



國立陽明交通大學

NATIONAL YANG MING CHIAO TUNG UNIVERSITY

# 前瞻積體電路設計實驗室

---

**ADFP Cloud 3.0**

## **Cell-based 設計使用者手冊**

**2026.03.08 Version 1**

**Author**

賴林鴻、蕭邦原

**Editor**

蕭邦原

# 目錄

ADFP Cloud 3.0 .....	1
1. RTL Simulation .....	3
2. Logic Synthesis.....	6
3. Gate-level Simulation.....	9
4. Power Analysis & STA.....	12
5. Clock Domain Crossing (CDC) Analysis .....	13
6. Physical Design : Auto Placement & Routing.....	14
7. Post-Layout Simulation.....	18
8. 查看 TSMC 提供之 CB 文件範例練習 .....	21
9. 查看 NYCU 課程提供之 CB 文件範例練習.....	22
Revision Record and Author List.....	24

# 1. RTL Simulation

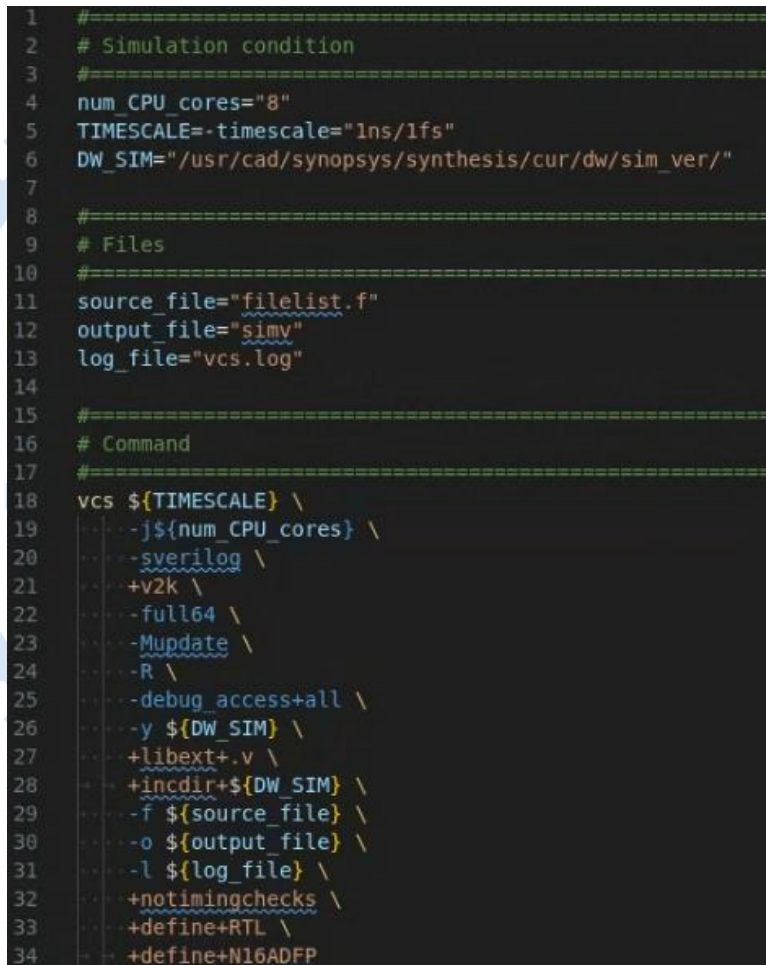
1.1 模擬前須準備好完整的 Verilog 檔案並將這些檔案寫入 filelist.f 之中

1.2 如何使用 Synopsys VCS 進行 RTL 模擬

(1) 建立執行檔案：01\_vcs\_rtl.sh，其內容請參考下方範例

```
$ touch 01_vcs_rtl.sh
```

```
$ chmod 755 01_vcs_rtl.sh
```



```
1 #=====  
2 # Simulation condition  
3 #=====  
4 num_CPU_cores="8"  
5 TIMESCALE=-timescale="1ns/1fs"  
6 DW_SIM="/usr/cad/synopsys/synthesis/cur/dw/sim_ver/"  
7  
8 #=====  
9 # Files  
10 #=====  
11 source_file="filelist.f"  
12 output_file="simv"  
13 log_file="vcs.log"  
14  
15 #=====  
16 # Command  
17 #=====  
18 vcs ${TIMESCALE} \  
19 --j${num_CPU_cores} \  
20 -sverilog \  
21 +v2k \  
22 -full64 \  
23 -Mupdate \  
24 -R \  
25 -debug_access+all \  
26 -y ${DW_SIM} \  
27 +libext+.v \  
28 +incdir+${DW_SIM} \  
29 -f ${source_file} \  
30 -o ${output_file} \  
31 -l ${log_file} \  
32 +notimingchecks \  
33 +define+RTL \  
34 +define+N16ADFP
```

