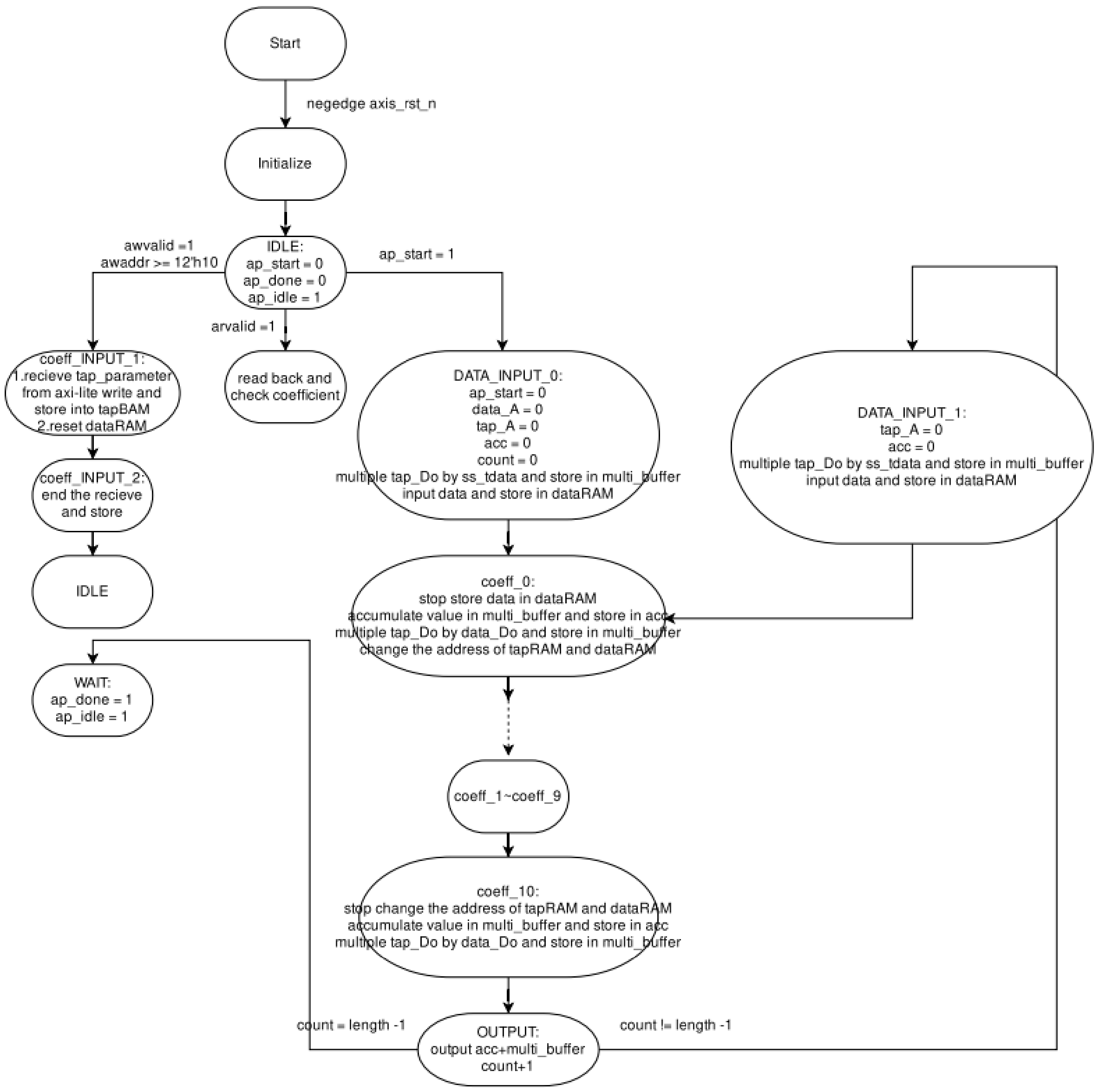
SOC Lab

Lab 3

電子碩二 311510060 李秉翰

Block Diagram


Start
negedge axis_rst_n
Initialize
IDLE: ap_start = 0 ap_done = 0 ap_idle = 1
awvalid =1 awaddr >= 12'h10
ap_start = 1
arvalid =1
coeff_INPUT_1: 1.recieve tap_parameter from axi-lite write and store into tapBAM 2.reset dataRAM
read back and check coefficient
DATA_INPUT_0: ap_start = 0 data_A = 0 tap_A = 0 acc = 0 count = 0 multiple tap_Do by ss_tdata and store in multi_buffer input data and store in dataRAM
DATA_INPUT_1: tap_A = 0 acc = 0 multiple tap_Do by ss_tdata and store in multi_buffer input data and store in dataRAM
coeff_INPUT_2: end the recieve and store
IDLE
coeff_0: stop store data in dataRAM accumulate value in multi_buffer and store in acc multiple tap_Do by data_Do and store in multi_buffer change the address of tapRAM and dataRAM
WAIT: ap_done = 1 ap_idle = 1
coeff_1~coeff_9
coeff_10: stop change the address of tapRAM and dataRAM accumulate value in multi_buffer and store in acc multiple tap_Do by data_Do and store in multi_buffer
count = length -1
OUTPUT: output acc+multi_buffer count+1
count != length -1

Describe operation

- How to receive data-in and tap parameters and place into SRAM
  When signal “awvalid” raises, check the “awaddr”. If the “awaddr” is 12’h00, raise the “awready” and “wready” signal to write the “ap_signal” register to get ap_start signal. If the “awaddr” is 12’h10, raise the “awready” and “wready” signal to write the “length" register to store the data length. If the “awaddr” is equal or bigger than 12’h20, raise the “awready”, “wready”, “tap_EN”, “tap_WE” and write the address

"awaddr – 12'h20" to reg: tap_A_buffer(connect to tap_A) and write the data to reg: tap_Di_buffer(connect to tap_Di). Then, the tap parameters can place into SRAM. After all, reset the "wready" and "awready" signal in order to receive the next tap parameter.

