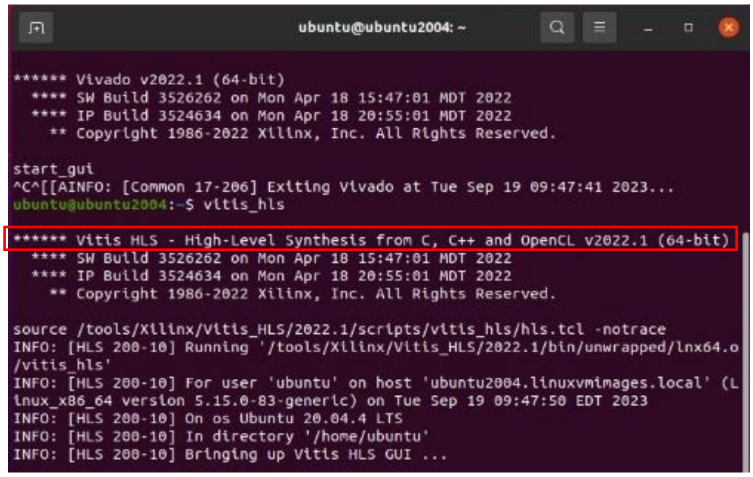
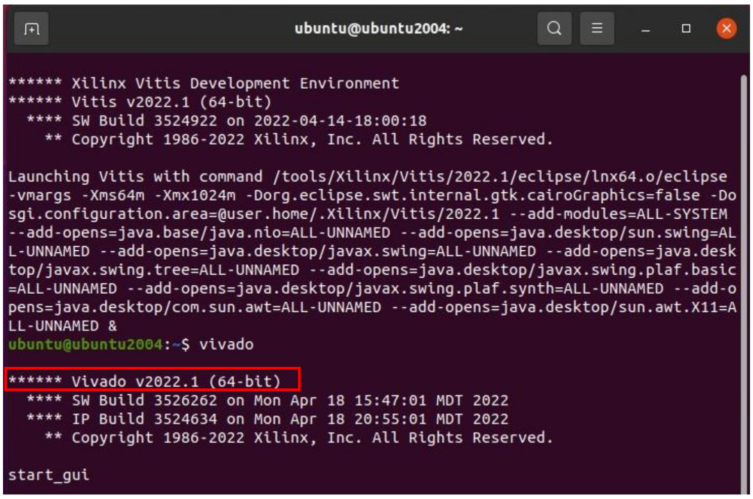
Lab1 - Tool installation

311651055_林柏宇

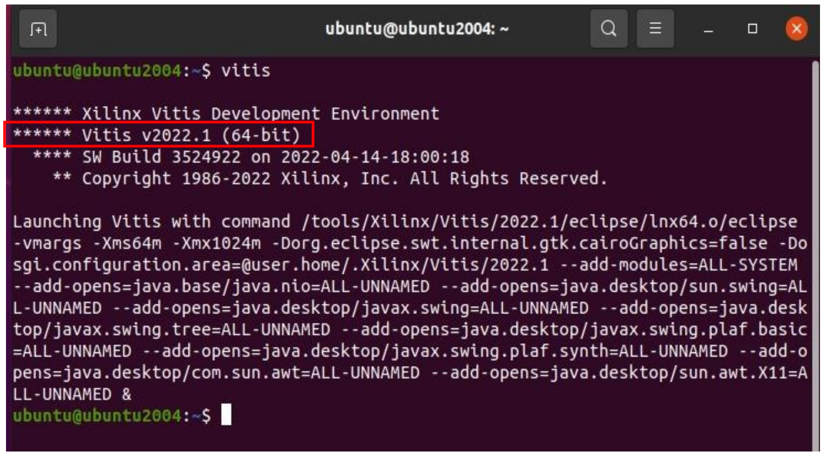
安裝好 Vistial_Box & Ubuntu，確定版本正確



```
****** Vivado v2022.1 (64-bit)
  **** SW Build 3526262 on Mon Apr 18 15:47:01 MDT 2022
  **** IP Build 3524634 on Mon Apr 18 20:55:01 MDT 2022
    ** Copyright 1986-2022 Xilinx, Inc. All Rights Reserved.

start_gui
^C^[[AINFO: [Common 17-206] Exiting Vivado at Tue Sep 19 09:47:41 2023...