(2) 模擬前須載入所需的模組（若已載入可忽略此步驟）

```
$ module load vcs
```

```
$ module load verdi/VCS_2025.06
```

(3) 執行 01\_vcs\_rtl.sh 進行 RTL 模擬

```
$ ./01_vcs_rtl.sh
```

### 1.3 如何使用 Cadence Xrun 進行 RTL 模擬

(1) 建立執行檔案：01\_xrun\_rtl.sh，其內容請參考下方範例

```
$ touch 01_xrun_rtl.sh
```

```
$ chmod 755 01_xrun_rtl.sh
```

```
1 #-----
2 # Simulation condition
3 #-----
4 num_CPU_cores="8"
5 TIMESCALE="-timescale 1ns/1fs"
6 DW_SIM="/usr/cad/synopsys/synthesis/cur/dw/sim_ver/"
7
8 #-----
9 # Files
10 #-----
11 source_file="filelist.f"
12 log_file="xrun.log"
13
14 #-----
15 # Command
16 #-----
17 xrun -override precision ${TIMESCALE} \
18 -mcl ${num_CPU_cores} \
19 -sv \
20 -y ${DW_SIM} \
21 -libext .v \
22 -access +rwc \
23 -f ${source_file} \
24 -log ${log_file} \
25 -notimingchecks \
26 +define+RTL \
27 +define+N16ADFP
```

(2) 模擬前須載入所需的模組（若已載入可忽略此步驟）

```
$ module load xrun/2024.09.006
```

```
$ module load verdi/NC_2025.06 如須要使用 verdi，須載入該模組之 NC 版本！
```

(3) 執行 01\_xrun\_rtl.sh 進行 RTL 模擬

```
$ ./01_xrun_rtl.sh
```

### 1.4 如何使用 Synopsys Verdi 開啟波形(.fsdb)

(1) 建立執行檔案：02\_verdi\_rtl.sh，其內容請參考下方範例

```
$ touch 02_verdi_rtl.sh
```

```
$ chmod 755 02_verdi_rtl.sh
```

```

1 #=====
2 # Files
3 #=====
4 source_file="filelist.f"
5
6 #=====
7 # Command
8 #=====
9 verdi -ssf *.fsdb \
10 -sv \
11 -f ${source_file} \
12 -nologo \
13 +define+RTL \
14 +define+N16ADFP \
15 +v2k \
16 -autoalias

```

(2) 開啟波形前須載入所需的模組 (若已載入可忽略此步驟)

```
$ module load verdi/VCS_2025.06
```

(3) 執行 02\_verdi\_rtl.sh 開啟波形

```
$ ./02_verdi_rtl.sh
```

(4) 若僅須要使用 nWave 開啟波形，請執行下方指令：

```
$ nWave -ssf *.fsdb
```

### 1.5 如何使用 Cadence SimVision 開啟波形(.shm)

(1) 建立執行檔案：02\_simvision\_rtl.sh，其內容請參考下方範例

```
$ touch 02_simvision_rtl.sh
```

```
$ chmod 755 02_simvision_rtl.sh
```

```

1 #=====
2 # Command
3 #=====
4 simvision *.shm

```

(2) 開啟波形前須載入所需的模組 (若已載入可忽略此步驟)

```
$ module load xrun/2024.09.006
```

(3) 執行 02\_simvision\_rtl.sh 開啟波形

```
$ ./02_simvision_rtl.sh
```

## 2. Logic Synthesis

### 2.1 ADFP 製程資料夾

- (1) Standard Cell Library (CCS Model)

```
/RAID2/PROCESS/ADFP/Executable_Package/Collaterals/IP/stdcell/  
N16ADFP_StdCell/CCS/
```

- (2) Standard Cell library (NLDM Model)

```
/RAID2/PROCESS/ADFP/Executable_Package/Collaterals/IP/stdcell/  
N16ADFP_StdCell/NLDM/
```

- (3) Standard IO library (NLDM Model)

```
/RAID2/PROCESS/ADFP/Executable_Package/Collaterals/IP/stdio/N  
16ADFP_StdIO/NLDM/
```

