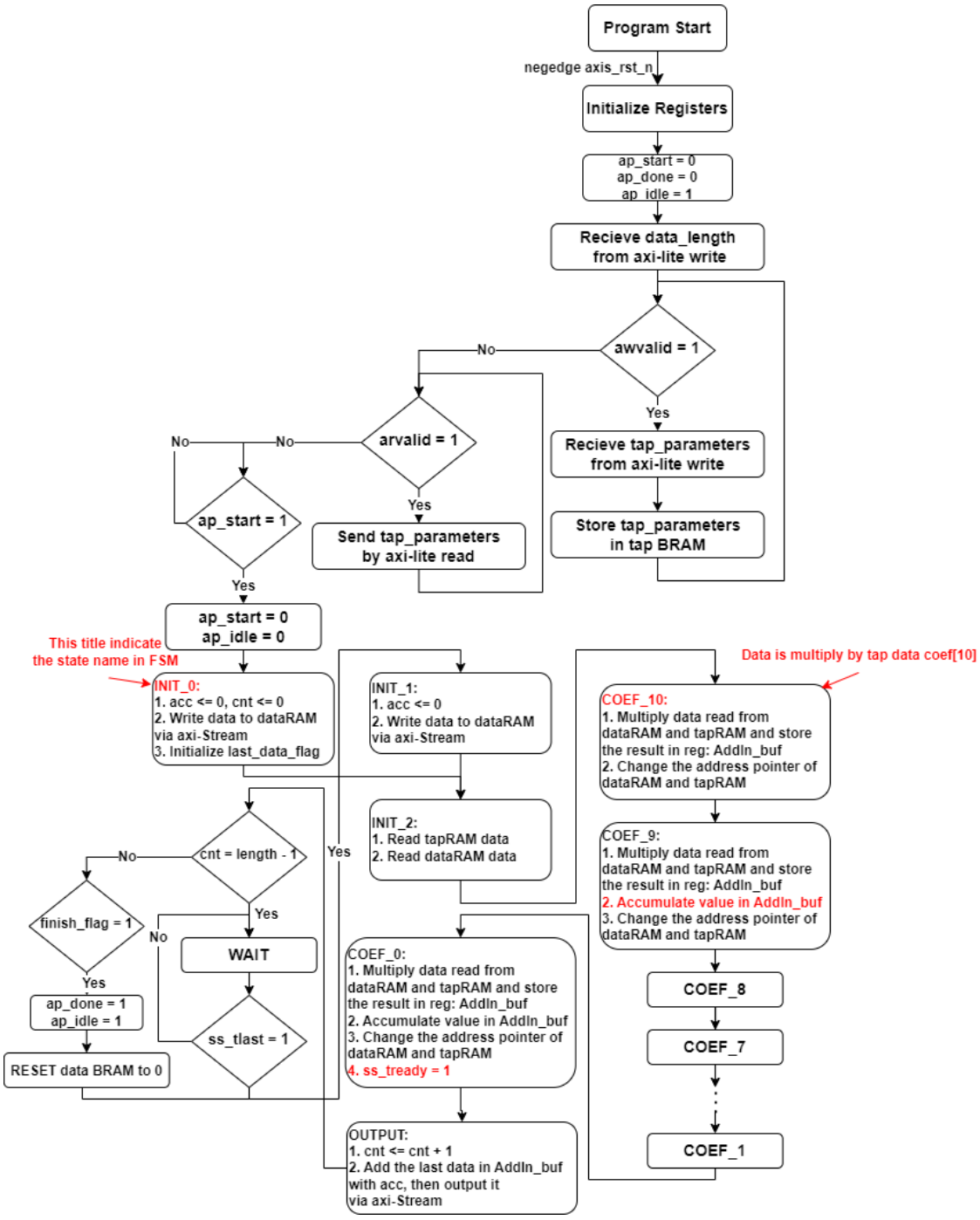
Lab 3 Report

310511082 王品棠

Block Diagram



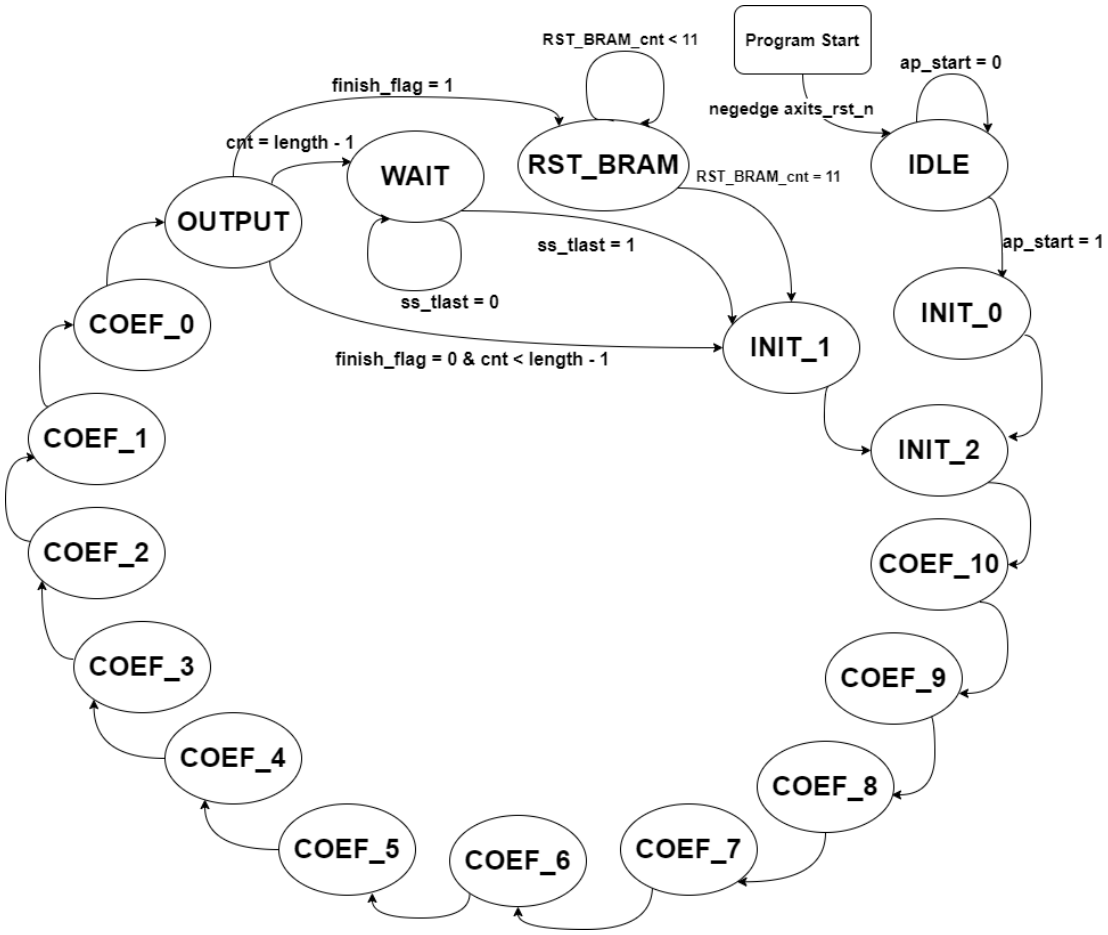
Describe Operation

How to receive data-in (axi-write) and tap parameters and place into SRAM

- When signal “awvalid” raises, raise “awready” and check the “awaddr” is 12’h00, 12’h10 or address equal or larger than 12’h20, determining the operation.
- If address is 12’h00, write the data to reg: ap_signals, if the address is 12’h10, write data to reg: length
- If the address is equal or larger than 12’h20, write the data to reg: write_buf(connected to tap_Di), write the mapped address to addr_buf(connected with tap_A) and raise the enable registers of tap_EN and tap_WE, then the tap parameters will be written into tapRAM in the next cycle.
- In this process, the signal “wready” is raised while the tap parameters are written into the SRAM, then the next tap parameter will be able to be received.

