

# MobaXtrem

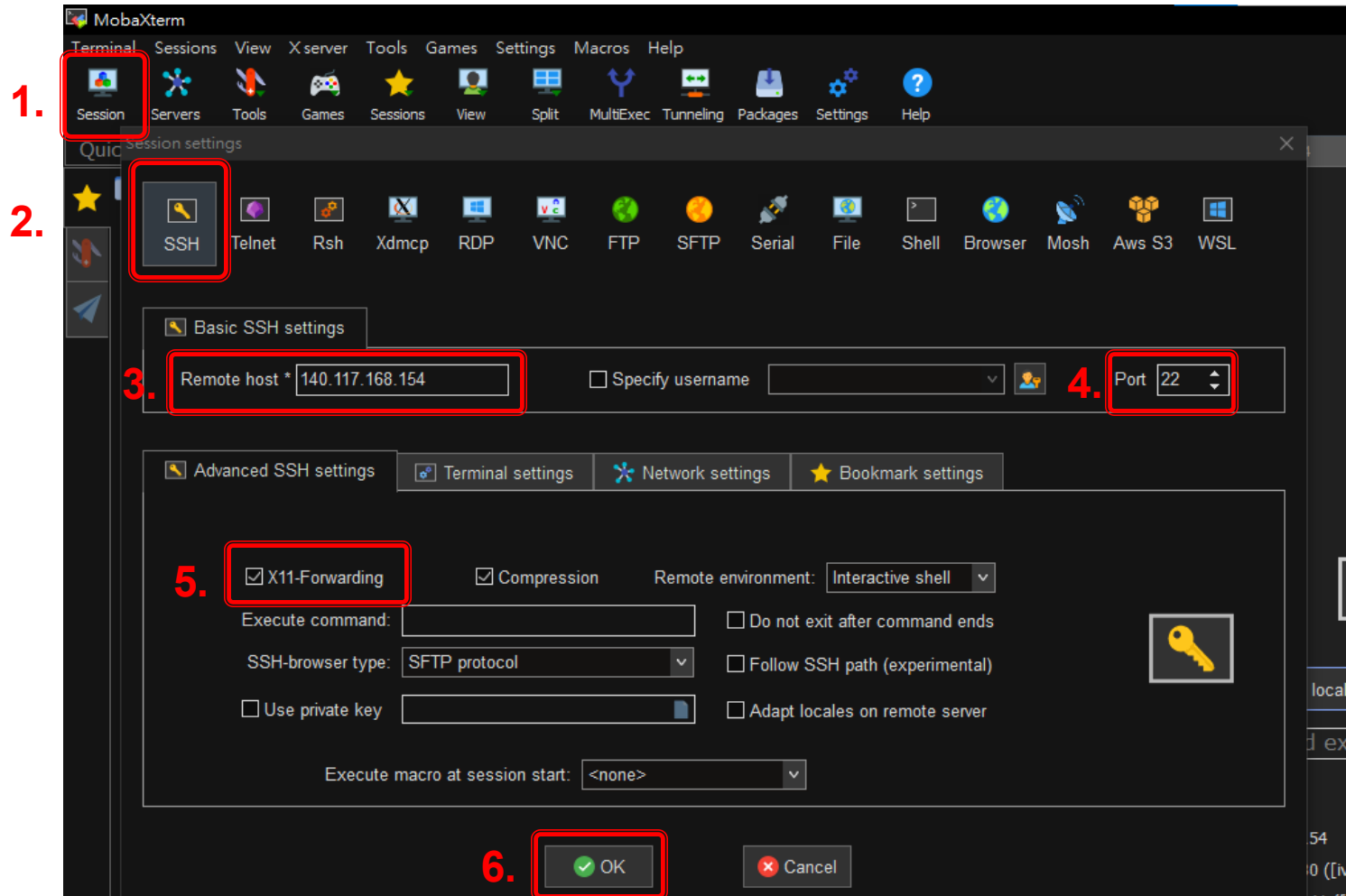
王敬翔 [hssiang@g-mail.nsysu.edu.tw](mailto:hssiang@g-mail.nsysu.edu.tw)

2025 ALU / HDL

# X Tools

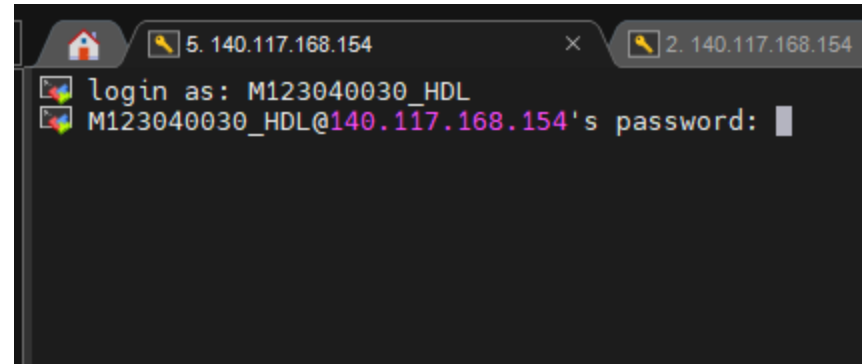
- 請使用遠端程式 MobaXterm 連入伺服器  
(以方便使用圖形化介面)
- 下載網址：  
<https://mobaxterm.mobatek.net/download-home-edition.html>

# SSH



# Login

- Username :
  - <Student ID>\_HDL
  - <Student ID>\_ALUEx: M123040033\_HDL
- Password :
  - 2025<Student ID>Ex: 2025M123040033



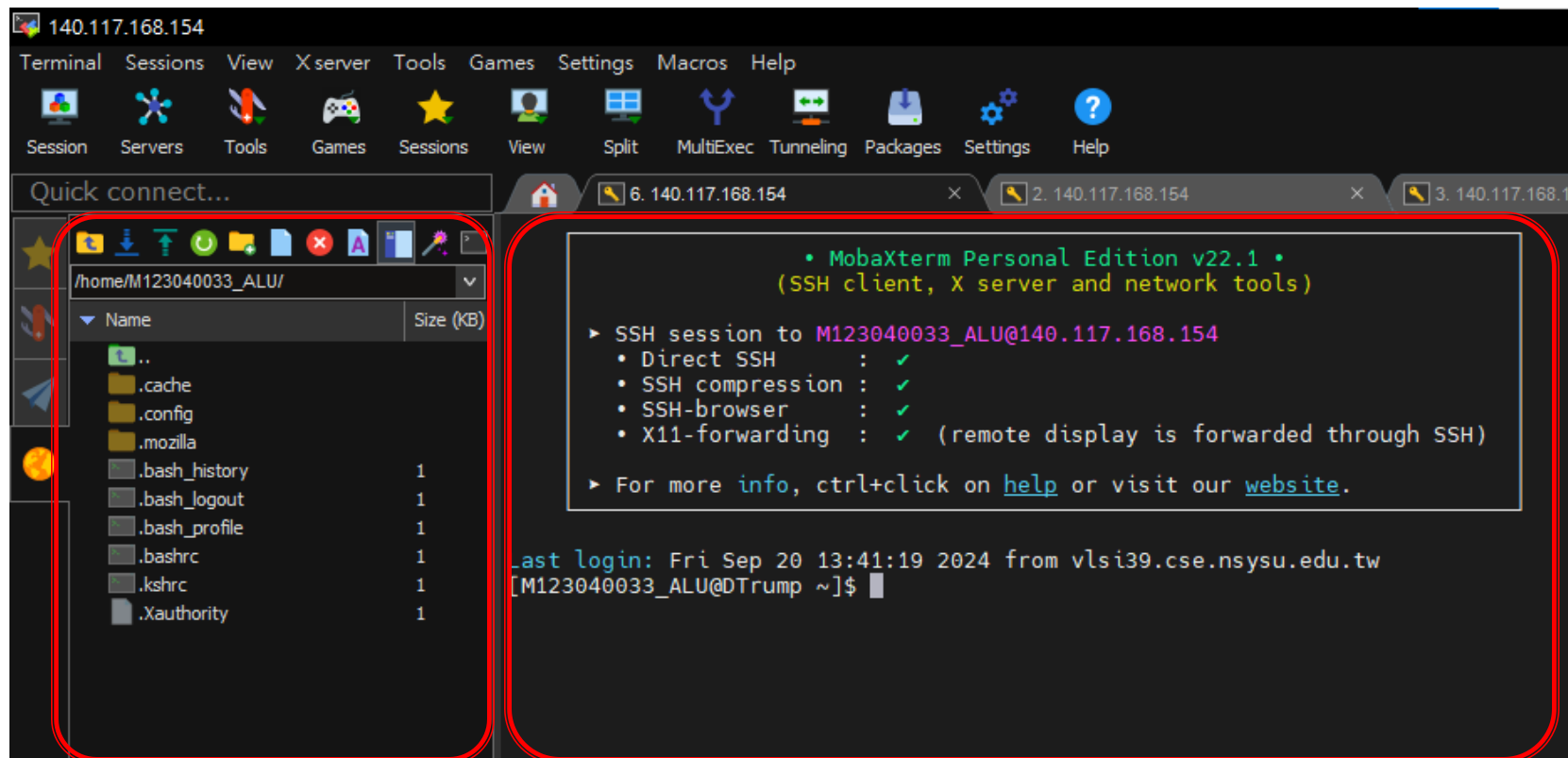
\*輸入密碼時畫面不會顯示任何字元  
看起來像沒反應是正常現象  
可以連續按**back**以重新輸入

