

Vitis HLS Import Vitis Library Flow

Field Application Engineer

Adaptive and Embedded Computing Group (AECG)

Revision History

Date	Version	Description
12/26/23	1.0	Initial version for flow introduction.

© Copyright 2021 Xilinx, Inc. Xilinx, the Xilinx logo, Artix, ISE, Kintex, Spartan, Virtex, Vivado, Zynq, and other designated brands included herein are trademarks of Xilinx in the United States and other countries. All other trademarks are the property of their respective owners.

NOTICE OF DISCLAIMER: The information disclosed to you hereunder (the "Information") is provided "AS-IS" with no warranty of any kind, express or implied. Xilinx does not assume any liability arising from your use of the Information. You are responsible for obtaining any rights you may require for your use of this Information. Xilinx reserves the right to make changes, at any time, to the Information without notice and at its sole discretion. Xilinx assumes no obligation to correct any errors contained in the Information or to advise you of any corrections or updates. Xilinx expressly disclaims any liability in connection with technical support or assistance that may be provided to you in connection with the Information. XILINX MAKES NO OTHER WARRANTIES, WHETHER EXPRESS, IMPLIED, OR STATUTORY, REGARDING THE INFORMATION, INCLUDING ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT OF THIRD-PARTY RIGHTS.

Vitis HLS Import Vitis Library Flow (Version 2021.1)

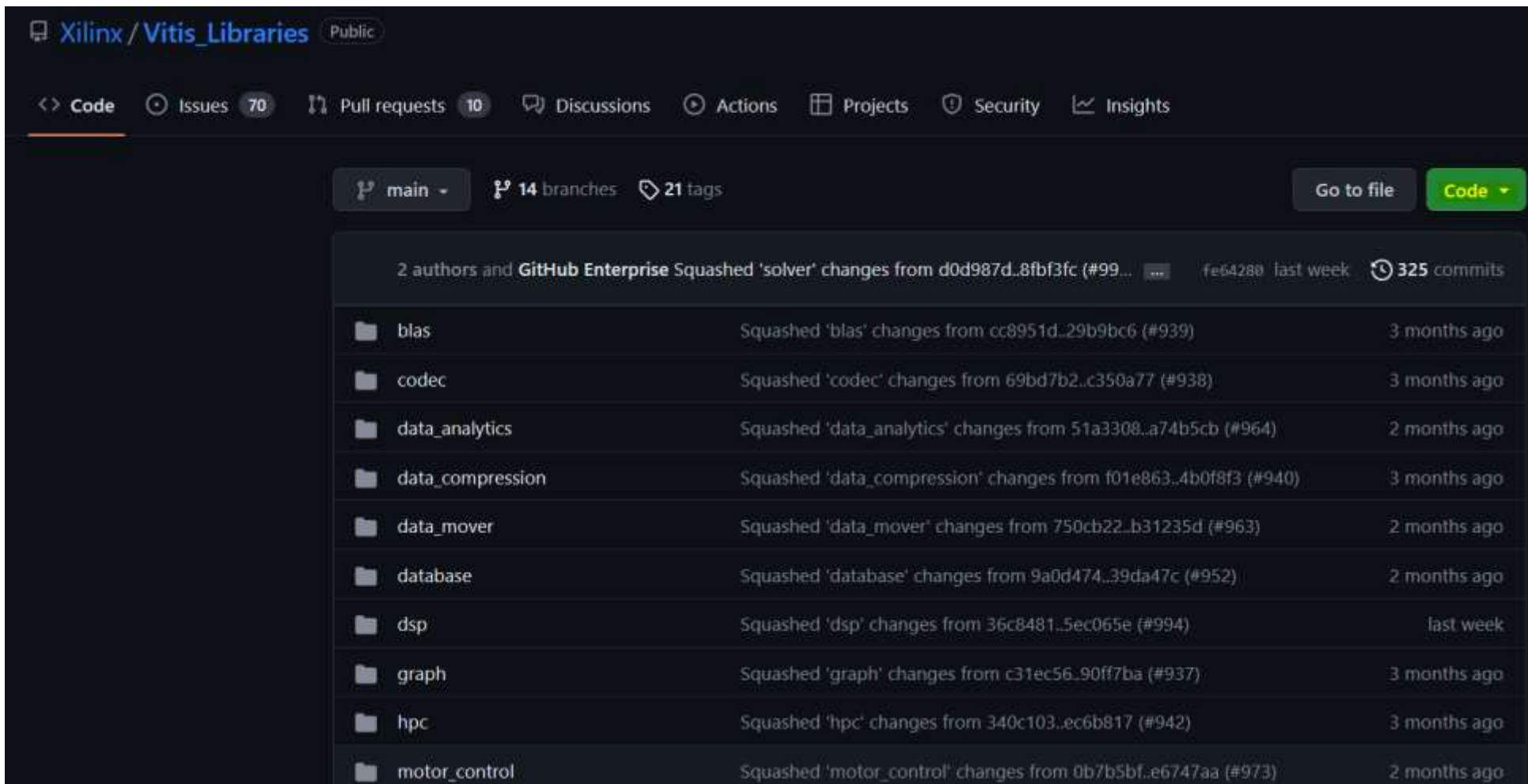
本篇主要介紹如何將 Vitis Library 中用 C/C++ 撰寫的 IP 利用不同版本的 Vitis HLS 匯出到自定義的 FPGA Chip 上