ubuntu@ubuntu2004:~$ vitis_hls

****** Vitis HLS - High-Level Synthesis from C, C++ and OpenCL v2022.1 (64-bit)
  **** SW Build 3526262 on Mon Apr 18 15:47:01 MDT 2022
  **** IP Build 3524634 on Mon Apr 18 20:55:01 MDT 2022
    ** Copyright 1986-2022 Xilinx, Inc. All Rights Reserved.

source /tools/Xilinx/Vitis_HLS/2022.1/scripts/vitis_hls/hls.tcl -notrace
INFO: [HLS 200-10] Running '/tools/Xilinx/Vitis_HLS/2022.1/bin/unwrapped/lnx64.o
/vitis_hls'
INFO: [HLS 200-10] For user 'ubuntu' on host 'ubuntu2004.linuxvmimages.local' (L
inux_x86_64 version 5.15.0-83-generic) on Tue Sep 19 09:47:50 EDT 2023
INFO: [HLS 200-10] On os Ubuntu 20.04.4 LTS
INFO: [HLS 200-10] In directory '/home/ubuntu'
INFO: [HLS 200-10] Bringing up Vitis HLS GUI ...
```

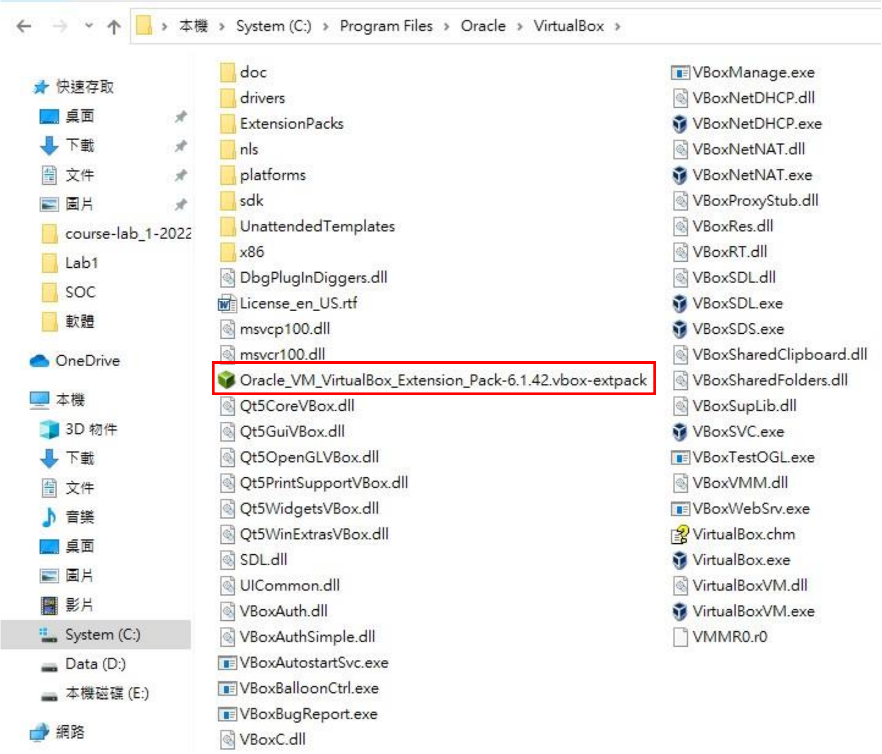


```
****** Xilinx Vitis Development Environment
****** Vitis v2022.1 (64-bit)
  **** SW Build 3524922 on 2022-04-14-18:00:18
    ** Copyright 1986-2022 Xilinx, Inc. All Rights Reserved.

Launching Vitis with command /tools/Xilinx/Vitis/2022.1/eclipse/lnx64.o/eclipse
-vmargs -Xms64m -Xmx1024m -Dorg.eclipse.swt.internal.gtk.cairoGraphics=false -Do
sgi.configuration.area=@user.home/.Xilinx/Vitis/2022.1 --add-modules=ALL-SYSTEM
--add-opens=java.base/java.nio=ALL-UNNAMED --add-opens=java.desktop/sun.swing=AL
L-UNNAMED --add-opens=java.desktop/javax.swing=ALL-UNNAMED --add-opens=java.desk
top/javax.swing.tree=ALL-UNNAMED --add-opens=java.desktop/javax.swing.plaf.basic
=ALL-UNNAMED --add-opens=java.desktop/javax.swing.plaf.synth=ALL-UNNAMED --add-o
pens=java.desktop/com.sun.awt=ALL-UNNAMED --add-opens=java.desktop/sun.awt.X11=A
LL-UNNAMED &
ubuntu@ubuntu2004:~$ vivado