- How to access shiftram and tapRAM to do computation

  State "DATA_INPUT_0": Save the first input data from axi-Stream into the shiftRAM (dataRAM) and set the address 12'h00 to reg: data_A_buffer (connect to data_A). Read the coeff at tap_A = 12'h00 which was stored in the tapRAM and multiplied by the input data "ss_tdata" and store the result in the reg: multi_buffer.

  State "DATA_INPUT_1": Save the input data except the first one from axi-Stream into the shiftRAM (dataRAM). Read the coeff at tap_A = 12'h00 which was stored in the tapRAM and multiplied by the input data "ss_tdata" and store the result in the reg: multi_buffer.

  State "coeff_0" to "coeff_9": Multiple the "tap_Do" by the "data_Do" and store in the reg: multi_buffer. Plus the reg: tap_A_buffer with 12'h04 to read the next coeff for the next state. Then check the reg: data_A_buffer. If it equal to 12'h00, write it to 12'h28. Otherwise, minus with 12'h04 read the previous data stored in the dataRAM. At the same time, save the data "acc + multi_buffer" into reg: acc.

  State "coeff_10": Because the last data_A and tap_A has changed at the previous state, this state only need to do multiple store in the reg: multi_buffer and accumulate "acc +multi_buffer".

  State "OUTPUT": raise the signal "sm_tvalid" and output the data with "acc+multi_buffer". Then the next state change to state "DATA_INPUT_1"

  Through state "DATA_INPUT_1" to "OUTPUT", the fir will start calculate with the coeff stored in the address "12'h00" in the tapRAM and the data which the address shift 12'h04 every cycle in the dataRAM.

- How ap_done is generated

After the last data is calculated and transferred, write the ap_done signal to 1.

