SOC Design hw2

108061207 陳伯丞

Part1: AXI-Master Interface

1. VITIS

```
Vitis HLS Console
>> Start test!
>> Comparing against output data...
>> Test passed!
-----------------------
INFO: [SIM 211-1] CSim done with 0 errors.
INFO: [SIM 211-3] *************** CSIM finish ***************
INFO: [HLS 200-111] Finished Command csim_design CPU user time: 0.2 seconds. CPU system time: 0.24 seconds. Elapsed time: 0.77
INFO: [HLS 200-112] Total CPU user time: 1.19 seconds. Total CPU system time: 2.1 seconds. Total elapsed time: 6.59 seconds; p
Finished C simulation.
```

Figure1. C simulation

FIR.cpp | directives.tcl × | Synthesis Summary(solution1) | Co-simulation Report(solution1)

```
1  ############################################################
2  ## This file is generated automatically by Vitis HLS.
3  ## Please DO NOT edit it.
4  ## Copyright 1986-2022 Xilinx, Inc. All Rights Reserved.
5  ############################################################
6  set_directive_interface -mode m_axi -depth 600 -offset slave "fir_n11_maxi" pn32HPInput
7  set_directive_interface -mode m_axi -depth 600 -offset slave "fir_n11_maxi" pn32HPOutput
8  set_directive_interface -mode s_axilite "fir_n11_maxi" an32Coef
9  set_directive_interface -mode s_axilite "fir_n11_maxi" regXferLeng
10 set_directive_interface -mode s_axilite -depth 600 "fir_n11_maxi" AXI-lite
11
```

Figure2. directives

FIR.cpp | FIR.h | FIRTester.cpp | Synthesis Summary(solution1) ×

Synthesis Summary Report of 'fir_n11_maxi'

General Information

Date:	Wed Oct 4 00:43:06 2023	Solution:	solution1 (Vivado IP Flow Target)
Version:	2022.1 (Build 3526262 on Mon Apr 18 15:47:01 MDT 2022)	Product family:	zynquplus
Project:	hsp_ip	Target device:	xck26-sfvc784-2LV-c

Timing Estimate

Target	Estimated	Uncertainty
10.00 ns	7.300 ns	2.70 ns

Performance & Resource Estimates

Modules ✓ Loops ✓

Modules & Loops	Issue Type	Violation Type	Distance	Slack	Latency(cycles)	Latency(ns)	Iteration Latenc
fir_n11_maxi				-	-	-	
fir_n11_maxi_Pipeline_XFER_LOOP				-	-	-	