****** Vivado v2022.1 (64-bit)
  **** SW Build 3526262 on Mon Apr 18 15:47:01 MDT 2022
  **** IP Build 3524634 on Mon Apr 18 20:55:01 MDT 2022
    ** Copyright 1986-2022 Xilinx, Inc. All Rights Reserved.

start_gui
```



```
ubuntu@ubuntu2004:~$ vitis

****** Xilinx Vitis Development Environment
****** Vitis v2022.1 (64-bit)
  **** SW Build 3524922 on 2022-04-14-18:00:18
    ** Copyright 1986-2022 Xilinx, Inc. All Rights Reserved.

Launching Vitis with command /tools/Xilinx/Vitis/2022.1/eclipse/lnx64.o/eclipse
-vmargs -Xms64m -Xmx1024m -Dorg.eclipse.swt.internal.gtk.cairoGraphics=false -Do
sgi.configuration.area=@user.home/.Xilinx/Vitis/2022.1 --add-modules=ALL-SYSTEM
--add-opens=java.base/java.nio=ALL-UNNAMED --add-opens=java.desktop/sun.swing=AL
L-UNNAMED --add-opens=java.desktop/javax.swing=ALL-UNNAMED --add-opens=java.desk
top/javax.swing.tree=ALL-UNNAMED --add-opens=java.desktop/javax.swing.plaf.basic
=ALL-UNNAMED --add-opens=java.desktop/javax.swing.plaf.synth=ALL-UNNAMED --add-o
pens=java.desktop/com.sun.awt=ALL-UNNAMED --add-opens=java.desktop/sun.awt.X11=A
LL-UNNAMED &
ubuntu@ubuntu2004:~$ 
```

大致上按照教學 pdf 走，但中間有遇到一些問題：



本機 › System (C:) › Program Files › Oracle › VirtualBox ›