Resource usage: including FF, LUT, BRAM

FF: 263

LUT: 248

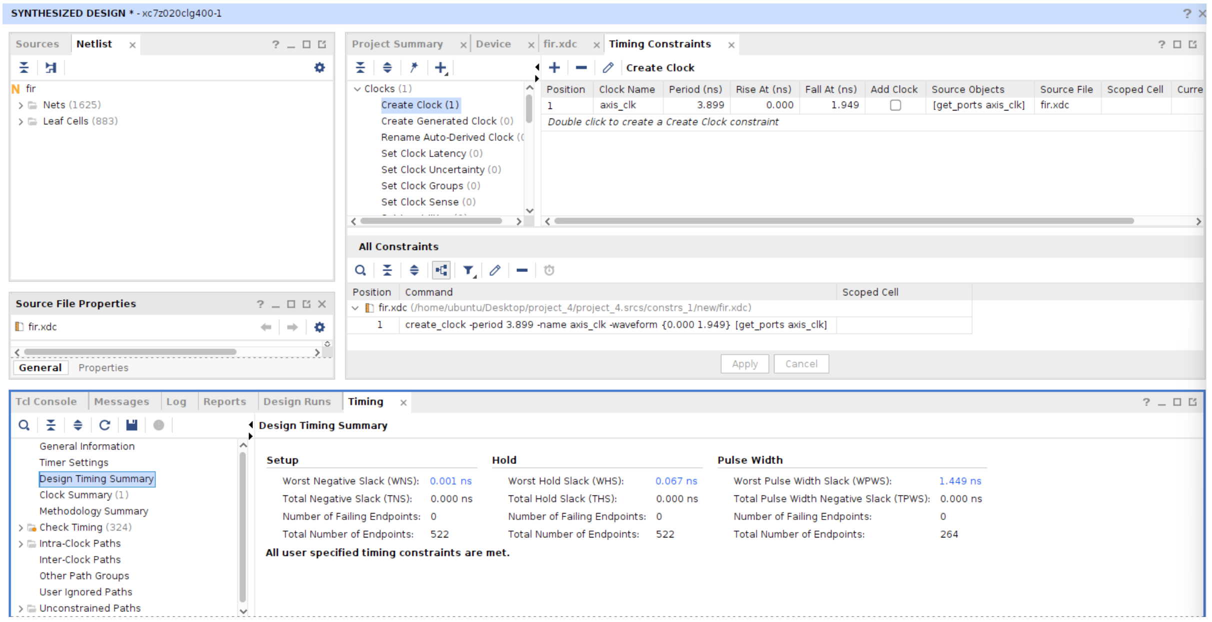
```
Report Cell Usage:
+------+--------+------+
|      |Cell    |Count |
+------+--------+------+
|1     |BUFG    |     1|
|2     |CARRY4  |    34|
|3     |DSP48E1 |     6|
|4     |LUT1    |    27|
|5     |LUT2    |    63|
|6     |LUT3    |     5|
|7     |LUT4    |    87|
|8     |LUT5    |    34|
|9     |LUT6    |    32|
|10    |FDCE    |   167|
|11    |FDRE    |    96|
|12    |IBUF    |   158|
|13    |OBUF    |   169|
+------+--------+------+
```

Timing Report

- **Try to synthesize the design with maximum frequency**

  The minimum period can achieve 3.899 ns.

  The maximum frequency is 256.476 MHz.



SYNTHESIZED DESIGN * - xc7z020clg400-1

Netlist: fir — Nets (1625), Leaf Cells (883)

Timing Constraints — Clocks (1): Create Clock (1), Create Generated Clock (0), Rename Auto-Derived Clock, Set Clock Latency (0), Set Clock Uncertainty (0), Set Clock Groups (0), Set Clock Sense (0)

Position	Clock Name	Period (ns)	Rise At (ns)	Fall At (ns)	Add Clock	Source Objects	Source File	Scoped Cell
1	axis_clk	3.899	0.000	1.949	☐	[get_ports axis_clk]	fir.xdc	