HW Interfaces

Figure3. C synthesis

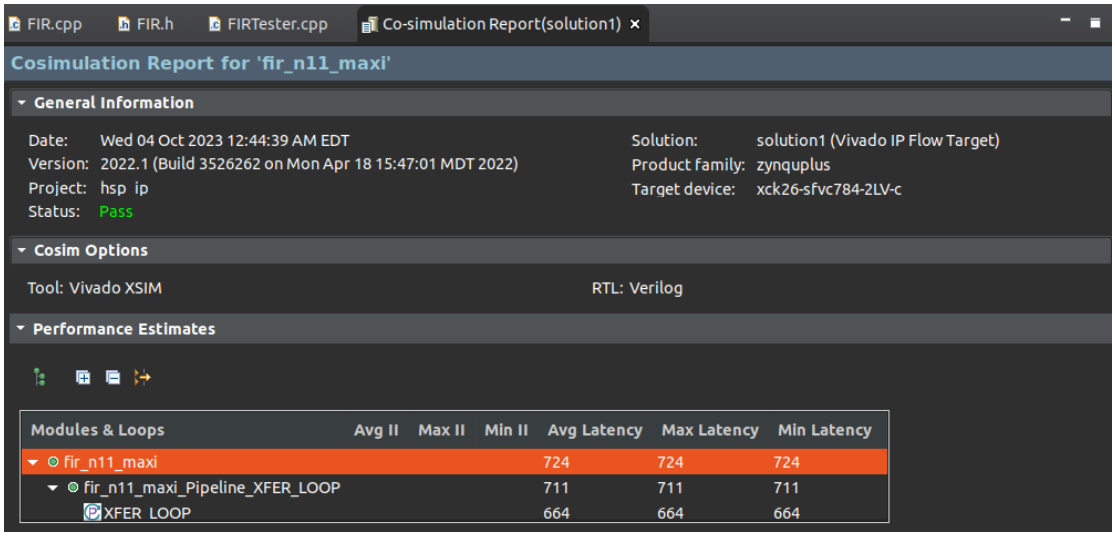


Figure4. Co-simulation

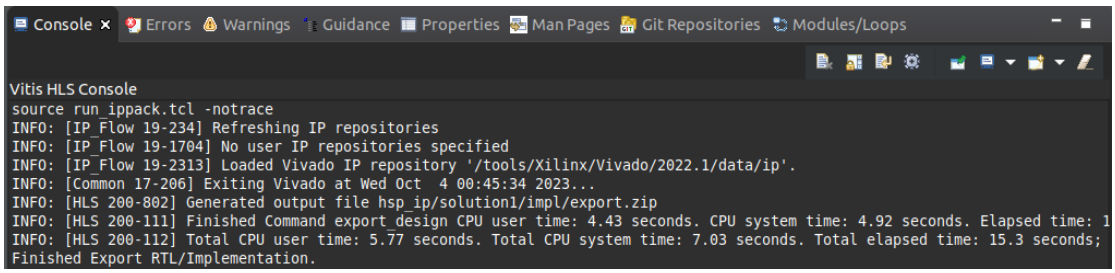


Figure5. Export RTL

2. VIVADO

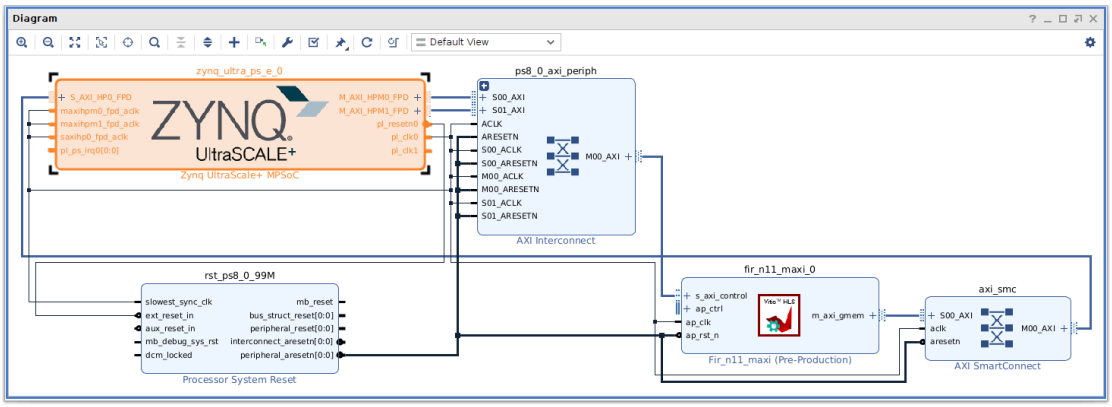


Figure6. Block Diagram

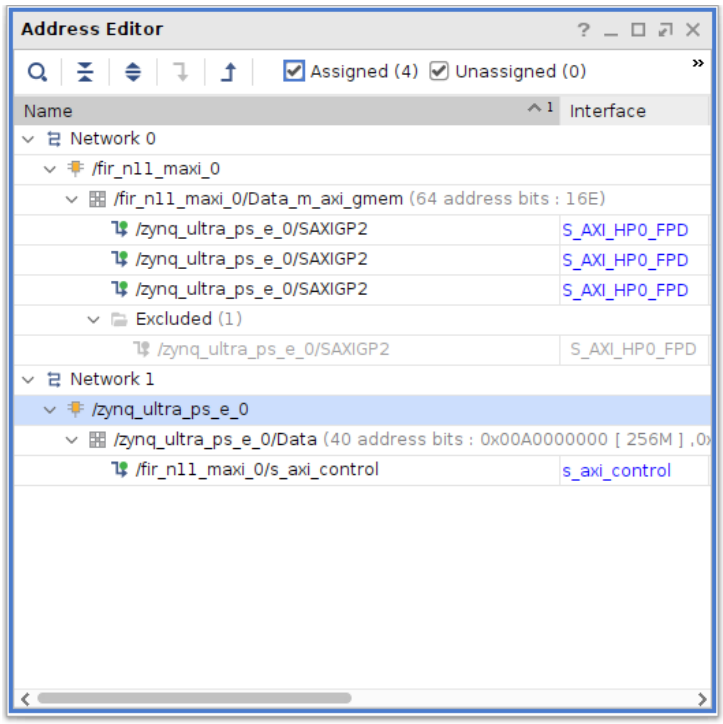


Figure7. Address Editor

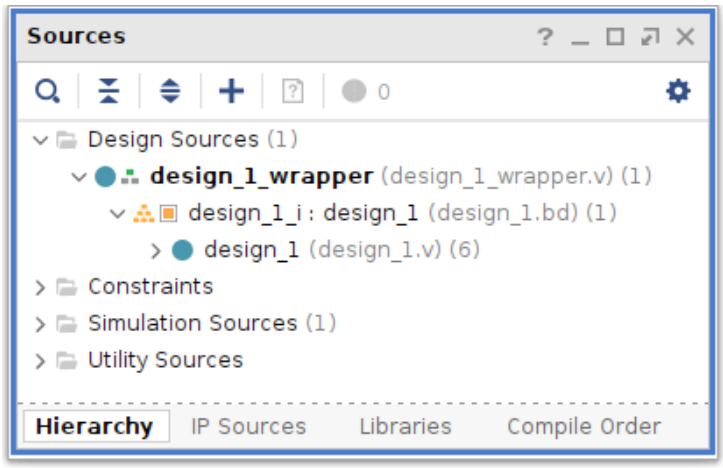


Figure8. Create Wrapper

