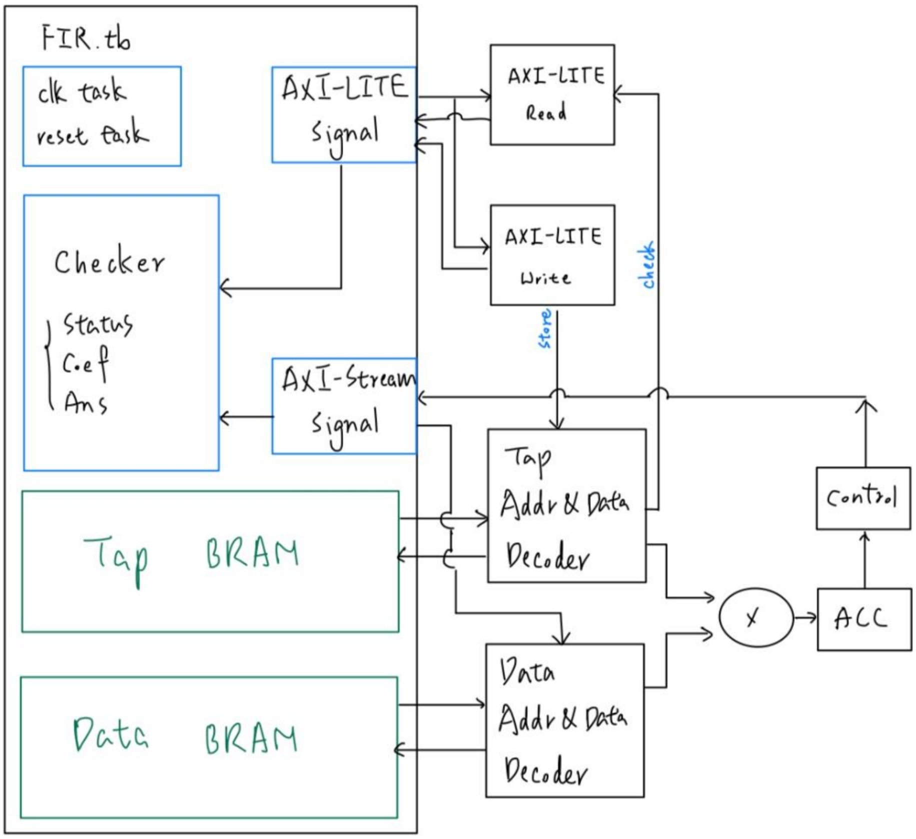
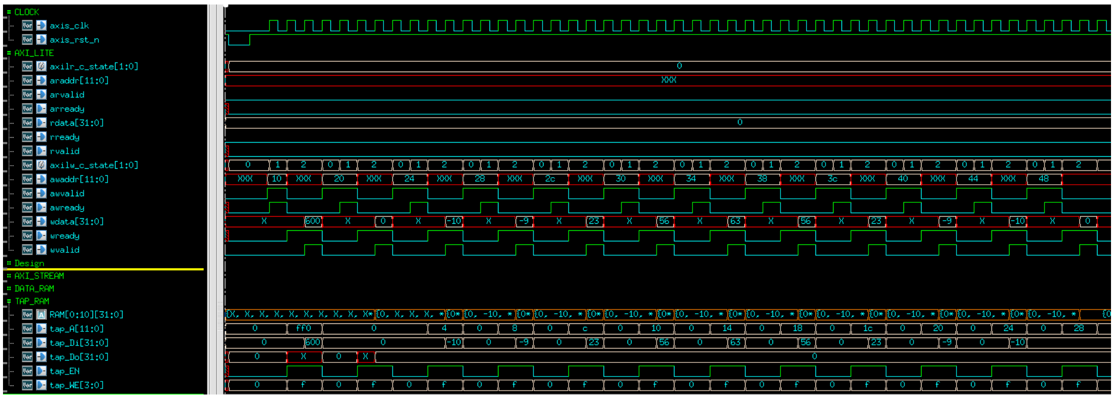
Report