Double click to create a Create Clock constraint

All Constraints

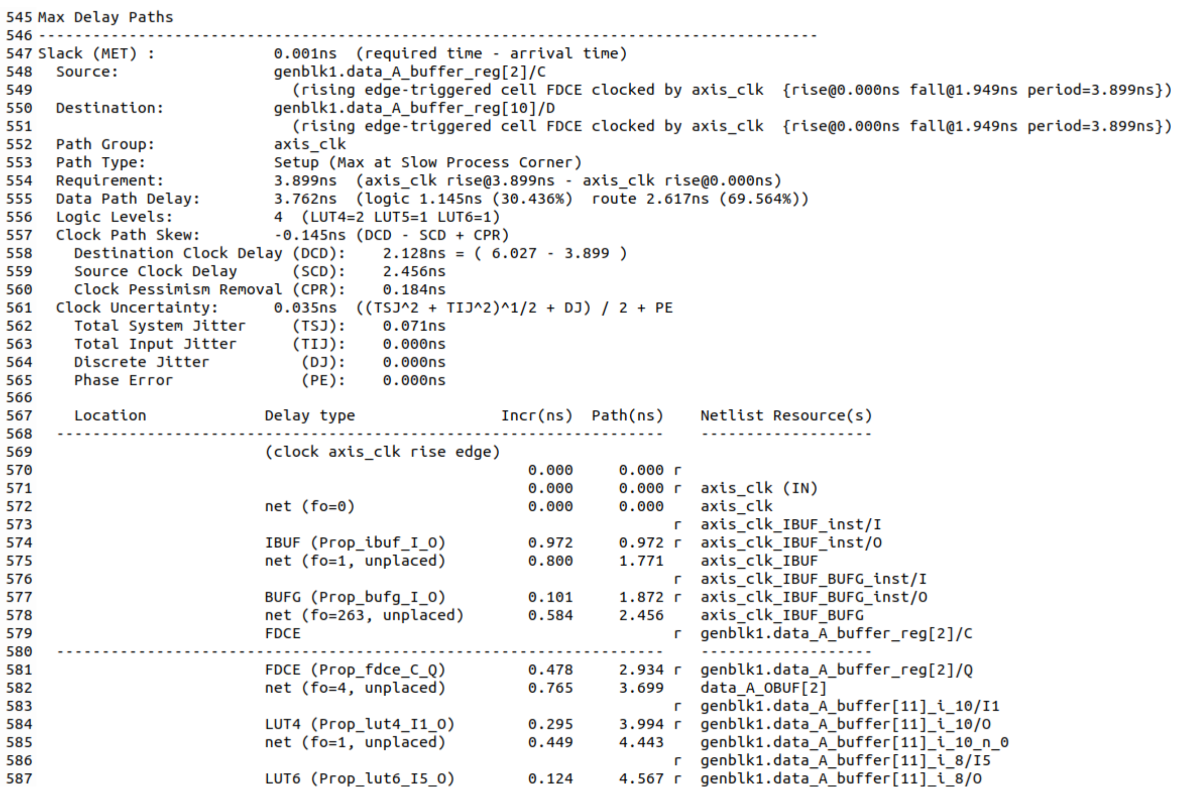
Position	Command	Scoped Cell
	fir.xdc (/home/ubuntu/Desktop/project_4/project_4.srcs/constrs_1/new/fir.xdc)	
1	create_clock -period 3.899 -name axis_clk -waveform {0.000 1.949} [get_ports axis_clk]	

Design Timing Summary

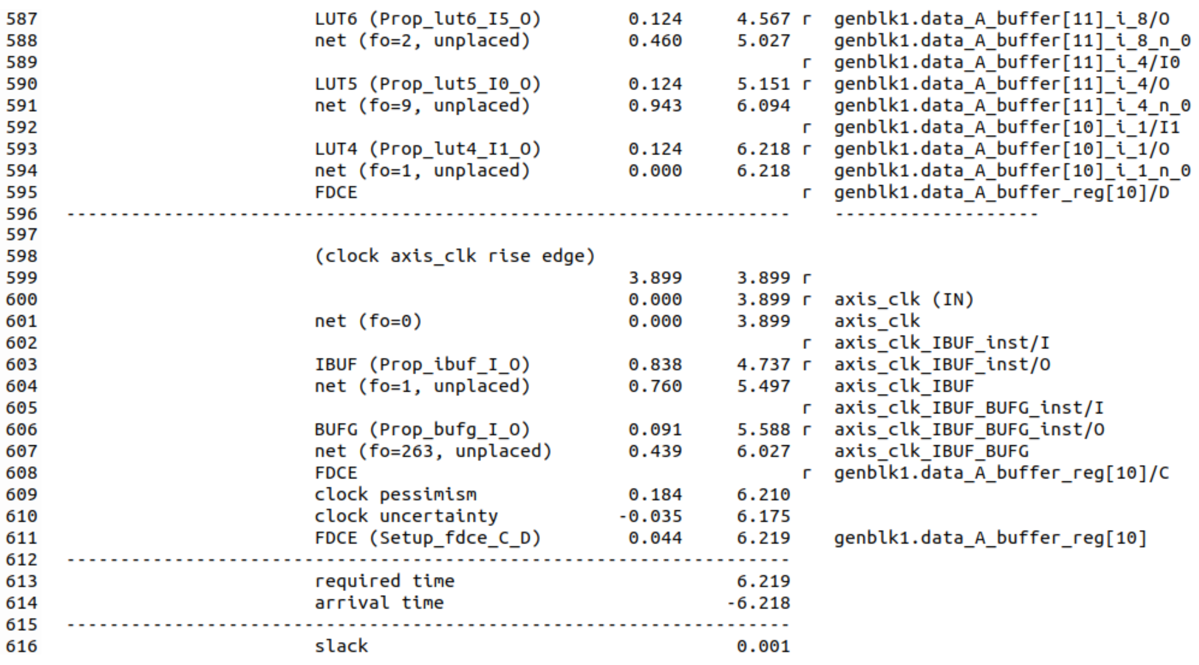
Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	0.001 ns	Worst Hold Slack (WHS):	0.067 ns	Worst Pulse Width Slack (WPWS):	1.449 ns
Total Negative Slack (TNS):	0.000 ns	Total Hold Slack (THS):	0.000 ns	Total Pulse Width Negative Slack (TPWS):	0.000 ns
Number of Failing Endpoints:	0	Number of Failing Endpoints:	0	Number of Failing Endpoints:	0
Total Number of Endpoints:	522	Total Number of Endpoints:	522	Total Number of Endpoints:	264