# Security

為避免外部惡意攻擊，作業用伺服器有兩層防護：

1. 僅允許校內網域聯入。  
在校外請使用本校 VPN；  
外校生請額外告知。
2. 3分鐘內連線失敗(輸入錯誤)五次，  
將會禁止連線**10**分鐘。

# 登入成功畫面



檔案

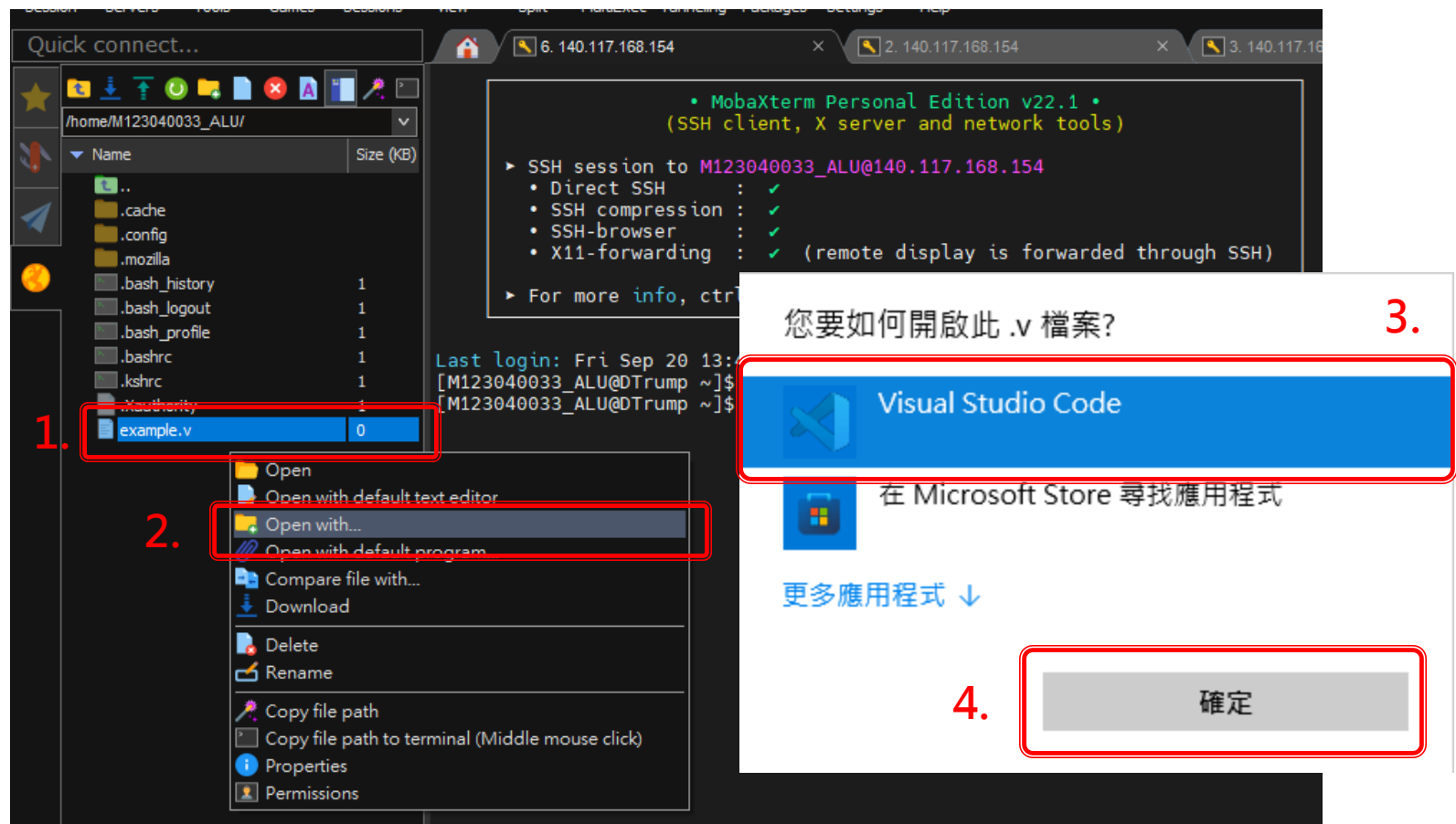
終端機

# 常用指令

- `touch example.v` //新增檔案
- `mv example.v Homework1/.` //移動/修改檔名
- `cp example.v example2.v` //複製檔案
- `passwd` //更改密碼
- 請自行更改密碼，防止他人登入抄襲作業

