# Homework tips

# Homework / Lab 流程

- HW跟Lab主要分成4個步驟: (可能有幾次Lab會不同)
  - (1) RTL simulation
  - (2) Synthesis
  - (3) Gate-level simulation
  - (4) upload
- 分別對應到每次作業的五個資料夾的其中四個
- LabXX/HWXX
  - 00_TESTBED
  - 01_RTL
  - 02_SYN
  - 03_GATE
  - 09_UPLOAD

# Directory tree structure

- LabXX/HWXX
  - 00_TESTBED
    - PATTERN.sv
    - TESTBENCH.sv
  - 01_RTL
    - 01_run
    - 09_clean_up
    - **XXX.sv**
  - 02_SYN
    - 01_run_dc
    - 09_clean_up
    - Netlist – XXX_SYN.v
    - Report
  - 03_GATE
    - 01_run
    - 09_clean_up
  - 09_UPLOAD
    - 01_upload
    - 02_download

# 00_TESTBED

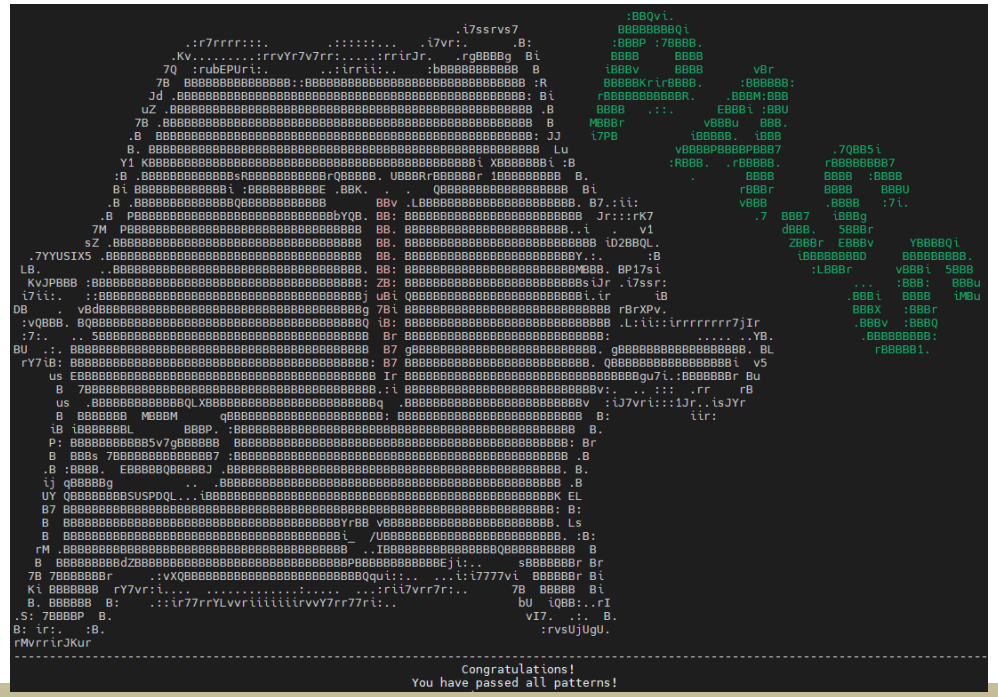
- 擺放驗證流程與測試資料的資料夾。(TESTBENCH.sv / Pattern.sv / input.txt / output.txt …)
- 大部分的Lab跟前三次的HW助教會幫你把Pattern寫好，
  提供同學驗證自己的Design。
- 如果題目有需求，則助教不會提供Pattern，
  Pattern將會由同學自行完成，以驗證自己的Design。
- 所有HW/Lab中，你的Design都必須通過所有Pattern才會給分，
  任何一筆測資錯誤都是0分，不會依照通過比例給分。