How to access shiftRam and tapRAM to do computation

- As shown in the following state diagram, the input data from axi-Stream will be stored into the shiftRAM (dataRAM) in the INIT_0 or INIT_1.
- The next state INIT_2, the enable signal and the address of shiftRAM and tapRAM will be calculated and assign to the register connected to their address signal wire.
- The next state COEF_10, the oldest data will be read from the shiftRAM and be multiplied by the last coef[10] that is read from tapRAM, and store the result in the reg: AddIn_buf.
- The following states, from COEF_9 to COEF_0, the same operation as COEF_10 is conducted but also accumulate the AddIn_buf to the accumulator “acc”.
- The next state OUTPUT, accumulate the AddIn_buf with the reg: acc for the last time of this output data, and assign the result to the register “sm_tdata_bug” (connected to sm_tdata). And also raise the signal sm_tvalid.
- In the program, the input data received via axi-Stream will be first stored in the shiftRAM in the first cycle, then the following 12 cycles are calculating the corresponding FIR output of this input data.
- Since we can design with only one multiplier and one adder for FIR calculation, I think the method I applied is the most efficient way analyzed with asymptotic analysis.



How ap_done is generated.

- The signal ap_done is generated after the last data is calculated and transferred.

Resource usage: including FF, LUT, BRAM

- From the synthesis log file, the following cell usage shows the FF and LUT usage.
- There are 263 LUTs and 315 FFs in total.

```
Report Cell Usage:
+------+--------+------+
|      |Cell    |Count |
+------+--------+------+
|1     |BUFG    |     1|
|2     |CARRY4  |    33|
|3     |DSP48E1 |     3|
|5     |LUT1    |    25|
|6     |LUT2    |    87|
|7     |LUT3    |    20|
|8     |LUT4    |    37|
|9     |LUT5    |    43|
|10    |LUT6    |    51|
|11    |FDCE    |   133|
|12    |FDPE    |     1|
|13    |FDRE    |   181|
|14    |IBUF    |   159|
|15    |OBUF    |   168|
|16    |OBUFT   |     1|
+------+--------+------+
```

Timing Report

Try to synthesize the design with maximum frequency

- The minimum period achieved by the design is 5.228ns, with the corresponding maximum frequency of 191MHz.

Timing Constraints

Create Clock

Position	Clock Name	Period (ns)	Rise At (ns)	Fall At (ns)	Add Clock	Source Objects	Source File
1	axis_clk	5.228	0.000	2.614	☐	[get_ports axis_clk]	fir.xdc

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	0.001 ns	Worst Hold Slack (WHS):	0.067 ns	Worst Pulse Width Slack (WPWS):	2.114 ns
Total Negative Slack (TNS):	0.000 ns	Total Hold Slack (THS):	0.000 ns	Total Pulse Width Negative Slack (TPWS):	0.000 ns
Number of Failing Endpoints:	0	Number of Failing Endpoints:	0	Number of Failing Endpoints:	0
Total Number of Endpoints:	507	Total Number of Endpoints:	507	Total Number of Endpoints:	318

All user specified timing constraints are met.