1. 先將整包 Vitis Library 下載下來 [GitHub - Xilinx/Vitis_Libraries: Vitis Libraries](#)
2. 開啟 Vitis HLS Command Prompt
3. 利用寫好的 tcl file include 相對應想要匯出的 Vitis Library
4. 在 Vivado 中測試匯出的 IP 是否可用

Vitis HLS Import Vitis Library Flow (Version 2021.1)

本篇主要介紹如何將 Vitis Library 中用 C/C++ 撰寫的 IP 利用不同版本的 Vitis HLS 匯出到自定義的 FPGA Chip 上

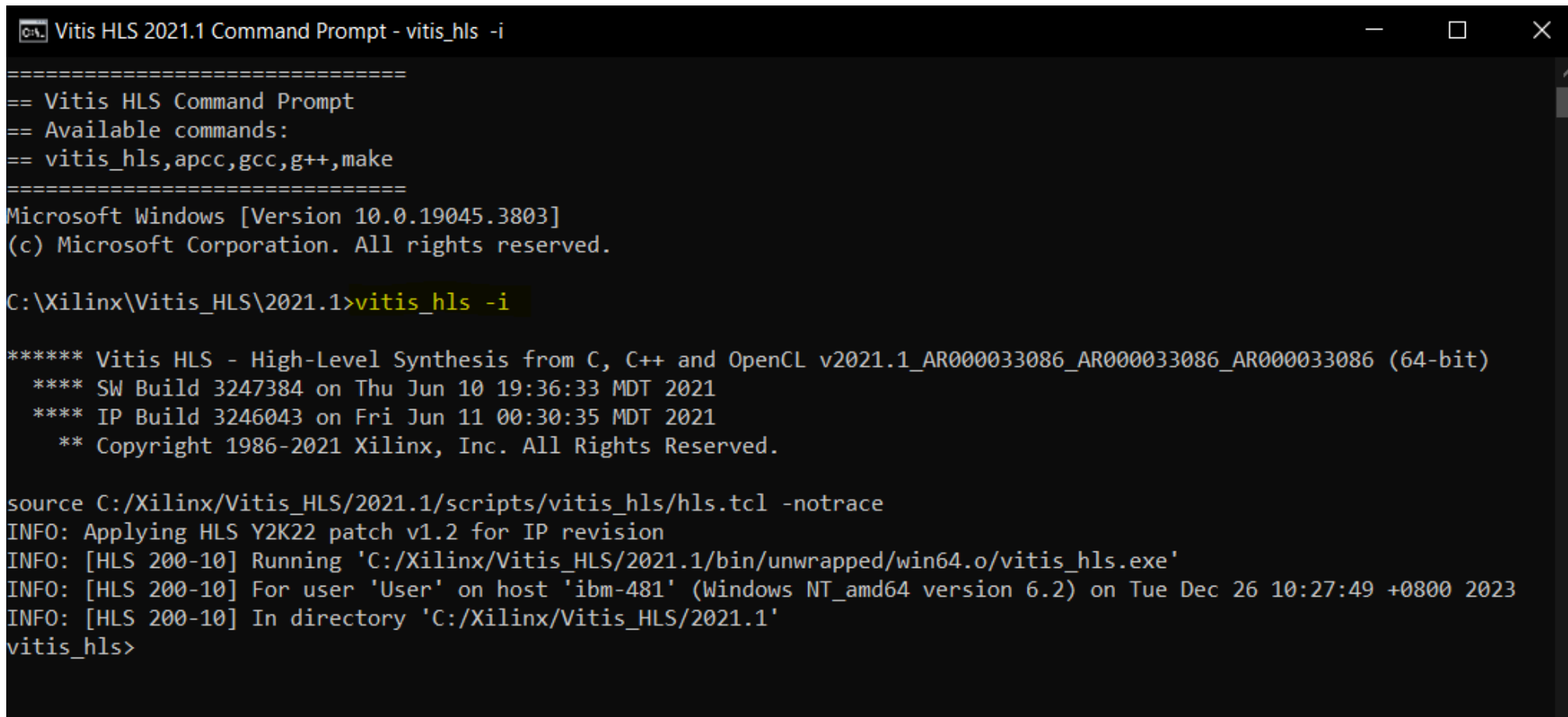
1. 先將整包 Vitis Library 下載下來 [GitHub - Xilinx/Vitis_Libraries: Vitis Libraries](#)