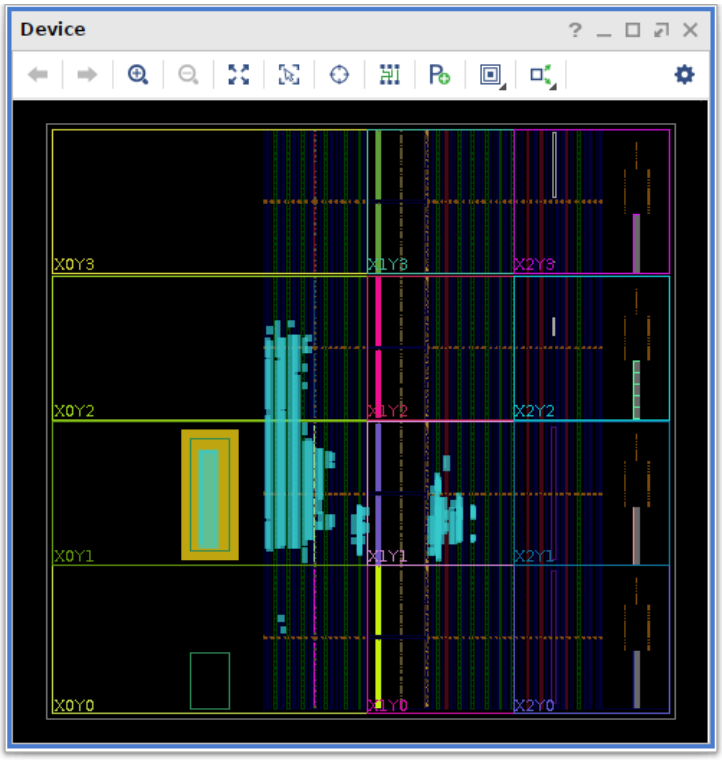


Figure9. Output of Bitstream Generating

依照 Lab1 和 Lab2 Workbook 的步驟完成了 VITIS 和 VIVADO 的操作，接著將產生的 bitstream file 上傳至 kv260 板進行模擬，但是卻無法產生預期的結果。

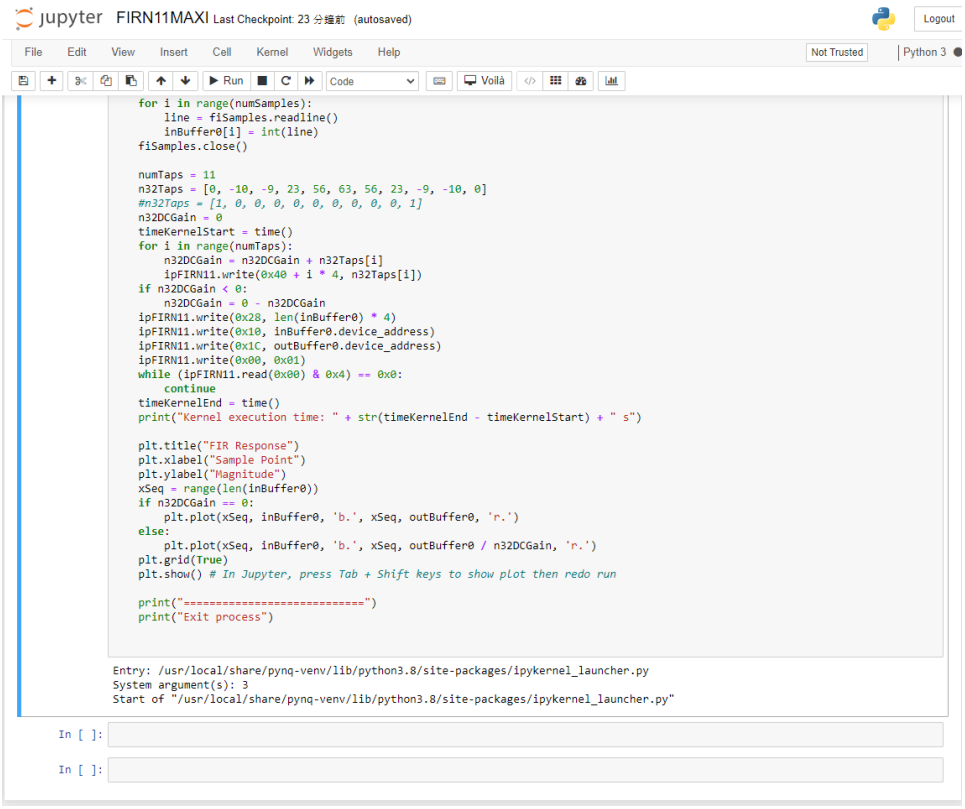


Figure10. Failed Jupyter output

經過反覆查看以及在討論區上討論，發現 Block Diagram 中多出了一個名叫”ap_ctl”，而最後的解決方法是更改 interface 的宣告方式，不再使用 directives 而是直接寫在 cpp 檔中。