- (4) SRAM library (NLDM Model)

```
/RAID2/PROCESS/ADFP/Executable_Package/Collaterals/IP/sram/N1  
6ADFP_SRAM/NLDM/
```

### 2.2 如何使用 Synopsys Design Compiler 進行電路合成

- (1) 準備檔案：syn.tcl 與 .synopsys\_dc.setup

- (2) 建立執行檔案：01\_dc\_shell.sh，其內容請參考下方範例

```
$ touch 01_dc_shell.sh
```

```
$ chmod 755 01_dc_shell.sh
```

```

1 #=====
2 # Simulation condition
3 #=====
4 num_CPU_cores="8"
5
6 #=====
7 # Files
8 #=====
9 syn_tcl="syn.tcl"
10 syn_log="syn.log"
11
12 #=====
13 # Command
14 #=====
15 dcnxt_shell -f ${syn_tcl} \
16 -x "set_host_options -max_cores ${num_CPU_cores}" \
17 -output_log_file ${syn_log}

```

(3) 合成電路前須載入所需的模組 (若已載入可忽略此步驟)

```
$ module load lc/2025.06
```

```
$ module load dc/2025.06 執行該指令即可同時載入 lc
```

(4) 執行 01\_dc\_shell.sh 進行電路合成

```
$ ./01_dc_shell.sh
```

## 2.3 如何使用 Cadence Genus 進行電路合成

(1) 準備檔案：syn.tcl 與 .cadence\_genus.setup

(2) 建立執行檔案：01\_genus.sh，其內容請參考下方範例

```
$ touch 01_genus.sh
```

```
$ chmod 755 01_genus.sh
```

```

6 #=====
7 # Files
8 #=====
9 syn_tcl="syn.tcl"
10 syn_log="syn.log"
11
12 #=====
13 # Command
14 #=====
15 genus -files ${syn_tcl} -log ${syn_log}

```

(3) 合成電路前須載入所需的模組 (若已載入可忽略此步驟)

\$ module load genus/DDI\_23.34.000 **Genus 最新版本已經合併至 DDI 中**

(4) 執行 01\_genus.sh 進行電路合成

\$ ./01\_genus.sh



### 3. Gate-level Simulation

3.1 模擬前須準備好完整的 Verilog 檔案(包含已合成的 Netlist)並將這些檔案寫入 filelist.f 之中

3.2 相關路徑

(1) Standard Cell Library Behavior Model

```
/RAID2/PROCESS/ADFP/Executable_Package/Collaterals/IP/stdcell/  
N16ADFP_StdCell/VERILOG/N16ADFP_StdCell.v
```

3.3 如何使用 Synopsys VCS 進行 Gate-level 模擬

(1) 建立執行檔案：01\_vcs\_gate.sh，其內容請參考下方範例

```
$ touch 01_vcs_gate.sh
```

```
$ chmod 755 01_vcs_gate.sh
```

```
1 #=====
2 # Simulation condition
3 #=====
4 num_CPU_cores="8"
5 TIMESCALE=-timescale="1ns/1fs"
6 N16ADFP_SIM="/RAID2/PROCESS/ADFP/Executable_Package/Collaterals/IP/stdcell/N16ADFP_StdCell/VERILOG/N16ADFP_StdCell.v"
7
8 #=====
9 # Files
10 #=====
11 source_file="filelist.f"
12 output_file="simv"
13 log_file="vcs.log"
14
15 #=====
16 # Command
17 #=====
18 vcs ${TIMESCALE} \
19   --js(num_CPU_cores) \
20   --sverilog \
21   +v2k \
22   --full64 \
23   --Mupdate \
24   --R \
25   --debug_access+all \
26   -f ${source_file} \
27   -o ${output_file} \
28   -l ${log_file} \
29   -v ${N16ADFP_SIM} \
30   +neg_tchk \
31   +nowarnNTCDN \
32   +define+GATE \
33   +define+N16ADFP
```

(2) 模擬前須載入所需的模組 (已載入可忽略此步驟)

```
$ module load vcs/2025.06
```

```
$ module load verdi/VCS_2025.06
```

