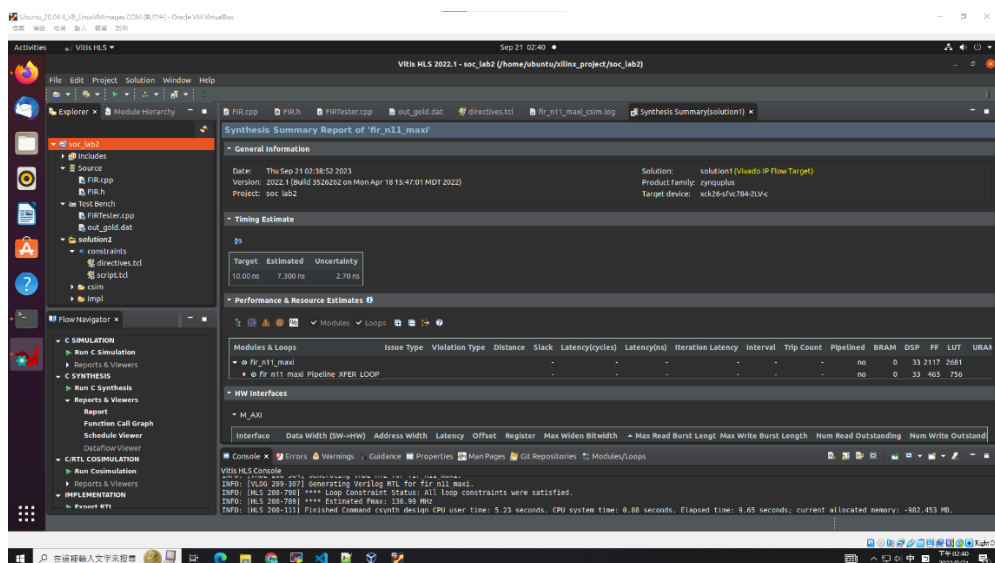


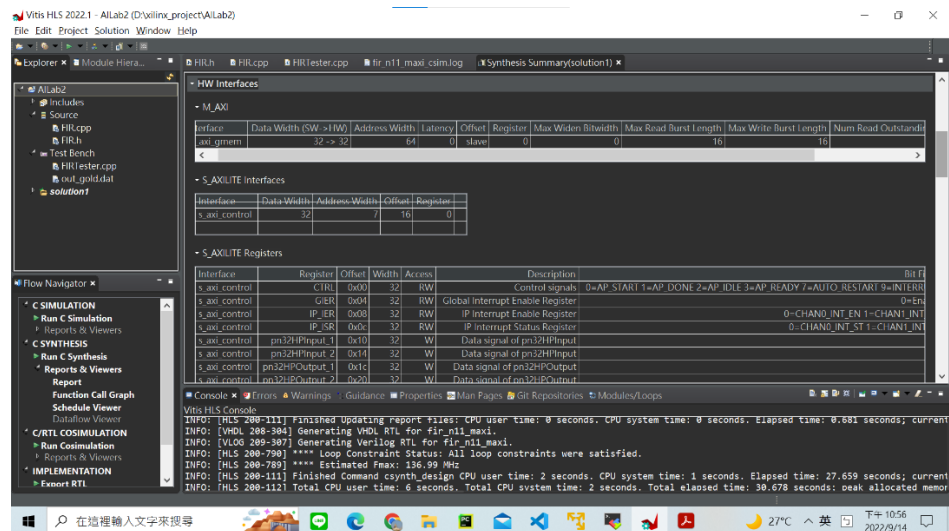
Lab2 Report

- Different between AXI Master and AXI Stream
 - ✓ The main advantages for m_axi interface are: separate and independent read and write channels, support burst-based accesses with potential performance of $\sim 17\text{GB/s}$ and provide support for outstanding transactions.
 - ✓ The axis interface has no address information and burst length information. Therefore, the signal transfers like a stream with the TLAST signal telling the end of the data transfer.
- Different between C-simulation and Co-simulation
 - ✓ C-simulation: This step is the algorithm-only simulation, which doesn't include any hardware information, namely, the simulation is high level simulation.
 - ✓ Co-simulation: This step is post-synthesis simulation, which means your high-level code has been translated into hardware description code. Therefore, the simulation is cycle accurate. You can see the signal at every cycle by opening the waveform report.
- Lab2-1 part