Block diagram:

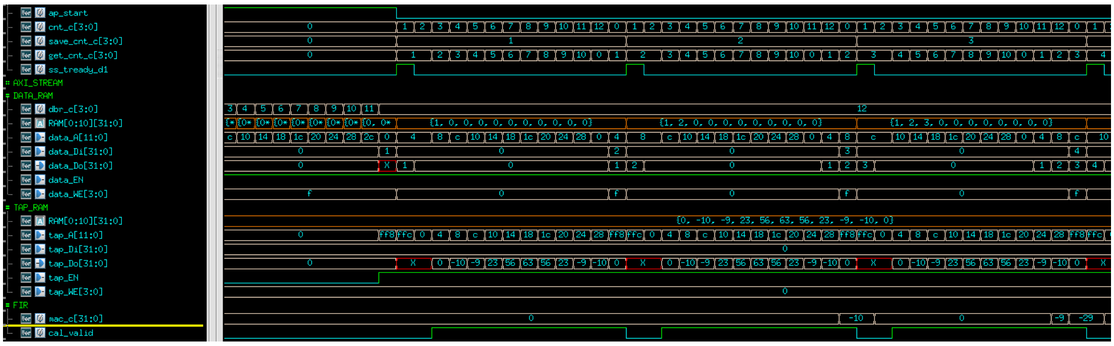


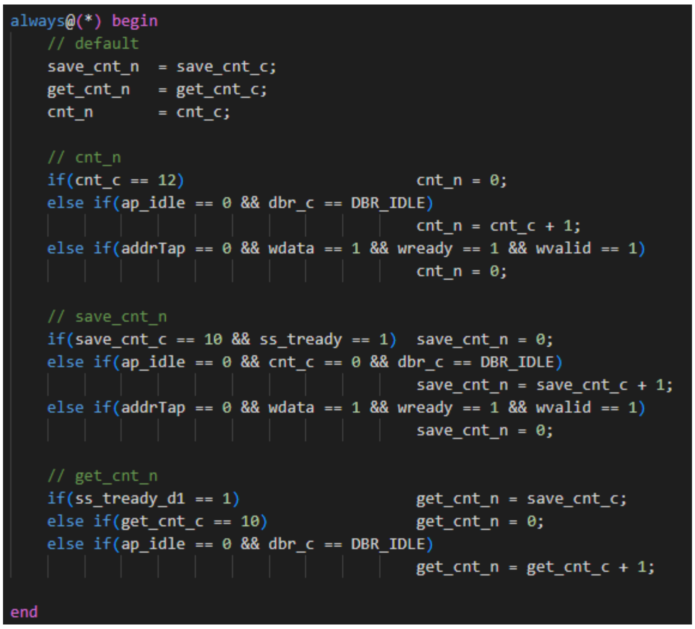
How to receive data-in and tap parameters and place into SRAM?



For AXI-Lite write transaction, I create the FSM to control the signal. When address handshake complete, I can get the correct address and wait for the wdata handshake. Here, I can control the tap_WE signal to write mode. Once handshake complete, the correct data is writing correctly into the tap SRAM.