# 編輯檔案

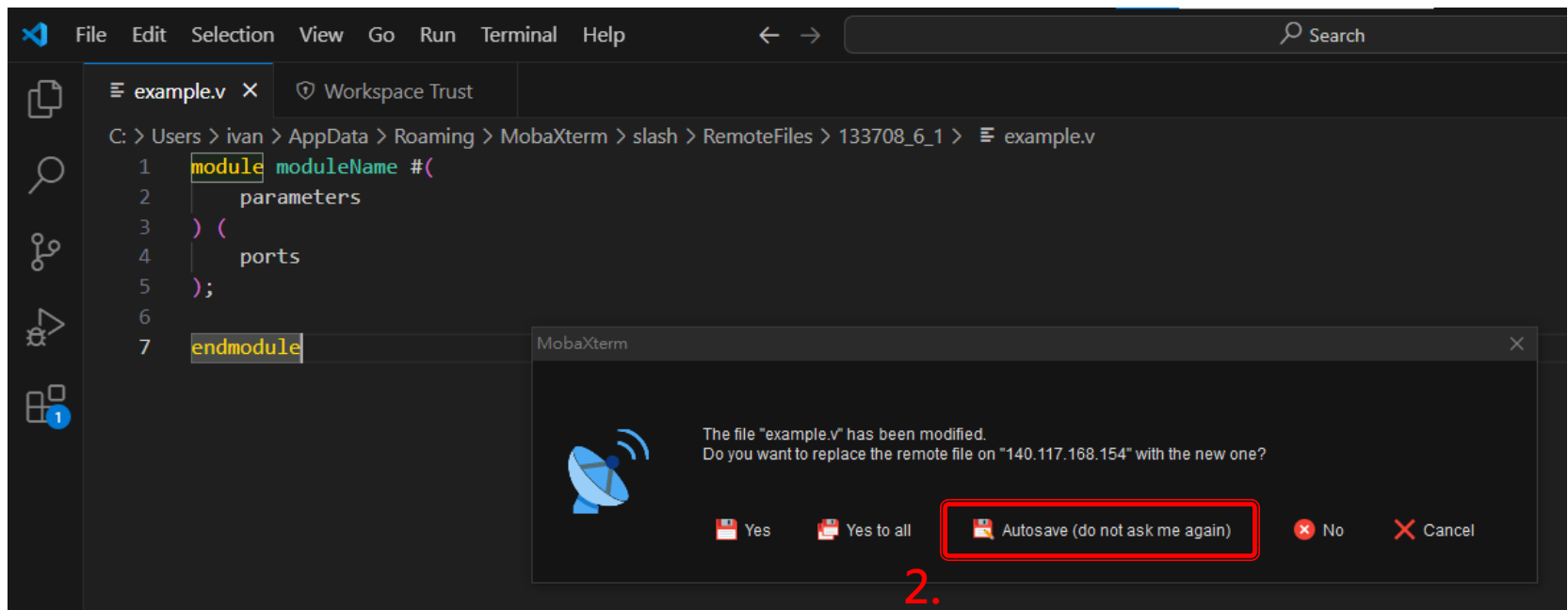
- 可以使用自己電腦上的文字編輯器編寫程式碼





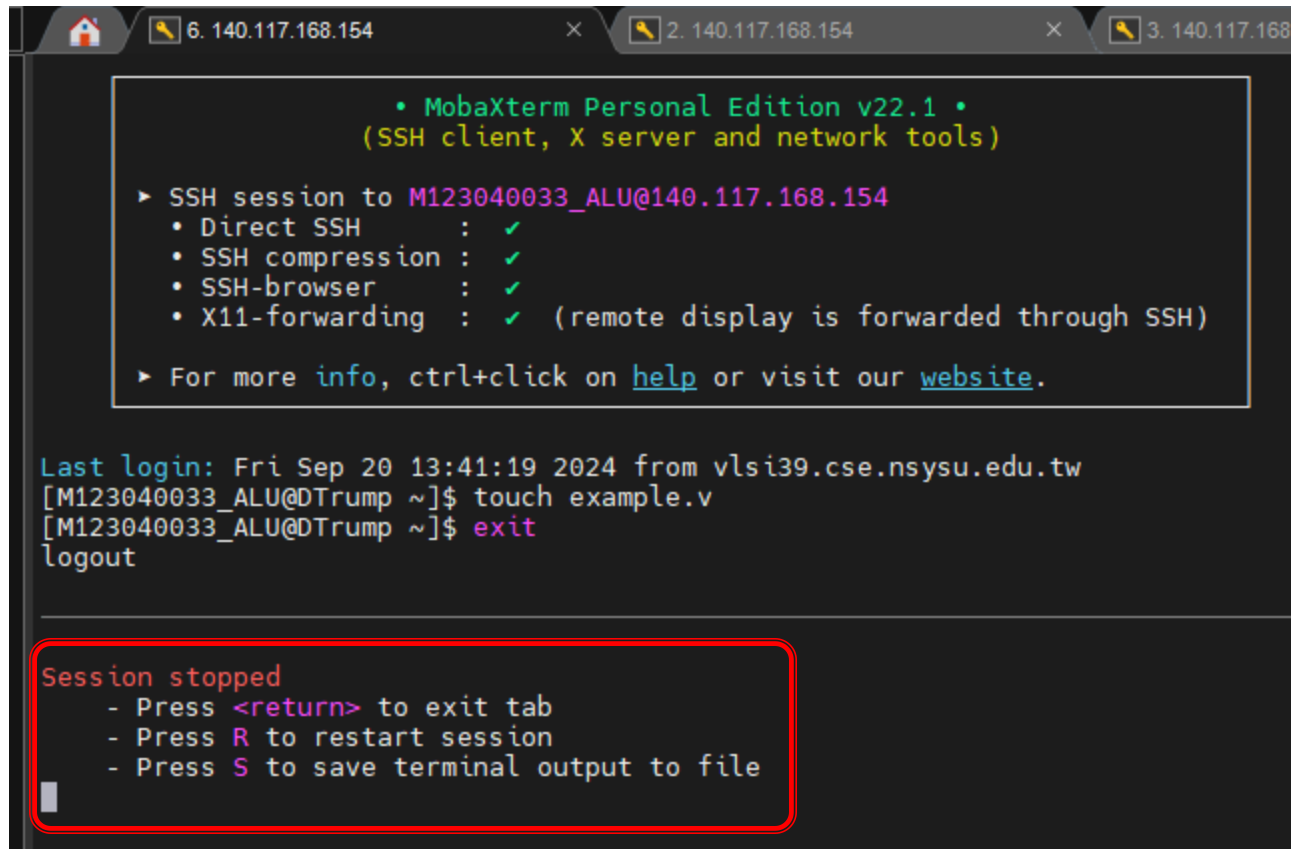
# 儲存檔案

## 1. Ctrl+S



# 關閉連線

- 使用完畢後請務必輸入 `exit` / (Ctrl + D)  
關閉連線以避免佔用伺服器資源



The screenshot shows a MobaXterm window with three tabs. The active tab is titled '6. 140.117.168.154'. Inside the terminal, a message box displays the MobaXterm version and session details. Below this, the terminal shows the user logging in, running 'touch example.v', and then typing 'exit'. The session ends with a 'logout' message. At the bottom, a red-bordered box highlights the 'Session stopped' message and instructions for the user.

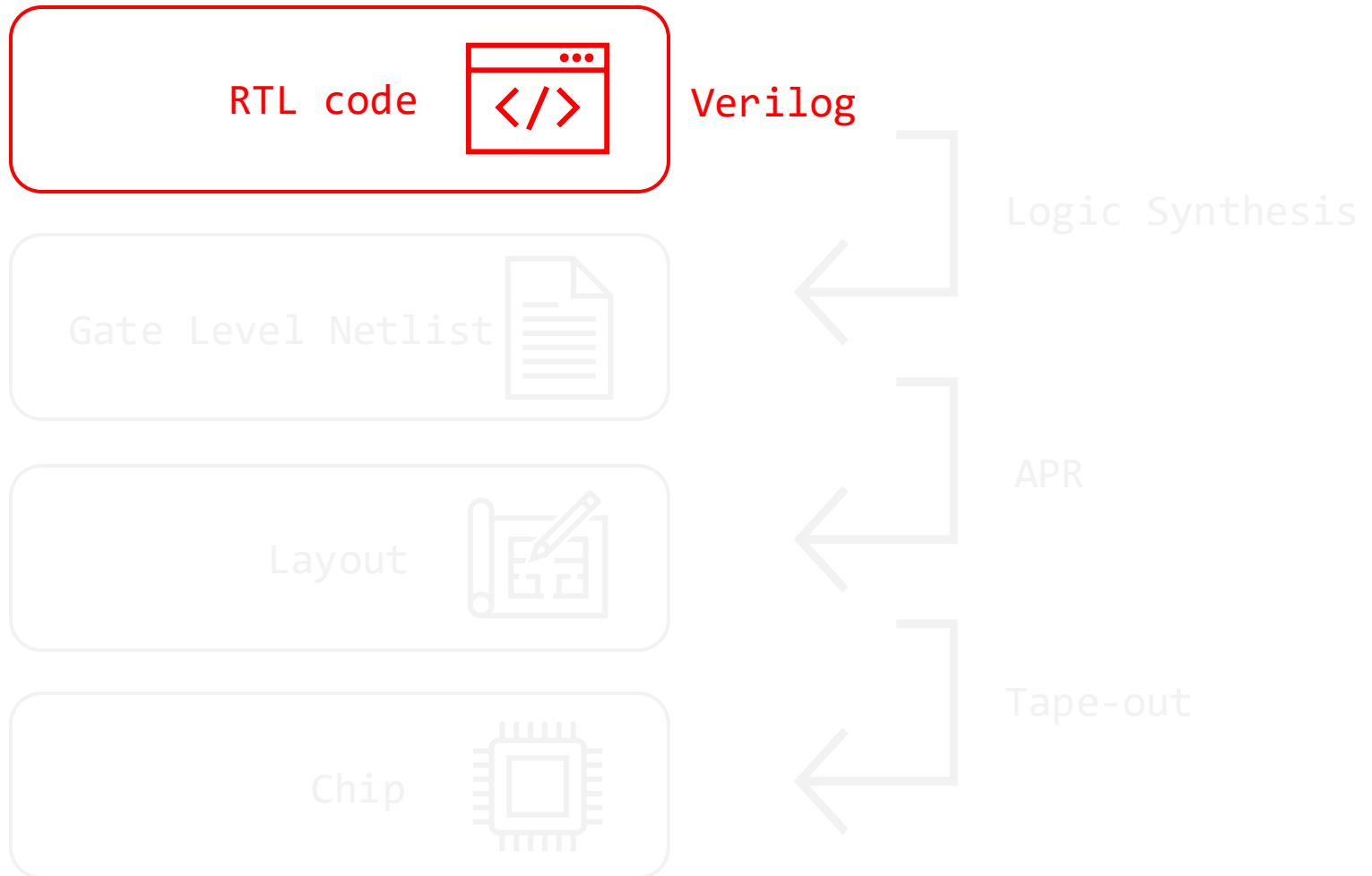
```
• MobaXterm Personal Edition v22.1 •  
(SSH client, X server and network tools)  
  
► SSH session to M123040033_ALU@140.117.168.154  
  • Direct SSH : ✓  
  • SSH compression : ✓  
  • SSH-browser : ✓  
  • X11-forwarding : ✓ (remote display is forwarded through SSH)  
  
► For more info, ctrl+click on help or visit our website.  
  
Last login: Fri Sep 20 13:41:19 2024 from vlsi39.cse.nsysu.edu.tw  
[M123040033_ALU@DTrump ~]$ touch example.v  
[M123040033_ALU@DTrump ~]$ exit  
logout  
  
Session stopped  
- Press <return> to exit tab  
- Press R to restart session  
- Press S to save terminal output to file
```

# Homework1說明

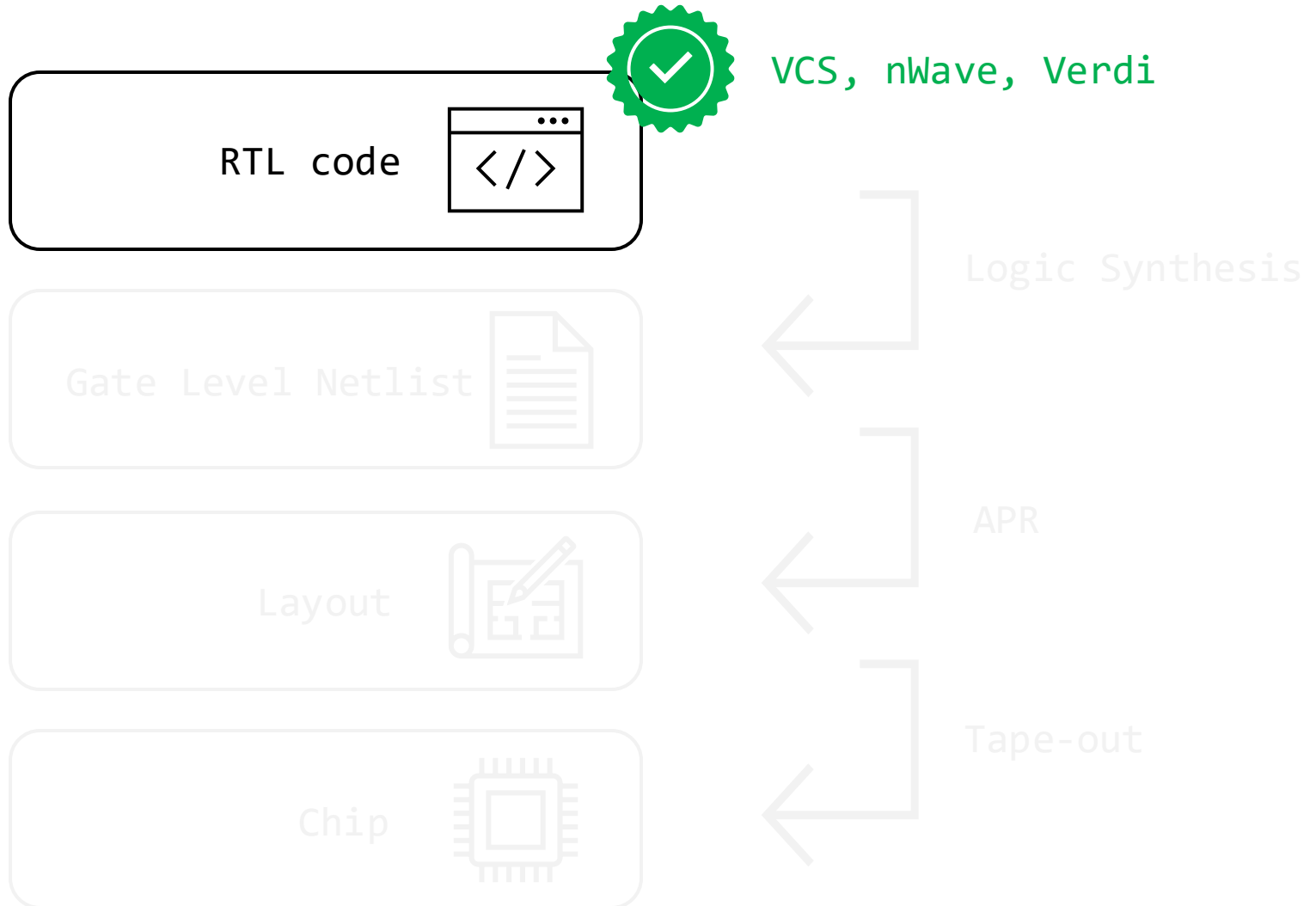
王敬翔 [hssiang@g-mail.nsysu.edu.tw](mailto:hssiang@g-mail.nsysu.edu.tw)