(3) 執行 01\_vcs\_gate.sh 進行 Gate-level 模擬

```
$ ./01_vcs_gate.sh
```

### 3.4 如何使用 Cadence Xrun 進行 Gate-level 模擬

(1) 建立執行檔案：01\_xrun\_gate.sh，其內容請參考下方範例

```
$ touch 01_xrun_gate.sh
```

```
$ chmod 755 01_xrun_gate.sh
```

```
1 # Simulation condition
2 #
3 #=====
4 num_CPU_cores="8"
5 TIMESCALE="-timescale 1ns/1fs"
6 N16ADFP_SIM=/RAID2/PROCESS/ADFP/Executable_Package/Collaterals/IP/stdcell/N16ADFP_StdCell/VERILOG/N16ADFP_StdCell.v
7
8 #=====
9 # Files
10 #
11 source_file="filelist.f"
12 log_file="xrun.log"
13
14 #=====
15 # Command
16 #=====
17 xrun -override precision ${TIMESCALE} \
18     -mcl ${num_CPU_cores} \
19     -sv \
20     -access +rwc \
21     -f ${source_file} \
22     -log ${log_file} \
23     -v ${N16ADFP_SIM} \
24     +define+GATE \
25     +define+N16ADFP
```

(2) 模擬前須載入所需的模組（已載入可忽略此步驟）

```
$ module load xrun
```

```
$ module load verdi/NC_2025.06 如須要使用 verdi，須載入該模組之 NC 版本！
```

(3) 執行 01\_xrun\_gate.sh 進行 Gate-level 模擬

```
$ ./01_xrun_gate.sh
```

### 3.5 如何使用 Synopsys Verdi 開啟波形(.fsdb)

(1) 建立執行檔案：02\_verdi\_gate.sh，其內容請參考下方範例

```
$ touch 02_verdi_gate.sh
```

```
$ chmod 755 02_verdi_gate.sh
```

```

1 #=====
2 # Files
3 #=====
4 source_file="filelist.f"
5
6 #=====
7 # Command
8 #=====
9 verdi -ssf *.fsdb \
10 -sv \
11 -f ${source_file} \
12 -nologo \
13 +define+GATE \
14 +define+N16ADFP \
15 +v2k \
16 -autoalias

```

(2) 模擬前須載入所需的模組 (已載入可忽略此步驟)

```
$ module load verdi/VCS_2025.06
```

(3) 執行 02\_verdi\_gate.sh 開啟波形

```
$ ./02_verdi_gate.sh
```

### 3.6 如何使用 Cadence SimVision 開啟波形(.shm)

(1) 建立執行檔案：02\_simvision\_gate.sh

```
$ touch 02_simvision_gate.sh
```

```
$ chmod 755 02_simvision_gate.sh
```

```

1 #=====
2 # Command
3 #=====
4 simvision *.shm

```

(2) 模擬前須載入所需的模組 (已載入可忽略此步驟)

```
$ module load xrun
```

(3) 執行 02\_simvision\_gate.sh 開啟波形

```
$ ./02_simvision_gate.sh
```

## 4. Power Analysis & STA

### 4.1 如何使用 Synopsys Prime Time 進行功率分析與靜態時序分析

(1) 準備檔案：pt.tcl 與 .synopsys\_pt.setup

(2) 建立執行檔案：01\_ptpx.sh，其內容請參考下方範例

```
$ touch 01_ptpx.sh
```

```
$ chmod 755 01_ptpx.sh
```

```
1 #=====  
2 # Simulation condition  
3 #=====  
4 num_CPU_cores="8"  
5  
6 #=====  
7 # Files  
8 #=====  
9 pt_tcl="pt.tcl"  
10 pt_log="pt.log"  
11  
12 #=====  
13 # Command  
14 #=====  
15 pt_shell -f ${pt_tcl} \  
16 -x "set host options -max_cores ${num_CPU_cores}" \  
17 -output_log_file ${pt_log}
```

(3) 分析前須載入所需的模組（已載入可忽略此步驟）

```
$ module load lc
```

```
$ module load primetime
```

(4) 執行 01\_ptpx.sh 進行分析

```
$ ./01_ptpx.sh
```

## 5. Clock Domain Crossing (CDC) Analysis

### 5.1 如何使用 Cadence Jasper 進行跨時域分析

- (1) 準備檔案：jg.tcl 與已合成電路的.sdc 檔案
- (2) 建立執行檔案：01\_jg.sh，其內容請參考下方範例

```
$ touch 01_jg.sh
```

```
$ chmod 755 01_jg.sh
```

```
1 #=====
2 # Files
3 #=====
4 jg_tcl="jg.tcl"
5
6 #=====
7 # Command 在 2025 後的版本可以不加
8 #=====
9 jg -cdc ${jg_tcl} -allow unsupported 05
```

