Brief introduction about the overall system

這次的 lab 是要用電路實現 finite impulse response(FIR)頻率響應。也就是基本的乘法累加。這次 lab 分成兩個 IP，其中一種是用 maxi 的協議，另外一種則是用 axi-stream 作為 IP 接口。

M-AXI:

FIR_N11_MAXI 直接用 m-axi 的方式直接對 DDR 做讀寫，所以不需要 DMA 的幫助。

S-AXI:

FIR_N11_Stream 是 axi-stream 為接口的 IP，並沒有 address 的資訊，所以需要 adapter 的幫助，也就是 dma 去協助對外部 DDR 做讀寫。這比起 m-axi 更模組化一些，將 address 的處理交給其他 IP。

What's observed & learned?

1. Differences between MAXI & Stream interface.
   兩者都屬於高效能傳輸的 protocol，主要的差別在於 axi stream 比起 maxi 少了 address 的接口。如此就不是涉及 data 讀寫的問題，單純是傳送與接收的概念。
   而 maxi 則有根據 address 去讀寫的功能。
   兩者解有 handshake 的機制(valid & ready)

   補充 summary:
   AXI-stream, AXI-FULL(MAXI, AXI-master-slave), axi-lite 是屬於 AMBA axi 4.0 裡面的協議。
   Axi-lite 主要負責少量數據傳輸(單一數據)，有 address, handshake。
   Axi-full 同 axi-lite 有 address, handshake，但多了 burst 的概念，也就可以在單一 handshake 之後傳遞大量資料。
   Axi-stream 想比於前兩者則沒有 address 概念，但有 handshake，只有收、送概念。也是用於大量資料傳輸的協議。

   AMBA (advanced microcontroller bus architecture)是由 arm 所提供的免費 open source 晶片通訊的協議，現金廣為使用，主要分為幾種:
   AHB (advanced high-performance bus)，用於高效能(高頻寬)接口。
   APB (advanced peripheral bus)用於周片 GPIO…較慢的接口。
   AXI (advanced eXtensible bus)用於高效能可拓展接口。

2. Differences between csim and cosim.
   C-sim 單純對 c/c++語言去做模擬，主要是看 code 的演算法有沒有錯

誤。Focus more on “functional verification”.
而 cosim 則是將 c-synthesis 合成出來的 RTL code 去進行電路模擬去跟原本的 C 語言做 co-simulation 比對結果是否一致。

以下是網路上的詳細說明:

C-Simulation (Cycle-Accurate Simulation):

- C-Simulation is a software-based simulation that does not involve hardware.
- It is a purely functional simulation that models the behavior of your C/C++ code as it would run on a general-purpose CPU.
- C-Simulation is typically faster than Co-Simulation because it doesn't model hardware details and executes code directly on the host CPU.
- It is mainly used for functional verification and debugging of your high-level synthesis code.
- C-Simulation provides a quick way to identify and fix functional issues in your code.

Co-Simulation (Cycle-Accurate Co-Simulation):

- Co-Simulation, on the other hand, is a hardware/software simulation that involves both the synthesized hardware and the host CPU.
- It models the interaction between the software running on the host CPU and the hardware blocks generated by HLS.
- Co-Simulation is more accurate than C-Simulation in terms of performance estimation because it accounts for the hardware/software interaction and timing.
- It is useful for validating the performance of your design, checking for potential bottlenecks, and ensuring that the hardware/software interfaces are correctly implemented.
- Co-Simulation can be slower than C-Simulation due to the need to model hardware and software interactions.