2025 HDL/ALU

# 1. Design Your Circuit by code



## 2. Pre-Simulation



# VCS simulation & nWave

王敬翔 [hssiang@g-mail.nsysu.edu.tw](mailto:hssiang@g-mail.nsysu.edu.tw)

2025 ALU / HDL

# VCS Simulation

- 利用Testbench提供測試訊號
- 模擬電路訊號與運作狀況
- 輸入檔案
  - Testbench
  - Your Design
- 輸出檔案
  - .fsdb Waveform file

# VCS Pre-Simulation

1. 輸入tcsh

2. 編輯pre\_sim.sh

```
[m123040033@DTrump pre_sim]$ tcsh
set vcs version: 2023.12 (default)
set verdi version: 2023.12 (default)
set synthesis version: 2023.12/ (default)
set lc version: 2023.12/ (default)
set formality version: 2023.12/ (default)
set primetime version: 2023.12/ (default)
set INNOVUS version: INNOVUS_21.17.000 (default)
```

testbench

Your Design

Output Waveform

可以修改成自己喜歡的檔名

```
$ sim.sh  X  Workspace Trust
C: > Users > ivan > AppData > Roaming > MobaXterm > slash > RemoteFiles > 133708_2_2 > $ sim.sh
1  #!/bin/tcsh
2
3  vcs -R \
4  /home/m123040033/HDL/pre_sim/TB.v \
5  /home/m123040033/HDL/rtl/adder_behavior_reg.v\
6  /home/m123040033/HDL/rtl/adder_behavior.v\
7  /home/m123040033/HDL/rtl/adder_dataflow_reg.v\
8  /home/m123040033/HDL/rtl/adder_dataflow.v\
9  /home/m123040033/HDL/rtl/adder_structure_reg.v\
10 /home/m123040033/HDL/rtl/adder_structure.v\
11 /home/m123040033/HDL/rtl/D_FF_32.v\
12 /home/m123040033/HDL/rtl/D_FF.v\
13 /home/m123040033/HDL/rtl/FullAdder.v \
14 +full64 \
15 +access+r +vcs+fsdbon +fsdb+mda +fsdbfile+6adder.fsdb
16
```

建議

1. 使用絕對路徑

2. 自己打路徑的內容



# VCS Pre-Simulation

## 3. source pre\_sim.sh

```
[m123040033@Elden pre_sim]$ source sim.sh
** Using c compiler gcc instead of cc ...
    Chronologic VCS (TM)
Version V-2023.12 -- Fri Sep 20 16:06:47 2024
```

Copyright (c) 1991 - 2023 Synopsys, Inc.  
This software and the associated documentation are proprietary to Synopsys, Inc. This software may only be used in accordance with the terms and conditions of a written license agreement with Synopsys, Inc. All other use, reproduction, or distribution of this software is strictly prohibited. Licensed Products communicate with Synopsys servers for the purpose of providing software updates, detecting software piracy and verifying that customers are using Licensed Products in conformity with the applicable License Key for such Licensed Products. Synopsys will use information gathered in connection with this process to deliver software updates and pursue software pirates and infringers.

Inclusivity & Diversity - Visit SolvNetPlus to read the "Synopsys Statement on Inclusivity and Diversity" (Refer to article 000036315 at <https://solvnetplus.synopsys.com>)

```
Parsing design file '/home/m123040033/HDL/pre_sim/TB.v'
Parsing design file '/home/m123040033/HDL/rtl/adder_behavior_reg.v'
Parsing design file '/home/m123040033/HDL/rtl/adder_behavior.v'
Parsing design file '/home/m123040033/HDL/rtl/adder_dataflow_reg.v'
Parsing design file '/home/m123040033/HDL/rtl/adder_dataflow.v'
Parsing design file '/home/m123040033/HDL/rtl/adder_structure_reg.v'
Parsing design file '/home/m123040033/HDL/rtl/adder_structure.v'
Parsing design file '/home/m123040033/HDL/rtl/D_FF_32.v'
Parsing design file '/home/m123040033/HDL/rtl/D_FF.v'
Parsing design file '/home/m123040033/HDL/rtl/FullAdder.v'
```

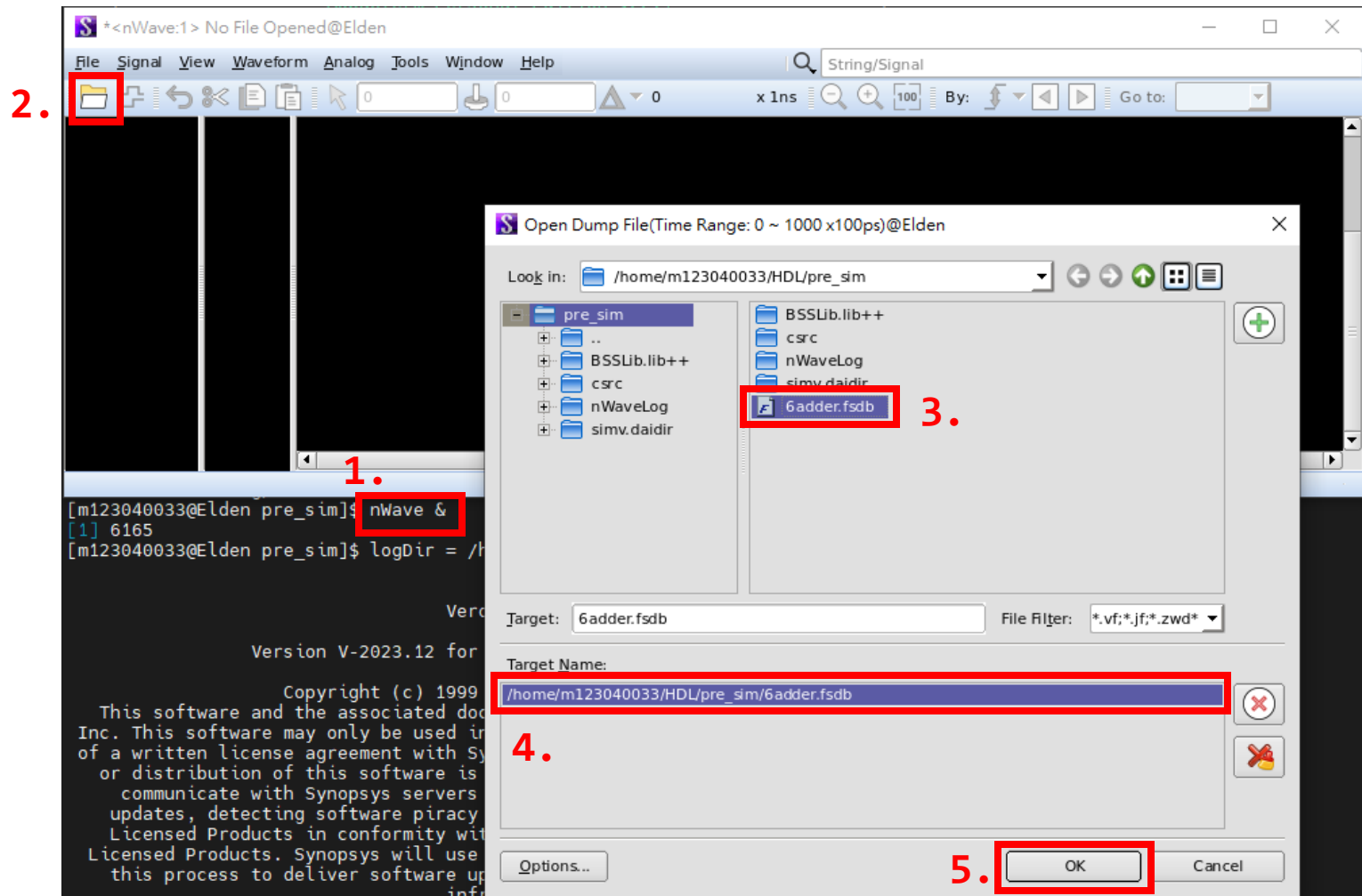
```
Top Level Modules:
  HW1_TB
TimeScale is 1 ns / 100 ps
Starting vcs inline pass...
```

```
3 modules and 0 UDP read.
recompiling module HW1_TB
recompiling module adder_structure
recompiling module FA
All of 3 modules done
```

```
3 modules and 0 UDP read.
recompiling module HW1_TB
recompiling module adder_structure
recompiling module FA
All of 3 modules done
rm -f _cuarc*.so _csrc*.so pre_vcsobj*.so share_vcsobj*.so
if [ -x ../simv ]; then chmod a-x ../simv; fi
g++ -o ../simv -m32 -m32 -rdynamic -Wl,-rpath='$_ORIGIN'/simv.daidir -Wl,-rpath=../simv.daidir -Wl,-rpath=/usr/cad/synopsys/vcs/2023.12/linux/lib -L/usr/cad/synopsys/vcs/2023.12/linux/lib -Wl,-rpath-link=../objs/amcQw_d.o _4401_archive_1.so _prev_archive_1.so SIM_l.o rmapats_mop.o rmapats.o rmar.o rmar_nd.o rmar_llvm_0_1.o rmar_llvm_0_0.o -lvrsim -lerrorinf -lsnpsmalloc -lvfs -lvcsnew -lsimprofile -luclinaive /usr/cad/synopsys/vcs/2023.12/linux/lib/vcs_tls.o -Wl,-whole-archive -lvcsucli -Wl,-no-whole-archive _vcs_pli_stub.o /usr/cad/synopsys/vcs/2023.12/linux/lib/vcs_save_restore_new.o /usr/cad/synopsys/verdi/2023.12/share/PLI/VCS/LINUX/pli.a /usr/cad/synopsys/vcs/2023.12/linux/lib/ctype-stubs_32.a -ldl -lc -lm -lpthread -ldl
../simv up to date
Info: [VCS_SAVE_RESTORE_INFO] ASLR (Address Space Layout Randomization) is detected on the machine. To enable $save functionality, ASLR will be switched off and simv re-executed. Please use '-no_save' simv switch to avoid re-execution or '-suppress=ASLR_DETECTED_INFO' to suppress this message.
Chronologic VCS simulator copyright 1991-2023
Contains Synopsys proprietary information.
Compiler version V-2023.12; Runtime version V-2023.12; Sep 20 16:07 2024
*Verdi* Loading libscore_vcs202312.so
FSDB Dumper for VCS, Release Verdi_V-2023.12, Linux, 11/15/2023
(C) 1996 - 2023 by Synopsys, Inc.
*Verdi* FSDB WARNING: The FSDB file already exists. Overwriting the FSDB file may crash the programs that are using this file.
*Verdi* : Create FSDB file '6adder.fsdb'
*Verdi* : Begin traversing the scopes, layer (0).
*Verdi* : Enable +mda dumping.
*Verdi* : End of traversing.
$finish called from file "/home/m123040033/HDL/pre_sim/TB.v", line 35.
$finish at simulation time 1000
VCS Simulation Report
Time: 100000 ps
CPU Time: 0.660 seconds; Data structure size: 0.0Mb
Fri Sep 20 16:07:10 2024
CPU time: .558 seconds to compile + .499 seconds to elab + .412 seconds to link + .722 seconds in simulation
```