All user specified timing constraints are met.

- **Report timing on longest path, slack**



```
Max Delay Paths
--------------------------------------------------------------------------------------
Slack (MET) :              0.001ns  (required time - arrival time)
  Source:                  genblk1.data_A_buffer_reg[2]/C
                             (rising edge-triggered cell FDCE clocked by axis_clk  {rise@0.000ns fall@1.949ns period=3.899ns})
  Destination:             genblk1.data_A_buffer_reg[10]/D
                             (rising edge-triggered cell FDCE clocked by axis_clk  {rise@0.000ns fall@1.949ns period=3.899ns})
  Path Group:              axis_clk
  Path Type:               Setup (Max at Slow Process Corner)
  Requirement:             3.899ns  (axis_clk rise@3.899ns - axis_clk rise@0.000ns)
  Data Path Delay:         3.762ns  (logic 1.145ns (30.436%)  route 2.617ns (69.564%))
  Logic Levels:            4  (LUT4=2 LUT5=1 LUT6=1)
  Clock Path Skew:         -0.145ns (DCD - SCD + CPR)
    Destination Clock Delay (DCD):    2.128ns = ( 6.027 - 3.899 )
    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=263, unplaced)       0.584     2.456    axis_clk_IBUF_BUFG
                         FDCE                                         r  genblk1.data_A_buffer_reg[2]/C
  -------------------------------------------------------------------    -------------------
                         FDCE (Prop_fdce_C_Q)         0.478     2.934 r  genblk1.data_A_buffer_reg[2]/Q
                         net (fo=4, unplaced)         0.765     3.699    data_A_OBUF[2]
                                                                      r  genblk1.data_A_buffer[11]_i_10/I1
                         LUT4 (Prop_lut4_I1_O)        0.295     3.994 r  genblk1.data_A_buffer[11]_i_10/O
                         net (fo=1, unplaced)         0.449     4.443    genblk1.data_A_buffer[11]_i_10_n_0
                                                                      r  genblk1.data_A_buffer[11]_i_8/I5
                         LUT6 (Prop_lut6_I5_O)        0.124     4.567 r  genblk1.data_A_buffer[11]_i_8/O
```