Screen Dump

AXI-Master:

Performance

```
================================================================
== Performance Estimates
================================================================
+ Timing:
    * Summary:
    +--------+----------+----------+------------+
    |  Clock |  Target  | Estimated| Uncertainty|
    +--------+----------+----------+------------+
    |ap_clk  |  10.00 ns|  7.300 ns|     2.70 ns|
    +--------+----------+----------+------------+

+ Latency:
    * Summary:
    +---------+---------+----------+----------+-----+-----+---------+
    |  Latency (cycles) |  Latency (absolute) |  Interval | Pipeline|
    |   min   |   max   |    min   |    max   | min | max |   Type  |
    +---------+---------+----------+----------+-----+-----+---------+
    |        ?|        ?|         ?|         ?|    ?|    ?|       no|
    +---------+---------+----------+----------+-----+-----+---------+

    + Detail:
        * Instance:
        +-----------------------------------------+---------------------------------+---------+---------+----------+----------+-----+-----+---------+
        |                                         |                                 |  Latency (cycles) |  Latency (absolute) |  Interval | Pipeline|
        |                 Instance                |              Module             |   min   |   max   |    min   |    max   | min | max |   Type  |
        +-----------------------------------------+---------------------------------+---------+---------+----------+----------+-----+-----+---------+
        |grp_fir_n11_maxi_Pipeline_XFER_LOOP_fu_242  |fir_n11_maxi_Pipeline_XFER_LOOP  |        ?|        ?|         ?|         ?|    ?|    ?|       no|
        +-----------------------------------------+---------------------------------+---------+---------+----------+----------+-----+-----+---------+

        * Loop:
        N/A
```

Utilization

Interface

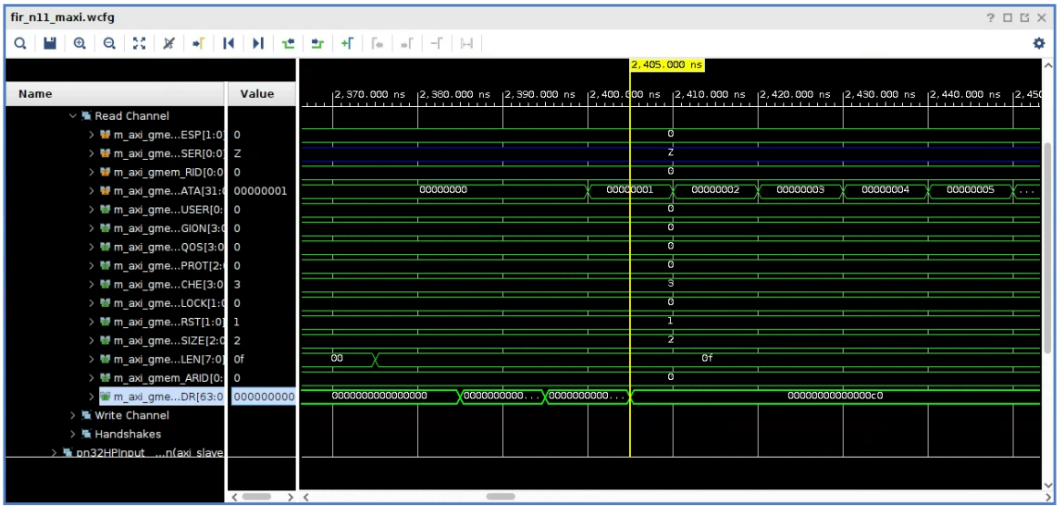
```
================================================================
== Interface
================================================================
* Summary:
```

