SoC Design Laboratory Lab3

EE 112061545 Chih-ling Wu

SoC_Lab3_Github_Link

1. Introduction:

In this lab, we implemented a FIR system in Verilog, using AXI-Lite interfaces for coefficients and Stream interfaces for data. We also collaborated with a BRAM module to handle data efficiently. The core focus of this lab is to master the control of handshake signals within these protocols and to create a well-designed data flow.

2. System Block diagrams:

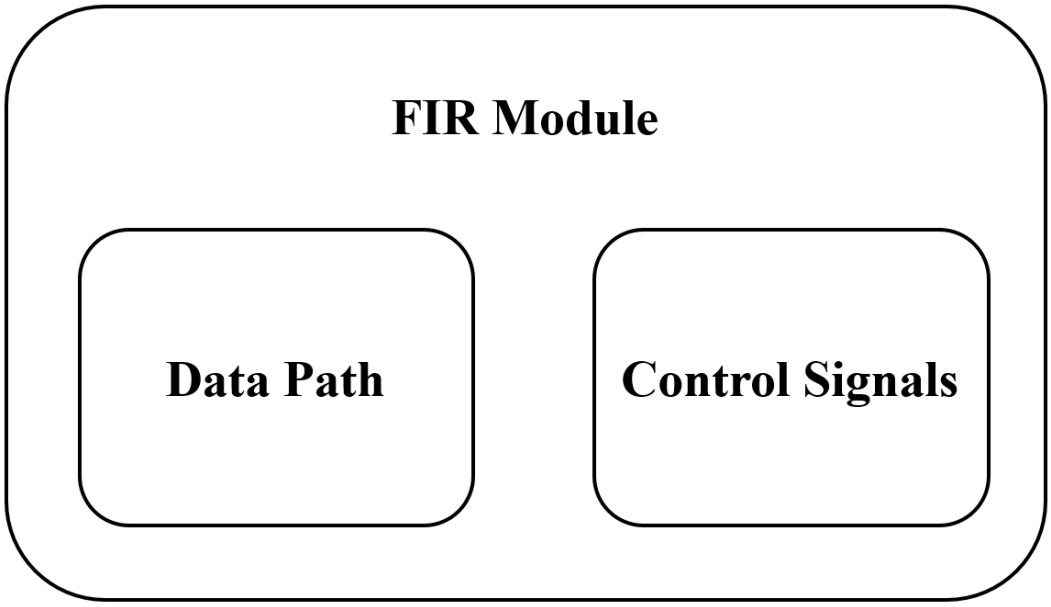

Fig1. FIR module block diagram

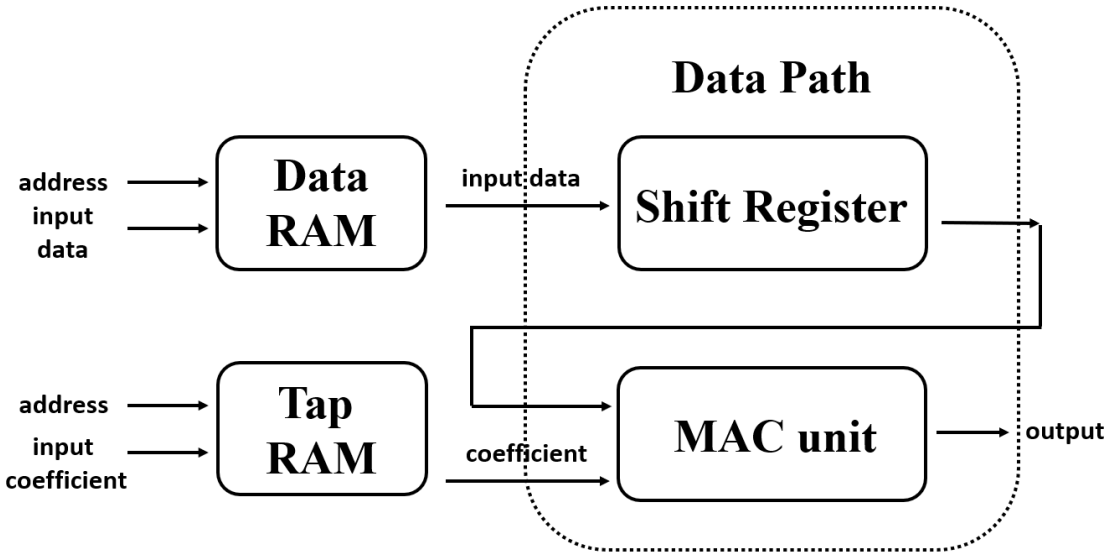

Fig2. Data path block diagram breakdown

(1) Dataflows:

- Shift Register:

  Input data is stored and shifted as new data arrives.