How to access shiftram and tapRAM to do computation?





```
always@(*) begin
    // default
    save_cnt_n  = save_cnt_c;
    get_cnt_n   = get_cnt_c;
    cnt_n       = cnt_c;

    // cnt_n
    if(cnt_c == 12)                                 cnt_n = 0;
    else if(ap_idle == 0 && dbr_c == DBR_IDLE)
                                                    cnt_n = cnt_c + 1;
    else if(addrTap == 0 && wdata == 1 && wready == 1 && wvalid == 1)
                                                    cnt_n = 0;

    // save_cnt_n
    if(save_cnt_c == 10 && ss_tready == 1)          save_cnt_n = 0;
    else if(ap_idle == 0 && cnt_c == 0 && dbr_c == DBR_IDLE)
                                                    save_cnt_n = save_cnt_c + 1;
    else if(addrTap == 0 && wdata == 1 && wready == 1 && wvalid == 1)
                                                    save_cnt_n = 0;

    // get_cnt_n
    if(ss_tready_d1 == 1)                           get_cnt_n = save_cnt_c;
    else if(get_cnt_c == 10)                        get_cnt_n = 0;
    else if(ap_idle == 0 && dbr_c == DBR_IDLE)
                                                    get_cnt_n = get_cnt_c + 1;

end
```

I use three counters to manipulate the SRAM address encoding. The first counter, cnt_c, I count from 0 to 12. When count to 0 anytime, the input data from axis-protocol stored into the data SRAM and the save counter plus one. At the same time, the third counter, get counter, counts starting from the save counter, and the get counter's value is used for the data BRAM address. After the above counters design, I can get the correct data and coefficient at every cycle.

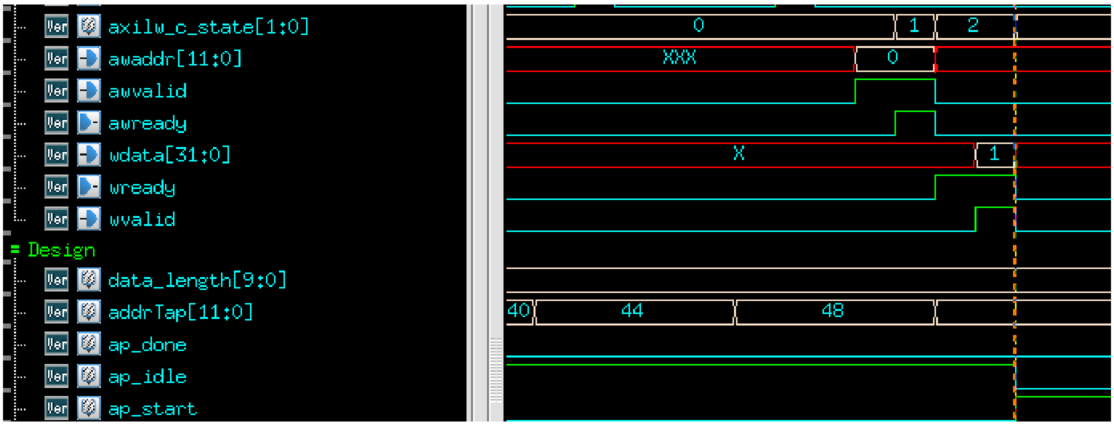
First 13 cycles (2 cycles reserved for read and write)