Report timing on longest path, slack

- The longest path is from cnt_reg[3] to cnt_reg{29}, which is the counter register used to accumulate how many data have already received.
- Since the carrier bit signal has to transmit through lots of bit in the register, the critical path may be long.
- With the maximum frequency, the slack for this maximum delay path is 0.001ns, still a positive number.

```
Max Delay Paths
--------------------------------------------------------------------------------------
Slack (MET) :             0.001ns  (required time - arrival time)
  Source:                 genblk1.cnt_reg[3]/C
                            (rising edge-triggered cell FDCE clocked by axis_clk  {rise@0.000ns fall@2.614ns period=5.228ns})
  Destination:            genblk1.cnt_reg[29]/D
                            (rising edge-triggered cell FDCE clocked by axis_clk  {rise@0.000ns fall@2.614ns period=5.228ns})
  Path Group:             axis_clk
  Path Type:              Setup (Max at Slow Process Corner)
  Requirement:            5.228ns  (axis_clk rise@5.228ns - axis_clk rise@0.000ns)
  Data Path Delay:        5.123ns  (logic 3.592ns (70.115%)  route 1.531ns (29.885%))
  Logic Levels:           13  (CARRY4=11 LUT1=1 LUT6=1)
  Clock Path Skew:        -0.145ns (DCD - SCD + CPR)
    Destination Clock Delay (DCD):    2.128ns = ( 7.356 - 5.228 )
    Source Clock Delay      (SCD):    2.456ns
    Clock Pessimism Removal (CPR):    0.184ns
  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
                                                      0.000     0.000 r  axis_clk (IN)
                         net (fo=0)                   0.000     0.000    axis_clk
                                                                     r  axis_clk_IBUF_inst/I
                         IBUF (Prop_ibuf_I_O)         0.972     0.972 r  axis_clk_IBUF_inst/O
                         net (fo=1, unplaced)         0.800     1.771    axis_clk_IBUF
                                                                     r  axis_clk_IBUF_BUFG_inst/I
                         BUFG (Prop_bufg_I_O)         0.101     1.872 r  axis_clk_IBUF_BUFG_inst/O
                         net (fo=317, unplaced)       0.584     2.456    axis_clk_IBUF_BUFG
                         FDCE                                           r  genblk1.cnt_reg[3]/C
  -------------------------------------------------------------------    -------------------
                         FDCE (Prop_fdce_C_Q)         0.478     2.934 r  genblk1.cnt_reg[3]/Q
                         net (fo=3, unplaced)         0.983     3.917    genblk1.cnt_reg[3]
                                                                     r  genblk1.ap_signals[2]_i_14/I0
                         LUT6 (Prop_lut6_I0_O)        0.295     4.212 r  genblk1.ap_signals[2]_i_14/O
                         net (fo=1, unplaced)         0.000     4.212    genblk1.ap_signals[2]_i_14_n_0
```