Vitis HLS Import Vitis Library Flow (Version 2021.1)

本篇主要介紹如何將 Vitis Library 中用 C/C++ 撰寫的 IP 利用不同版本的 Vitis HLS 匯出到自定義的 FPGA Chip 上

2. 開啟 Vitis HLS Command Prompt



```
Vitis HLS 2021.1 Command Prompt - vitis_hls -i

=====
== Vitis HLS Command Prompt
== Available commands:
== vitis_hls,apcc,gcc,g++,make
=====

Microsoft Windows [Version 10.0.19045.3803]
(c) Microsoft Corporation. All rights reserved.

C:\Xilinx\Vitis_HLS\2021.1>vitis_hls -i

***** Vitis HLS - High-Level Synthesis from C, C++ and OpenCL v2021.1_AR000033086_AR000033086_AR000033086 (64-bit)
**** SW Build 3247384 on Thu Jun 10 19:36:33 MDT 2021
**** IP Build 3246043 on Fri Jun 11 00:30:35 MDT 2021
** Copyright 1986-2021 Xilinx, Inc. All Rights Reserved.

source C:/Xilinx/Vitis_HLS/2021.1/scripts/vitis_hls/hls.tcl -notrace
INFO: Applying HLS Y2K22 patch v1.2 for IP revision
INFO: [HLS 200-10] Running 'C:/Xilinx/Vitis_HLS/2021.1/bin/unwrapped/win64.o/vitis_hls.exe'
INFO: [HLS 200-10] For user 'User' on host 'ibm-481' (Windows NT_amd64 version 6.2) on Tue Dec 26 10:27:49 +0800 2023
INFO: [HLS 200-10] In directory 'C:/Xilinx/Vitis_HLS/2021.1'
vitis_hls>
```

Vitis HLS Import Vitis Library Flow (Version 2021.1)

本篇主要介紹如何將 Vitis Library 中用 C/C++ 撰寫的 IP 利用不同版本的 Vitis HLS 匯出到自定義的 FPGA Chip 上

3. 利用寫好的 tcl file include 相對應想要匯出的 Vitis Library

```
set VITIS_LIBS ../../../../Vitis_Libraries/motor_control
set SVPWM_IP ${VITIS_LIBS}/L1/tests/IP_SVPWM
```

設定要 include Vitis Library 路徑

```
set PROJ "ip_svpwm_duty.prj"
set SOLN "sol1"
```

Project Name & Solution Name

```
if {[info exists CLKP]} {
    set CLKP 10
}
```

Clock 速度，10 = 100MHz，5 = 200MHz

```
if {[info exists XPART]} {
    set XPART xc7z020-clg484-1
}
```

