



# 112-1 SoC Design Laboratory

## Lab3 submission guide

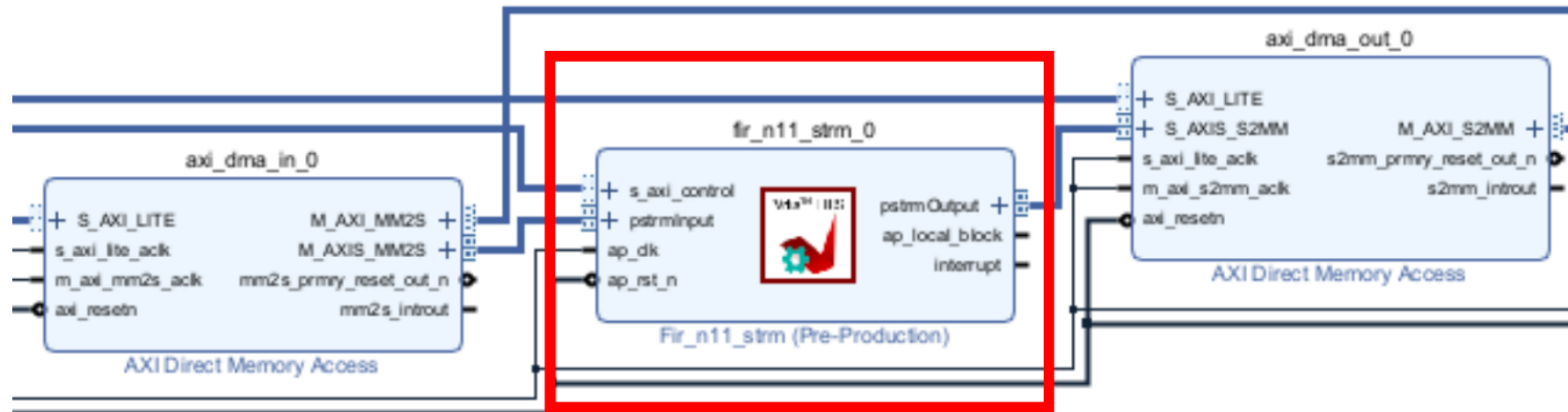
Teacher : Jiin Lai

TA : 楊嶺騏、蘇子源



# Function specification

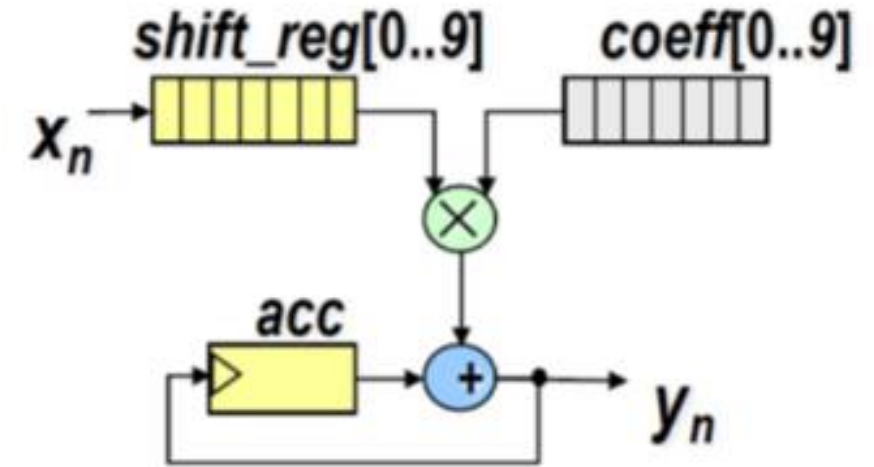
- Same as course-lab\_2(FIRN11stream)
- $y[t] = \sum (h[i] * x[t - i])$





# Design specification

- Data Width 32
- Tape Num 11
- Data Num TBD - Based on size of data file
- Interface
  - data\_in stream (  $X_n$  )
  - data\_out: stream (  $Y_n$  )
  - coef[Tape Num-1:0] axilite
  - len: axilite
  - ap\_start: axilite
  - ap\_done: axilite
- Using **one Multiplier and one Adder**
- Shift register implemented with SRAM (Shift\_RAM, size = 10 DW) - size = 10 DW
- Tap coefficient implemented with SRAM (Tap\_RAM = 11 DW) and initialized by axilite write
- Operation
  - ap\_start to initiate FIR engine (ap\_start valid for one clock cycle)
  - Stream-in  $X_n$ . The rate is depending on the FIR processing speed. Use axi-stream valid/ready for flow control
  - Stream out  $Y_n$ , the output rate depends on FIR processing speed.





# Deliver module header

---

- The I/O signals are listed in fir.v.
- You are requested to use simplified AXI-lite and AXI-stream protocol.
- FIR module interface
  - reference:
    - AXI-lite:
      - <https://www.realdigital.org/doc/a9fee931f7a172423e1ba73f66ca4081>
      - <https://docs.xilinx.com/r/en-US/pg202-mipi-dphy/AXI4-Lite-Interface>
    - AXI-stream:
      - <https://developer.arm.com/documentation/ih0051/latest/>
      - <https://docs.xilinx.com/r/en-US/pg256-sdfec-integrated-block/AXI4-Stream-Interface>



# Configuration Register Address map

---

- 0x00 – [0] - ap\_start   -> set when ap\_start signal asserts
  - > reset, when start data transfer, i.e. 1<sup>st</sup> axi-stream data come in
- [1] – ap\_done   -> when FIR process all the dataset, i.e. receive tlast, and last Y is generated
- [2] – ap\_idle   -> indicate FIR is actively processing data
- 0x10-14 - data-length
- 0x20-FF – Tap parameters, (e.g., 0x20-24 Tap0, in sequence ...)