Data	0	0	0	0	0	0	0	0	0	0	1
Tap	0	-10	-9	23	56	63	56	23	-9	-10	0

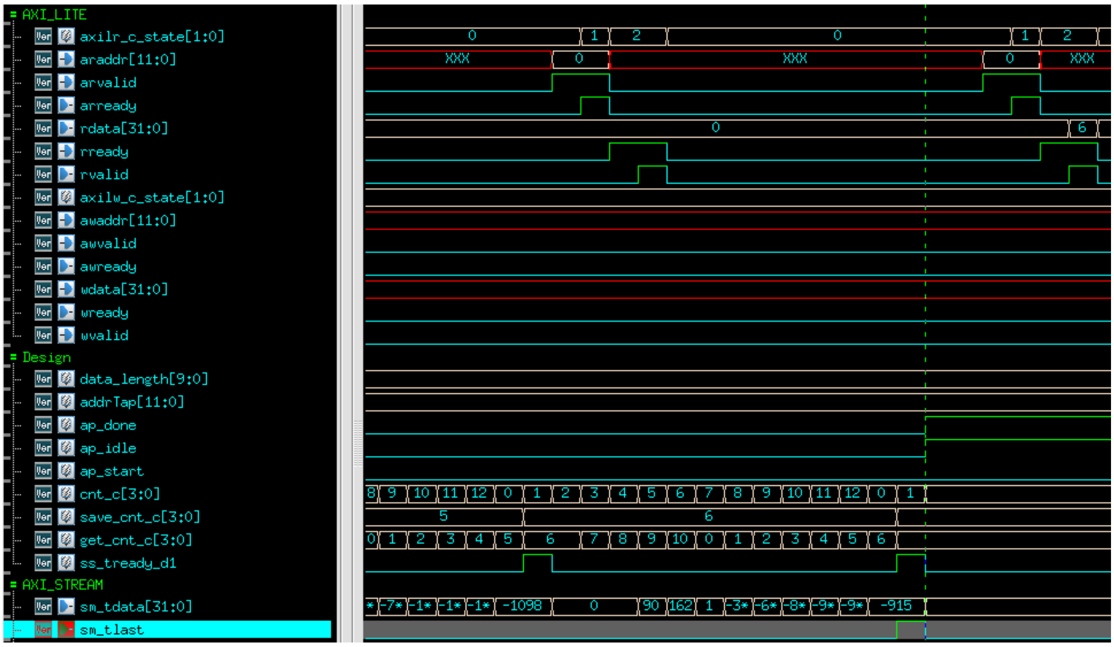
Second 13 cycles

Data	0	0	0	0	0	0	0	0	0	1	2
Tap	0	-10	-9	23	56	63	56	23	-9	-10	0

How to control your status signals?



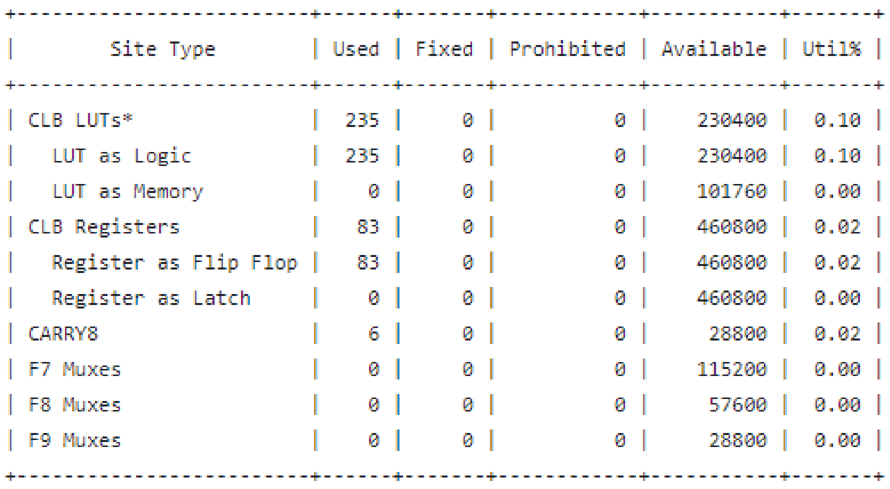
Start signal



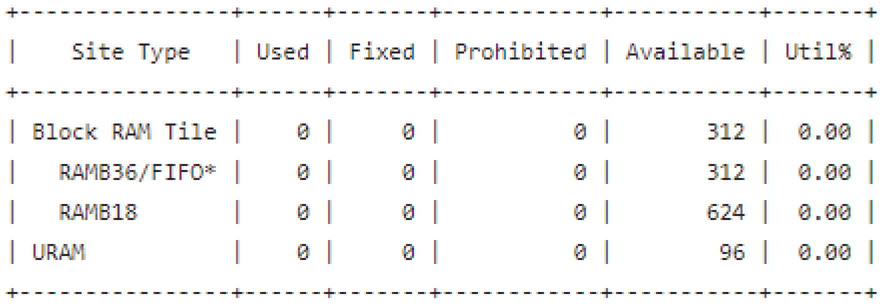
Idle and Done signal

Start signal pulls up when AXI-Lite write transaction writes the 0x00 address and 0 as wdata. Idle and done signals pulls down when the last answer has been transferring out through AXI-Stream transaction. And my status reset method is to wait the fir system get the next start signal then to reset the done and idle signal.