設定 FPGA Parts

```
open_project -reset $PROJ
```

```
add_files "${SVPWM_IP}/src/ip_svpwm.cpp" -cflags "-I${VITIS_LIBS}/L1/include/hw -I${SVPWM_IP}/src"
add_files -tb "${SVPWM_IP}/src/test_svpwm.cpp" -cflags "-I${VITIS_LIBS}/L1/include/hw -I${SVPWM_IP}/src"
set_top hls_svpwm_duty
```

```
open_solution -reset $SOLN
```

```
set_part $XPART
create_clock -period $CLKP
set_clock_uncertainty 1.25
```

```
csynth_design
config_export -ipname hls_svpwm_duty
export_design -rtl verilog -format ip_catalog
```

```
exit
```

這邊會直接跑完 Synthesis、Implementation 和 export IP

Vitis HLS Import Vitis Library Flow (Version 2021.1)

本篇主要介紹如何將 Vitis Library 中用 C/C++ 撰寫的 IP 利用不同版本的 Vitis HLS 匯出到自定義的 FPGA Chip 上

3. source 寫好的 tcl file

```
Vitis HLS 2021.1 Command Prompt
=====
== Vitis HLS Command Prompt
== Available commands:
== vitis_hls,apcc,gcc,g++,make
=====
Microsoft Windows [Version 10.0.19045.3803]
(c) Microsoft Corporation. All rights reserved.

C:\Xilinx\Vitis_HLS\2021.1>vitis_hls -i

***** Vitis HLS - High-Level Synthesis from C, C++ and OpenCL v2021.1_AR000033086_AR000033086_AR000033086 (64-bit)
***** SW Build 3247384 on Thu Jun 10 19:36:33 MDT 2021
***** IP Build 3246043 on Fri Jun 11 00:30:35 MDT 2021
***** Copyright 1986-2021 Xilinx, Inc. All Rights Reserved.

source C:/Xilinx/Vitis_HLS/2021.1/scripts/vitis_hls/hls.tcl -notrace
INFO: Applying HLS Y2K22 patch v1.2 for IP revision
INFO: [HLS 200-10] Running 'C:/Xilinx/Vitis_HLS/2021.1/bin/unwrapped/win64.o/vitis_hls.exe'
INFO: [HLS 200-10] For user 'User' on host 'ibm-481' (Windows NT_amd64 version 6.2) on Tue Dec 26 10:27:49 +0800 2023
INFO: [HLS 200-10] In directory 'C:/Xilinx/Vitis_HLS/2021.1'
vitis_hls> source ./run_hls.tcl
INFO: [HLS 200-1510] Running: open_project -reset ip_svpwm_duty.prj
INFO: [HLS 200-10] Creating and opening project 'C:/Xilinx/Vitis_HLS/2021.1/ip_svpwm_duty.prj'.
INFO: [HLS 200-1510] Running: add_files ../../Vitis_Libraries/motor_control/L1/tests/IP_SVPWM/src/ip_svpwm.cpp -cflags
-I../../Vitis_Libraries/motor_control/L1/include/hw -I../../Vitis_Libraries/motor_control/L1/tests/IP_SVPWM/src
INFO: [HLS 200-10] Adding design file '../../Vitis_Libraries/motor_control/L1/tests/IP_SVPWM/src/ip_svpwm.cpp' to the
project
INFO: [HLS 200-1510] Running: add_files -tb ../../Vitis_Libraries/motor_control/L1/tests/IP_SVPWM/src/test_svpwm.cpp
-cflags -I../../Vitis_Libraries/motor_control/L1/include/hw -I../../Vitis_Libraries/motor_control/L1/tests/IP_SVPWM/src
```

Vitis HLS Import Vitis Library Flow (Version 2021.1)