- MAC:

  Performing the multiplication and addition operations for the FIR filter.

- Data RAM:

  Storing input data.

- Tap RAM:

  Storing filter coefficients (tap parameters).

(2) Control signals:

- Handshake Signals:

  Facilitating communication with the AXI-Lite interface and the stream interface.

- Flags:

  Indicating the configuration and operation status of the filter (ap_start, ap_idle, ap_done).

- State Machine:

  To control the operation of the FIR filter with states (IDLE, LOAD, MAC, DONE).

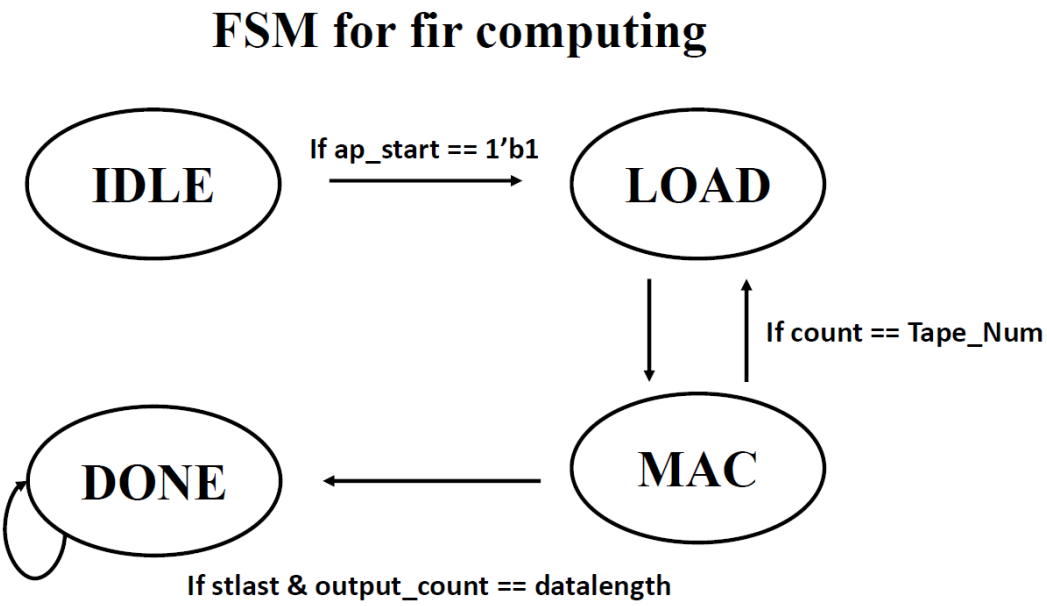

Fig3. Design for finite state machine for fir computing

3. Operations:

(1) How to Receive Data-In and Tap Parameters and Place into SRAM?

Input data is received through the "Stream Interface" (ss_tvalid, ss_tdata, ss_tlast). Data is loaded into the "Data RAM" for processing and stored in the "Shift Register."

Filter coefficients (tap parameters) are configured through the AXI-Lite interface using write transactions. The coefficients are loaded into the "Tap RAM" for use in the FIR filter.

(2) How to Access Shift Register and Tap RAM to Do Computation?

During the "MAC" state, the FIR filter retrieves data from the "Shift Register" to perform the computation. The data is used as input to the "MAC Unit."

The FIR filter accesses the "Tap RAM" to retrieve the filter coefficients (tap parameters) corresponding to the current computation. These coefficients are also used by the "MAC Unit" for the multiplication operation.

(3) How to generate ap signals?

ap_start: This signal is generated when the coefficient read is done. It is set to 1 for one clock cycle when the address for reading coefficients is detected.

ap_done: This signal is generated based on conditions, including the output count matching the data length and the current state not being IDLE.

ap_idle: It is set to 1 when the FIR filter is idle or not actively processing data. It transitions to 0 when a new operation starts (ap_start is 1) and returns to 1 when the operation is complete.

(4) How to generate handshake signals in two interfaces?