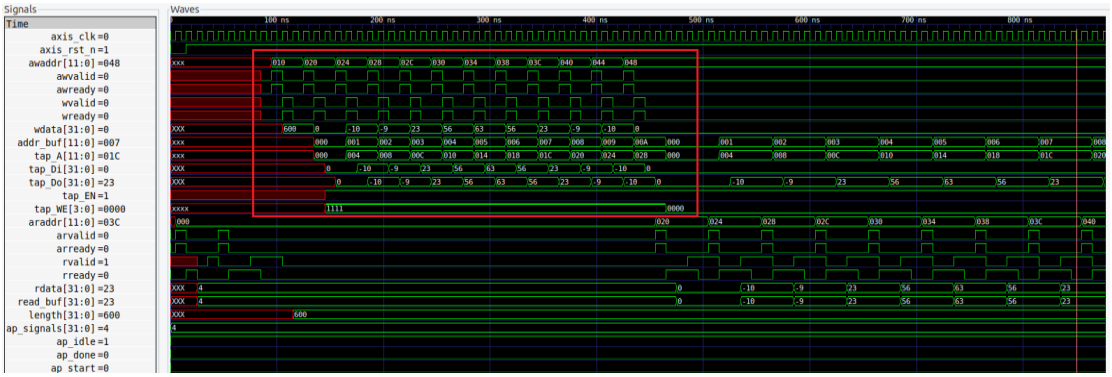
```
                                                  r  genblk1.ap_signals[2]_i_14/I0
LUT6 (Prop_lut6_I0_O)        0.295     4.212 r  genblk1.ap_signals[2]_i_14/O
net (fo=1, unplaced)         0.000     4.212    genblk1.ap_signals[2]_i_14_n_0
                                                  r  genblk1.ap_signals_reg[2]_i_7/S[1]
CARRY4 (Prop_carry4_S[1]_CO[3])
                             0.533     4.745 r  genblk1.ap_signals_reg[2]_i_7/CO[3]
net (fo=1, unplaced)         0.009     4.754    genblk1.ap_signals_reg[2]_i_7_n_0
                                                  r  genblk1.ap_signals_reg[2]_i_3/CI
CARRY4 (Prop_carry4_CI_CO[3])
                             0.117     4.871 r  genblk1.ap_signals_reg[2]_i_3/CO[3]
net (fo=1, unplaced)         0.000     4.871    genblk1.ap_signals_reg[2]_i_3_n_0
                                                  r  genblk1.ap_signals_reg[2]_i_2/CI
CARRY4 (Prop_carry4_CI_CO[2])
                             0.252     5.123 f  genblk1.ap_signals_reg[2]_i_2/CO[2]
net (fo=42, unplaced)        0.530     5.653    genblk1.ap_signals_reg[2]_i_2_n_1
                                                  f  genblk1.cnt[0]_i_3/I0
LUT1 (Prop_lut1_I0_O)        0.302     5.955 r  genblk1.cnt[0]_i_3/O
net (fo=1, unplaced)         0.000     5.955    genblk1.cnt[0]_i_3_n_0
                                                  r  genblk1.cnt_reg[0]_i_2/DI[0]
CARRY4 (Prop_carry4_DI[0]_CO[3])
                             0.576     6.531 r  genblk1.cnt_reg[0]_i_2/CO[3]
net (fo=1, unplaced)         0.009     6.540    genblk1.cnt_reg[0]_i_2_n_0
                                                  r  genblk1.cnt_reg[4]_i_1/CI
CARRY4 (Prop_carry4_CI_CO[3])
                             0.117     6.657 r  genblk1.cnt_reg[4]_i_1/CO[3]
net (fo=1, unplaced)         0.000     6.657    genblk1.cnt_reg[4]_i_1_n_0
                                                  r  genblk1.cnt_reg[8]_i_1/CI
CARRY4 (Prop_carry4_CI_CO[3])
                             0.117     6.774 r  genblk1.cnt_reg[8]_i_1/CO[3]
net (fo=1, unplaced)         0.000     6.774    genblk1.cnt_reg[8]_i_1_n_0
                                                  r  genblk1.cnt_reg[12]_i_1/CI
CARRY4 (Prop_carry4_CI_CO[3])
                             0.117     6.891 r  genblk1.cnt_reg[12]_i_1/CO[3]
net (fo=1, unplaced)         0.000     6.891    genblk1.cnt_reg[12]_i_1_n_0
                                                  r  genblk1.cnt_reg[16]_i_1/CI
CARRY4 (Prop_carry4_CI_CO[3])
                             0.117     7.008 r  genblk1.cnt_reg[16]_i_1/CO[3]
net (fo=1, unplaced)         0.000     7.008    genblk1.cnt_reg[16]_i_1_n_0
                                                  r  genblk1.cnt_reg[20]_i_1/CI
CARRY4 (Prop_carry4_CI_CO[3])
                             0.117     7.125 r  genblk1.cnt_reg[20]_i_1/CO[3]
net (fo=1, unplaced)         0.000     7.125    genblk1.cnt_reg[20]_i_1_n_0
                                                  r  genblk1.cnt_reg[24]_i_1/CI
CARRY4 (Prop_carry4_CI_CO[3])
                             0.117     7.242 r  genblk1.cnt_reg[24]_i_1/CO[3]
net (fo=1, unplaced)         0.000     7.242    genblk1.cnt_reg[24]_i_1_n_0
                                                  r  genblk1.cnt_reg[28]_i_1/CI
CARRY4 (Prop_carry4_CI_O[1])
                             0.337     7.579 r  genblk1.cnt_reg[28]_i_1/O[1]
net (fo=1, unplaced)         0.000     7.579    genblk1.cnt_reg[28]_i_1_n_6
FDCE                                           r  genblk1.cnt_reg[29]/D
-------------------------------------------    -------------------

(clock axis_clk rise edge)
                             5.228     5.228 r
                             0.000     5.228 r  axis_clk (IN)
net (fo=0)                   0.000     5.228    axis_clk
                                                  r  axis_clk_IBUF_inst/I
IBUF (Prop_ibuf_I_O)         0.838     6.066 r  axis_clk_IBUF_inst/O
net (fo=1, unplaced)         0.760     6.826    axis_clk_IBUF
                                                  r  axis_clk_IBUF_BUFG_inst/I
BUFG (Prop_bufg_I_O)         0.091     6.917 r  axis_clk_IBUF_BUFG_inst/O
net (fo=317, unplaced)       0.439     7.356    axis_clk_IBUF_BUFG
FDCE                                           r  genblk1.cnt_reg[29]/C
clock pessimism              0.184     7.539
clock uncertainty           -0.035     7.504
FDCE (Setup_fdce_C_D)        0.076     7.580    genblk1.cnt_reg[29]
-------------------------------------------
required time                          7.580
arrival time                          -7.579
-------------------------------------------
slack                                  0.001
```

