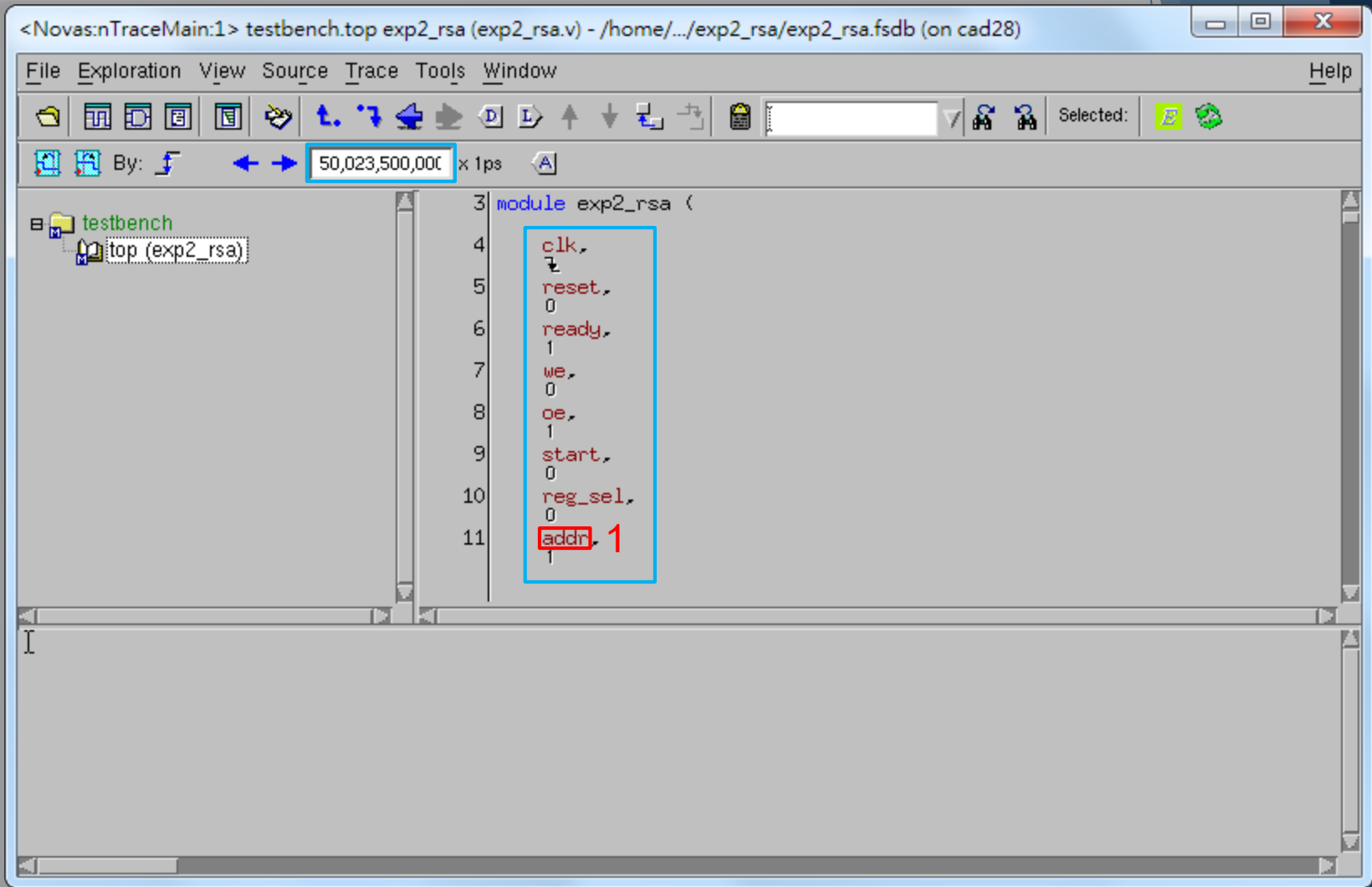


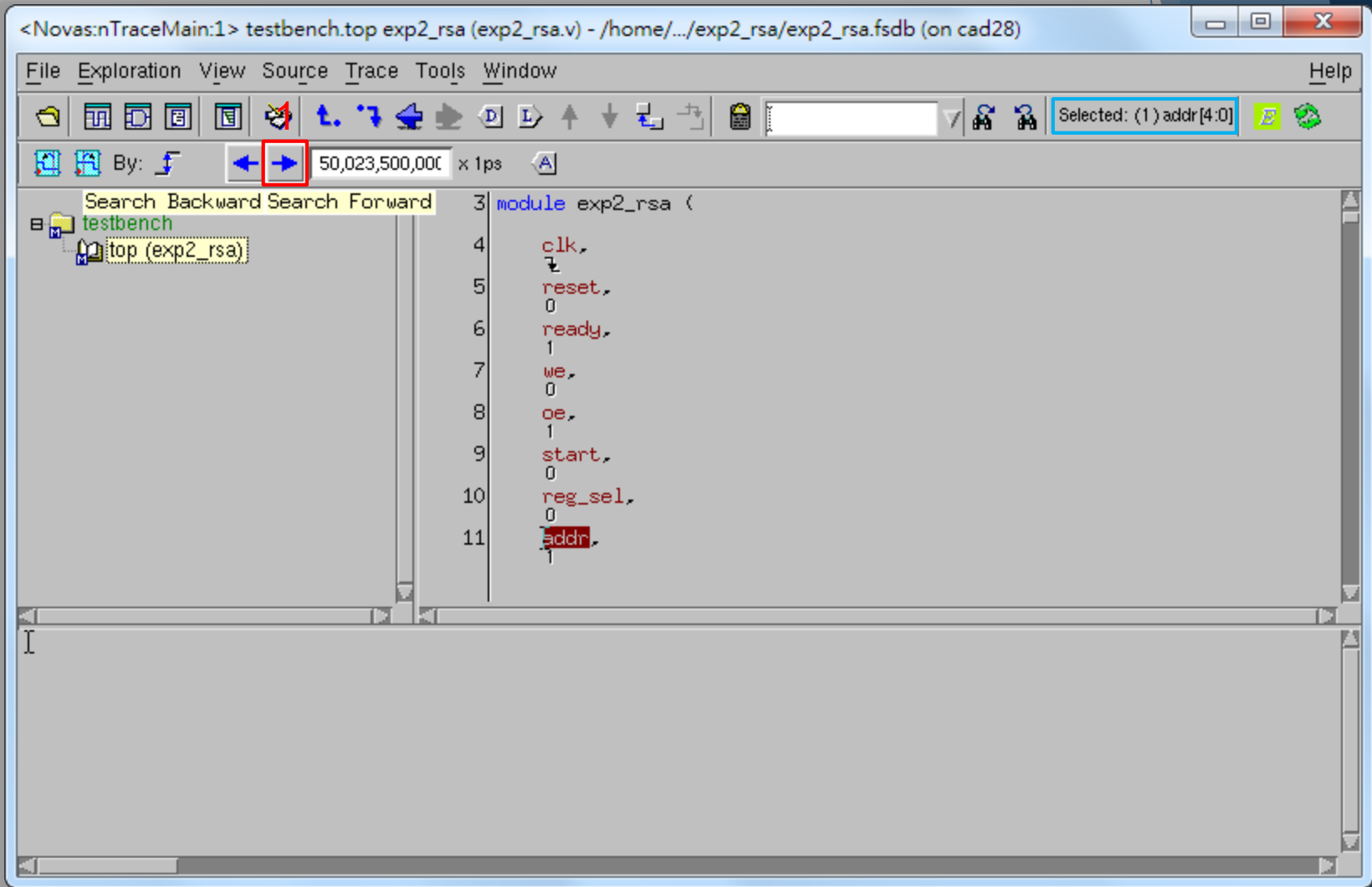


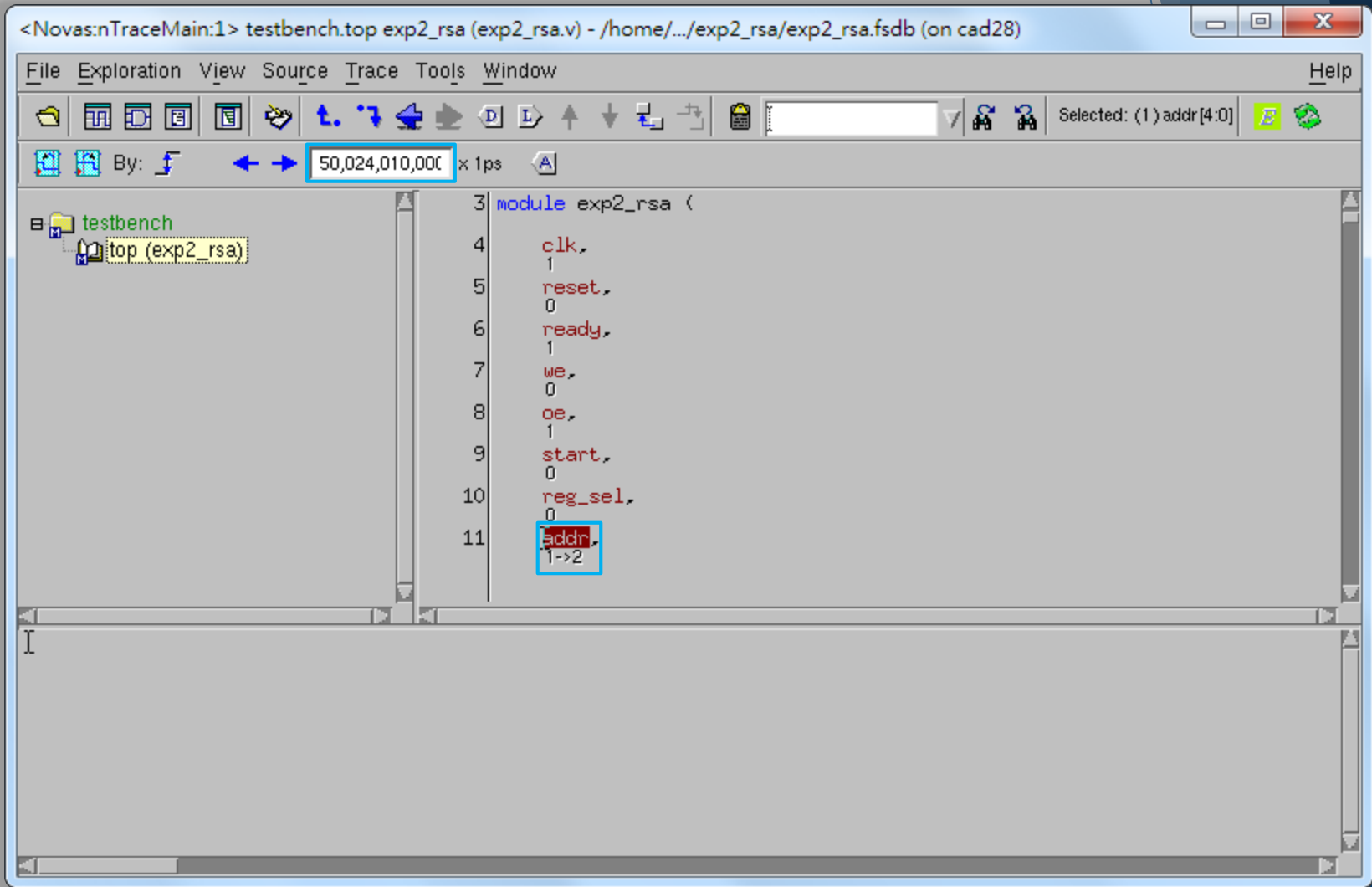


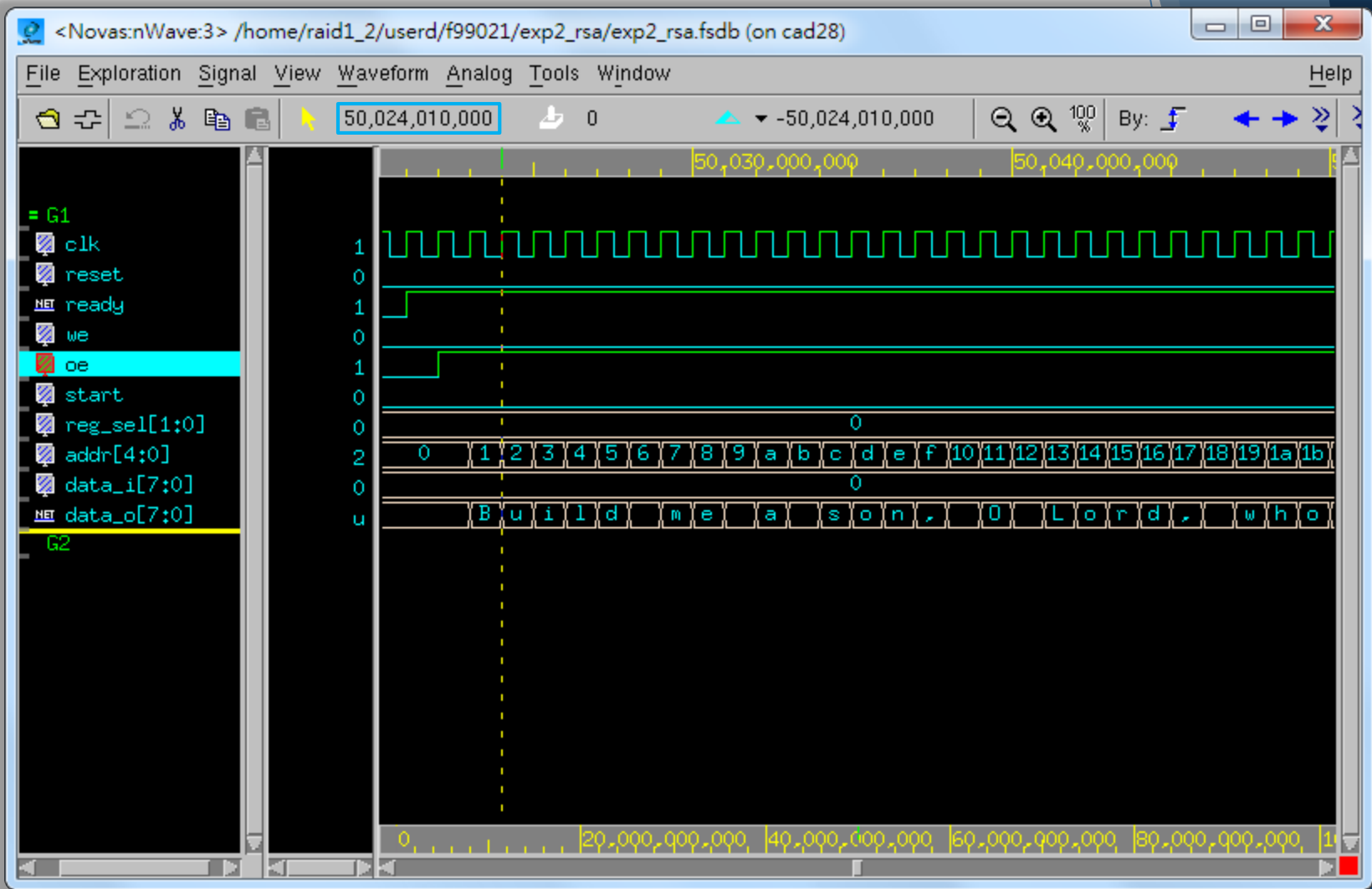












The background features abstract, overlapping geometric shapes in various shades of blue, primarily on the left and right sides, framing a central white area. The shapes include triangles and polygons, some with thin white outlines.

The End

Reference

1. "MobaXterm User Manual" by The Centre for eResearch, University of Auckland.
2. "Cadence NC-Verilog Simulator Tutorial" by Cadence
3. "Quick Start: an nLint Tutorial" by NOVAS
4. "Introduction to Verdi" by Abel Hu
5. "Verdi³ datasheet" by Synopsys