RTL Ports	Dir	Bits	Protocol	Source Object	C Type
s_axi_control_AWVALID	in	1	s_axi	control	array
s_axi_control_AWREADY	out	1	s_axi	control	array
s_axi_control_AWADDR	in	7	s_axi	control	array
s_axi_control_WVALID	in	1	s_axi	control	array
s_axi_control_WREADY	out	1	s_axi	control	array
s_axi_control_WDATA	in	32	s_axi	control	array
s_axi_control_WSTRB	in	4	s_axi	control	array
s_axi_control_ARVALID	in	1	s_axi	control	array
s_axi_control_ARREADY	out	1	s_axi	control	array
s_axi_control_ARADDR	in	7	s_axi	control	array
s_axi_control_RVALID	out	1	s_axi	control	array
s_axi_control_RREADY	in	1	s_axi	control	array
s_axi_control_RDATA	out	32	s_axi	control	array
s_axi_control_RRESP	out	2	s_axi	control	array
s_axi_control_BVALID	out	1	s_axi	control	array
s_axi_control_BREADY	in	1	s_axi	control	array
s_axi_control_BRESP	out	2	s_axi	control	array
ap_clk	in	1	ap_ctrl_hs	fir_n11_maxi	return value
ap_rst_n	in	1	ap_ctrl_hs	fir_n11_maxi	return value
interrupt	out	1	ap_ctrl_hs	fir_n11_maxi	return value
m_axi_gmem_AWVALID	out	1	m_axi	gmem	pointer
m_axi_gmem_AWREADY	in	1	m_axi	gmem	pointer
m_axi_gmem_AWADDR	out	64	m_axi	gmem	pointer
m_axi_gmem_AWID	out	1	m_axi	gmem	pointer
m_axi_gmem_AWLEN	out	8	m_axi	gmem	pointer
m_axi_gmem_AWSIZE	out	3	m_axi	gmem	pointer
m_axi_gmem_AWBURST	out	2	m_axi	gmem	pointer
m_axi_gmem_AWLOCK	out	2	m_axi	gmem	pointer
m_axi_gmem_AWCACHE	out	4	m_axi	gmem	pointer
m_axi_gmem_AWPROT	out	3	m_axi	gmem	pointer
m_axi_gmem_AWQOS	out	4	m_axi	gmem	pointer
m_axi_gmem_AWREGION	out	4	m_axi	gmem	pointer
m_axi_gmem_AWUSER	out	1	m_axi	gmem	pointer
m_axi_gmem_WVALID	out	1	m_axi	gmem	pointer

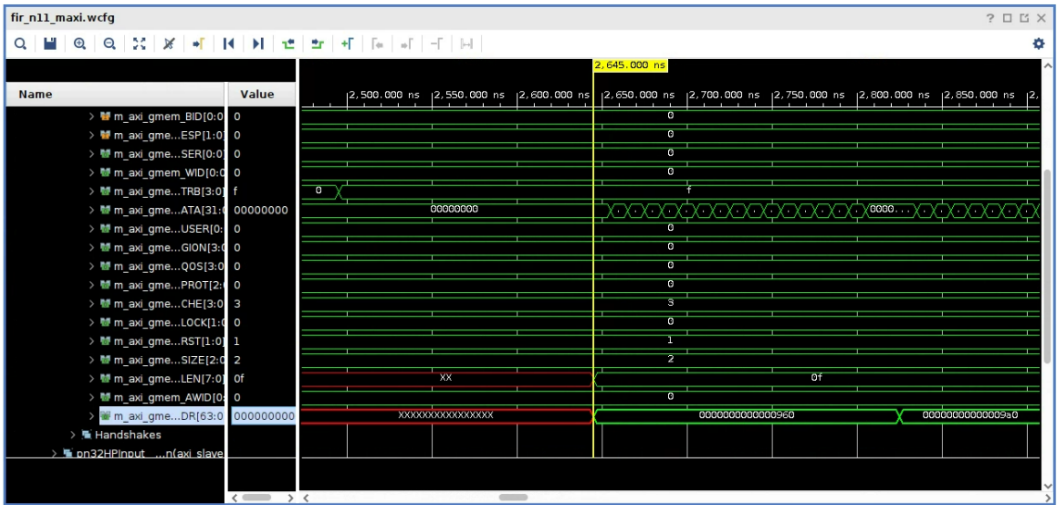
m_axi_gmem_WREADY	in	1	m_axi	gmem	pointer
m_axi_gmem_WDATA	out	32	m_axi	gmem	pointer
m_axi_gmem_WSTRB	out	4	m_axi	gmem	pointer
m_axi_gmem_WLAST	out	1	m_axi	gmem	pointer
m_axi_gmem_WID	out	1	m_axi	gmem	pointer
m_axi_gmem_WUSER	out	1	m_axi	gmem	pointer
m_axi_gmem_ARVALID	out	1	m_axi	gmem	pointer
m_axi_gmem_ARREADY	in	1	m_axi	gmem	pointer
m_axi_gmem_ARADDR	out	64	m_axi	gmem	pointer
m_axi_gmem_ARID	out	1	m_axi	gmem	pointer
m_axi_gmem_ARLEN	out	8	m_axi	gmem	pointer
m_axi_gmem_ARSIZE	out	3	m_axi	gmem	pointer
m_axi_gmem_ARBURST	out	2	m_axi	gmem	pointer
m_axi_gmem_ARLOCK	out	2	m_axi	gmem	pointer
m_axi_gmem_ARCACHE	out	4	m_axi	gmem	pointer
m_axi_gmem_ARPROT	out	3	m_axi	gmem	pointer
m_axi_gmem_ARQOS	out	4	m_axi	gmem	pointer
m_axi_gmem_ARREGION	out	4	m_axi	gmem	pointer
m_axi_gmem_ARUSER	out	1	m_axi	gmem	pointer
m_axi_gmem_RVALID	in	1	m_axi	gmem	pointer
m_axi_gmem_RREADY	out	1	m_axi	gmem	pointer
m_axi_gmem_RDATA	in	32	m_axi	gmem	pointer
m_axi_gmem_RLAST	in	1	m_axi	gmem	pointer
m_axi_gmem_RID	in	1	m_axi	gmem	pointer
m_axi_gmem_RUSER	in	1	m_axi	gmem	pointer
m_axi_gmem_RRESP	in	2	m_axi	gmem	pointer
m_axi_gmem_BVALID	in	1	m_axi	gmem	pointer
m_axi_gmem_BREADY	out	1	m_axi	gmem	pointer
m_axi_gmem_BRESP	in	2	m_axi	gmem	pointer
m_axi_gmem_BID	in	1	m_axi	gmem	pointer
m_axi_gmem_BUSER	in	1	m_axi	gmem	pointer