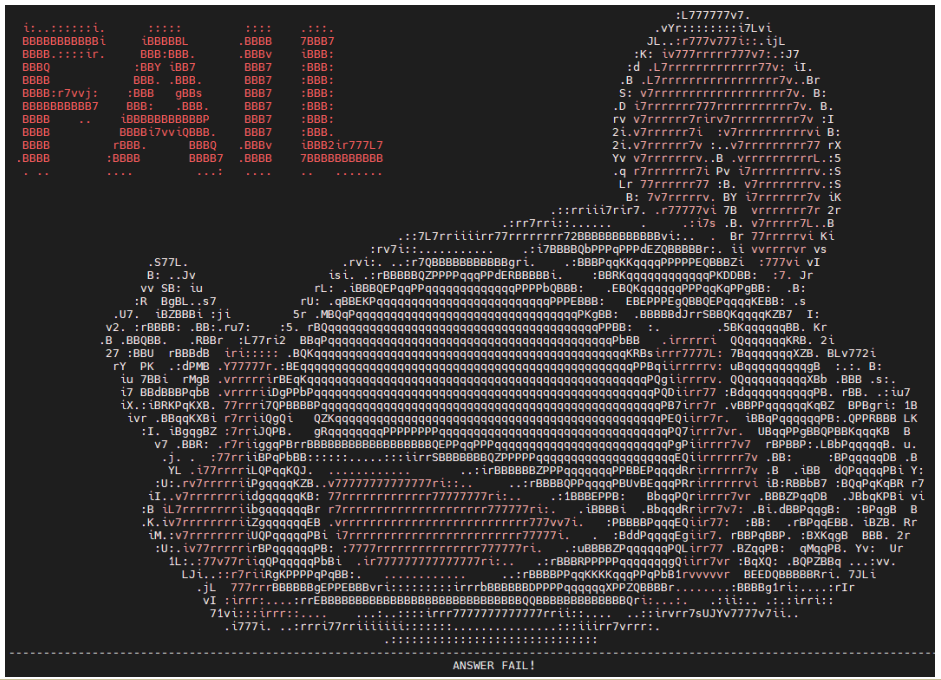
# 01_RTL

- 進行RTL simulation的資料夾，主要包含當次要完成的Design檔案 (XXX.sv)。
- 也包含01_run與09_clean_up兩個Script。
  - 01_run: 進行電路模擬
  - 09_clean_up: 清除波型檔
- RTL simulation會用pattern去驗證你的Design的functional正確性。
- 如果通過所有pattern會得到一隻PASS貓貓，有任何一筆pattern錯誤會得到一隻FAIL蠑螈。

script 使用範例:

```
ee01 [Lab01/01_RTL]% ./01_run
```



# 02_SYN

- 進行電路合成的資料夾，主要包含合成的tcl檔。(syn.tcl)
- 也包含01_run_dc與09_clean_up兩個Script。
  - 01_run_dc: 進行電路合成
  - 09_clean_up: 清除合成結果
- 合成結果包含Timing report、Area report、Netlist、Delay format。
- 合成結果不能有Error、不能有Latch、要有Area report、Timing report slack MET。 (slack ≧ 0)
- Synthesis可以驗證你的timing、是否有使用到不可合成的語法。

```
Number of ports:                         48
Number of nets:                         185
Number of cells:                        161
Number of combinational cells:          161
Number of sequential cells:               0
Number of macros/black boxes:             0
Number of buf/inv:                       38
Number of references:                    18

Combinational area:            2864.030431
Buf/Inv area:                   379.209614
Noncombinational area:            0.000000
Macro/Black Box area:             0.000000
Net Interconnect area:      undefined  (No wire load specified)

Total cell area:               2864.030431
Total area:                 undefined
```

Area report

```
U875/Y (NAND2X4)                 0.08       5.05 r
U974/Y (NAND3X4)                 0.16       5.21 f
out_n[7] (out)                   0.00       5.21 f
data arrival time                           5.21

max_delay                        5.00       5.00
output external delay            0.00       5.00
data required time                          5.00
-----------------------------------------------------
data required time                          5.00
data arrival time                          -5.21
-----------------------------------------------------
slack (VIOLATED)                           -0.21
```

Fail !

```
Startpoint: in_num0[1] (input port)
Endpoint: out_num1[0]
          (output port)
Path Group: default
Path Type: max

Point                                    Incr       Path
-----------------------------------------------------------
input external delay                     0.00       0.00 r
in_num0[1] (in)                          0.00       0.00 r
U250/Y (INVXL)                           0.06       0.06 f
U166/Y (AOI222XL)                        0.58       0.64 r
U218/Y (AOI222XL)                        0.34       0.98 f
   ...
U170/Y (INVX2)                           0.20       9.28 f
U332/Y (AOI22X1)                         0.72      10.00 r
out_num1[0] (out)                        0.00      10.00 r
data arrival time                                  10.00

max_delay                               10.00      10.00
output external delay                    0.00      10.00
data required time                                 10.00
-----------------------------------------------------------
data required time                                 10.00
data arrival time                                 -10.00
-----------------------------------------------------------
slack (MET)                                        0.00
```

Pass !