- (3) 分析前須載入所需的模組 (已載入可忽略此步驟)

```
$ module load jasper
```

- (4) 執行 01\_jg.sh 進行分析

```
$ ./01_jg.sh
```

## 6. Physical Design : Auto Placement & Routing

### 6.1 ADFP 製程資料夾 – Library

- (1) Standard Cell Library CCS Model (\_ccs.lib)

```
/RAID2/PROCESS/ADFP/Executable_Package/Collaterals/IP/stdcell/  
N16ADFP_StdCell/CCS/
```

- (2) Standard Cell library NLDM Model (.lib)

```
/RAID2/PROCESS/ADFP/Executable_Package/Collaterals/IP/stdcell/  
N16ADFP_StdCell/NLDM/
```

- (3) Standard IO library NLDM Model (.lib)

```
/RAID2/PROCESS/ADFP/Executable_Package/Collaterals/IP/stdio/N  
16ADFP_StdIO/NLDM/
```

- (4) SRAM library NLDM Model (.lib)

```
/RAID2/PROCESS/ADFP/Executable_Package/Collaterals/IP/sram/N  
16ADFP_SRAM/NLDM/
```

### 6.2 ADFP 製程資料夾 – LEF (.lef)

- (1) Technology LEF (.tlef)

```
/RAID2/PROCESS/ADFP/Executable_Package/Collaterals/Tech/APR/  
N16ADFP_APR_Innovus/
```

- (2) Standard Cell LEF (.lef)

```
/RAID2/PROCESS/ADFP/Executable_Package/Collaterals/IP/stdcell/  
N16ADFP_StdCell/LEF/
```

- (3) Standard IO LEF (.lef)

/RAID2/PROCESS/ADFP/Executable\_Package/Collaterals/IP/stdio/N16ADFP\_StdIO/LEF/

- (4) SRAM LEF (.lef)

/RAID2/PROCESS/ADFP/Executable\_Package/Collaterals/IP/sram/N16ADFP\_SRAM/LEF/

- (5) Bond Pad LEF (.lef)

/RAID2/PROCESS/ADFP/Executable\_Package/Collaterals/IP/bondpad/N16ADFP\_BondPad/LEF/

### 6.3 ADFP 製程資料夾 – RC Extraction

- (1) Cadence QRC Extraction (qrcTechFile)

/RAID2/PROCESS/ADFP/Executable\_Package/Collaterals/Tech/RC/N16ADFP\_QRC/

### 6.4 ADFP 製程資料夾 – GDS

- (1) Technology Mapping file (.map)

/RAID2/PROCESS/ADFP/Executable\_Package/Collaterals/Tech/APR/N16ADFP\_APR\_Innovus/

- (2) Standard Cell GDS (.gds)

/RAID2/PROCESS/ADFP/Executable\_Package/Collaterals/IP/stdcell/N16ADFP\_StdCell/GDS/

- (3) Standard IO GDS (.gds)

/RAID2/PROCESS/ADFP/Executable\_Package/Collaterals/IP/stdio/N16ADFP\_StdIO/GDS/

- (4) SRAM GDS (.gds)

/RAID2/PROCESS/ADFP/Executable\_Package/Collaterals/IP/sram/N16ADFP\_SRAM/GDS/

- (5) Bond Pad GDS (.gds)

/RAID2/PROCESS/ADFP/Executable\_Package/Collaterals/IP/bondpad/N16ADFP\_BondPad/GDS/

## 6.5 ADFP 製程資料夾 – Verilog Model Including Power/Ground

- (1) Standard Cell (\_pwr.v)

/RAID2/PROCESS/ADFP/Executable\_Package/Collaterals/IP/stdcell/N16ADFP\_StdCell/VERILOG/

- (2) Standard IO (\_pwr.v)

/RAID2/PROCESS/ADFP/Executable\_Package/Collaterals/IP/stdio/N16ADFP\_StdIO/VERILOG/

- (3) SRAM (\_pwr.v)

/RAID2/PROCESS/ADFP/Executable\_Package/Collaterals/IP/sram/N16ADFP\_SRAM/VERILOG/

## 6.6 ADFP 製程資料夾 – SPICE

- (1) Standard Cell (.spi)

/RAID2/PROCESS/ADFP/Executable\_Package/Collaterals/IP/stdcell/N16ADFP\_StdCell/SPICE/

- (2) Standard IO (.spi)