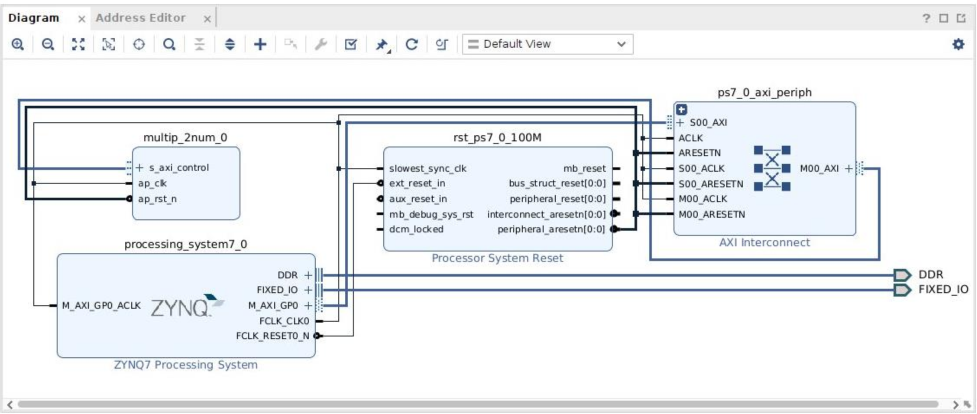
快速存取
桌面
下載
文件
圖片
course-lab_1-2022
Lab1
SOC
軟體
OneDrive
本機
3D 物件
下載
文件
音樂
桌面
圖片
影片
System (C:)
Data (D:)
本機磁碟 (E:)
網路

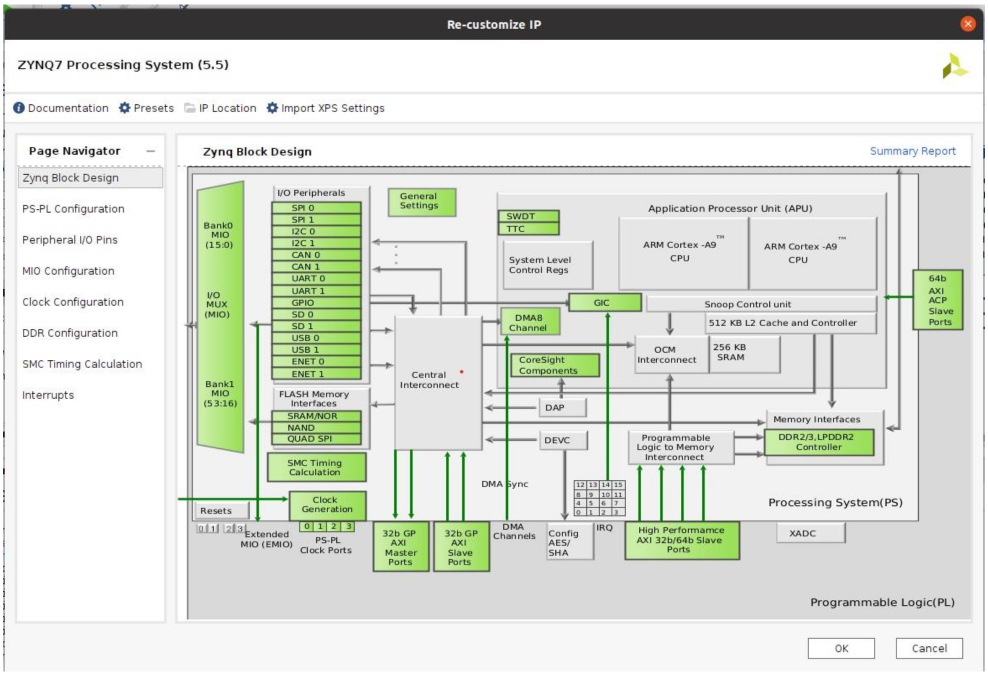
doc
drivers
ExtensionPacks
nls
platforms
sdk
UnattendedTemplates
x86
DbgPlugInDiggers.dll
License_en_US.rtf
msvcp100.dll
msvcr100.dll
Oracle_VM_VirtualBox_Extension_Pack-6.1.42.vbox-extpack
Qt5CoreVBox.dll
Qt5GuiVBox.dll
Qt5OpenGLVBox.dll
Qt5PrintSupportVBox.dll
Qt5WidgetsVBox.dll
Qt5WinExtrasVBox.dll
SDL.dll
UICommon.dll
VBoxAuth.dll
VBoxAuthSimple.dll
VBoxAutostartSvc.exe
VBoxBalloonCtrl.exe
VBoxBugReport.exe
VBoxC.dll
VBoxManage.exe
VBoxNetDHCP.dll
VBoxNetDHCP.exe
VBoxNetNAT.dll
VBoxNetNAT.exe
VBoxProxyStub.dll
VBoxRes.dll
VBoxRT.dll
VBoxSDL.dll
VBoxSDL.exe
VBoxSDS.exe
VBoxSharedClipboard.dll
VBoxSharedFolders.dll
VBoxSupLib.dll
VBoxSVC.exe
VBoxTestOGL.exe
VBoxVMM.dll
VBoxWebSrv.exe
VirtualBox.chm
VirtualBox.exe
VirtualBoxVM.dll
VirtualBoxVM.exe
VMMR0.r0

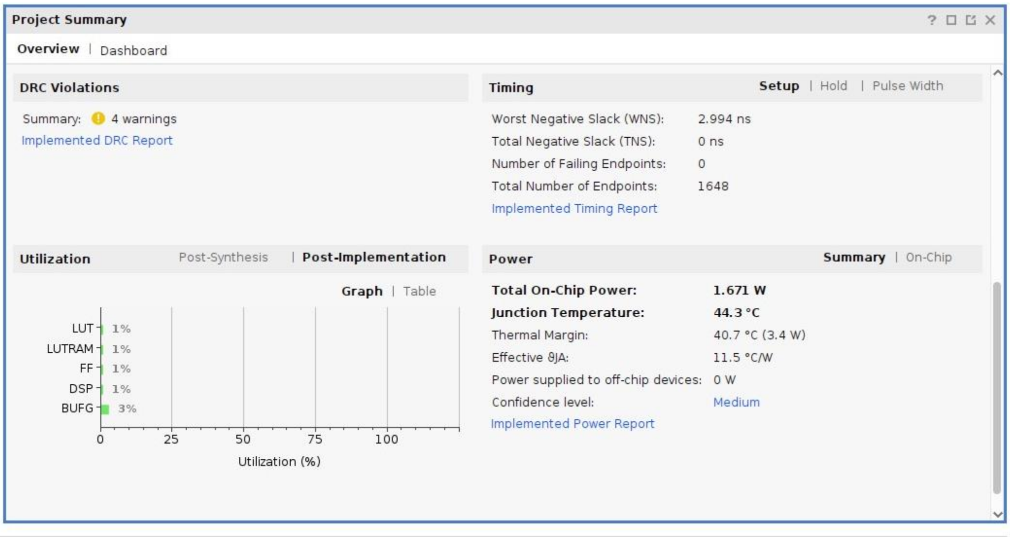
擴充包要放在對的路徑，否則在 sudo apt update 的時候會無法進行下去

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get install libncurses5
sudo apt-get install libtinfo5
sudo apt-get install libncurses5-dev libncursesw5-dev