Co-simulation transcript/waveform

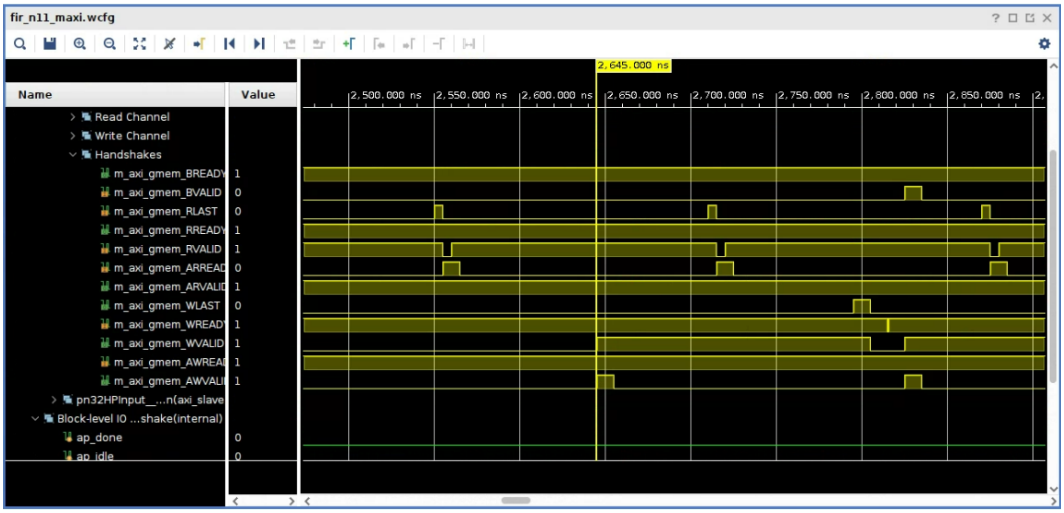
Read channel:



Write channel

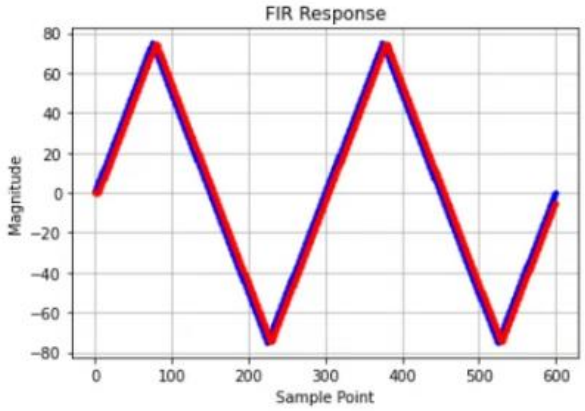


Hand shake



Jupyter notebook execution

```
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"
Kernel execution time: 0.0002777576446533203 s
```



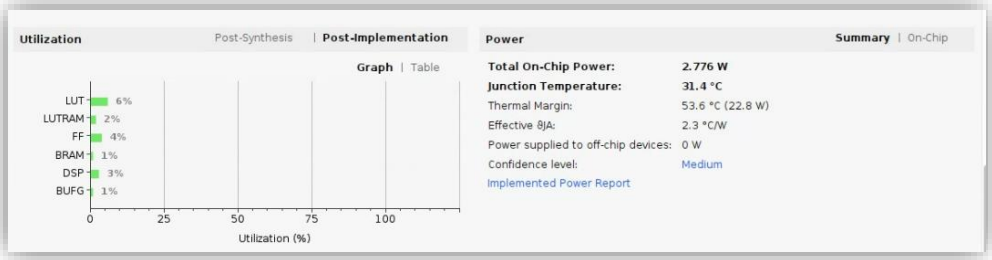

```
==============================
Exit process
```

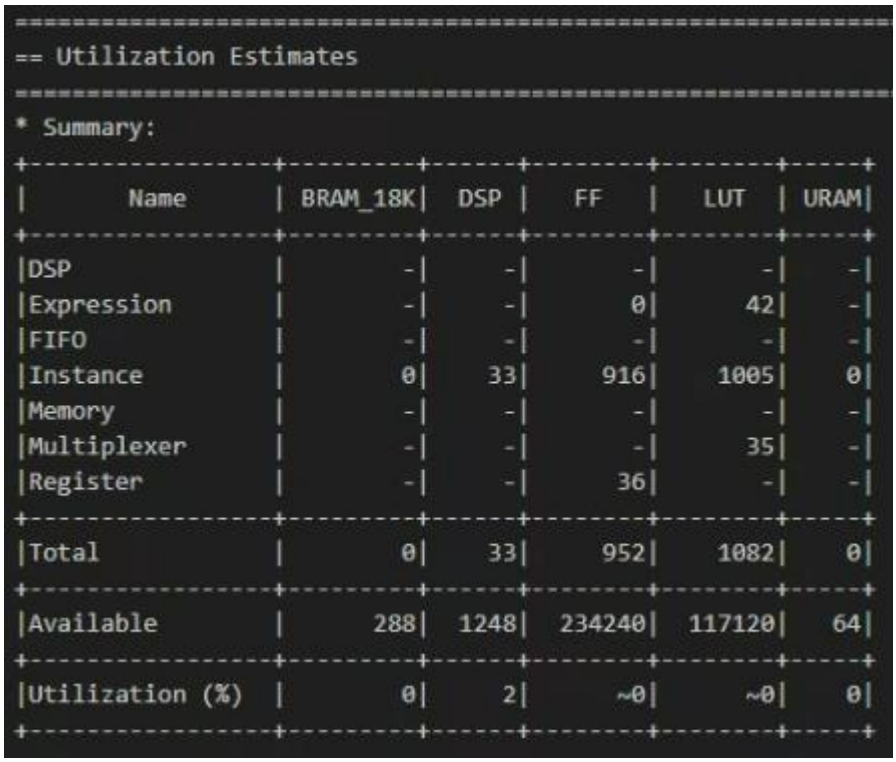
AXI-Stream:

Performance

```
================================================================
== Performance Estimates
================================================================
+ Timing:
    * Summary:
    +--------+----------+----------+------------+
    |  Clock |  Target  | Estimated| Uncertainty|
    +--------+----------+----------+------------+
    |ap_clk  |  10.00 ns|  6.290 ns|     2.70 ns|
    +--------+----------+----------+------------+

+ Latency:
    * Summary:
    +---------+---------+----------+----------+-----+-----+---------+
    |  Latency (cycles) |  Latency (absolute) |  Interval | Pipeline|
    |   min   |   max   |    min   |    max   | min | max |   Type  |
    +---------+---------+----------+----------+-----+-----+---------+
    |        ?|        ?|         ?|         ?|    ?|    ?|       no|
    +---------+---------+----------+----------+-----+-----+---------+

    + Detail:
        * Instance:
        +----------------------------------------------+--------------------------------+---------+---------+----------+----------+-----+-----+---------+
        |                                              |                                |  Latency (cycles) |  Latency (absolute) |  Interval | Pipeline|
        |                   Instance                   |             Module             |   min   |   max   |    min   |    max   | min | max |   Type  |
        +----------------------------------------------+--------------------------------+---------+---------+----------+----------+-----+-----+---------+
        |grp_fir_n11_strm_Pipeline_XFER_LOOP_fu_112    |fir_n11_strm_Pipeline_XFER_LOOP |        ?|        ?|         ?|         ?|    ?|    ?|       no|
        +----------------------------------------------+--------------------------------+---------+---------+----------+----------+-----+-----+---------+

        * Loop:
        N/A
```

Utilization





```
================================================================
== 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|
+-----------------+---------+------+--------+--------+-----+
```

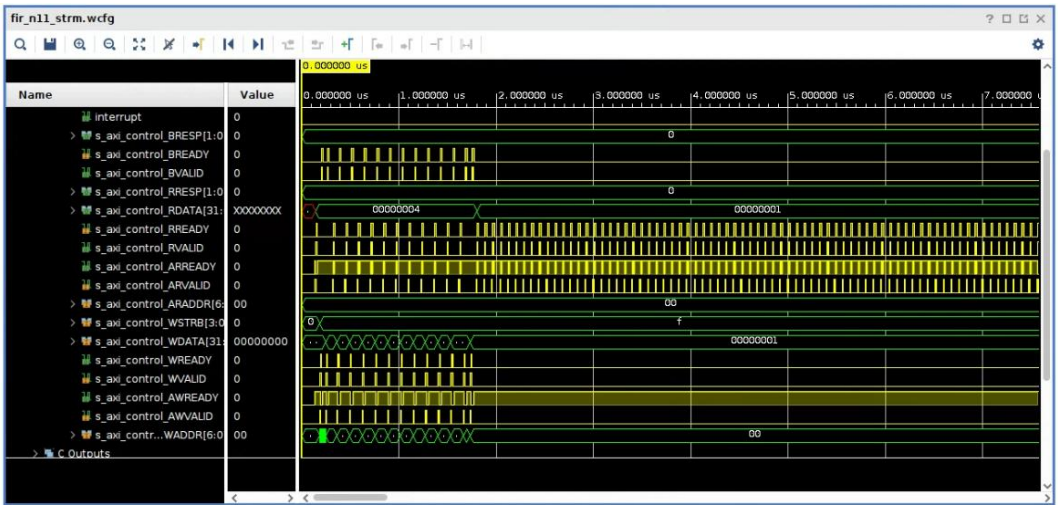
Interface

```
================================================================
== Interface
================================================================
* Summary:
```