/RAID2/PROCESS/ADFP/Executable\_Package/Collaterals/IP/stdio/N16ADFP\_StdIO/SPICE/

- (3) SRAM (.spi)

/RAID2/PROCESS/ADFP/Executable\_Package/Collaterals/IP/sram/N  
16ADFP\_SRAM/SPICE/

## 6.7 如何使用 Cadence Innovus 進行實體電路設計

- (1) 實作前須載入所需的模組 (已載入可忽略此步驟)

**\$ module load innovus** **Innovus 最新版本已經合併至 DDI 中**

- (2) 執行軟體進行實體電路設計

**\$ innovus**

- (3) 詳細 APR 步驟可以參考 TSMC 或 NYCU 課程提供的範例練習



## 7. Post-Layout Simulation

7.1 模擬前須準備好完整的 Verilog 檔案(包含 APR 後的 Netlist)並將這些檔案寫入 filelist.f 之中

7.2 相關路徑

(1) Standard Cell Behavior Model

```
/RAID2/PROCESS/ADFP/Executable_Package/Collaterals/IP/stdcell/  
N16ADFP_StdCell/VERILOG/N16ADFP_StdCell.v
```

(2) Standard IO Behavior Model

```
/RAID2/PROCESS/ADFP/Executable_Package/Collaterals/IP/stdio/  
N16ADFP_StdIO/VERILOG/N16ADFP_StdIO.v
```

7.3 如何使用 Synopsys VCS 進行 Post-layout 模擬

(1) 準備執行檔案：01\_vcs\_post.sh，其內容請參考下方範例

```
$ touch 01_vcs_post.sh
```

```
$ chmod 755 01_vcs_post.sh
```

```
1 #=====  
2 # Simulation condition  
3 #=====  
4 num CPU cores="8"  
5 TIMESCALE=- timescale="1ns/1fs"  
6 N16ADFP_SIM="/RAID2/PROCESS/ADFP/Executable_Package/Collaterals/IP/stdcell/N16ADFP_StdCell/VERILOG/N16ADFP_StdCell.v"  
7 N16ADFP_IO_SIM="/RAID2/PROCESS/ADFP/Executable_Package/Collaterals/IP/stdio/N16ADFP_StdIO/VERILOG/N16ADFP_StdIO.v"  
8  
9 #=====  
10 # Files  
11 #=====  
12 source file="filelist.f"  
13 output_file="simv"  
14 log_file="vcs.log"  
15  
16 #=====  
17 # Command  
18 #=====  
19 vcs ${TIMESCALE} \  
20 --j${num CPU cores} \  
21 --sverilog \  
22 --v2k \  
23 --full64 \  
24 --Mupdate \  
25 --R \  
26 --debug access+all \  
27 --f ${source_file} \  
28 --o ${output_file} \  
29 --l ${log_file} \  
30 --v ${N16ADFP_SIM} \  
31 --v ${N16ADFP_IO_SIM} \  
32 --neg tchk \  
33 --define+POST \  
34 --define+N16ADFP
```

(2) 模擬前須載入所需的模組 (已載入可忽略此步驟)

```
$ module load vcs
```

```
$ module load verdi/VCS_2025.06
```

(3) 執行 01\_vcs\_post.sh 進行 Post-layout 模擬

```
$ ./01_vcs_post.sh
```

## 7.4 如何使用 Cadence Xrun 進行 Post-layout 模擬

(1) 準備執行檔案：01\_xrun\_post.sh，其內容請參考下方範例

```
$ touch 01_xrun_post.sh
```

```
$ chmod 755 01_xrun_post.sh
```

```
1 #=====  
2 # Simulation condition  
3 #=====  
4 num_CPU_cores="8"  
5 TIMESCALE="-timescale 1ns/1fs"  
6 N16ADFP_SIM="/RAID2/PROCESS/ADFP/Executable_Package/Collaterals/IP/stdcell/N16ADFP_StdCell/VERILOG/N16ADFP_StdCell.v  
7 N16ADFP_IO_SIM="/RAID2/PROCESS/ADFP/Executable_Package/Collaterals/IP/stdio/N16ADFP_StdIO/VERILOG/N16ADFP_StdIO.v  
8  
9 #=====  
10 # Files  
11 #=====  
12 source_file="filelist.f"  
13 log_file="xrun.log"  
14  
15 #=====  
16 # Command  
17 #=====  
18 xrun -override precision ${TIMESCALE} \  
19 -mcl ${num_CPU_cores} \  
20 -sv \  
21 -access +rwc \  
22 -f ${source_file} \  
23 -log ${log_file} \  
24 -v ${N16ADFP_SIM} \  
25 -v ${N16ADFP_IO_SIM} \  
26 +define+POST \  
27 +define+N16ADFP
```