本篇主要介紹如何將 Vitis Library 中用 C/C++ 撰寫的 IP 利用不同版本的 Vitis HLS 匯出到自定義的 FPGA Chip 上

3. 成功會像下面這樣

```
Vitis HLS 2021.1 Command Prompt
INFO: [VHDL 208-304] Generating VHDL RTL for hls_svpwm_duty.
INFO: [VLOG 209-307] Generating Verilog RTL for hls_svpwm_duty.
INFO: [HLS 200-789] **** Estimated Fmax: 100.97 MHz
INFO: [HLS 200-111] Finished Command csynth_design CPU user time: 8 seconds. CPU system time: 1 seconds. Elapsed time: 1
9.717 seconds; current allocated memory: 468.131 MB.
INFO: [HLS 200-1510] Running: config_export -ipname hls_svpwm_duty
INFO: [HLS 200-1510] Running: export_design -rtl verilog -format ip_catalog
INFO: [IMPL 213-8] Exporting RTL as a Vivado IP.

***** Vivado v2021.1_AR000033086_AR000033086 (64-bit)
***** SW Build 3247384 on Thu Jun 10 19:36:33 MDT 2021
***** IP Build 3246043 on Fri Jun 11 00:30:35 MDT 2021
** Copyright 1986-2021 Xilinx, Inc. All Rights Reserved.

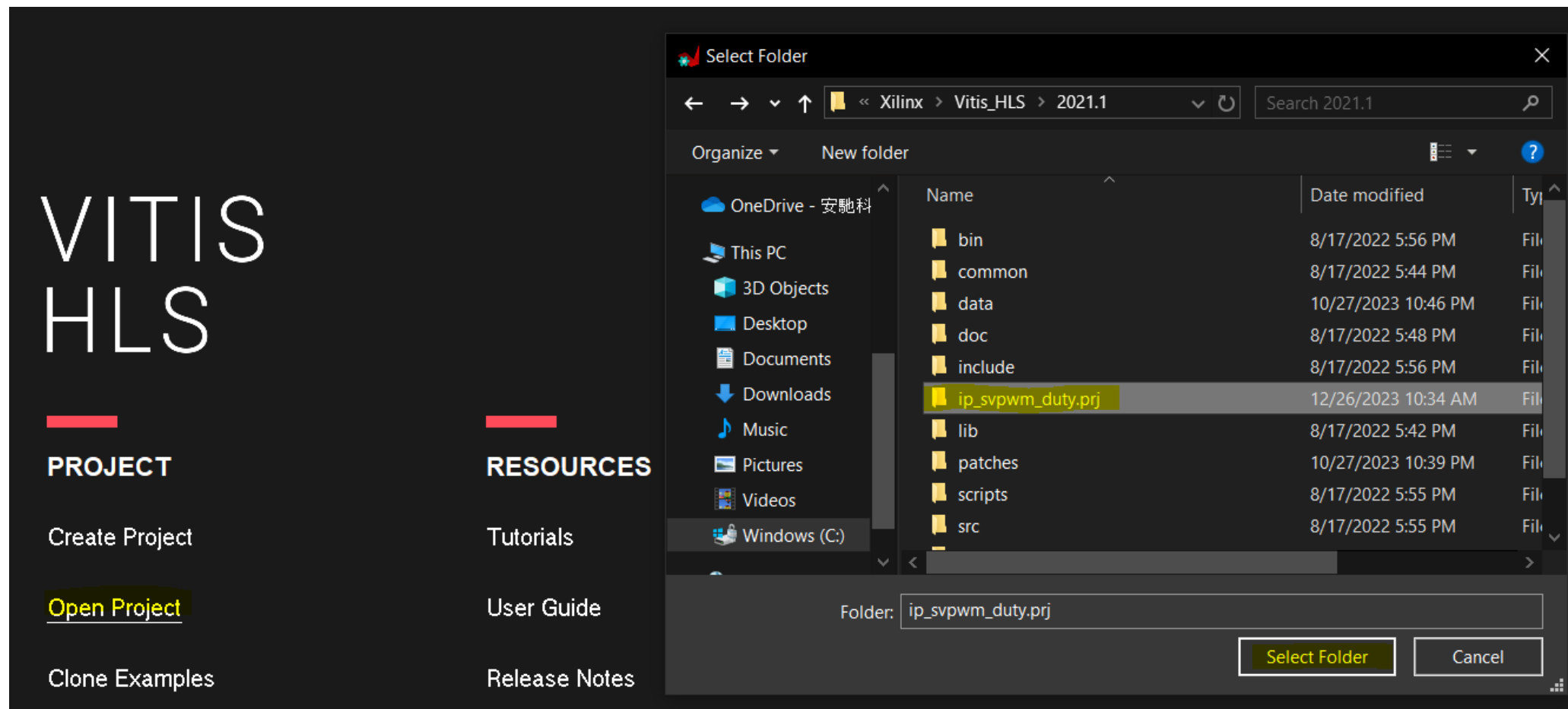
Sourcing tcl script 'C:/Xilinx/Vivado/2021.1/scripts/Vivado_init.tcl'
source run_ippack.tcl -notrace
INFO: [IP_Flow 19-234] Refreshing IP repositories
INFO: [IP_Flow 19-1704] No user IP repositories specified
INFO: [IP_Flow 19-2313] Loaded Vivado IP repository 'C:/Xilinx/Vivado/2021.1/data/ip'.
INFO: [Common 17-206] Exiting Vivado at Tue Dec 26 10:34:41 2023...
INFO: [HLS 200-802] Generated output file ip_svpwm_duty.prj/sol1/impl/export.zip
INFO: [HLS 200-111] Finished Command export_design CPU user time: 1 seconds. CPU system time: 0 seconds. Elapsed time: 8
.582 seconds; current allocated memory: 471.738 MB.
INFO: [HLS 200-112] Total CPU user time: 12 seconds. Total CPU system time: 2 seconds. Total elapsed time: 411.822 secon
ds; peak allocated memory: 468.144 MB.
INFO: [Common 17-206] Exiting vitis_hls at Tue Dec 26 10:34:41 2023...

C:\Xilinx\Vitis_HLS\2021.1>
```