# nWave – Select File

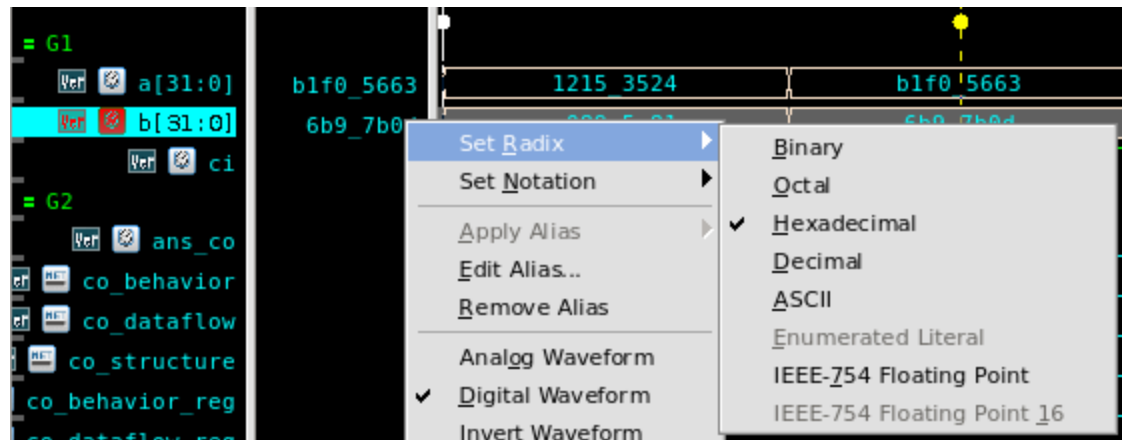
- nWave & //&是在背景執行





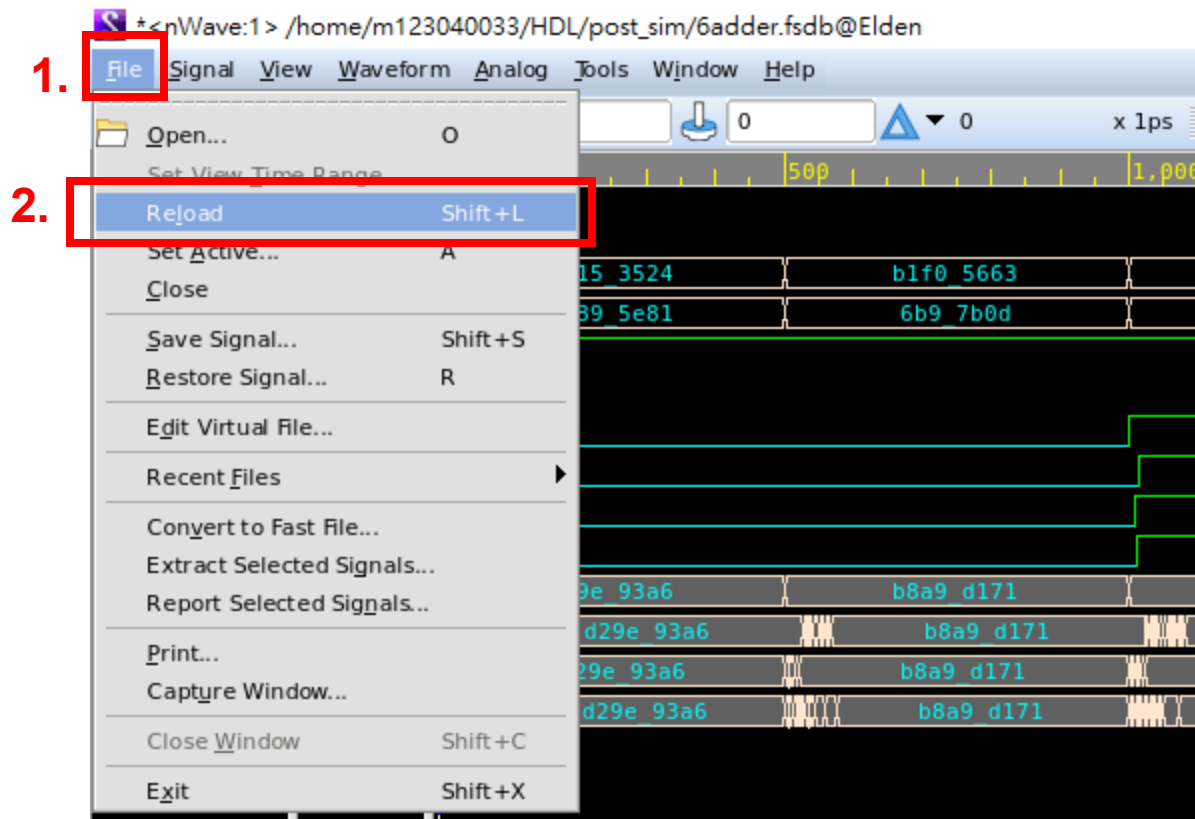
# nWave - 常用操作

- 縮放 : Ctrl + 滑鼠滾輪
- 調整信號位置 : 按住滑鼠中鍵拖曳
- 調整數值表示 : 右鍵 > Set Radix



# nWave - Reload

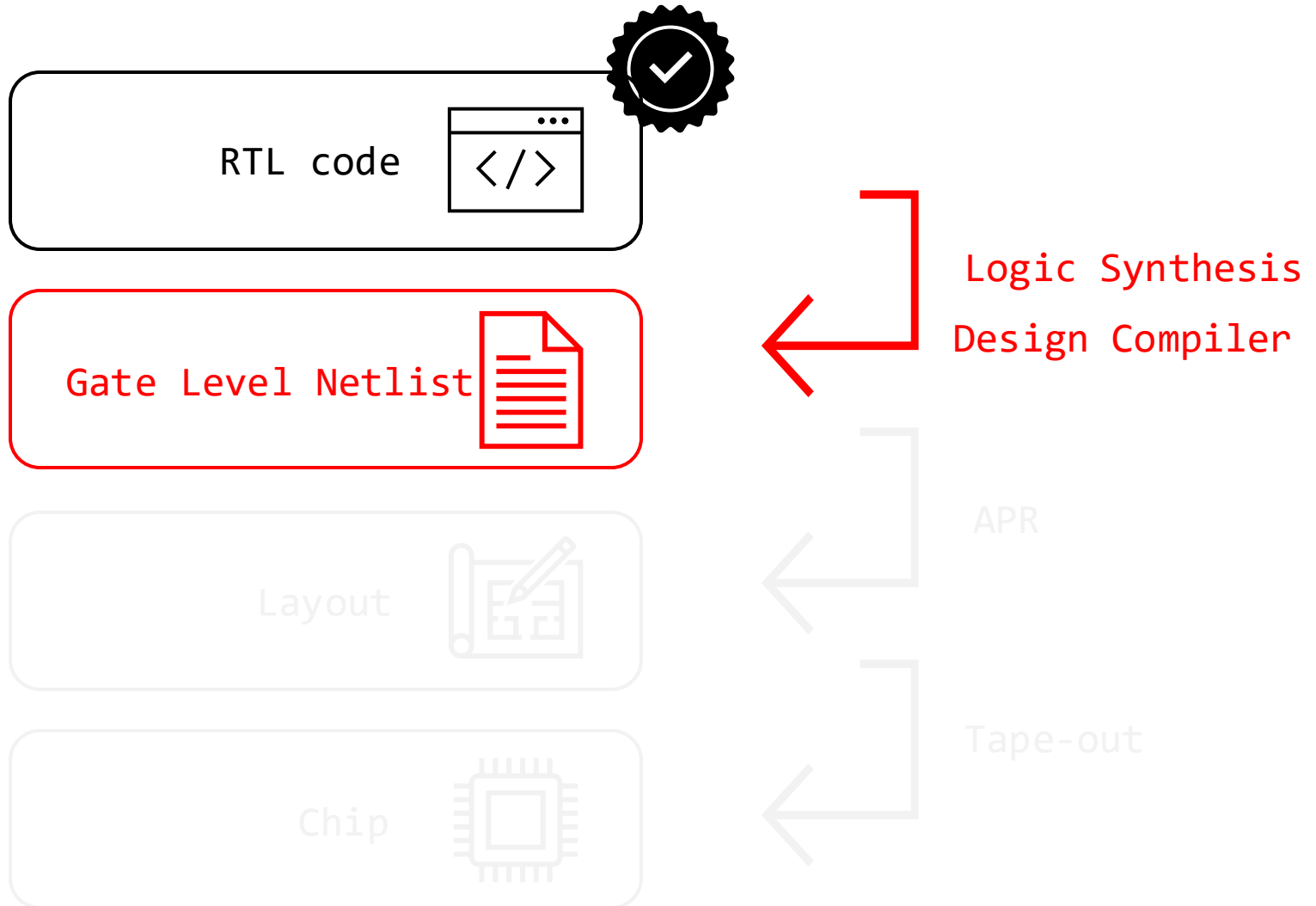
- 重新執行 VCS 模擬後，不必重開nWave和fsdb



# Verdi

- Debug Tool，有需要的同學請自行到 YouTube 觀看。
- <https://youtu.be/wTpzvZs4xZU>

# 3. Logic Synthesis



# Design Compiler

王敬翔 [hssiang@g-mail.nsysu.edu.tw](mailto:hssiang@g-mail.nsysu.edu.tw)

2025 ALU / HDL



# 所需檔案

- RTL code (你的程式碼, 不包含testbench)
- 執行Design compiler之腳本 (作業檔案提供)
- TSMC ADFP Cell Library File (伺服器內部)
- EDA Tools (伺服器內部)

# 腳本說明

- dc.tcl

```
4 #設定ADFP(16nm)製程路徑
5 set search_path      "/cad/CBDK/ADFP/Executable_Package/Collaterals/IP/stdcell/N16ADFP_StdCell/CCS/ $search_path"
6 #設定ADFP(16nm)製程路徑檔，如果有memory compiler的檔案db檔的路徑，記得在這邊設定
7 set target_library   "N16ADFP_StdCellss0p72vm40c_ccs.db N16ADFP_StdCellff0p88v125c_ccs.db"
8
9 set link_library      "* $target_library dw_foundation.sldb"
10 set symbol_library   "tsmc040.sdb generic.sdb"
11 set synthetic_library "dw_foundation.sldb"
12 set hdlin_translate_off_skip_text "TRUE"
13 set edifout_netlist_only "TRUE"
14 set verilogout_no_tri true
15 set hdlin_enable_presto_for_vhdl "TRUE"
16 set sh_enable_line_editing true
17 set sh_line_editing_mode emacs
18 history keep 100
19 alias h history
```

# 腳本說明

- 修改檔案存放位置、top module name及clock

```
21 #Path_Top:Verilog放置的位置
22 #Path_Syn:合成後report.txt檔案要放置的根位置，需自行在目錄下創建名為dc_out_file之資料夾
23 #Dump_file_name:合成後產生檔案之名字
24 set Path_Top      "."
25 set Path_Syn      "../structure_reg/"
26 set Dump_file_name "adder_structure_reg"
27 #設定Top module 名稱，需跟自行設計之電路的top module name相同
28 set Top           "adder_structure_reg"
29 #Specify Clock，clock名需和top module中clk port相同
30 set Clk_pin       "clk"
31 set Clk_period    "0.49"
```

- 將所有要合成的檔案之完整路徑放入 {} 內

```
33 # [ Read Design File ]
34 #如果設計有parameter設計，read_file指定不能用，需使用analyze + elaborate指令並自行更改路徑
35 # read_file -format verilog {/home/m123040049/HDL_HW/multiplier.v}
36 # current_design $Top
37 analyze -format verilog {
38     /home/m123040033/HDL/rtl/adder_structure_reg.v
39     /home/m123040033/HDL/rtl/D_FF.v
40     /home/m123040033/HDL/rtl/D_FF_32.v
41     /home/m123040033/HDL/rtl/FullAdder.v
42 }
43 elaborate $Top
44 current_design $Top
45 #檢查是否讀取成功
46 link
```

# 腳本說明

- 作業要求合成 *delay-optimization*, *area-optimization*, and *in-between* 之三種電路，需修改31行中的period

```
21  #Path_Top:Verilog放置的位置
22  #Path_Syn:合成後report.txt檔案要放置的根位置，需自行在目錄下創建名為dc_out_file之資料夾
23  #Dump_file_name:合成後產生檔案之名字
24  set Path_Top      "/"
25  set Path_Syn      "../structure_reg/"
26  set Dump_file_name "adder_structure_reg"
27  #設定Top module 名稱，需跟自行設計之電路的top module name相同
28  set Top           "adder_structure_reg"
29  #Specify Clock, clock名需和top module中clk port相同
30  set Clk_pin       "clk"
31  set Clk_period    "0.49"
```

# 腳本說明

- 作業要求需合成 *delay-optimization*, *area-optimization*, and *in-between* 之三種電路，需修改31行中的period
  - 先將period設一個很大的值(假設10)進行合成，合成後report timing 發現最快可以跑2，則將period改設2再合一次，如果slack  $\geq 0$  (MET) 則 22 即為 *area-optimization* 之period
  - 接著改設period為一個很小的值(假設0)進行合成，合成後report timing發現slack = -0.5 (Violate)，則將period改設0.5再合一次，如果slack  $\geq 0$  (MET)則0.5即為 *delay-optimization* 之period，讓 slack 盡可能越接近0