(2) 模擬前須載入所需的模組（已載入可忽略此步驟）

```
$ module load xrun
```

```
$ module load verdi/NC_2025.06 如須要使用 verdi，須載入該模組之 NC 版本！
```

(3) 執行 01\_xrun\_post.sh 進行 Post-layout 模擬

```
$ ./01_xrun_post.sh
```

## 7.5 如何使用 Synopsys Verdi 開啟波形(.fsdb)

(1) 建立執行檔案：02\_verdi\_post.sh，其內容請參考下方範例

```
$ touch 02_verdi_post.sh
```

```
$ chmod 755 02_verdi_post.sh
```

```
1 #=====  
2 # Files  
3 #=====  
4 source_file="filelist.f"  
5  
6 #=====  
7 # Command  
8 #=====  
9 verdi -ssf *.fsdb \  
10 -sv \  
11 -f ${source_file} \  
12 -nologo \  
13 +define+POST \  
14 +define+N16ADFP \  
15 +v2k \  
16 -autoalias
```

(2) 開啟前須載入所需的模組 (已載入可忽略此步驟)

```
$ module load verdi/VCS_2025.06
```

## 7.6 如何使用 Cadence SimVision 開啟波形(.shm)

(1) 建立執行檔案：02\_simvision\_post.sh，其內容請參考下方範例

```
$ touch 02_simvision_post.sh
```

```
$ chmod 755 02_simvision_post.sh
```

```
1 #=====  
2 # Command  
3 #=====  
4 simvision *.shm
```

(2) 開啟前須載入所需的模組 (已載入可忽略此步驟)

```
$ module load xrun
```


(3) 執行 02\_simvision\_post.sh 開啟波形

```
$ ./02_simvision_post.sh
```

## 8. 查看 TSMC 提供之 CB 文件範例練習

8.1 TSMC 有提供 Lab 供大家參考，特別是有關 Physical Design 有需多 Library 和 mmmc 檔案要設定，建議可以參考 Lab。路徑如下：

/RAID2/PROCESS/ADFP/Executable\_Package/Lab

8.2 請使用「MyPDF 」軟體參閱浮水印文件，有關 CB 說明：

(1) TSMC ADFP 製程簡要說明

ADFP046\_TSMC\_N16ADFP\_Introduction\_20220118\_wmc.pdf

(2) APR 須要注意的事項

ADFP045\_TSMC\_N16ADFP\_Chip\_20220118\_wmc.pdf

(3) Lab 練習文件，使用 Innovus 跑 IP level 和 Full Chip level 之 APR

ADFP043\_TSMC\_N16ADFP\_Lab\_Chip\_20220211\_wmc.pdf

## 9. 查看 NYCU 課程提供之 CB 文件範例練習


9.1 NYCU 課程 - ICLAB 有提供一個完整的 Cell-Based IC Design 教學，以一個實際電路：Pseudo Random Number Generator (PRNG)，讓學生操作整個數位 IC 設計流程，包含前端設計與後端設計。

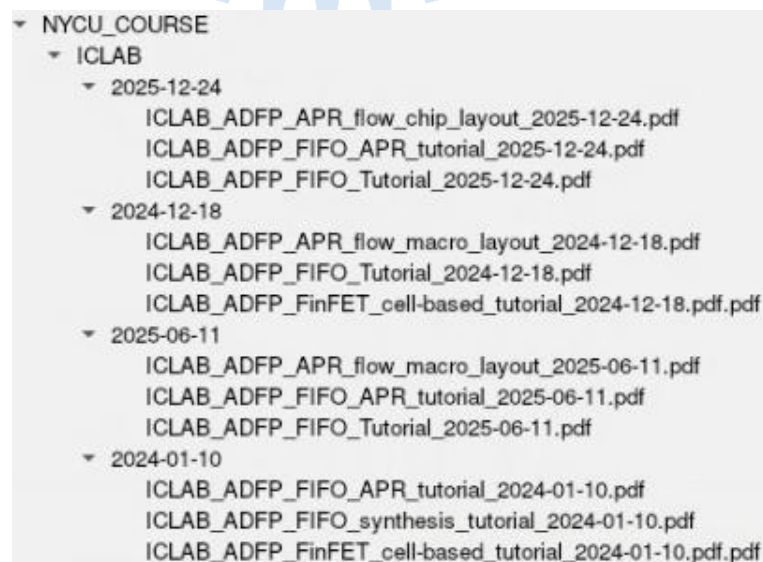
### (1) 前端設計

- RTL 撰寫：基礎 Verilog 技能與進階技巧應用 (如 SRAM, CDC, Low Power, Formal Verification...)。
- 電路合成：比較成熟製程與先進製程對於電路 PPA 之比較。
- 電路分析：使用對應的工具對電路進行功能模擬、Power 分析與 CDC 檢查。