Vitis HLS Import Vitis Library Flow (Version 2021.1)

本篇主要介紹如何將 Vitis Library 中用 C/C++ 撰寫的 IP 利用不同版本的 Vitis HLS 匯出到自定義的 FPGA Chip 上

3. 可以用 Vitis HLS GUI 開看看剛剛 tcl 建立好的專案



Vitis HLS Import Vitis Library Flow (Version 2021.1)

本篇主要介紹如何將 Vitis Library 中用 C/C++ 撰寫的 IP 利用不同版本的 Vitis HLS 匯出到自定義的 FPGA Chip 上

3. 這邊就會給出相對應 synthesis 的報告以及整個專案的 source code 和 constraint

The screenshot displays the Vitis HLS 2021.1 Synthesis Summary report for the project 'hls_svpwm_duty'. The interface includes a File menu, Explorer, Module Hierarchy, Flow Navigator, and Synthesis Summary panels.

Synthesis Summary Report of 'hls_svpwm_duty'

General Information

- Date: Tue Dec 26 10:34:30 2023
- Version: 2021.1_AR000033086_AR000033086 (Build 3247384 on Thu Jun 10 19:36:33 MDT 2021)
- Project: ip_svpwm_duty.prj
- Solution: sol1 (Vivado IP Flow Target)
- Product family: zynq
- Target device: xc7z020-dlg484-1

Timing Estimate

Target	Estimated	Uncertainty
10.00 ns	9.904 ns	1.25 ns

Performance & Resource Estimates

Modules & Loops	Issue Type	Violation Type	Distance	Slack	Latency(cycles)	Latency(ns)	Iteration Latency	Interval	Trips Count	Pipelined	BRAM	DSR	FF	LUT	URAM
hls_svpwm_duty															
hls_svpwm_duty_axi_ap_fixed_24_8_5_3_0_ap_ufixed_16_0_5_3_0	Timing Violation														