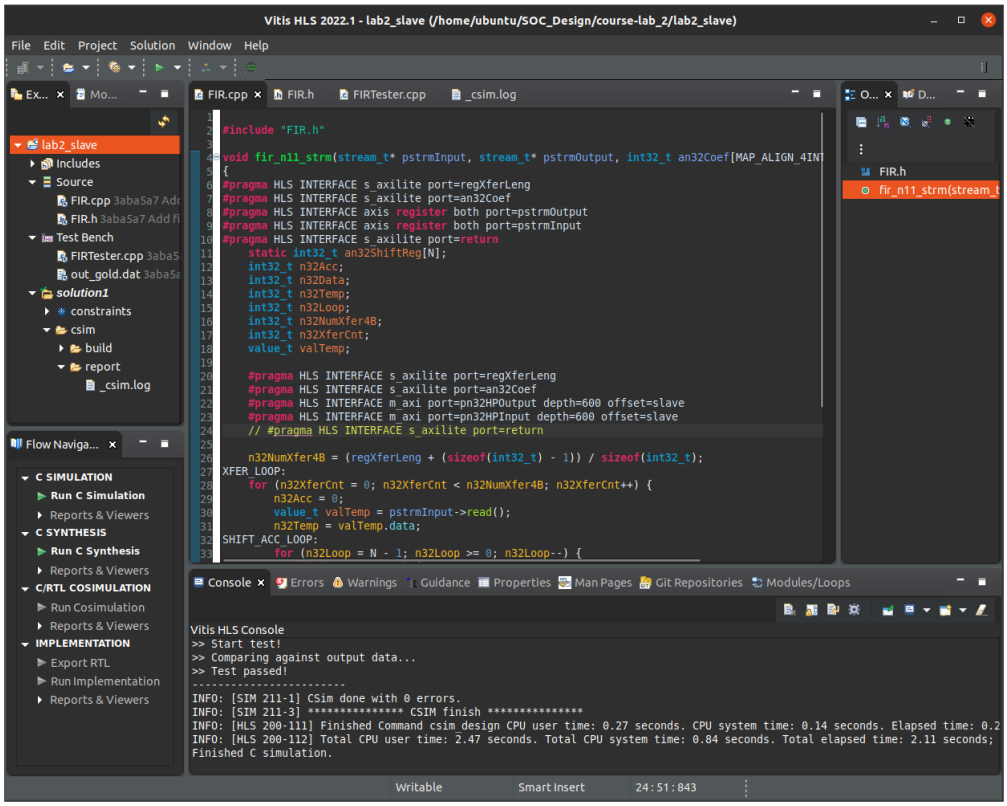


Figure11. Using pragma to define interface

更改後再重新完成Figure3~Figure9的步驟。

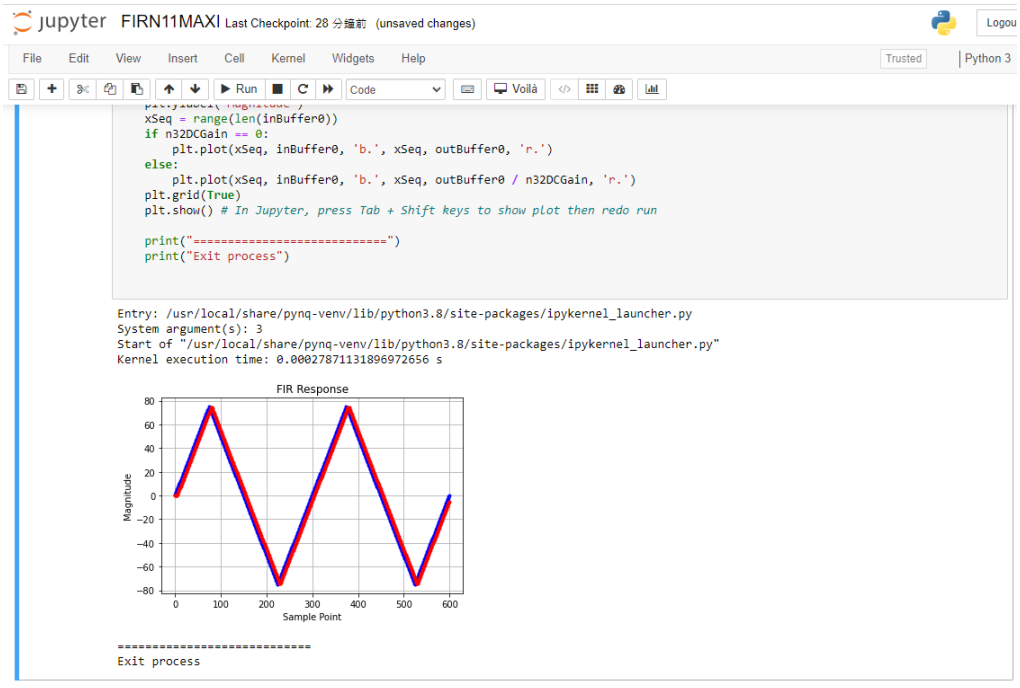


Figure12. Jupyter Output

Part2: Stream Interface

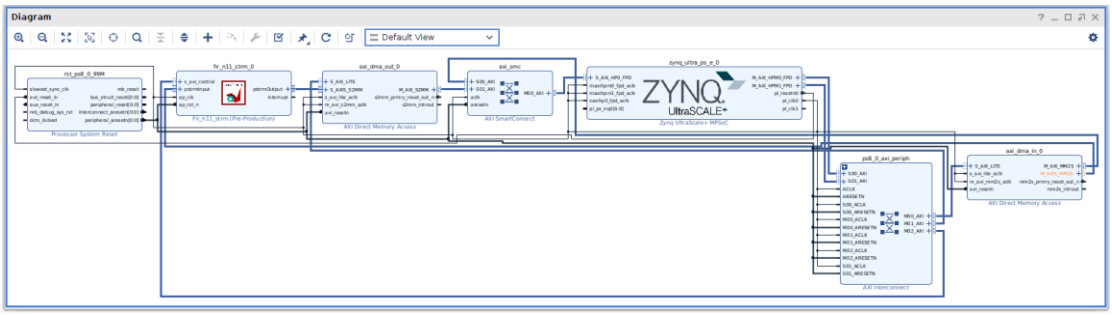
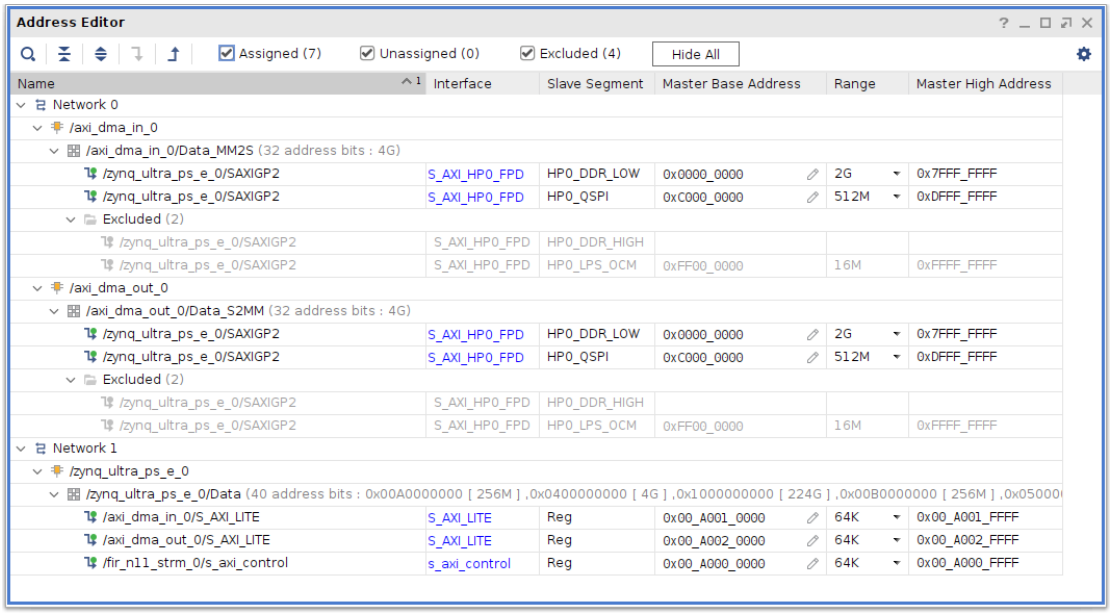


Figure13. Block Diagram



Name	Interface	Slave Segment	Master Base Address	Range	Master High Address
Network 0					
/axi_dma_in_0					
/axi_dma_in_0/Data_MM2S (32 address bits : 4G)					
/zynq_ultra_ps_e_0/SAXIGP2	S_AXI_HP0_FPD	HP0_DDR_LOW	0x0000_0000	2G	0x7FFF_FFFF
/zynq_ultra_ps_e_0/SAXIGP2	S_AXI_HP0_FPD	HP0_QSPI	0xC000_0000	512M	0xDFFF_FFFF
Excluded (2)					
/zynq_ultra_ps_e_0/SAXIGP2	S_AXI_HP0_FPD	HP0_DDR_HIGH			
/zynq_ultra_ps_e_0/SAXIGP2	S_AXI_HP0_FPD	HP0_LPS_OCM	0xFF00_0000	16M	0xFFFF_FFFF