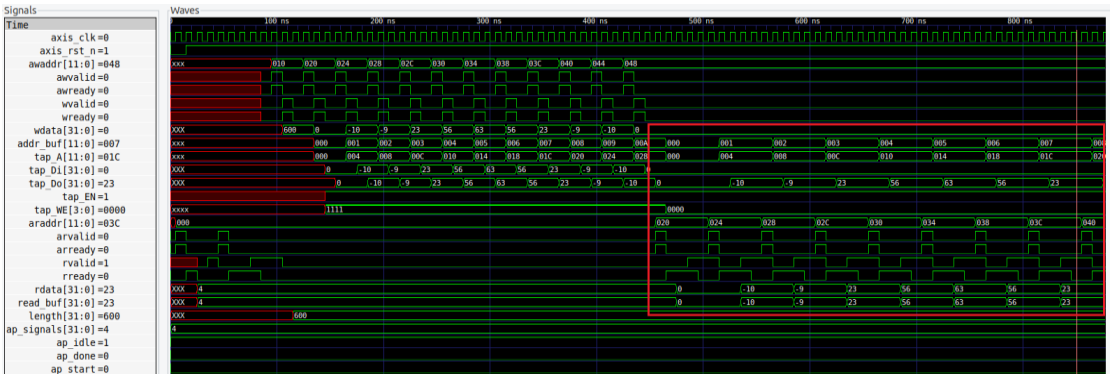
Simulation Waveform

Coefficient program, and read back

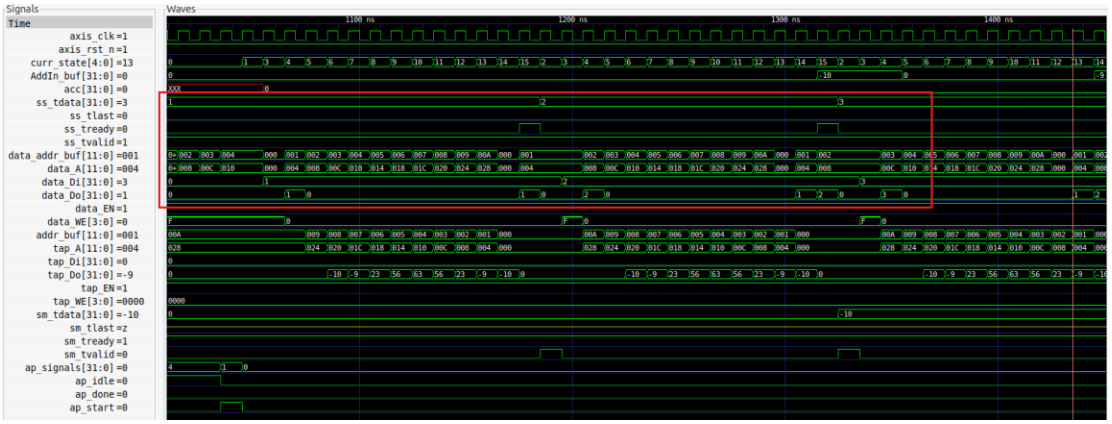
- Receive tap parameters from axi-write and write to tapRAM