HW Interfaces

Interface	Data Width	Address Width	Offset	Register
AXI4LITE	32	32	0	1

Console

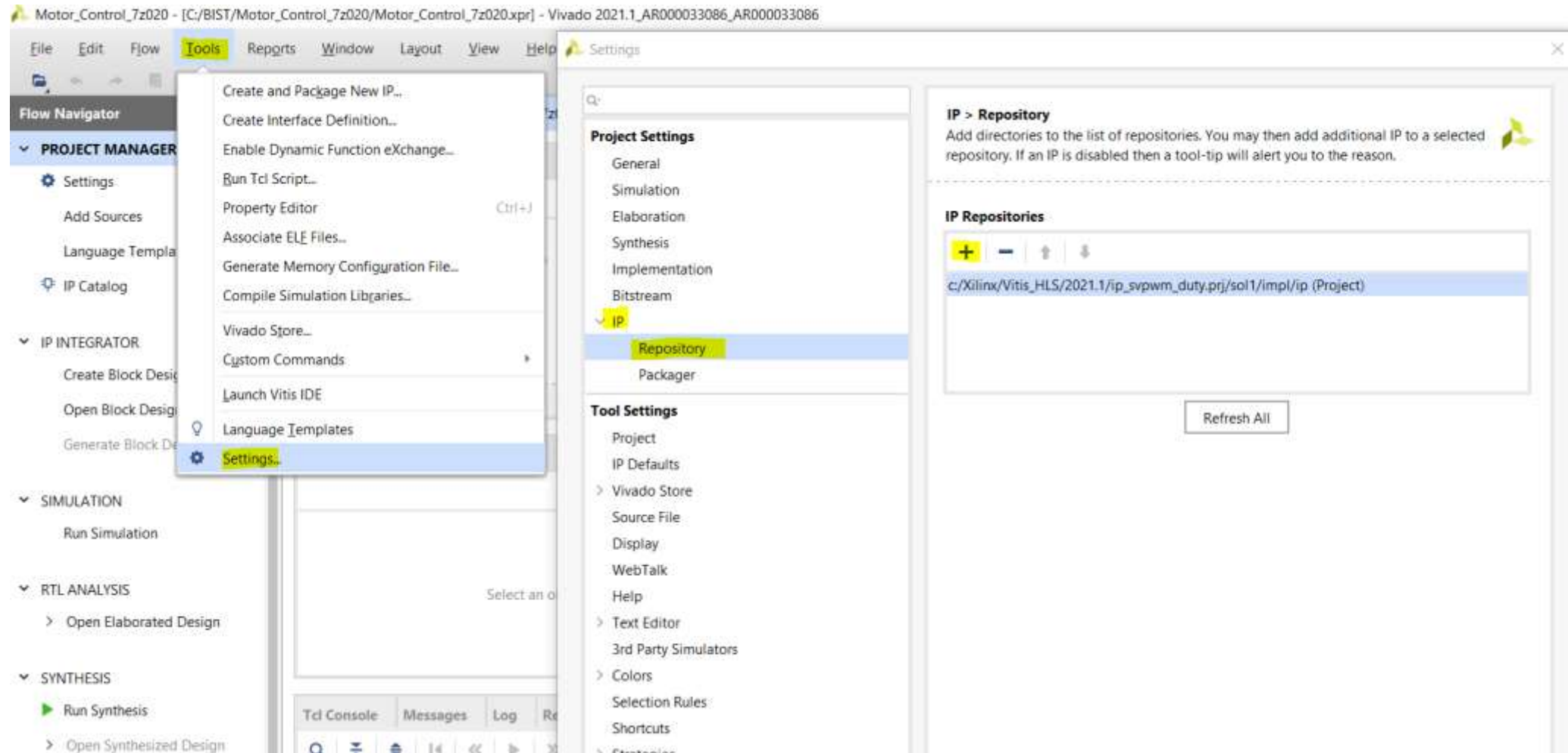
29 Guidance-Infos, 2 Guidance-Warnings, 0 Guidance-Errors

Applying dataflow to function 'xt::motorcontrol-hls_svpwm_duty_axi-ap_fixed<24, 8, (ap_o_model)>, (ap_o_model)>, (ap_o_model)>, (ap_o_model)>'.
ZedBoard: xc7z020-dlg484-1, Freq: 100.0 MHz, Depth: 1, Issue: 1000000000, 1000000000.

Vitis HLS Import Vitis Library Flow (Version 2021.1)