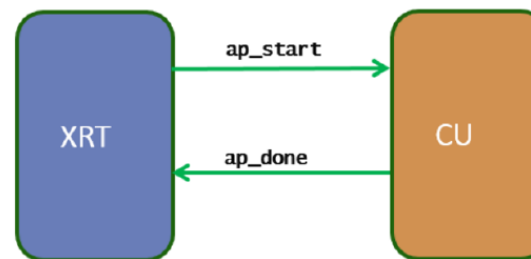


# Testbench(1/2)

- fir\_tb(Please name your top module same as **fir** for simulation)
- The testbench should keep in the same directory with Makefile
- Block-level Protocol (ap\_start, ap\_done):

AP\_CTRL\_HS (Sequential Executed Kernel)

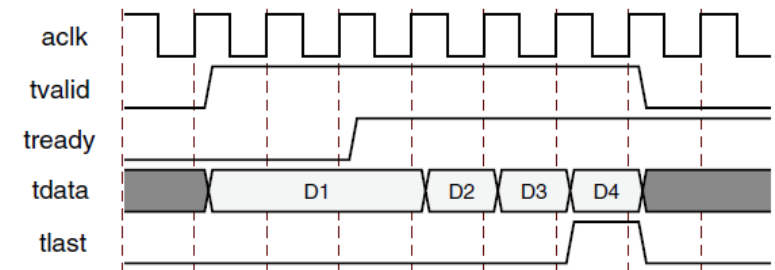
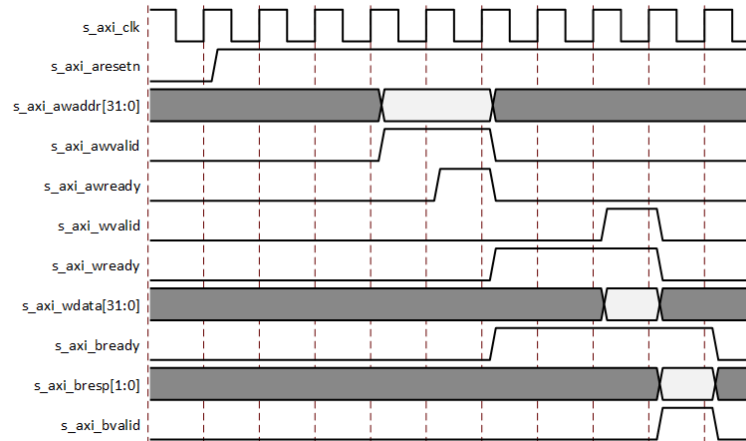
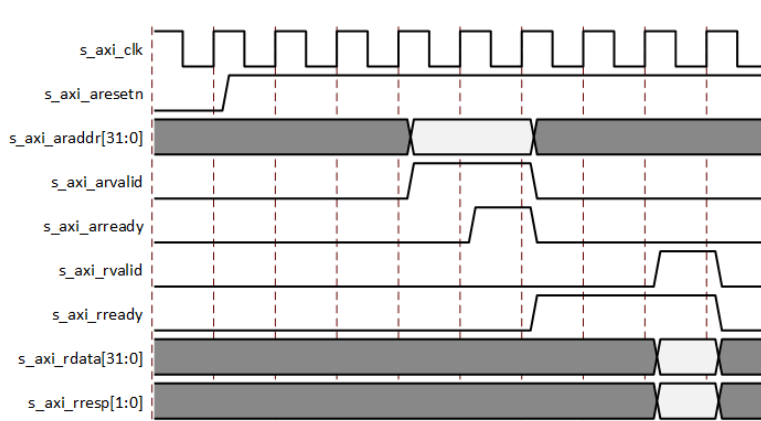
- Host and Kernel Synchronization by
  - ap\_start
  - ap\_done
- Kernel can only be restarted (ap\_start), after it completes the current execution (ap\_done)
- Serving one execution request a time





# Testbench(2/2)

- Test dataset
  - Samples\_triangular\_wave.dat
  - out\_gold.dat
- FIR Kernel waveform





# Submission (1/2)

---

- Hierarchy:
  - StudentID\_lab3/
    - Waveform
    - Simulation.log
    - Report.pdf
    - Synthesis report
    - Github link
- Your Github link should attach the file
  - Design, Synthesis report(Including FF, LUT, Bram ), Waveform, simulation.log, makefile,...etc.
  - Report
- Location of design (**If use vivado to design**)
  - `hostproject/hostproject.srcs/sources_1/new/`





# Submission (2/2)

---

- Compress all above files in a single zip file named
  - StudentID\_lab3.zip
- Submit to Submit to NYCU E3
- Deadline: **10/19 (Thu.) 23:59**
  - 20% off for the late submission penalty within 3 days



# What is included in the report

---

- Block Diagram
  - Datapath – dataflow
  - Control signals
- Describe operation, e.g.
  - How to receive data-in and tap parameters and place into SRAM
  - How to access shiftram and tapRAM to do computation
  - How ap\_done is generated.
  - .....
- Resource usage: including FF, LUT, BRAM
- Timing Report
  - Try to synthesize the design with maximum frequency
  - Report timing on longest path, slack
- Simulation Waveform, show
  - Coefficient program, and read back
  - Data-in stream-in
  - Data-out stream-out
  - Bram access control
  - FSM