Resource usage: including FF, LUT, BRAM



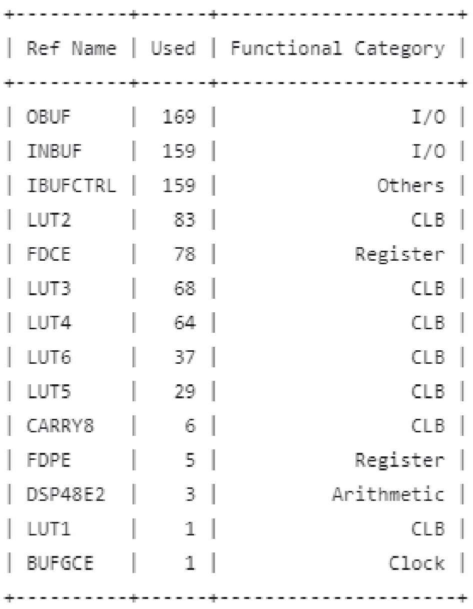
```
+-------------------------+------+-------+------------+-----------+-------+
|        Site Type        | Used | Fixed | Prohibited | Available | Util% |
+-------------------------+------+-------+------------+-----------+-------+
| CLB LUTs*               |  235 |     0 |          0 |    230400 |  0.10 |
|   LUT as Logic          |  235 |     0 |          0 |    230400 |  0.10 |
|   LUT as Memory         |    0 |     0 |          0 |    101760 |  0.00 |
| CLB Registers           |   83 |     0 |          0 |    460800 |  0.02 |
|   Register as Flip Flop |   83 |     0 |          0 |    460800 |  0.02 |
|   Register as Latch     |    0 |     0 |          0 |    460800 |  0.00 |
| CARRY8                  |    6 |     0 |          0 |     28800 |  0.02 |
| F7 Muxes                |    0 |     0 |          0 |    115200 |  0.00 |
| F8 Muxes                |    0 |     0 |          0 |     57600 |  0.00 |
| F9 Muxes                |    0 |     0 |          0 |     28800 |  0.00 |
+-------------------------+------+-------+------------+-----------+-------+
```



```
+----------------+------+-------+------------+-----------+-------+
|    Site Type   | Used | Fixed | Prohibited | Available | Util% |
+----------------+------+-------+------------+-----------+-------+
| Block RAM Tile |    0 |     0 |          0 |       312 |  0.00 |
|   RAMB36/FIFO* |    0 |     0 |          0 |       312 |  0.00 |
|   RAMB18       |    0 |     0 |          0 |       624 |  0.00 |
| URAM           |    0 |     0 |          0 |        96 |  0.00 |
+----------------+------+-------+------------+-----------+-------+
```

BRAM usage

Since this design’s RAMs are not the inner design’s RAM, the synthesis report might not generate the BRAM in used.



```
+----------+------+---------------------+
| Ref Name | Used | Functional Category |
+----------+------+---------------------+
| OBUF     |  169 |                 I/O |
| INBUF    |  159 |                 I/O |
| IBUFCTRL |  159 |              Others |
| LUT2     |   83 |                 CLB |
| FDCE     |   78 |            Register |
| LUT3     |   68 |                 CLB |
| LUT4     |   64 |                 CLB |
| LUT6     |   37 |                 CLB |
| LUT5     |   29 |                 CLB |
| CARRY8   |    6 |                 CLB |
| FDPE     |    5 |            Register |
| DSP48E2  |    3 |          Arithmetic |
| LUT1     |    1 |                 CLB |
| BUFGCE   |    1 |               Clock |
+----------+------+---------------------+
```