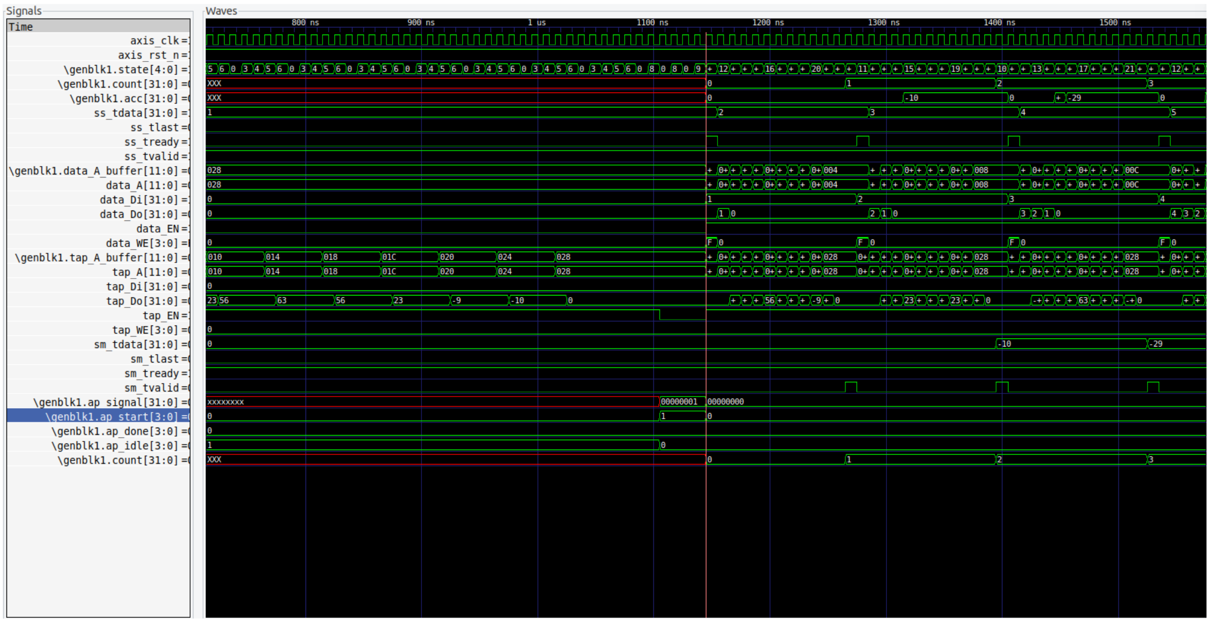
```
                         LUT6 (Prop_lut6_I5_O)        0.124     4.567 r  genblk1.data_A_buffer[11]_i_8/O
                         net (fo=2, unplaced)         0.460     5.027    genblk1.data_A_buffer[11]_i_8_n_0
                                                                      r  genblk1.data_A_buffer[11]_i_4/I0
                         LUT5 (Prop_lut5_I0_O)        0.124     5.151 r  genblk1.data_A_buffer[11]_i_4/O
                         net (fo=9, unplaced)         0.943     6.094    genblk1.data_A_buffer[11]_i_4_n_0
                                                                      r  genblk1.data_A_buffer[10]_i_1/I1
                         LUT4 (Prop_lut4_I1_O)        0.124     6.218 r  genblk1.data_A_buffer[10]_i_1/O
                         net (fo=1, unplaced)         0.000     6.218    genblk1.data_A_buffer[10]_i_1_n_0
                         FDCE                                         r  genblk1.data_A_buffer_reg[10]/D
  -------------------------------------------------------------------    -------------------

                         (clock axis_clk rise edge)
                                                      3.899     3.899 r
                                                      0.000     3.899 r  axis_clk (IN)
                         net (fo=0)                   0.000     3.899    axis_clk
                                                                      r  axis_clk_IBUF_inst/I
                         IBUF (Prop_ibuf_I_O)         0.838     4.737 r  axis_clk_IBUF_inst/O
                         net (fo=1, unplaced)         0.760     5.497    axis_clk_IBUF
                                                                      r  axis_clk_IBUF_BUFG_inst/I
                         BUFG (Prop_bufg_I_O)         0.091     5.588 r  axis_clk_IBUF_BUFG_inst/O
                         net (fo=263, unplaced)       0.439     6.027    axis_clk_IBUF_BUFG
                         FDCE                                         r  genblk1.data_A_buffer_reg[10]/C
                         clock pessimism              0.184     6.210
                         clock uncertainty           -0.035     6.175
                         FDCE (Setup_fdce_C_D)        0.044     6.219    genblk1.data_A_buffer_reg[10]
  -------------------------------------------------------------------
                         required time                          6.219
                         arrival time                          -6.218
  -------------------------------------------------------------------
                         slack                                  0.001
```