  - 最後設period為 $(2+0.5)/2=1.25$ 進行合成，假設slack = -0.1(Violate)則設1.35再合一次，直到slack  $\geq 0$ (MET)，最終period值即為*in-between* 電路之period

# 腳本說明

- 如果電路中無clk設計，使用50、51行，註解54~57行

```
49 # [ Setting Clock Constraints, Combinational Circurt USED ]
50 set_max_delay $Clk_period -from [all_inputs] -to [all_outputs]
51 create_clock -name $Clk_pin -period $Clk_period
52
53 # [ Setting Clock Constraints ]
54 # create_clock -name $Clk_pin -period $Clk_period [get_ports $Clk_pin]
55 # set_fix_hold [get_clocks $Clk_pin]
56 # set_dont_touch_network [get_clocks $Clk_pin]
57 # set_ideal_network [get_ports $Clk_pin]
```

- 如果電路中有clk設計，使用54~57行，註解50、51行

```
49 # [ Setting Clock Constraints, Combinational Circurt USED ]
50 # set_max_delay $Clk_period -from [all_inputs] -to [all_outputs]
51 # create_clock -name $Clk_pin -period $Clk_period
52
53 # [ Setting Clock Constraints ]
54 create_clock -name $Clk_pin -period $Clk_period [get_ports $Clk_pin]
55 set_fix_hold [get_clocks $Clk_pin]
56 set_dont_touch_network [get_clocks $Clk_pin]
57 set_ideal_network [get_ports $Clk_pin]
```

# 腳本說明

- 設定合成時命名規則(不須修改)
- 輸出timing、area和power report，輸出路徑可至25行更改
- 產生合成後.v檔、ddc、sdf和sdc 檔，產生之名稱可至26行更改

```
77 #Change Naming Rule
78 set bus_inference_style {%s[%d]}
79 set bus_naming_style {%s[%d]}
80 set hdlout_internal_busses true
81 change_names -hierarchy -rule verilog
82 define_name_rules name_rule -allowed "A-Z a-z 0-9 _" -max_length 255 -type cell
83 define_name_rules name_rule -allowed "A-Z a-z 0-9 _[]" -max_length 255 -type net
84 define_name_rules name_rule -map {"\\*cell\\*" "cell"}
85 define_name_rules name_rule -case_insensitive
86 change_names -hierarchy -rules name_rule
87 remove_unconnected_ports -blast_buses [get_cells -hierarchical *]
88
89 report_timing -significant_digits 6 -sort_by group
90
91 #Report
92 report_timing -path full -delay max -significant_digits 6 -sort_by group > $Path_Syn/1.timing_report_${Dump_file_name}.txt
93 report_area -hier -nosplit > $Path_Syn/2.area_report_${Dump_file_name}.txt
94 report_power -analysis_effort low > $Path_Syn/3.power_report_${Dump_file_name}.txt
95
96 #Write out
97 write -hierarchy -format ddc -output $Path_Syn/${Dump_file_name}.ddc
98 write -format verilog -hierarchy -output $Path_Syn/${Dump_file_name}.v
99 write_sdf -version 2.1 -context verilog $Path_Syn/${Dump_file_name}.sdf
100 write_sdc $Path_Syn/${Dump_file_name}.sdc
```

# 使用Design Compiler合成

- 輸入`dcnxt_shell -f dc.tcl`，開始執行tcl檔、進行合成

```
[m103040049@devotion ~/ALU_HW1]$ dc_shell -f dc.tcl

Design Compiler Graphical
DC Ultra (TM)
DFTMAX (TM)
Power Compiler (TM)
DesignWare (R)
DC Expert (TM)
Design Vision (TM)
HDL Compiler (TM)
VHDL Compiler (TM)
DFT Compiler
Design Compiler(R)

Version 0-2018.06 for linux64 - May 21, 2018

Copyright (c) 1988 - 2018 Synopsys, Inc.
This software and the associated documentation are proprietary to Synopsys,
Inc. This software may only be used in accordance with the terms and conditions
of a written license agreement with Synopsys, Inc. All other use, reproduction,
or distribution of this software is strictly prohibited.
Initializing...
set Company "VLSI5015"
VLSI5015
set Designer "default"
default
#設定40nm製程路徑
set search_path "/EDA_Tools/CBDK/CBDK TSMC40_Arm_f2.0/CIC/SynopsysDC/db/sc9_b
```