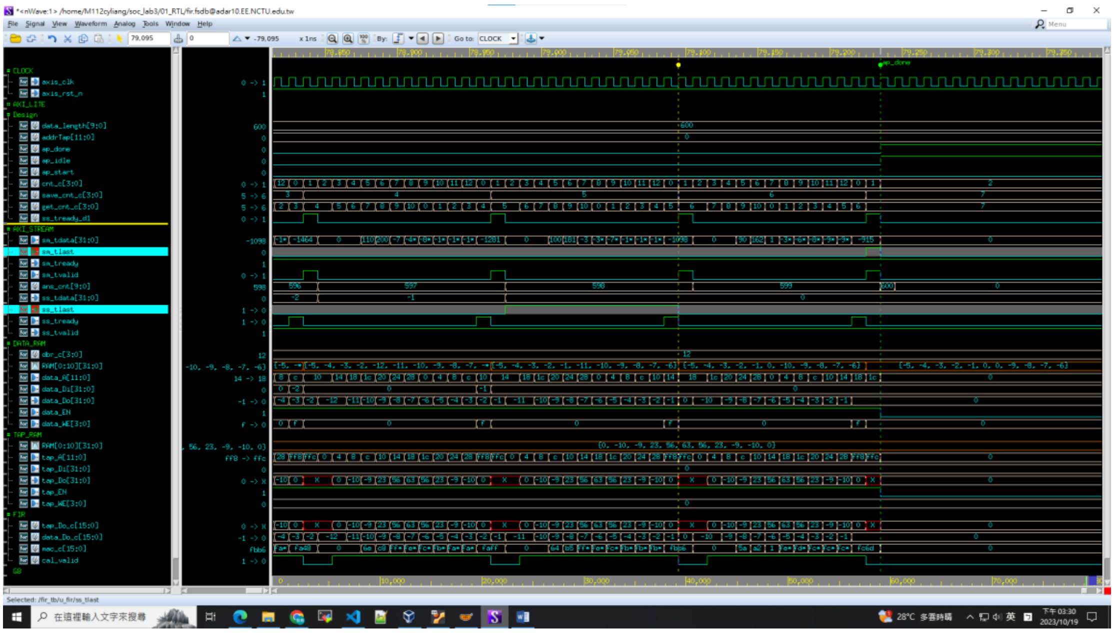
Nearly all the logics being synthesized are generated into LUT.

Timing report

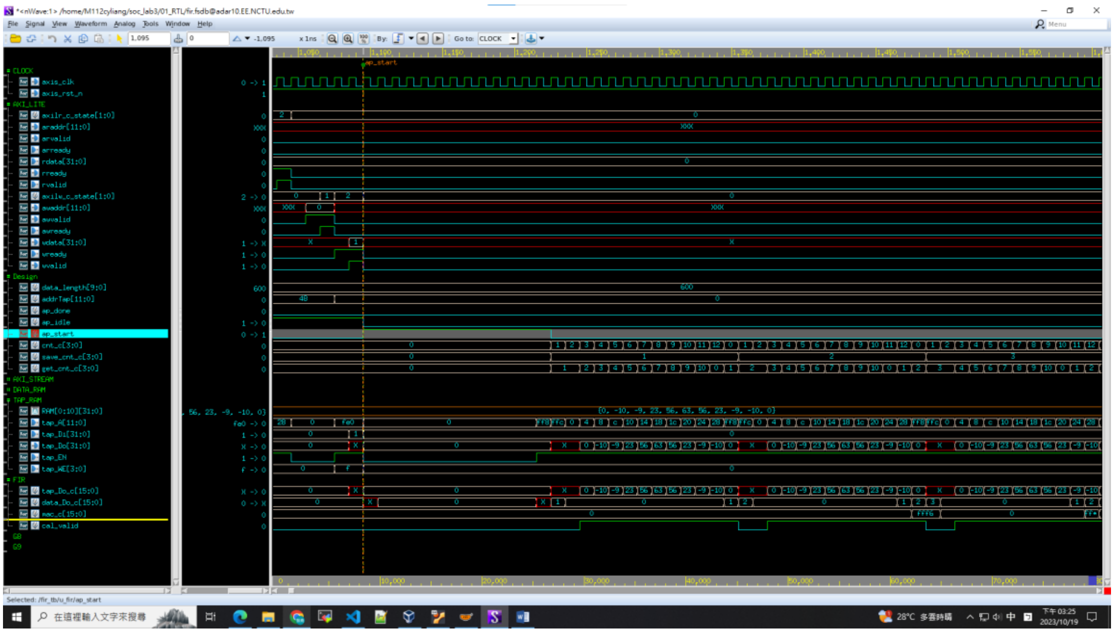
```
-------------------------------------------------------------------------------------------------
From Clock:  axis_clk
  To Clock:  axis_clk

Setup :            0  Failing Endpoints,  Worst Slack        1.306ns,  Total Violation        0.000ns
Hold  :           NA  Failing Endpoints,  Worst Slack           NA  ,  Total Violation             NA
PW    :            0  Failing Endpoints,  Worst Slack        5.725ns,  Total Violation        0.000ns
-------------------------------------------------------------------------------------------------
```