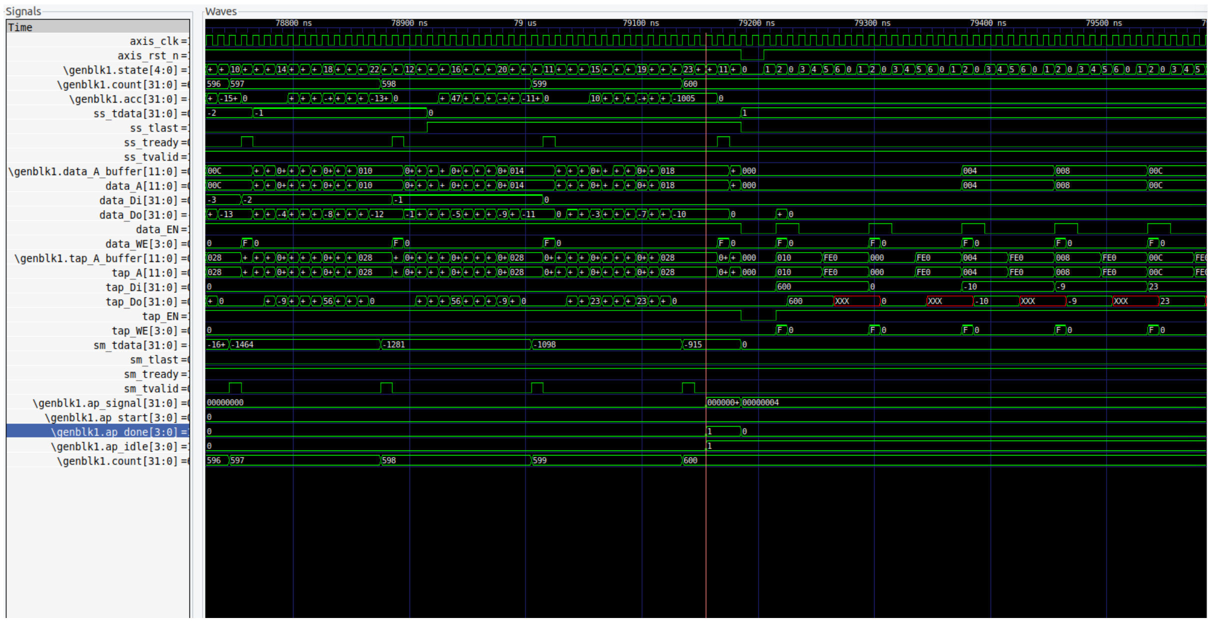
Simulation Waveform

- **ap signal**

  ap_start at 1145 ns and ap_done at 79155 ns

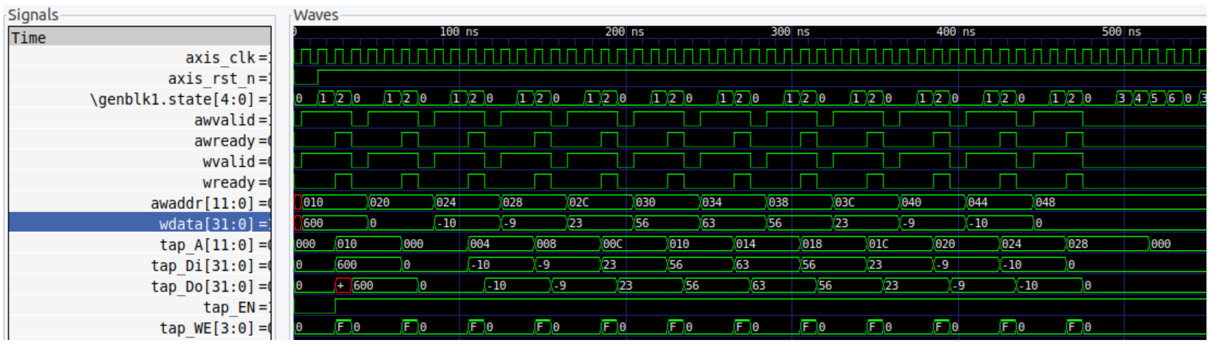
  It takes 78010 ns, that is 7801 clock cycles with cycles time 10 ns.