# Report Timing

|  |       |        |
|--|-------|--------|
| stage2/adder/add_7/U1_23/CO (ADDF_X1M_A9TR)              | 0.08  | 2.07 r |
| stage2/adder/add_7/U1_24/CO (ADDF_X1M_A9TR)              | 0.08  | 2.15 r |
| stage2/adder/add_7/U1_25/CO (ADDF_X1M_A9TR)              | 0.08  | 2.24 r |
| stage2/adder/add_7/U1_26/CO (ADDF_X1M_A9TR)              | 0.08  | 2.32 r |
| stage2/adder/add_7/U1_27/CO (ADDF_X1M_A9TR)              | 0.08  | 2.40 r |
| stage2/adder/add_7/U1_28/S (ADDF_X1M_A9TR)               | 0.18  | 2.58 f |
| stage2/adder/add_7/SUM[28] (adder_DW01_add_0)            | 0.00  | 2.58 f |
| stage2/adder/sign_d (adder)                              | 0.00  | 2.58 f |
| stage2/conversion_sign/sign_d (conversion_sign)          | 0.00  | 2.58 f |
| stage2/conversion_sign/U34/Y (BUFH_X1M_A9TR)             | 0.08  | 2.67 f |
| stage2/conversion_sign/U5/Y (INV_X1M_A9TR)               | 0.03  | 2.70 r |
| stage2/conversion_sign/U3/Y (BUFH_X1M_A9TR)              | 0.08  | 2.78 r |
| stage2/conversion_sign/U15/Y (AO22_X1M_A9TR)             | 0.09  | 2.87 r |
| stage2/conversion_sign/mantissa_out[9] (conversion_sign) | 0.00  | 2.87 r |
| stage2/mantissa_unsigned_reg_9_/D (DFFRPQ_X2M_A9TR)      | 0.00  | 2.87 r |
| data arrival time  |       | 2.87   |
| 表格 Delay 欄位需要填入的部分                                       |       |        |
| clock clk (rise edge)                                    | 3.00  | 3.00   |
| clock network delay (ideal)                              | 0.00  | 3.00   |
| stage2/mantissa_unsigned_reg_9_/CK (DFFRPQ_X2M_A9TR)     | 0.00  | 3.00 r |
| library setup time                                       | -0.05 | 2.95   |
| data required time                                       |       | 2.95   |
| -----  |       |        |
| data required time                                       |       | 2.95   |
| data arrival time  |       | 2.87   |
| 需要 $\geq 0$ 且有 slack (MET)                               |       |        |
| slack (MET)  |       | 0.09   |

# Report Area

```
*****
Report : area
Design : FLP_adder
Version: 0-2018.06
Date   : Sat Oct  8 17:10:59 2022
*****

Library(s) Used:

  sc9_cln40g_base_rvt_ss_typical_max_0p81v_125c (File: /EDA_Tools/CBDK/CBDK_TSMC40_Arm_f2.0/CIC/SynopsysDC/db/sc9_ba

Number of ports:                1545
Number of nets:                 3576
Number of cells:                2029
Number of combinational cells:  1834
Number of sequential cells:     172
Number of macros/black boxes:   0
Number of buf/inv:              409
Number of references:           4

Combinational area: 2369.833178
Buf/Inv area: 218.635199
Noncombinational area: 780.191961
Macro/Black Box area: 0.000000
Net Interconnect area: undefined (Wire load has zero net area)

Total cell area: 3150.025139
Total area: undefined

Hierarchical area distribution
-----
```

Combinational Logic ( 表格 CL 欄位 )

Sequential Logic ( 表格 SL 欄位 )

Total Area ( 表格 Total 欄位 )

# Report Power

Design

Wire Load Model

Library

-----

FLP\_adder

Zero

sc9\_cln40g\_base\_rvt\_ss\_typical\_max\_0p81v\_125c

Global Operating Voltage = 0.81

Power-specific unit information :

Voltage Units = 1V

Capacitance Units = 1.000000pf

Time Units = 1ns

Dynamic Power Units = 1mW (derived from V,C,T units)

Leakage Power Units = 1uW

Cell Internal Power = 287.2819 uW (86%)

Net Switching Power = 46.9000 uW (14%)

-----

Total Dynamic Power = 334.1819 uW (100%)

Cell Leakage Power = 29.8890 uW

請轉換單位: uW → W (表格要求)

Dynamic Power ( 表格 Dynamic 欄位 )

Leakage Power ( 表格 Leakage 欄位 )

Dynamic + Leakage = Total ( 表格 Total 欄位 )

| Power Group   | Internal Power | Switching Power | Leakage Power | Total Power ( % ) | Attrs |
|---------------|----------------|-----------------|---------------|-------------------|-------|
| io_pad        | 0.0000         | 0.0000          | 0.0000        | 0.0000 ( 0.00%)   |       |
| memory        | 0.0000         | 0.0000          | 0.0000        | 0.0000 ( 0.00%)   |       |
| black_box     | 0.0000         | 0.0000          | 0.0000        | 0.0000 ( 0.00%)   |       |
| clock_network | 0.0000         | 0.0000          | 0.0000        | 0.0000 ( 0.00%)   |       |
| register      | 0.2055         | 4.9244e-03      | 7.7363        | 0.2182 ( 59.93%)  |       |
| sequential    | 0.0000         | 0.0000          | 0.0000        | 0.0000 ( 0.00%)   |       |
| combinational | 8.1768e-02     | 4.1976e-02      | 22.1417       | 0.1459 ( 40.07%)  |       |
| -----         |                |                 |               |                   |       |
| Total         | 0.2873 mW      | 4.6900e-02 mW   | 29.8780 uW    | 0.3641 mW         |       |

請轉換單位: uW → W (表格要求)

Dynamic Power ( 表格 Dynamic 欄位 )

Leakage Power ( 表格 Leakage 欄位 )

Dynamic + Leakage = Total ( 表格 Total 欄位 )

# VCS Post-Simulation

```
1  #!/bin/tcsh
2
3  vcs -R -error=noMPD \
4  /home/m123040033/HDL/post_sim/TB.v \
5  /home/m123040033/HDL/behavior/adder_behavior.v \
6  /home/m123040033/HDL/behavior_reg/adder_behavior_reg.v \
7  /home/m123040033/HDL/dataflow/adder_dataflow.v \
8  /home/m123040033/HDL/dataflow_reg/adder_dataflow_reg.v \
9  /home/m123040033/HDL/structure/adder_structure.v \
10 /home/m123040033/HDL/structure_reg/adder_structure_reg.v \
11 /cad/CBDK/ADFP/Executable_Package/Collaterals/IP/stdcell/N16ADFP_StdCell/VERILOG/N16ADFP_StdCell.v \
12 +full64 \
13 +access+r +vcs+fsdbon +fsdb+mda +fsdbfile+adder.fsdb +neg_tchk
```

testbench

Gate  
Level  
Netlist

Standard Cell

Output Waveform  
可以修改成自己喜歡的檔名

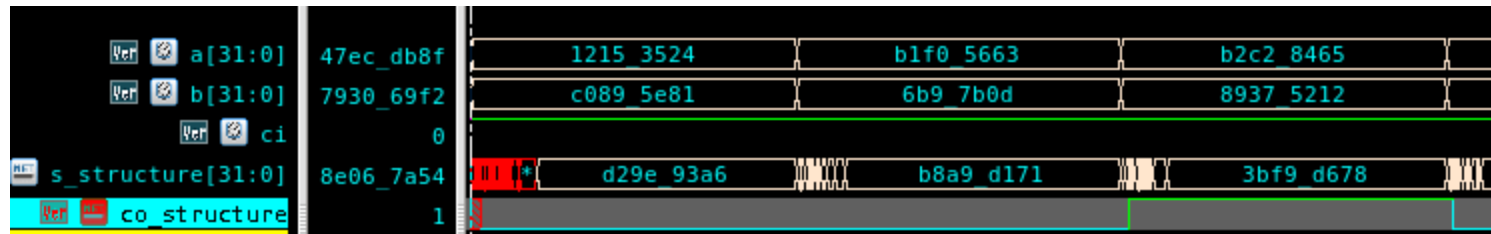
# VCS Post-Simulation - TB

需要在 TB 內加入 Standard Delay Format (SDF)