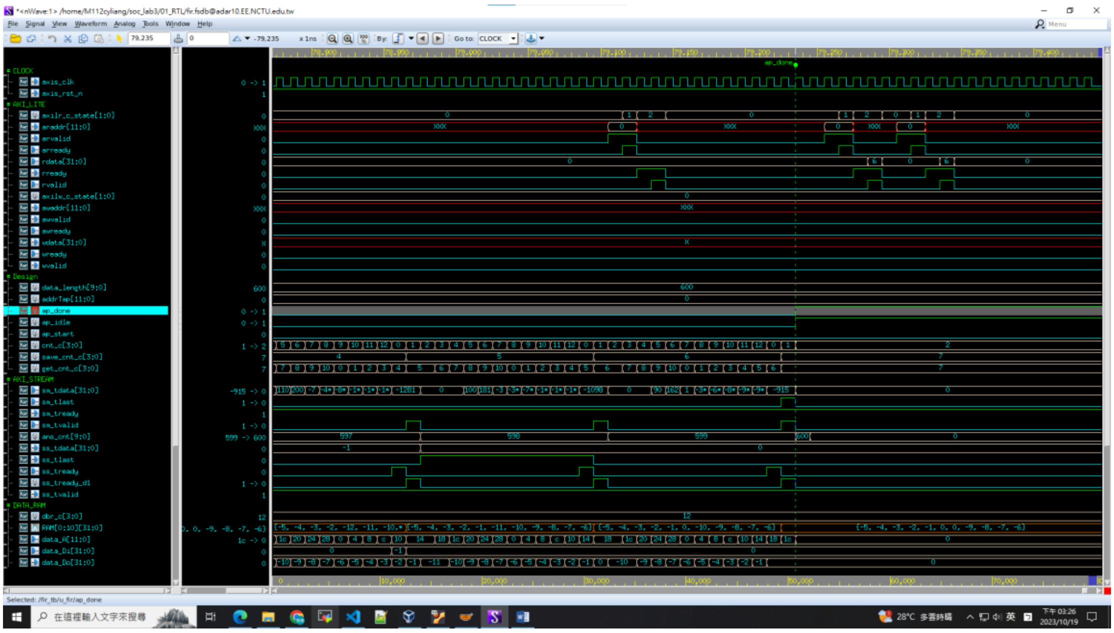
Waveform



AXI-Stream in out last



ap_start at 1095 ns



ap_done at 79235 ns

The total computing latency: (79235 – 1095) / 10ns = 7814 cycles