/axi_dma_out_0					
/axi_dma_out_0/Data_S2MM (32 address bits : 4G)					
/zynq_ultra_ps_e_0/SAXIGP2	S_AXI_HP0_FPD	HP0_DDR_LOW	0x0000_0000	2G	0x7FFF_FFFF
/zynq_ultra_ps_e_0/SAXIGP2	S_AXI_HP0_FPD	HP0_QSPI	0xC000_0000	512M	0xDFFF_FFFF
Excluded (2)					
/zynq_ultra_ps_e_0/SAXIGP2	S_AXI_HP0_FPD	HP0_DDR_HIGH			
/zynq_ultra_ps_e_0/SAXIGP2	S_AXI_HP0_FPD	HP0_LPS_OCM	0xFF00_0000	16M	0xFFFF_FFFF
Network 1					
/zynq_ultra_ps_e_0					
/zynq_ultra_ps_e_0/Data (40 address bits : 0x00A0000000 [256M] ,0x0400000000 [4G] ,0x1000000000 [224G] ,0x00B0000000 [256M] ,0x0500					
/axi_dma_in_0/S_AXI_LITE	S_AXI_LITE	Reg	0x00_A001_0000	64K	0x00_A001_FFFF
/axi_dma_out_0/S_AXI_LITE	S_AXI_LITE	Reg	0x00_A002_0000	64K	0x00_A002_FFFF
/fir_n11_strm_0/s_axi_control	s_axi_control	Reg	0x00_A000_0000	64K	0x00_A000_FFFF