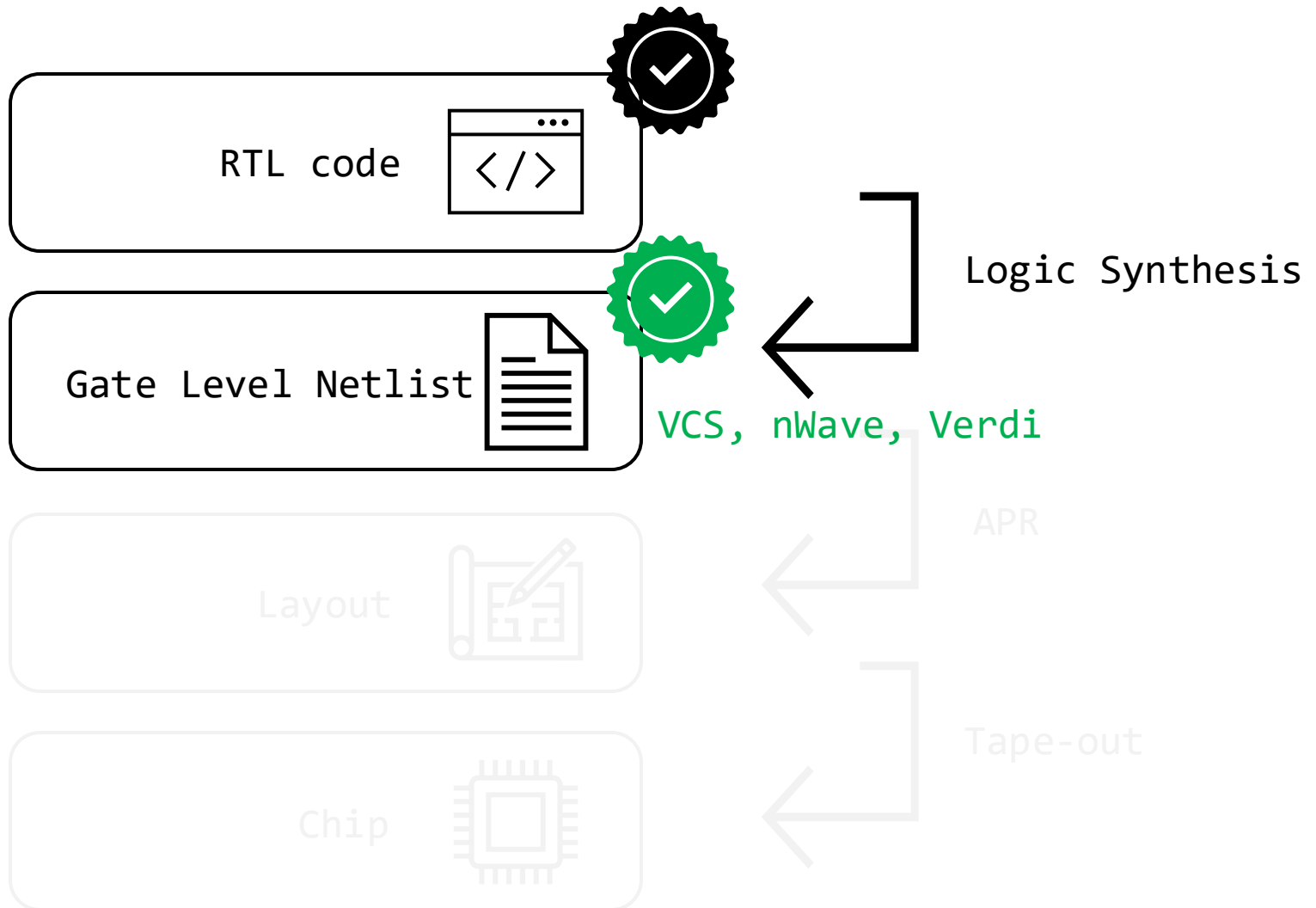
```
adder_behavior behavior(s_behavior, co_behavior, a, b, ci);
adder_behavior_reg behavior_reg(s_behavior_reg, co_behavior_reg, a, b, ci, clk);
adder_dataflow dataflow(s_dataflow, co_dataflow, a, b, ci);
adder_dataflow_reg dataflow_reg(s_dataflow_reg, co_dataflow_reg, a, b, ci, clk);
adder_structure structure(s_structure, co_structure, a, b, ci);
adder_structure_reg structure_reg(s_structure_reg, co_structure_reg, a, b, ci, clk);
```

**SDF File path** 這邊要和呼叫的 **Module Instance name** 一樣

```
initial $sdf_annotate("/home/m123040033/HDL/behavior/adder_behavior.sdf", behavior);
initial $sdf_annotate("/home/m123040033/HDL/behavior_reg/adder_behavior_reg.sdf", behavior_reg);
initial $sdf_annotate("/home/m123040033/HDL/dataflow/adder_dataflow.sdf", dataflow);
initial $sdf_annotate("/home/m123040033/HDL/dataflow_reg/adder_dataflow_reg.sdf", dataflow_reg);
initial $sdf_annotate("/home/m123040033/HDL/structure/adder_structure.sdf", structure);
initial $sdf_annotate("/home/m123040033/HDL/structure_reg/adder_structure_reg.sdf", structure_reg);
```



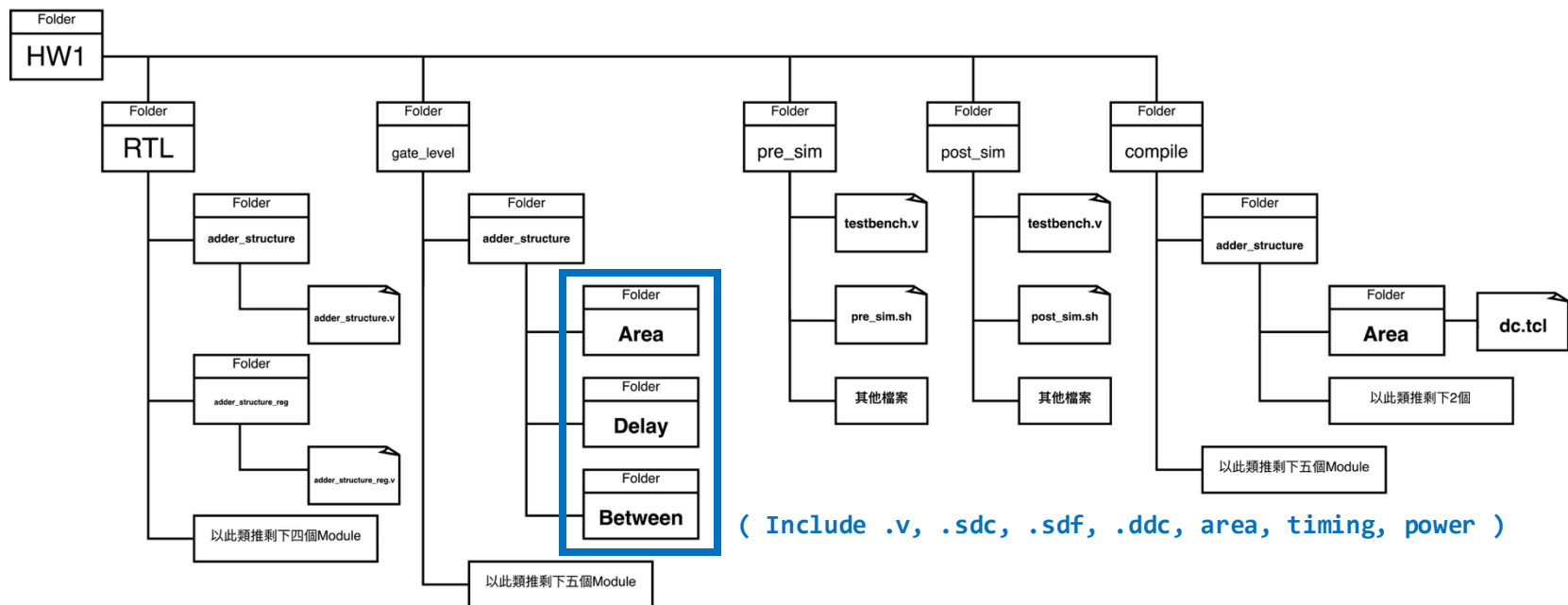
# 4. Post-Simulation



# HW1 流程

1. Write Verilog Code
2. Write Testbench
3. Run Pre Synthesis Simulation - VCS
4. Snapshot Waveform (Pre-sim) - nWave
5. Logic Synthesis - Design Compiler
6. Run Post Synthesis Simulation - VCS
7. Snapshot Waveform (Post-sim) - nWave
8. Complete Report

# HW1 目錄建立方式





# Report.pdf

- 模擬波形圖 (只需delay optimize, 不用其他optimize)
  - RTL level
  - Gate level
- 數據表格
- 觀察三種modeling之電路的數據/波型是否相同。  
並解釋你認為的原因
- 心得

# Report.pdf

- 數據表格

|                     |         | Area ( $\mu\text{m}^2$ ) |    |       | Delay<br>(ns) | Power (W) |         |       |
|---------------------|---------|--------------------------|----|-------|---------------|-----------|---------|-------|
|                     |         | CL                       | SL | Total |               | dynamic   | leakage | total |
| adder_structure     | delay   |                          |    |       |               |           |         |       |
|                     | area    |                          |    |       |               |           |         |       |
|                     | between |                          |    |       |               |           |         |       |
| adder_structure_reg | delay   |                          |    |       |               |           |         |       |
|                     | area    |                          |    |       |               |           |         |       |
|                     | between |                          |    |       |               |           |         |       |
| adder_dataflow      | delay   |                          |    |       |               |           |         |       |
|                     | area    |                          |    |       |               |           |         |       |
|                     | between |                          |    |       |               |           |         |       |
| adder_dataflow_reg  | delay   |                          |    |       |               |           |         |       |
|                     | area    |                          |    |       |               |           |         |       |
|                     | between |                          |    |       |               |           |         |       |
| adder_behavior      | delay   |                          |    |       |               |           |         |       |
|                     | area    |                          |    |       |               |           |         |       |
|                     | between |                          |    |       |               |           |         |       |
| adder_behavior_reg  | delay   |                          |    |       |               |           |         |       |
|                     | area    |                          |    |       |               |           |         |       |
|                     | between |                          |    |       |               |           |         |       |



# 繳交檔案

- 直接複製整個HW1資料夾到Server內繳交

- `cp -r /home/M143040045_HDL/HW1/ /MasterClass/Homework-Submit/M143040045_HDL/`

- 請替換 M143040045\_HDL 成自己的學號

- 上傳加入封面的Report.pdf到網大作業區

- Due:10/13 23:59

# 評分說明

- (15%) Testbench
  - 10% Testbench
  - 5% 需使用 `$random` 產生 10 筆亂數輸出
- (60%) Verilog RTL codes and Synthesized gate-level codes
  - 10% adder\_structure, 10% adder\_structure\_reg
  - 10% adder\_dataflow , 10% adder\_dataflow\_reg
  - 10% adder\_behavior , 10% adder\_behavior\_reg
- (25%) Report (\*只收PDF)