※ AXI-Lite Interface Handshake Signals:

● awready and arready (write and Read Ready): These signals indicate readiness to accept addresses for write and read operations, respectively. They are updated based on the presence of valid address signals.

● wready and rready (write and Read Ready): These signals reflect readiness to accept data for writing or to send data for reading. They are directly derived from the presence of valid data signals.

※Stream Interface Handshake Signals:

● ss_tready (ready to Receive Data): This signal represents readiness to receive data through the Stream interface. Its status depends on the current state and the

presence of valid data.

- sm_tvalid (valid for Sending Back Output Data): This signal indicates when the module is ready to send back output data through the Stream interface. It becomes valid when specific conditions are met, allowing the module to transmit data.

4. Simulation waveform & results:

(1) Coefficient program, and read back:

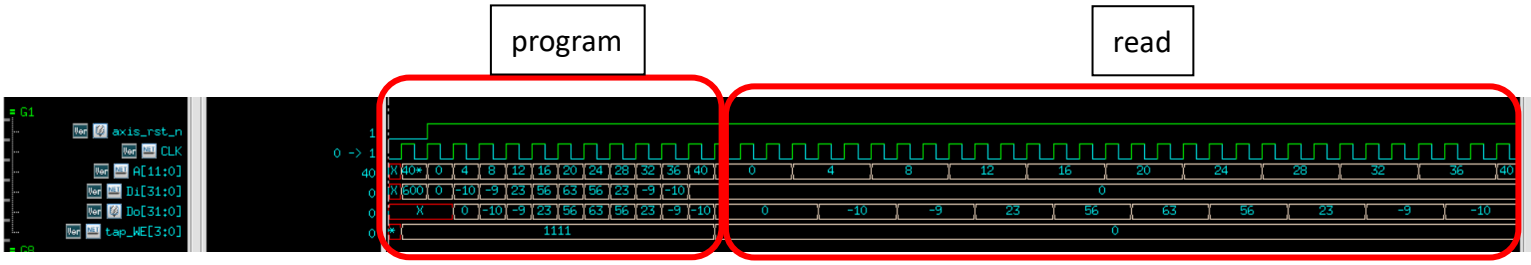

Fig4. Coefficient read / program waveform

```
-----------Start simulation-----------
----Start the data input(AXI-Stream)----
----Start the coefficient input(AXI-lite)----
 Check Coefficient ...
OK: exp =           0, rdata =           0
OK: exp =         -10, rdata =         -10
OK: exp =          -9, rdata =          -9
OK: exp =          23, rdata =          23
OK: exp =          56, rdata =          56
OK: exp =          63, rdata =          63
OK: exp =          56, rdata =          56
OK: exp =          23, rdata =          23
OK: exp =          -9, rdata =          -9
OK: exp =         -10, rdata =         -10
OK: exp =           0, rdata =           0
```

Fig5. Coefficient read / program simulation result

(2) Data-in stream-in:

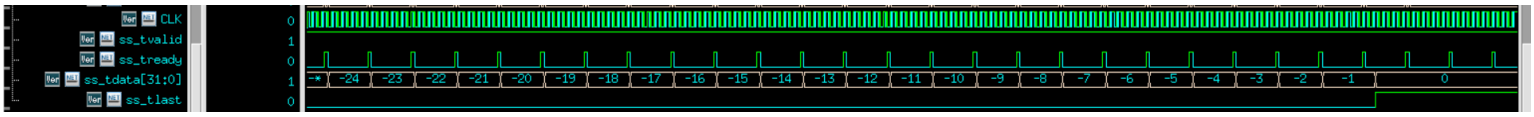

Fig6. Data stream in waveform

(3) Data-out stream-out:

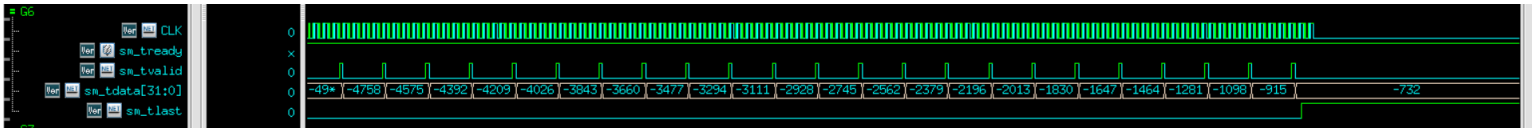

Fig7. Data stream out waveform

```
[PASS] [Pattern        568] Golden answer:       -6588, Your answer:       -6588
[PASS] [Pattern        569] Golden answer:       -6405, Your answer:       -6405
[PASS] [Pattern        570] Golden answer:       -6222, Your answer:       -6222
[PASS] [Pattern        571] Golden answer:       -6039, Your answer:       -6039
[PASS] [Pattern        572] Golden answer:       -5856, Your answer:       -5856
[PASS] [Pattern        573] Golden answer:       -5673, Your answer:       -5673
[PASS] [Pattern        574] Golden answer:       -5490, Your answer:       -5490
[PASS] [Pattern        575] Golden answer:       -5307, Your answer:       -5307
[PASS] [Pattern        576] Golden answer:       -5124, Your answer:       -5124
[PASS] [Pattern        577] Golden answer:       -4941, Your answer:       -4941
[PASS] [Pattern        578] Golden answer:       -4758, Your answer:       -4758
[PASS] [Pattern        579] Golden answer:       -4575, Your answer:       -4575
[PASS] [Pattern        580] Golden answer:       -4392, Your answer:       -4392
[PASS] [Pattern        581] Golden answer:       -4209, Your answer:       -4209
[PASS] [Pattern        582] Golden answer:       -4026, Your answer:       -4026
[PASS] [Pattern        583] Golden answer:       -3843, Your answer:       -3843
[PASS] [Pattern        584] Golden answer:       -3660, Your answer:       -3660
[PASS] [Pattern        585] Golden answer:       -3477, Your answer:       -3477
[PASS] [Pattern        586] Golden answer:       -3294, Your answer:       -3294
[PASS] [Pattern        587] Golden answer:       -3111, Your answer:       -3111
[PASS] [Pattern        588] Golden answer:       -2928, Your answer:       -2928
[PASS] [Pattern        589] Golden answer:       -2745, Your answer:       -2745
[PASS] [Pattern        590] Golden answer:       -2562, Your answer:       -2562
[PASS] [Pattern        591] Golden answer:       -2379, Your answer:       -2379
[PASS] [Pattern        592] Golden answer:       -2196, Your answer:       -2196
[PASS] [Pattern        593] Golden answer:       -2013, Your answer:       -2013
[PASS] [Pattern        594] Golden answer:       -1830, Your answer:       -1830
[PASS] [Pattern        595] Golden answer:       -1647, Your answer:       -1647
OK: exp =           0, rdata =           0
[PASS] [Pattern        596] Golden answer:       -1464, Your answer:       -1464
------End the data input(AXI-Stream)------
[PASS] [Pattern        597] Golden answer:       -1281, Your answer:       -1281
[PASS] [Pattern        598] Golden answer:       -1098, Your answer:       -1098
[PASS] [Pattern        599] Golden answer:        -915, Your answer:        -915
OK: exp =           2, rdata =           6
OK: exp =           4, rdata =           6
-----------------------------------------------
----------Congratulations! Pass-------------
Simulation complete via $finish(1) at time 78675 NS + 0
./tb/fir_tb.v:200         $finish;
ncsim> exit
[chihlingqg@node01 fir]$
```

Fig8. Data-out match golden results

(4) RAM access control:

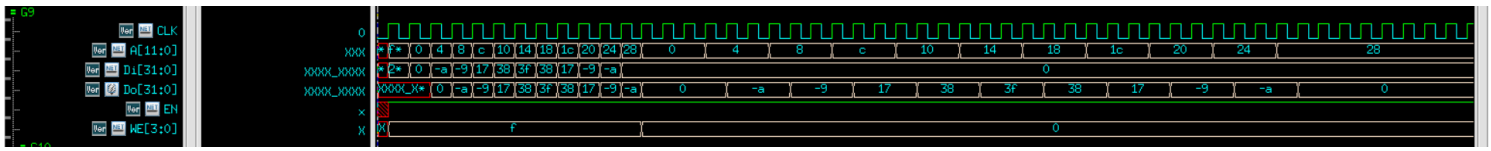

Fig9. Tap RAM access control

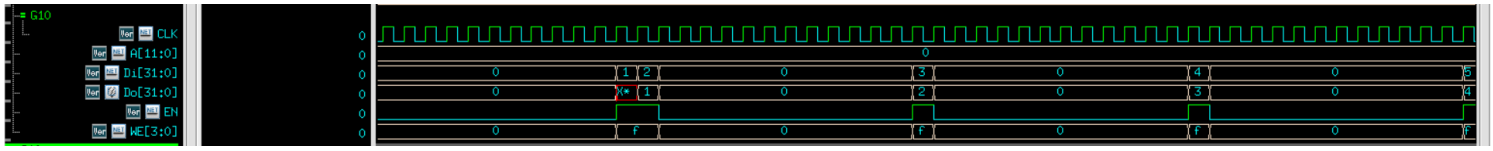

Fig10. Data RAM access control

(5) Finite state machine transition:

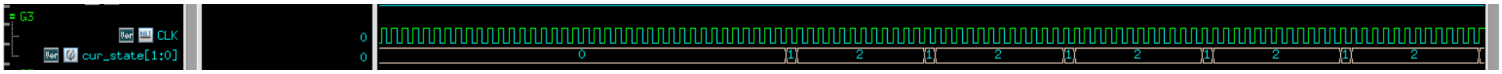

Fig11. FSM transition

5. Resource Usage:

```
---------------------------------------------------------------------------------
Start RTL Component Statistics
---------------------------------------------------------------------------------
Detailed RTL Component Info :
+---Adders :
     2 Input   32 Bit       Adders := 2
     2 Input   12 Bit       Adders := 2
     2 Input   10 Bit       Adders := 1
     2 Input    4 Bit       Adders := 1
+---Registers :
                 32 Bit    Registers := 27
                 10 Bit    Registers := 1
                  4 Bit    Registers := 1
                  2 Bit    Registers := 1
                  1 Bit    Registers := 6
+---Multipliers :
               32x32  Multipliers := 1
+---Muxes :
     2 Input   32 Bit        Muxes := 9
     2 Input   12 Bit        Muxes := 3
     2 Input    6 Bit        Muxes := 1
     3 Input    4 Bit        Muxes := 1
     2 Input    4 Bit        Muxes := 2
     3 Input    2 Bit        Muxes := 1
     4 Input    2 Bit        Muxes := 2
     2 Input    1 Bit        Muxes := 6
     3 Input    1 Bit        Muxes := 1
    13 Input    1 Bit        Muxes := 11
---------------------------------------------------------------------------------
Finished RTL Component Statistics
---------------------------------------------------------------------------------
```

Fig12. Resource: one 32bit multiplier and one 32bit adder

```
1. Slice Logic
--------------

+-------------------------+------+-------+------------+-----------+-------+
|        Site Type        | Used | Fixed | Prohibited | Available | Util% |
+-------------------------+------+-------+------------+-----------+-------+
| Slice LUTs*             |  516 |     0 |          0 |     53200 |  0.97 |
|   LUT as Logic          |  516 |     0 |          0 |     53200 |  0.97 |
|   LUT as Memory         |    0 |     0 |          0 |     17400 |  0.00 |
| Slice Registers         |  887 |     0 |          0 |    106400 |  0.83 |
|   Register as Flip Flop |  887 |     0 |          0 |    106400 |  0.83 |
|   Register as Latch     |    0 |     0 |          0 |    106400 |  0.00 |
| F7 Muxes                |    0 |     0 |          0 |     26600 |  0.00 |
| F8 Muxes                |    0 |     0 |          0 |     13300 |  0.00 |
+-------------------------+------+-------+------------+-----------+-------+
```

Fig13. Resource: LUT and FFs

```
2. Memory
---------

+----------------+------+-------+------------+-----------+-------+
|    Site Type   | Used | Fixed | Prohibited | Available | Util% |
+----------------+------+-------+------------+-----------+-------+
| Block RAM Tile |    0 |     0 |          0 |       140 |  0.00 |
|   RAMB36/FIFO* |    0 |     0 |          0 |       140 |  0.00 |
|   RAMB18       |    0 |     0 |          0 |       280 |  0.00 |
+----------------+------+-------+------------+-----------+-------+
```