Figure14. Address Editor

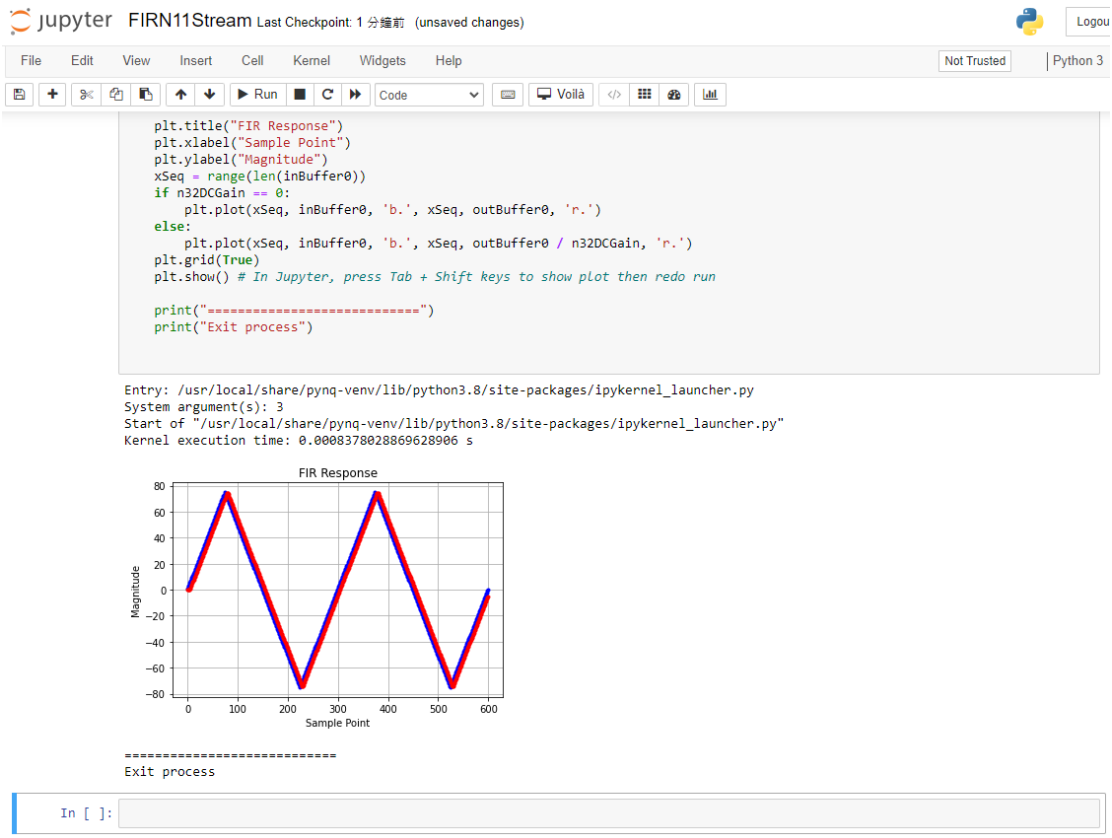
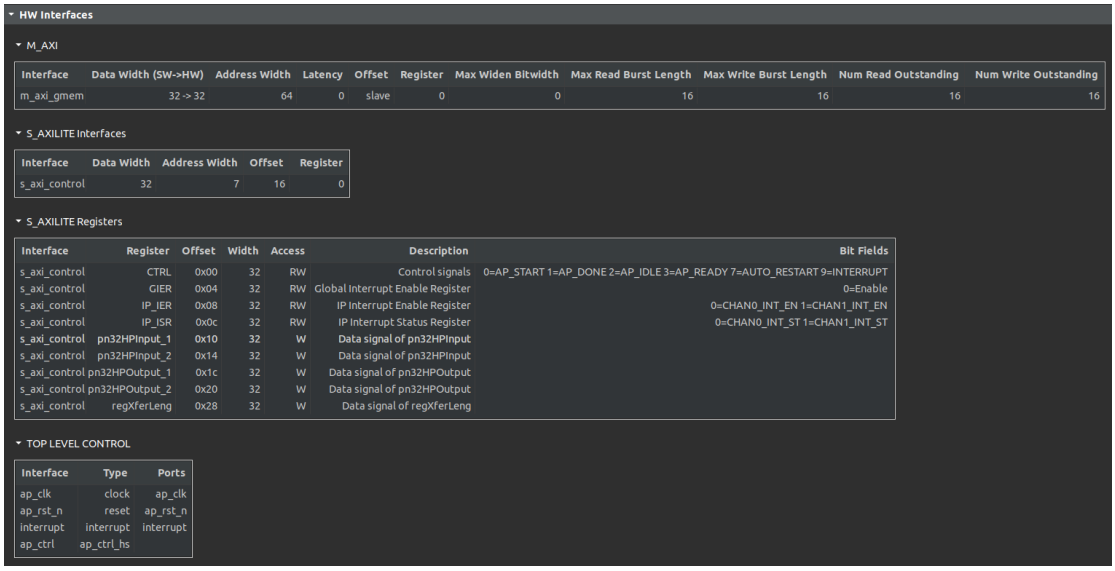


