EE6094
CAD for VLSI Design

# Compiler and Checker

Andy, Yu-Guang Chen
Assistant Professor, Department of EE
National Central University
andyygchen@ee.ncu.edu.tw
Slides Credit: TA黃仕成



## Outline

- Compiler
- Checker

# Outline

- Compiler
- Checker



# Compiler

- How to compile and execute your source code on workstation?
  - Step1: Use g++ command.

```
[109521021@eda359_forclass ~/isc2v-master]$ g++ -std=c++11 test.cpp -o test
```

  - Step2: Execute your output file.

```
[109521021@eda359_forclass ~]$ ./test
Hello World!
[109521021@eda359_forclass ~]$
```

./Your output file name

# Outline

- Compiler
- Checker

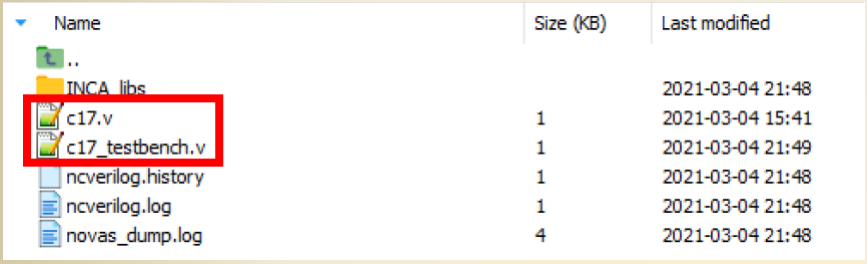


# Checker

- How to check verilog code that dumped by your c++ code is correct?
  - We need to use testbench to check the result.

# Checker

- How to use the testbench?
  - Step1: Make sure you put your verilog code and testbench code files at the same folder.






# Checker

- How to use the testbench?
  - Step2: Key in the following commands.
    source /usr/cad/cadence/CIC/incisiv.cshrc
    source /usr/cad/synopsys/CIC/verdi.cshrc

```
[108521118@eda359_forclass PA1]$ source /usr/cad/cadence/CIC/incisiv.cshrc
[108521118@eda359_forclass PA1]$ source /usr/cad/synopsys/CIC/verdi.cshrc
```

  - Step3: Step3 Use “ncverilog” command .

```
[108521118@eda359_forclass PA1]$ ncverilog +access+r c17.v c17_testbench.v
```

Verilog code file | Testbench code file

# Checker

◆Result

```
ncverilog: 15.20-s039: (c) Copyright 1995-2017 Cadence Design Systems, Inc.
Recompiling... reason: file './c17_testbench.v' is newer than expected.
        expected: Thu Mar  4 21:48:39 2021
        actual:   Thu Mar  4 21:49:44 2021
file: c17_testbench.v
        module worklib.c17_tb:v
                errors: 0, warnings: 0
                Caching library 'worklib' ....... Done
        Elaborating the design hierarchy:
        Building instance overlay tables: ..................... Done
        Generating native compiled code:
                worklib.c17_tb:v <0x5ca70917>
                        streams:   7, words: 12208
        Building instance specific data structures.
        Loading native compiled code:     ..................... Done
        Design hierarchy summary:
                                    Instances  Unique
                Modules:                    2       2
                Primitives:                 6       1
                Registers:                  0       0
                Scalar wires:               5       -
                Initial blocks:             1       1
                Simulation timescale:  1ps
        Writing initial simulation snapshot: worklib.c17_tb:v
Loading snapshot worklib.c17_tb:v ..................... Done
*Verdi* Loading libsscore_ius152.so
ncsim> source /usr/cad/cadence/INCISIV/cur/tools/inca/files/ncsimrc
ncsim> run
 input pattern = 00000 --> golden value = 00
 your answer = 00
 input pattern = 10101 --> golden value = 11
 your answer = 11
 input pattern = 01010 --> golden value = 11
 your answer = 11
 input pattern = 11011 --> golden value = 11
 your answer = 11
 input pattern = 11111 --> golden value = 10
 your answer = 10
You're all correct!!!
  ***     ***
 *****   *****
***************
 ***********
  *********
   *******
    *****
     ***
      *
ncsim: *W,RNQUIE: Simulation is complete.
ncsim> exit
```