Fig14. No use of BRAM in module

6. Timing Reports:

Clock period: 12ns

```
Max Delay Paths
--------------------------------------------------------------------------------------
Slack (MET) :             0.037ns  (required time - arrival time)
  Source:                 araddr[5]
                            (input port clocked by axis_clk  {rise@0.000ns fall@6.000ns period=12.000ns})
  Destination:            tap_A[11]
                            (output port clocked by axis_clk  {rise@0.000ns fall@6.000ns period=12.000ns})
  Path Group:             axis_clk
  Path Type:              Max at Slow Process Corner
  Requirement:            12.000ns  (axis_clk rise@12.000ns - axis_clk rise@0.000ns)
  Data Path Delay:        7.128ns  (logic 4.901ns (68.763%)  route 2.226ns (31.237%))
  Logic Levels:           6  (CARRY4=2 IBUF=1 LUT1=1 LUT4=1 OBUF=1)
  Input Delay:            2.400ns
  Output Delay:           2.400ns
  Clock Uncertainty:      0.035ns  ((TSJ^2 + TIJ^2)^1/2 + DJ) / 2 + PE
    Total System Jitter     (TSJ):    0.071ns
    Total Input Jitter      (TIJ):    0.000ns
    Discrete Jitter          (DJ):    0.000ns
    Phase Error              (PE):    0.000ns

    Location             Delay type                Incr(ns)  Path(ns)    Netlist Resource(s)
  -------------------------------------------------------------------    -------------------
                         (clock axis_clk rise edge)
                                                      0.000     0.000 r
                         input delay                  2.400     2.400
                                                      0.000     2.400 f  araddr[5] (IN)
                         net (fo=0)                   0.000     2.400    araddr[5]
                                                                      f  araddr_IBUF[5]_inst/I
                         IBUF (Prop_ibuf_I_O)         0.972     3.372 f  araddr_IBUF[5]_inst/O
                         net (fo=44, unplaced)        0.800     4.171    araddr_IBUF[5]
                                                                      f  tap_A_OBUF[7]_inst_i_9/I0
                         LUT1 (Prop_lut1_I0_O)        0.124     4.295 r  tap_A_OBUF[7]_inst_i_9/O
                         net (fo=1, unplaced)         0.000     4.295    tap_A_OBUF[7]_inst_i_9_n_0
                                                                      r  tap_A_OBUF[7]_inst_i_3/S[1]
                         CARRY4 (Prop_carry4_S[1]_CO[3])
                                                      0.533     4.828 r  tap_A_OBUF[7]_inst_i_3/CO[3]
                         net (fo=1, unplaced)         0.009     4.837    tap_A_OBUF[7]_inst_i_3_n_0
                                                                      r  tap_A_OBUF[11]_inst_i_4/CI
                         CARRY4 (Prop_carry4_CI_O[3])
                                                      0.331     5.168 r  tap_A_OBUF[11]_inst_i_4/O[3]
                         net (fo=1, unplaced)         0.618     5.786    addr_reg02_in[11]
                                                                      r  tap_A_OBUF[11]_inst_i_1/I3
                         LUT4 (Prop_lut4_I3_O)        0.307     6.093 r  tap_A_OBUF[11]_inst_i_1/O
                         net (fo=1, unplaced)         0.800     6.893    tap_A_OBUF[11]
                                                                      r  tap_A_OBUF[11]_inst/I
                         OBUF (Prop_obuf_I_O)         2.634     9.528 r  tap_A_OBUF[11]_inst/O
                         net (fo=0)                   0.000     9.528    tap_A[11]
                                                                      r  tap_A[11] (OUT)
  -------------------------------------------------------------------    -------------------

                         (clock axis_clk rise edge)
                                                     12.000    12.000 r
                         clock pessimism              0.000    12.000
                         clock uncertainty           -0.035    11.965
                         output delay                -2.400     9.565
  ---------------------------------------------------------------------
                         required time                          9.565
                         arrival time                          -9.528
  ---------------------------------------------------------------------
                         slack                                  0.037
```

Fig15. Critical path

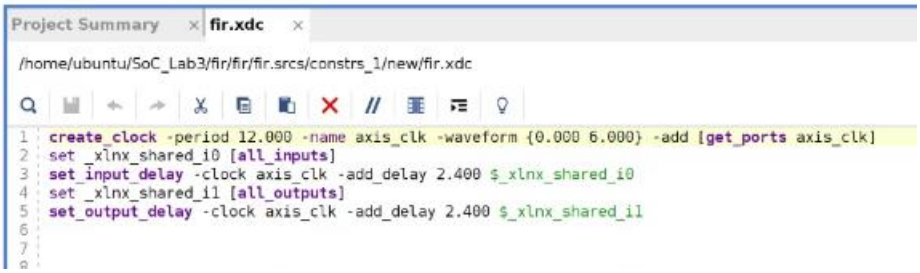

Fig16. Synthesize Constraint

https://github.com/Chih-Ling-Wu/SoC_Lab3.git