Figure15. Jupyter output

Part3: Performance

1. Master Interface



HW Interfaces

M_AXI

Interface	Data Width (SW->HW)	Address Width	Latency	Offset	Register	Max Widen Bitwidth	Max Read Burst Length	Max Write Burst Length	Num Read Outstanding	Num Write Outstanding
m_axi_gmem	32 -> 32	64	0	slave	0	0	16	16	16	16

S_AXILITE Interfaces

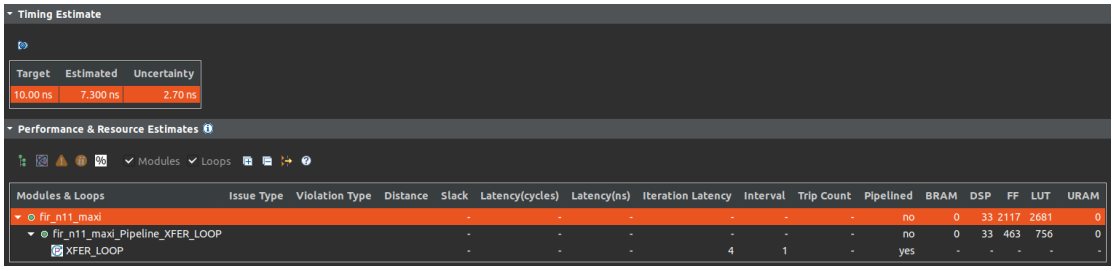
Interface	Data Width	Address Width	Offset	Register
s_axi_control	32	7	16	0