sudo apt-get install ncurses-compat-libs
```

sudo apt update 在某些電腦需要改成” sudo apt －get update”







DRC Violations

Summary: 4 warnings

Implemented DRC Report

Timing — Setup | Hold | Pulse Width

Worst Negative Slack (WNS): 2.994 ns

Total Negative Slack (TNS): 0 ns

Number of Failing Endpoints: 0

Total Number of Endpoints: 1648

Implemented Timing Report

Utilization — Post-Synthesis | Post-Implementation

Graph | Table

Power — Summary | On-Chip

Total On-Chip Power: **1.671 W**

Junction Temperature: **44.3 °C**

Thermal Margin: 40.7 °C (3.4 W)

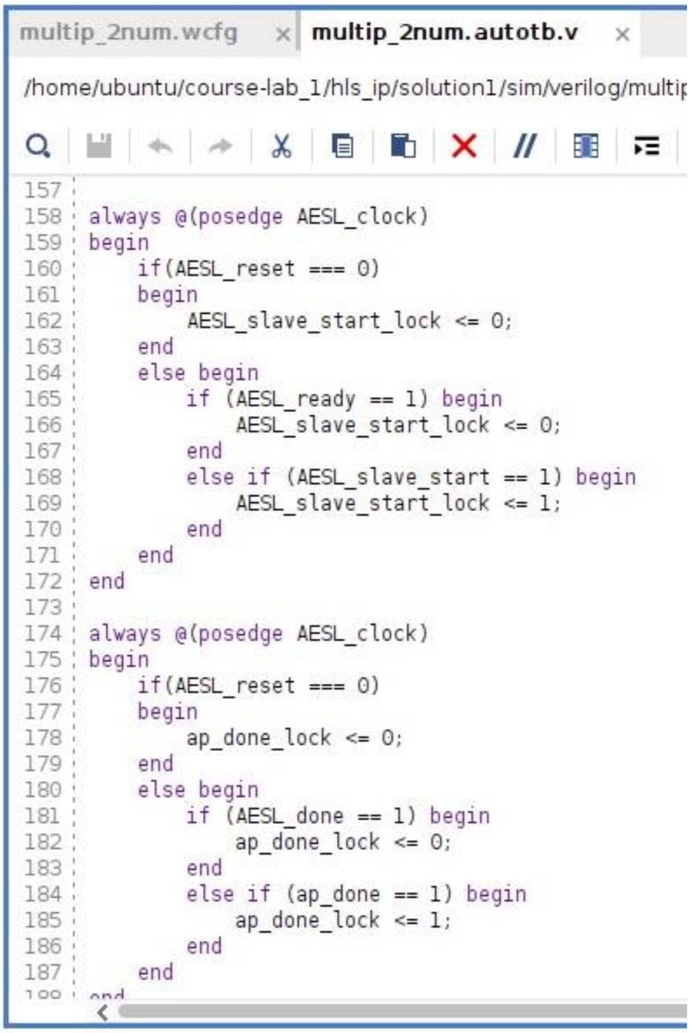
Effective θJA: 11.5 °C/W

Power supplied to off-chip devices: 0 W

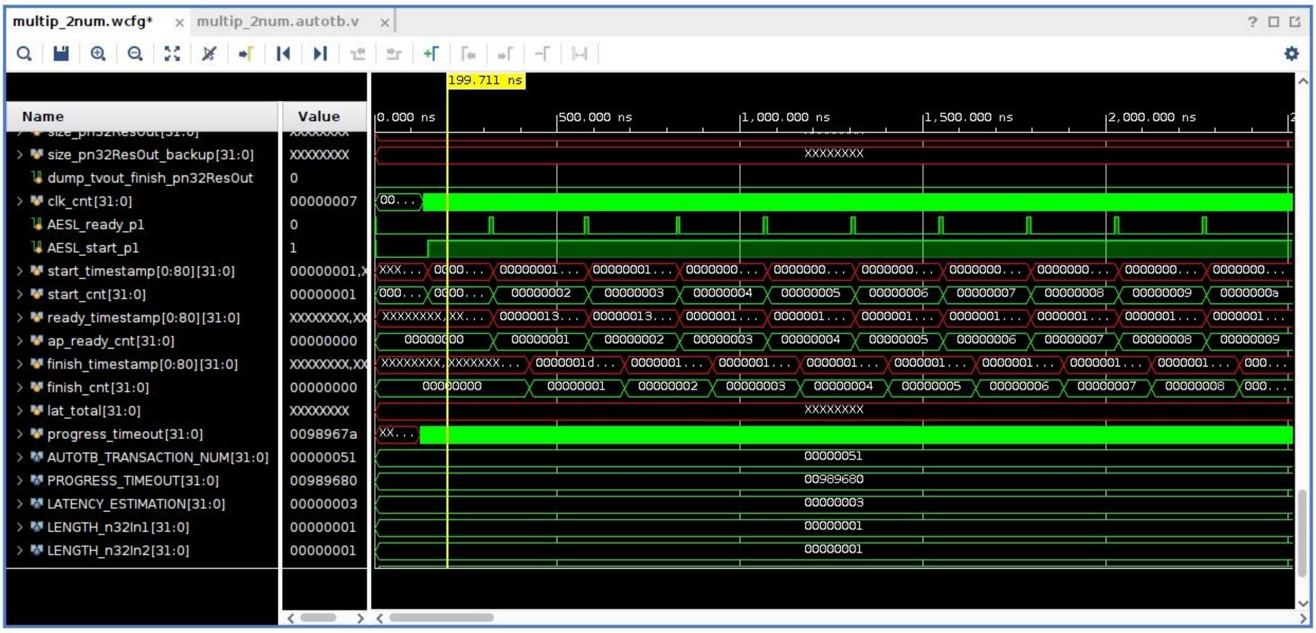
Confidence level: Medium

Implemented Power Report

Vitis 呈現的 power consumption 和 schematic

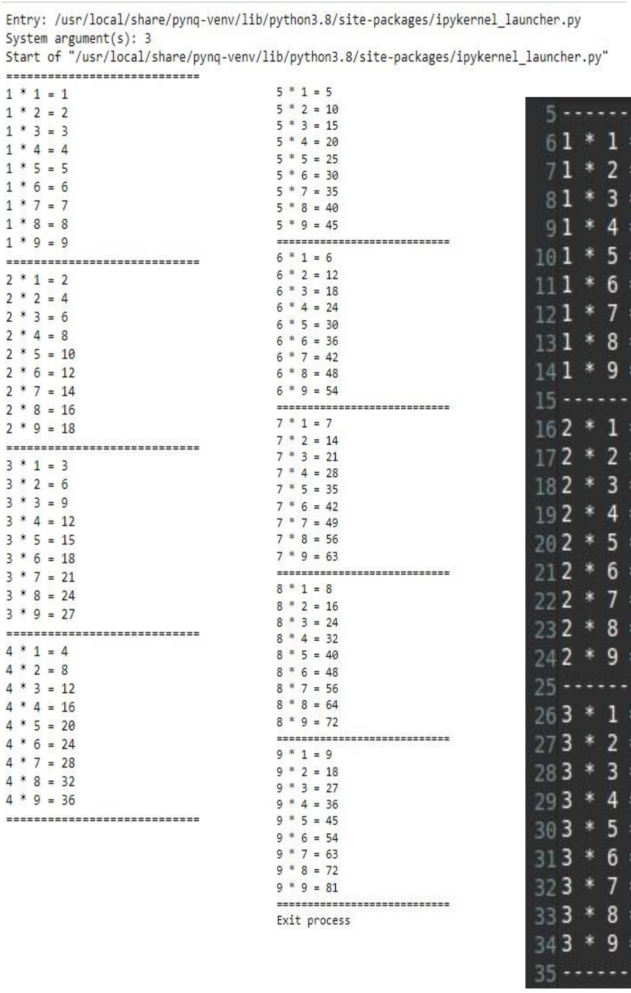

multip_2num.wcfg
multip_2num.autotb.v
/home/ubuntu/course-lab_1/hls_ip/solution1/sim/verilog/multip

```verilog

always @(posedge AESL_clock)
begin
    if(AESL_reset === 0)
    begin
        AESL_slave_start_lock <= 0;
    end
    else begin
        if (AESL_ready == 1) begin
            AESL_slave_start_lock <= 0;
        end
        else if (AESL_slave_start == 1) begin