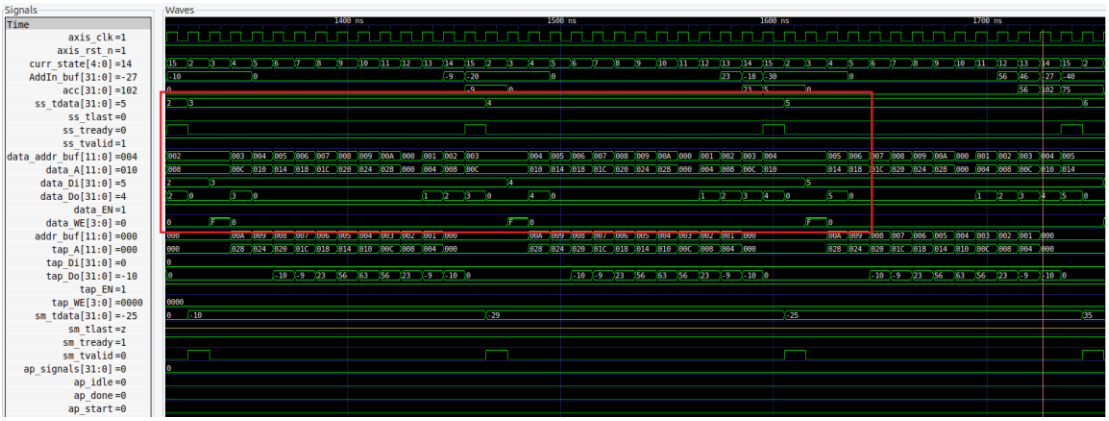


- Read tap parameters from tapRAM and send to testbench via axi-read

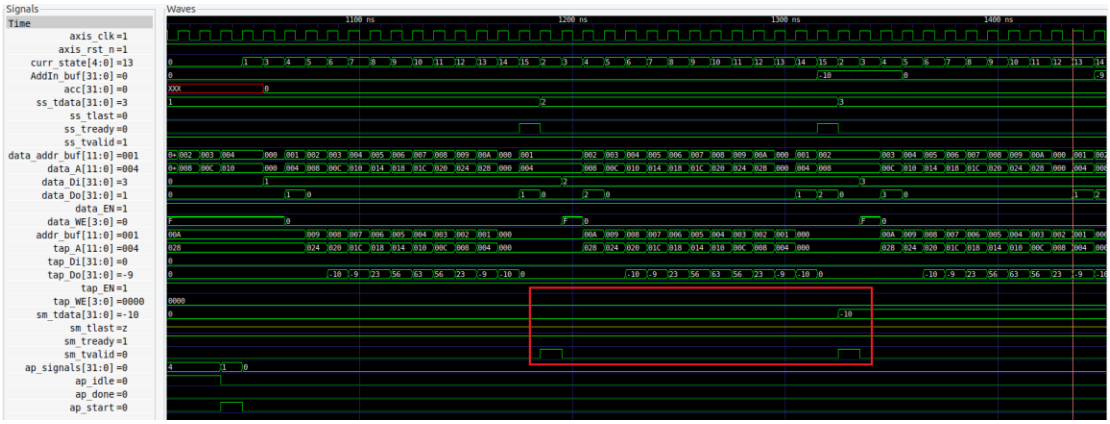


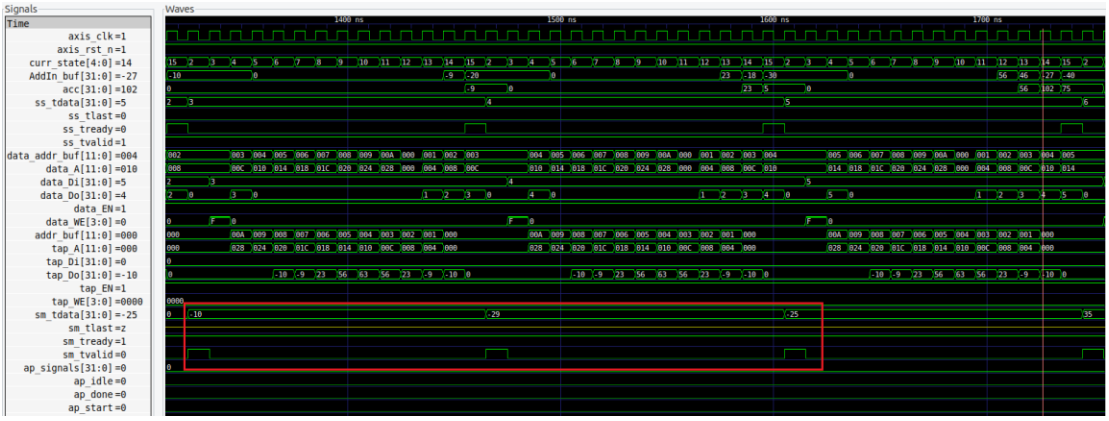