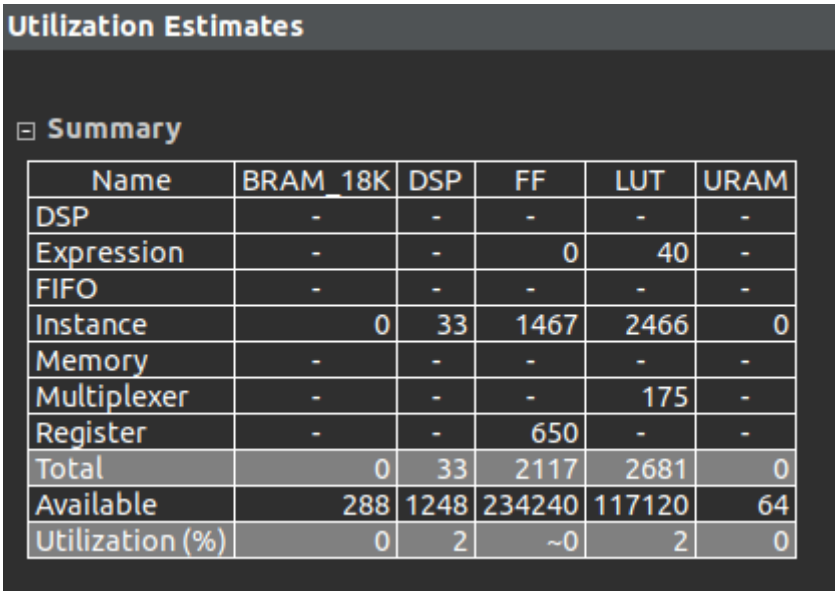
S_AXILITE Registers

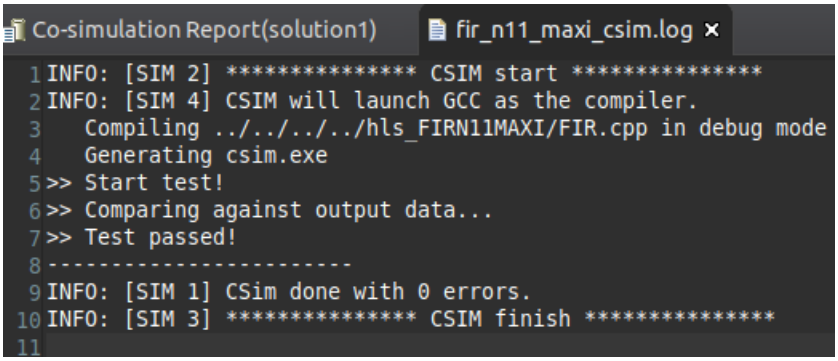
Interface	Register	Offset	Width	Access	Description	Bit Fields
s_axi_control	CTRL	0x00	32	RW	Control signals	0=AP_START 1=AP_DONE 2=AP_IDLE 3=AP_READY 7=AUTO_RESTART 9=INTERRUPT
s_axi_control	GIER	0x04	32	RW	Global Interrupt Enable Register	0=Enable
s_axi_control	IP_IER	0x08	32	RW	IP Interrupt Enable Register	0=CHAN0_INT_EN 1=CHAN1_INT_EN
s_axi_control	IP_ISR	0x0c	32	RW	IP Interrupt Status Register	0=CHAN0_INT_ST 1=CHAN1_INT_ST
s_axi_control	pn32HPInput_1	0x10	32	W	Data signal of pn32HPInput	
s_axi_control	pn32HPInput_2	0x14	32	W	Data signal of pn32HPInput	
s_axi_control	pn32HPOutput_1	0x1c	32	W	Data signal of pn32HPOutput	
s_axi_control	pn32HPOutput_2	0x20	32	W	Data signal of pn32HPOutput	
s_axi_control	regXferLeng	0x28	32	W	Data signal of regXferLeng	

TOP LEVEL CONTROL

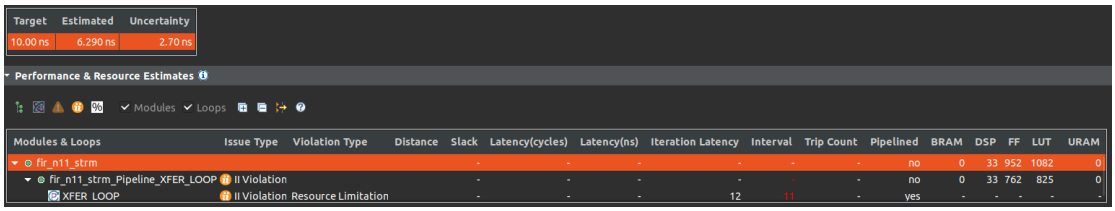
Interface	Type	Ports
ap_clk	clock	ap_clk
ap_rst_n	reset	ap_rst_n
interrupt	interrupt	interrupt
ap_ctrl	ap_ctrl_hs	







2. Stream Interface



HW Interfaces

S_AXILITE Interfaces

Interface	Data Width	Address Width	Offset	Register
s_axi_control	32	7	64	0

S_AXILITE Registers

Interface	Register	Offset	Width	Access	Description	Bit Fields
s_axi_control	CTRL	0x00	32	RW	Control signals	0=AP_START 1=AP_DONE 2=AP_IDLE 3=AP_READY 7=AUTO_RESTART 9=INTERRUPT
s_axi_control	GIER	0x04	32	RW	Global Interrupt Enable Register	0=Enable
s_axi_control	IP_IER	0x08	32	RW	IP Interrupt Enable Register	0=CHAN0_INT_EN 1=CHAN1_INT_EN
s_axi_control	IP_ISR	0x0c	32	RW	IP Interrupt Status Register	0=CHAN0_INT_ST 1=CHAN1_INT_ST
s_axi_control	regXferLeng	0x10	32	W	Data signal of regXferLeng	

AXIS

Interface	Register Mode	TDATA	TDEST	TID	TKEEP	TLAST	TREADY	TSTRB	TUSER	TVALID
pstrmInput	both	32	1	1	4	1	1	4	1	1
pstrmOutput	both	32	1	1	4	1	1	4	1	1

TOP LEVEL CONTROL

Interface	Type	Ports
ap_clk	clock	ap_clk
ap_rst_n	reset	ap_rst_n
interrupt	interrupt	interrupt
ap_ctrl	ap_ctrl_hs	

Utilization Estimates

Summary

Name	BRAM_18K	DSP	FF	LUT	URAM
DSP	-	-	-	-	-
Expression	-	-	0	42	-
FIFO	-	-	-	-	-
Instance	0	33	916	1005	0
Memory	-	-	-	-	-
Multiplexer	-	-	-	35	-
Register	-	-	36	-	-
Total	0	33	952	1082	0
Available	288	1248	234240	117120	64
Utilization (%)	0	2	~0	~0	0

Synthesis Summary(solution1) | Synthesis Details(solution1)(fir_n11_strm_csynth.rpt) | fir_n11_strm_csim.log

```
1 INFO: [SIM 2] *************** CSIM start ***************
2 INFO: [SIM 4] CSIM will launch GCC as the compiler.
3    Compiling ../../../../hls_FIRN11Stream/FIR.cpp in debug mode
4    Generating csim.exe
5 >> Start test!
6 >> Comparing against output data...
7 >> Test passed!
8 -----------------------
9 INFO: [SIM 1] CSim done with 0 errors.
10 INFO: [SIM 3] *************** CSIM finish ***************
11
```