            AESL_slave_start_lock <= 1;
        end
    end
end

always @(posedge AESL_clock)
begin
    if(AESL_reset === 0)
    begin
        ap_done_lock <= 0;
    end
    else begin
        if (AESL_done == 1) begin
            ap_done_lock <= 0;
        end
        else if (ap_done == 1) begin
            ap_done_lock <= 1;
        end
    end
end
```

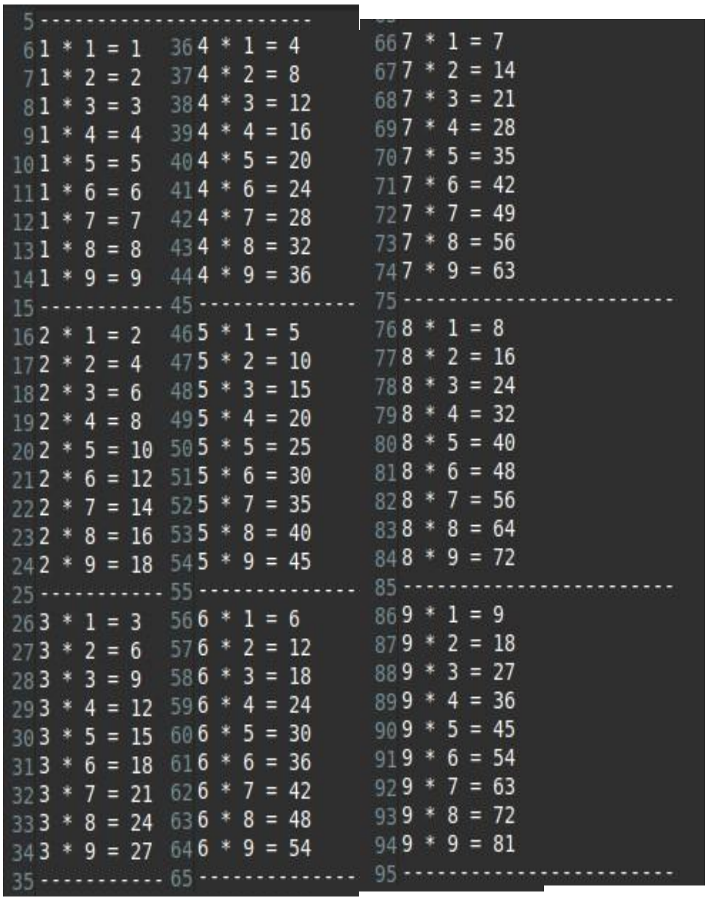


Name	Value
size_pn32ResOut_backup[31:0]	XXXXXXXX
dump_tvout_finish_pn32ResOut	0
clk_cnt[31:0]	00000007
AESL_ready_p1	0
AESL_start_p1	1
start_timestamp[0:80][31:0]	00000001,X
start_cnt[31:0]	00000001
ready_timestamp[0:80][31:0]	XXXXXXXX,XX
ap_ready_cnt[31:0]	00000000
finish_timestamp[0:80][31:0]	XXXXXXXX,XX
finish_cnt[31:0]	00000000
lat_total[31:0]	XXXXXXXX
progress_timeout[31:0]	0098967a
AUTOTB_TRANSACTION_NUM[31:0]	00000051
PROGRESS_TIMEOUT[31:0]	00989680
LATENCY_ESTIMATION[31:0]	00000003
LENGTH_n32In1[31:0]	00000001
LENGTH_n32In2[31:0]	00000001

範例 verilog code 是一個用 HLS 寫出來的高階乘法器，而上圖則是在軟體 Vivado 呈現出來的波型。



```
Entry: /usr/local/share/pynq-venv/lib/python3.8/site-packages/ipykernel_launcher.py
System argument(s): 3
Start of "/usr/local/share/pynq-venv/lib/python3.8/site-packages/ipykernel_launcher.py"