本篇主要介紹如何將 Vitis Library 中用 C/C++ 撰寫的 IP 利用不同版本的 Vitis HLS 匯出到自定義的 FPGA Chip 上

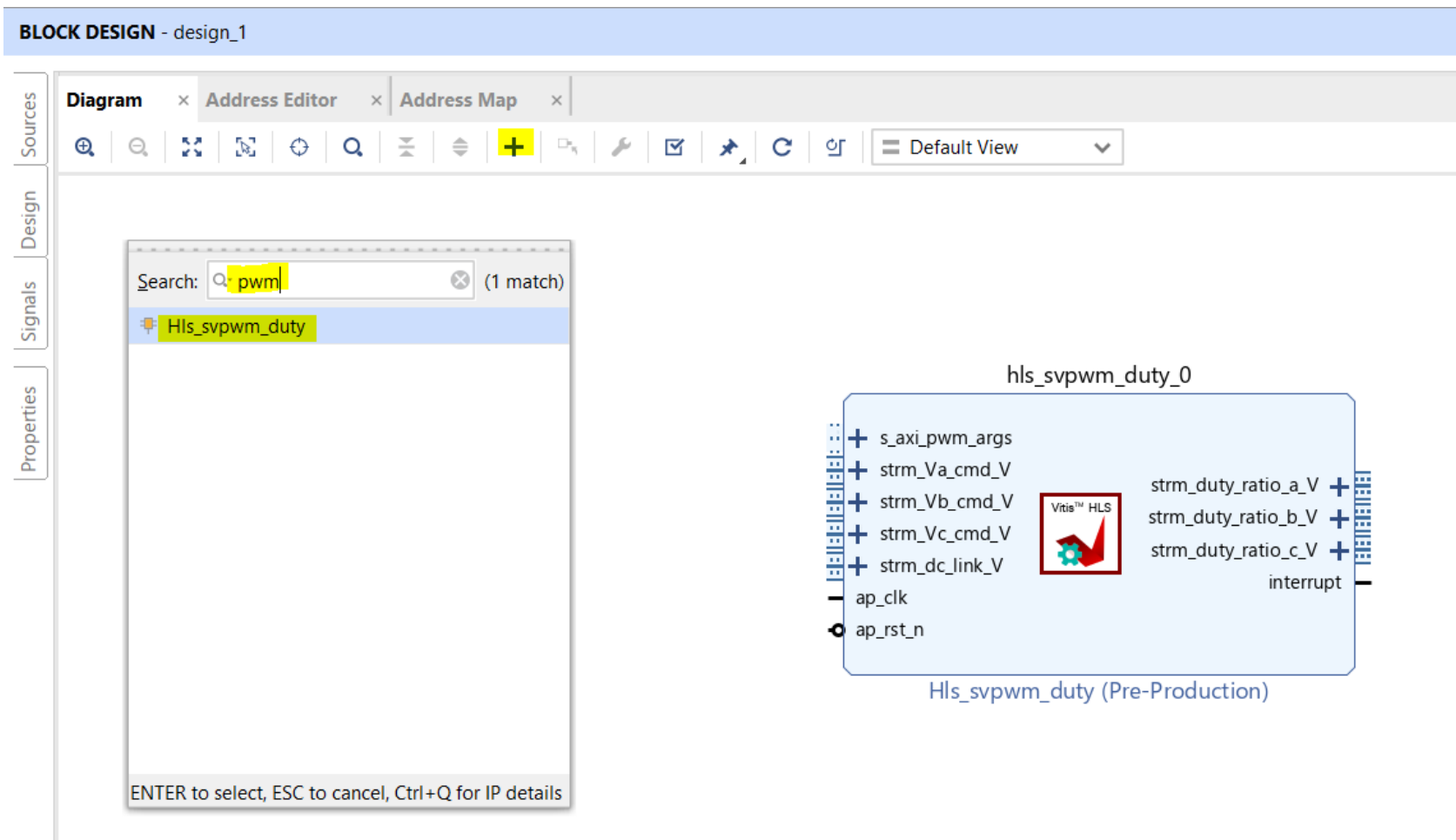
4. 在 Vivado 中測試匯出的 IP 是否可用



Vitis HLS Import Vitis Library Flow (Version 2021.1)

本篇主要介紹如何將 Vitis Library 中用 C/C++ 撰寫的 IP 利用不同版本的 Vitis HLS 匯出到自定義的 FPGA Chip 上

4. 在 Vivado 中測試匯出的 IP 是否可用



Reference

1. [GitHub - Xilinx/Vitis_Libraries: Vitis Libraries](#)
2. [kria-vitis-platforms/kd240/platforms/vivado/ip/svpwm_duty/run_hls.tcl at main · Xilinx/kria-vitis-platforms · GitHub](#)