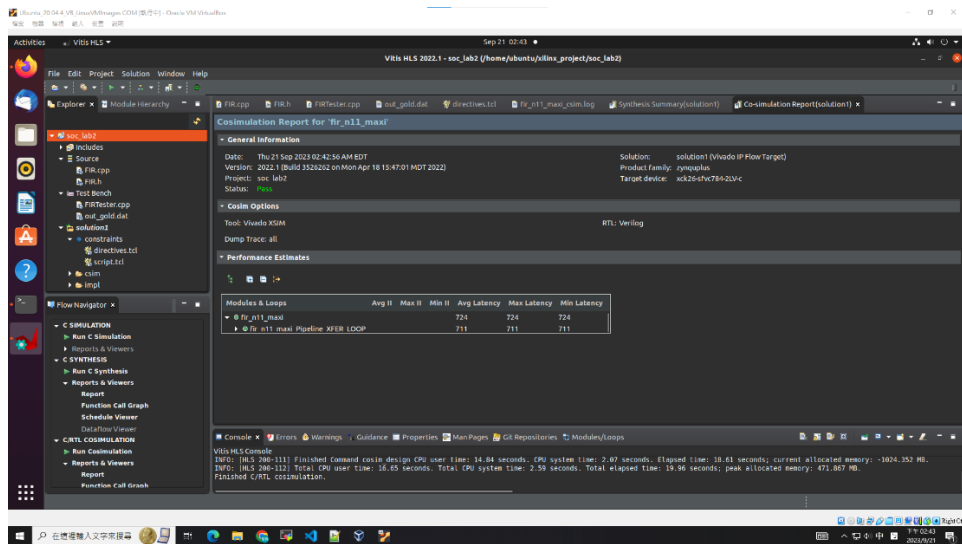
In lab2, we use FIR function to implement the HLS process. Lab2-1, we use AXI-master interface to build the system. The important thing to be aware is that m_axi protocol doesn't support scalar input and return. By using this protocol, the resource usages may higher than using the AXI-stream interface.