Data-in stream-in





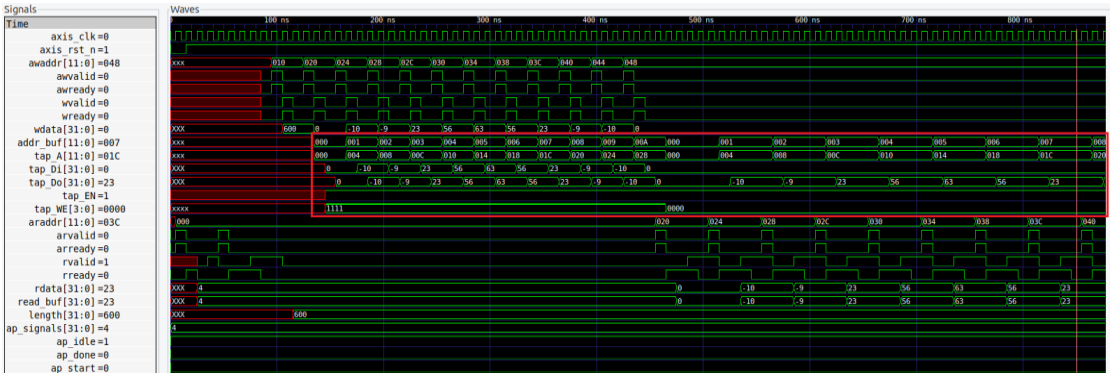
Data-out stream-out



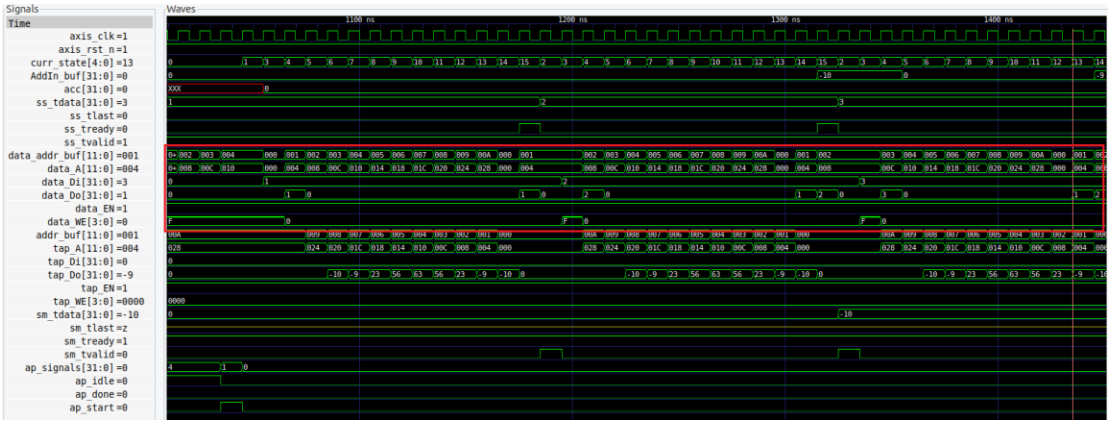


RAM access control

- Tap RAM



- Data RAM



FSM