### (2) 後端設計

- APR 流程教學 1：先進製程下的 Chip layout 實作
- APR 流程教學 2：先進製程下的 Macro layout 實作
- 比較成熟製程與先進製程對於電路實體設計有何不同
- Sign-off 流程教學：Dummy / DRC / LVS / Layout in Virtuoso

9.2 請使用「MyPDF 」參閱相關浮水印文件，有關 CB 說明：



Note：建議參考當前最新版本之文件。

(1) TSMC ADFP 製程簡要說明

ICLAB\_ADFP\_FinFET\_cell-based\_tutorial\_<Date>.pdf

(2) 如何 Step by step 跑完 Logic Synthesis 流程

ICLAB\_ADFP\_FIFO\_synthesis\_tutorial\_<Date>.pdf

(3) 如何 Step by step 跑完 Macro layout 的 APR 流程

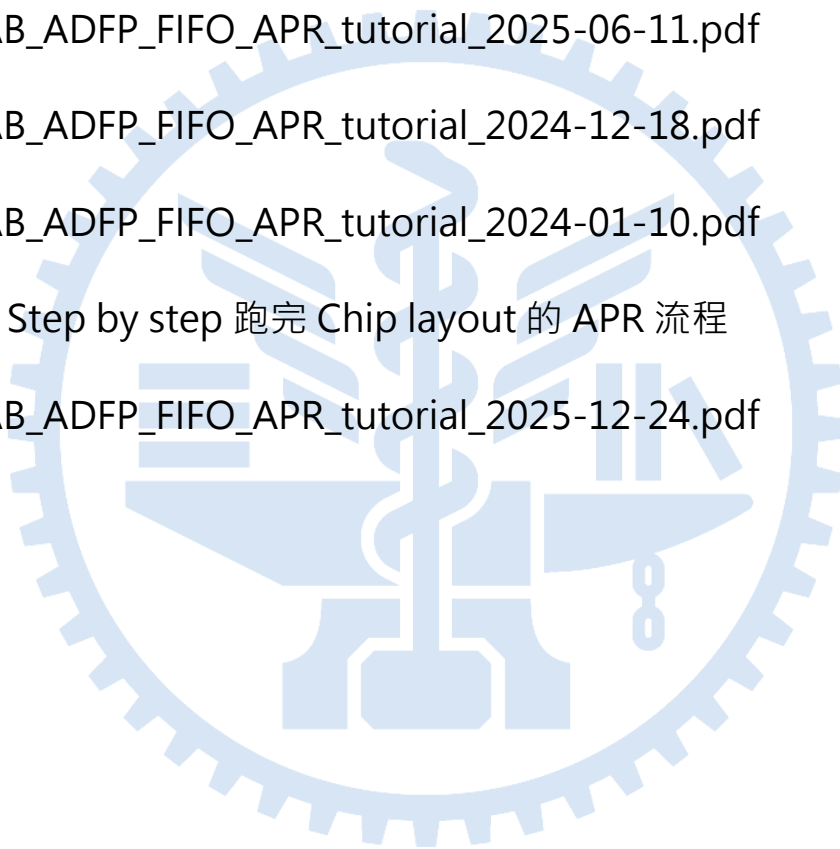
ICLAB\_ADFP\_FIFO\_APR\_tutorial\_2025-06-11.pdf

ICLAB\_ADFP\_FIFO\_APR\_tutorial\_2024-12-18.pdf

ICLAB\_ADFP\_FIFO\_APR\_tutorial\_2024-01-10.pdf

(4) 如何 Step by step 跑完 Chip layout 的 APR 流程

ICLAB\_ADFP\_FIFO\_APR\_tutorial\_2025-12-24.pdf



## Revision Record and Author List

Original Tutorial by 賴林鴻、蕭邦原

2026.03.08 Version 1 edited by Bang-Yuan Xiao