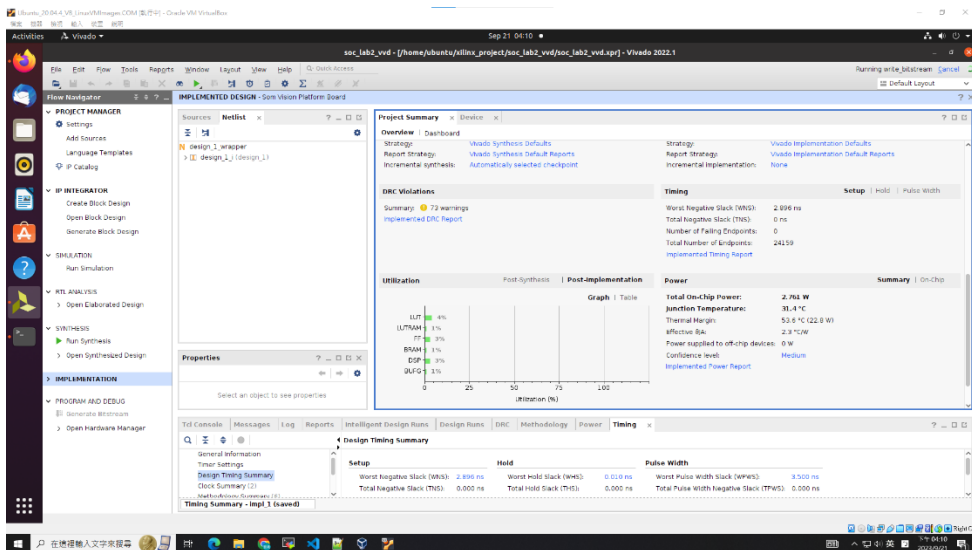
Synthesis report



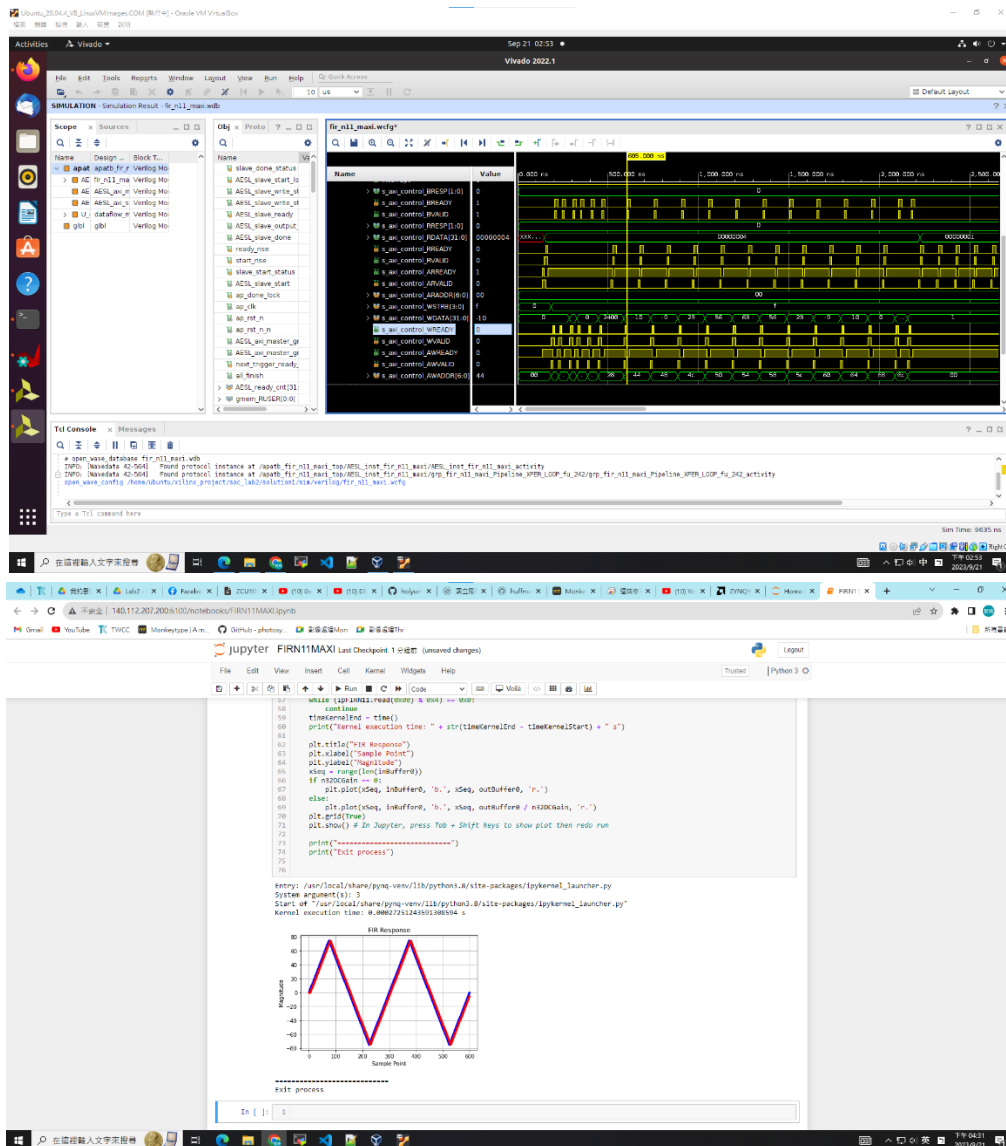
HW interface



Co simulation



utilization



➤ Lab2-2 part

An AXI-stream interface can be applied to any input argument and any array or pointer output argument. It transfers data in a sequential streaming manner, so it cannot be used with arguments that are both read and written. The other important thing is we need to include `<hls_stream.h>` to use the `hls::stream` object for internal streams.