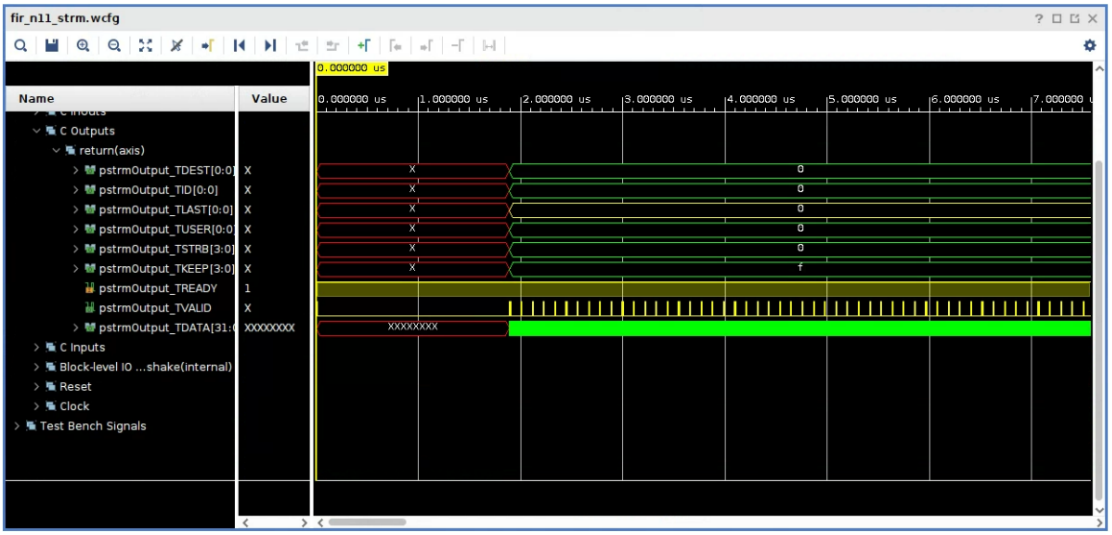
RTL Ports	Dir	Bits	Protocol	Source Object	C Type
s_axi_control_AWVALID	in	1	s_axi	control	array
s_axi_control_AWREADY	out	1	s_axi	control	array
s_axi_control_AWADDR	in	7	s_axi	control	array
s_axi_control_WVALID	in	1	s_axi	control	array
s_axi_control_WREADY	out	1	s_axi	control	array
s_axi_control_WDATA	in	32	s_axi	control	array
s_axi_control_WSTRB	in	4	s_axi	control	array
s_axi_control_ARVALID	in	1	s_axi	control	array
s_axi_control_ARREADY	out	1	s_axi	control	array
s_axi_control_ARADDR	in	7	s_axi	control	array
s_axi_control_RVALID	out	1	s_axi	control	array
s_axi_control_RREADY	in	1	s_axi	control	array
s_axi_control_RDATA	out	32	s_axi	control	array
s_axi_control_RRESP	out	2	s_axi	control	array
s_axi_control_BVALID	out	1	s_axi	control	array
s_axi_control_BREADY	in	1	s_axi	control	array
s_axi_control_BRESP	out	2	s_axi	control	array
ap_clk	in	1	ap_ctrl_hs	fir_n11_strm	return value
ap_rst_n	in	1	ap_ctrl_hs	fir_n11_strm	return value
interrupt	out	1	ap_ctrl_hs	fir_n11_strm	return value
pstrmInput_TDATA	in	32	axis	pstrmInput_V_data_V	pointer
pstrmInput_TVALID	in	1	axis	pstrmInput_V_dest_V	pointer
pstrmInput_TREADY	out	1	axis	pstrmInput_V_dest_V	pointer
pstrmInput_TDEST	in	1	axis	pstrmInput_V_dest_V	pointer
pstrmInput_TKEEP	in	4	axis	pstrmInput_V_keep_V	pointer
pstrmInput_TSTRB	in	4	axis	pstrmInput_V_strb_V	pointer
pstrmInput_TUSER	in	1	axis	pstrmInput_V_user_V	pointer
pstrmInput_TLAST	in	1	axis	pstrmInput_V_last_V	pointer
pstrmInput_TID	in	1	axis	pstrmInput_V_id_V	pointer
pstrmOutput_TDATA	out	32	axis	pstrmOutput_V_data_V	pointer
pstrmOutput_TVALID	out	1	axis	pstrmOutput_V_dest_V	pointer
pstrmOutput_TREADY	in	1	axis	pstrmOutput_V_dest_V	pointer
pstrmOutput_TDEST	out	1	axis	pstrmOutput_V_dest_V	pointer
pstrmOutput_TKEEP	out	4	axis	pstrmOutput_V_keep_V	pointer
pstrmOutput_TSTRB	out	4	axis	pstrmOutput_V_strb_V	pointer
pstrmOutput_TUSER	out	1	axis	pstrmOutput_V_user_V	pointer
pstrmOutput_TLAST	out	1	axis	pstrmOutput_V_last_V	pointer
pstrmOutput_TID	out	1	axis	pstrmOutput_V_id_V	pointer