Timing report

# 02_SYN

- 記得檢查是否合成出Latch
  - 可以在syn.log用ctrl+F尋找關鍵字Latch
- 如果出現Latch → Demo Fail

```
#=================================================
#  Read RTL Code
#=================================================
read_sverilog {$DESIGN\.sv}
Loading db file '/usr/synthesis/libraries/syn/dw_foundation.sldb'
Loading db file '/usr/synthesis/libraries/syn/standard.sldb'
Loading db file '/RAID2/COURSE/iclab/iclabta01/umc018/Synthesis/slow.db'
Loading db file '/usr/synthesis/libraries/syn/gtech.db'
  Loading link library 'slow'
  Loading link library 'gtech'
Loading sverilog file '/home/RAID2/COURSE/dcs/dcsta02/TA/Lab02/01_RTL/Sort.sv'
Detecting input file type automatically (-rtl or -netlist).
Reading with Presto HDL Compiler (equivalent to -rtl option).
Running PRESTO HDLC
Compiling source file /home/RAID2/COURSE/dcs/dcsta02/TA/Lab02/01_RTL/Sort.sv

Inferred memory devices in process
        in routine Sort line 52 in file
                '/home/RAID2/COURSE/dcs/dcsta02/TA/Lab02/01_RTL/Sort.sv'.
===========================================================================
|    Register Name     | Type  | Width | Bus | MB | AR | AS | SR | SS | ST |
===========================================================================
|    out_num2_reg      | Latch |   6   | Y   | N  | N  | N  | -  | -  | -  |
===========================================================================
```

# 03_GATE

- 進行Gate-level simulation的資料夾。
- 包含01_run與09_clean_up兩個Script。
  - 01_run: 進行電路模擬
  - 09_clean_up: 清除波型
- 與RTL simulation相似，會使用與01_RTL相同的pattern，
  輸入合成出來的電路加上cell delay進行模擬。
- 同時驗證電路functional與timing的正確性。
- 如果在01_RTL階段的code寫法不當，合成出來的電路就會不正常，在這一步驟高機率翻車。

# 09_UPLAOD

- 上傳你的Design (XXX.sv)到助教帳號底下，
- 包含01_upload與02_download兩個Script。
  - 01_upload: 上傳code
  - 02_download :下載上傳結果
- 上傳前請確認在01_RTL的XXX.sv是否為你要繳交的版本
- 會依照當前時間上傳至1de或2de，過了2de的Deadline將無法上傳
- ./02_download [argument] 可以確認你上傳的檔案
  - [argument] = demo1 or demo2

```
linux01 [Lab02/09_UPLOAD]% ./02_download demo1
Download done!
linux31 [Lab02/09_upload]% ./02_download demo2
Download done!
linux31 [Lab02/09_upload]% ./02_download demo2
You haven't uploaded yet!
```

Download your code to 09_UPLOAD directory

```
linux31 [Lab02/09_upload]% ls
01_upload  02_download  Counter.sv  Counter_dcsta02.sv
```

02_download 使用方式

01_upload 使用方式

```
linux01 [Lab02/09_UPLOAD]% ./01_upload

module Sort(
    // Input signals
 :
endmodule
-----------------------------------------------------
The 1st demo deadline is Thu Mar 18 16:25:00 CST 2021 ,
It is Fri Mar 12 18:15:48 CST 2021 now!
It will upload to demo1.
It will overwrite your file if you have uploaded before.
Is this the file you want to upload?(y/n):y
Upload done!
linux01 [Lab02/09_UPLOAD]%
```

如果超過2de期限

```
The 1st demo deadline is Wed Mar 20 12:55:59 CST 2019 , and the 2nd demo dea
It is Wed Mar 20 13:00:27 CST 2019 now!
The submission is not accepted since the 2nd demo deadline is over.
```