=============================
1 * 1 = 1
1 * 2 = 2
1 * 3 = 3
1 * 4 = 4
1 * 5 = 5
1 * 6 = 6
1 * 7 = 7
1 * 8 = 8
1 * 9 = 9
=============================
2 * 1 = 2
2 * 2 = 4
2 * 3 = 6
2 * 4 = 8
2 * 5 = 10
2 * 6 = 12
2 * 7 = 14
2 * 8 = 16
2 * 9 = 18
=============================
3 * 1 = 3
3 * 2 = 6
3 * 3 = 9
3 * 4 = 12
3 * 5 = 15
3 * 6 = 18
3 * 7 = 21
3 * 8 = 24
3 * 9 = 27
=============================
4 * 1 = 4
4 * 2 = 8
4 * 3 = 12
4 * 4 = 16
4 * 5 = 20
4 * 6 = 24
4 * 7 = 28
4 * 8 = 32
4 * 9 = 36
=============================
5 * 1 = 5
5 * 2 = 10
5 * 3 = 15
5 * 4 = 20
5 * 5 = 25
5 * 6 = 30
5 * 7 = 35
5 * 8 = 40
5 * 9 = 45
=============================
6 * 1 = 6
6 * 2 = 12
6 * 3 = 18
6 * 4 = 24
6 * 5 = 30
6 * 6 = 36
6 * 7 = 42
6 * 8 = 48
6 * 9 = 54
=============================
7 * 1 = 7
7 * 2 = 14
7 * 3 = 21
7 * 4 = 28
7 * 5 = 35
7 * 6 = 42
7 * 7 = 49
7 * 8 = 56
7 * 9 = 63
=============================
8 * 1 = 8
8 * 2 = 16
8 * 3 = 24
8 * 4 = 32
8 * 5 = 40
8 * 6 = 48
8 * 7 = 56
8 * 8 = 64
8 * 9 = 72
=============================
9 * 1 = 9
9 * 2 = 18
9 * 3 = 27
9 * 4 = 36
9 * 5 = 45
9 * 6 = 54
9 * 7 = 63
9 * 8 = 72
9 * 9 = 81
=============================
Exit process
```



```
------------------------
1 * 1 = 1
1 * 2 = 2