Co-simulation transcript/waveform

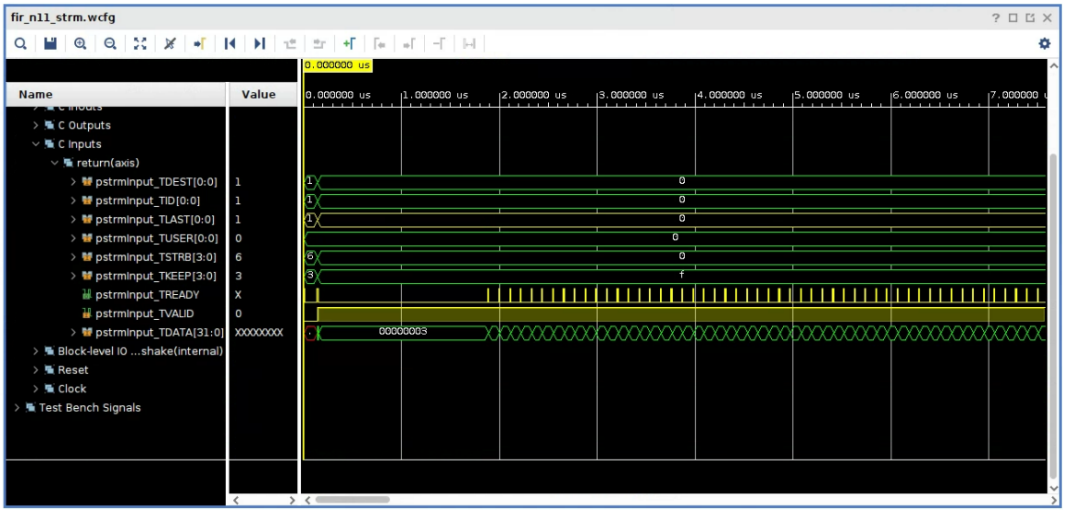
C inouts:



C outputs:

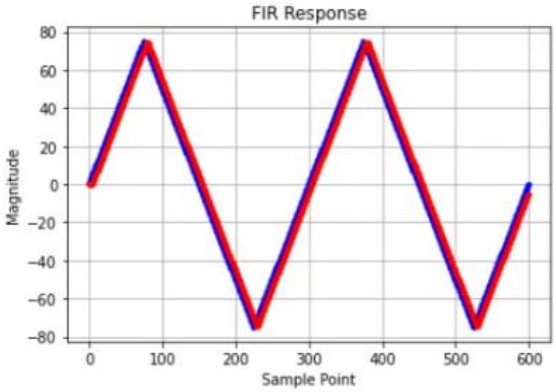


C inputs:



Jupyter notebook execution

```
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"
Kernel execution time: 0.0009963512420654297 s
```




```
============================
Exit process
```