1 * 3 = 3
1 * 4 = 4
1 * 5 = 5
1 * 6 = 6
1 * 7 = 7
1 * 8 = 8
1 * 9 = 9
------------------------
2 * 1 = 2
2 * 2 = 4
2 * 3 = 6
2 * 4 = 8
2 * 5 = 10
2 * 6 = 12
2 * 7 = 14
2 * 8 = 16
2 * 9 = 18
------------------------
3 * 1 = 3
3 * 2 = 6
3 * 3 = 9
3 * 4 = 12
3 * 5 = 15
3 * 6 = 18
3 * 7 = 21
3 * 8 = 24
3 * 9 = 27
------------------------
4 * 1 = 4
4 * 2 = 8
4 * 3 = 12
4 * 4 = 16
4 * 5 = 20
4 * 6 = 24
4 * 7 = 28
4 * 8 = 32
4 * 9 = 36
------------------------
5 * 1 = 5
5 * 2 = 10
5 * 3 = 15
5 * 4 = 20
5 * 5 = 25
5 * 6 = 30
5 * 7 = 35
5 * 8 = 40
5 * 9 = 45
------------------------
6 * 1 = 6
6 * 2 = 12
6 * 3 = 18
6 * 4 = 24
6 * 5 = 30
6 * 6 = 36
6 * 7 = 42
6 * 8 = 48
6 * 9 = 54
------------------------
7 * 1 = 7
7 * 2 = 14
7 * 3 = 21
7 * 4 = 28
7 * 5 = 35
7 * 6 = 42
7 * 7 = 49
7 * 8 = 56
7 * 9 = 63
------------------------
8 * 1 = 8
8 * 2 = 16
8 * 3 = 24
8 * 4 = 32
8 * 5 = 40
8 * 6 = 48
8 * 7 = 56
8 * 8 = 64
8 * 9 = 72
------------------------
9 * 1 = 9
9 * 2 = 18
9 * 3 = 27
9 * 4 = 36
9 * 5 = 45
9 * 6 = 54
9 * 7 = 63
9 * 8 = 72
9 * 9 = 81
------------------------
```

這是租借 FPGA 版後，得到的輸出。

對照 vitis 軟體模擬的輸出，基本吻合

我有詳細對 HLS C code 程式的註解筆記，放在 GitHub 上：
https://github.com/leolin0501/soclab.github.io/tree/main