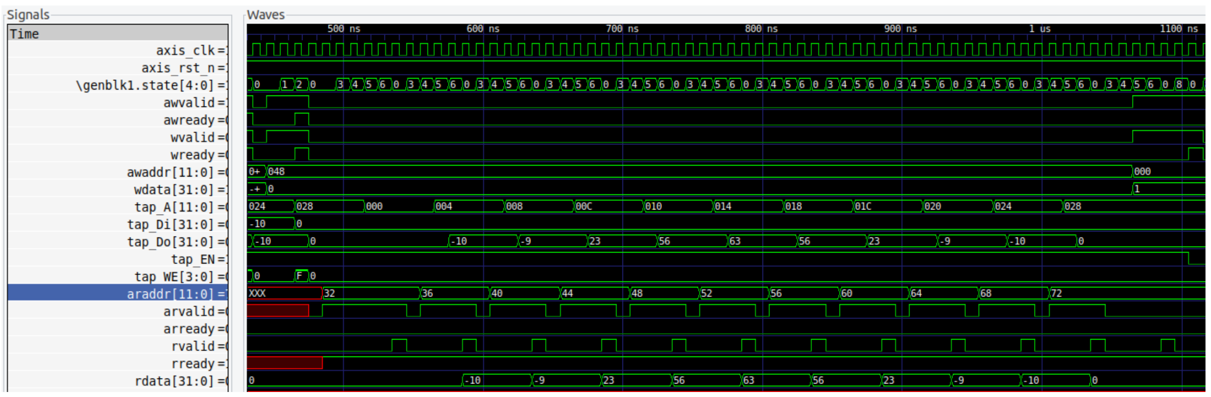


- **coefficient program, and read back(axi-Lite)**

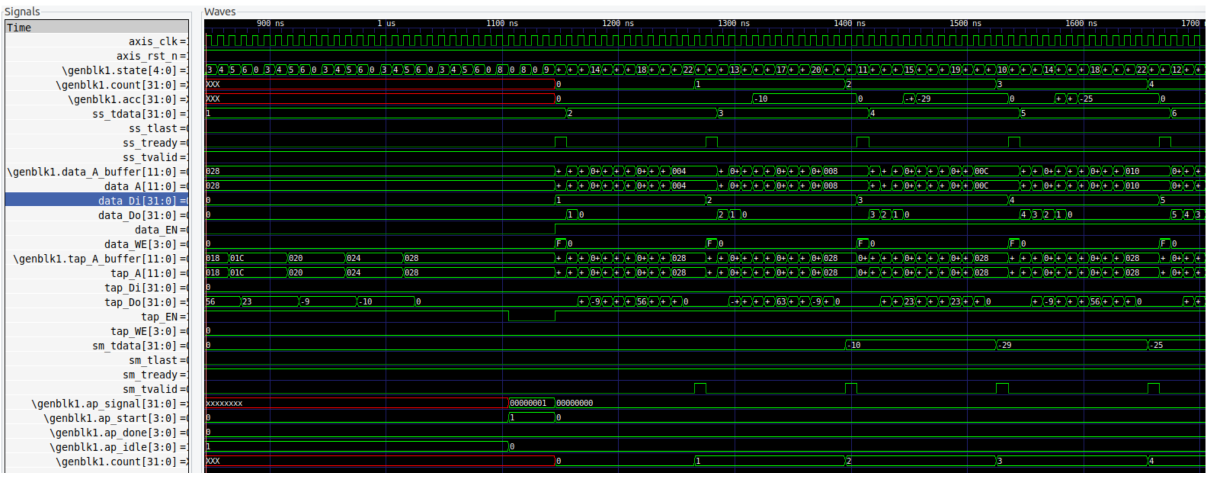
receive coefficient from axi-write and write into tapRAM

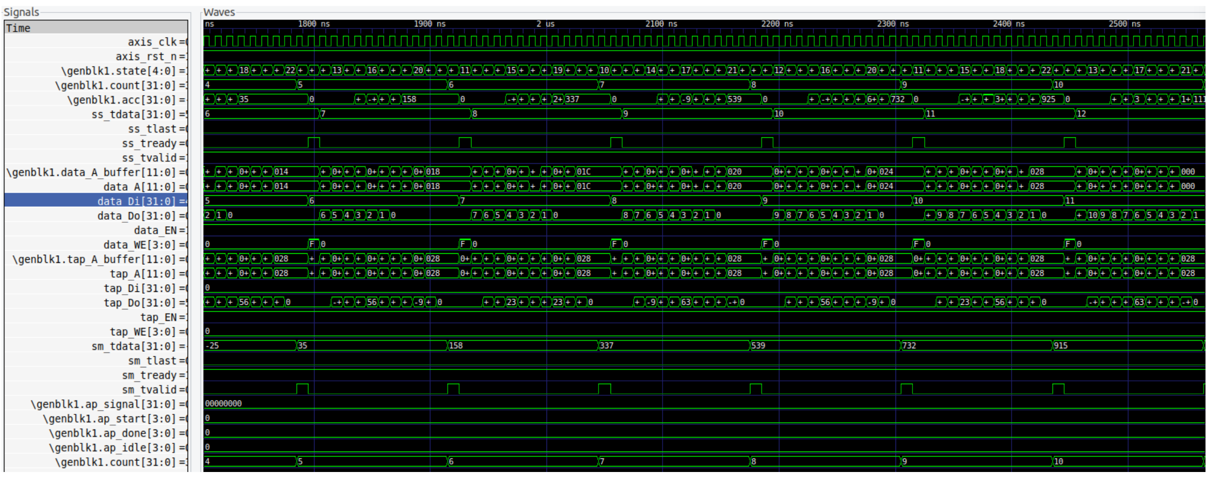


Read back

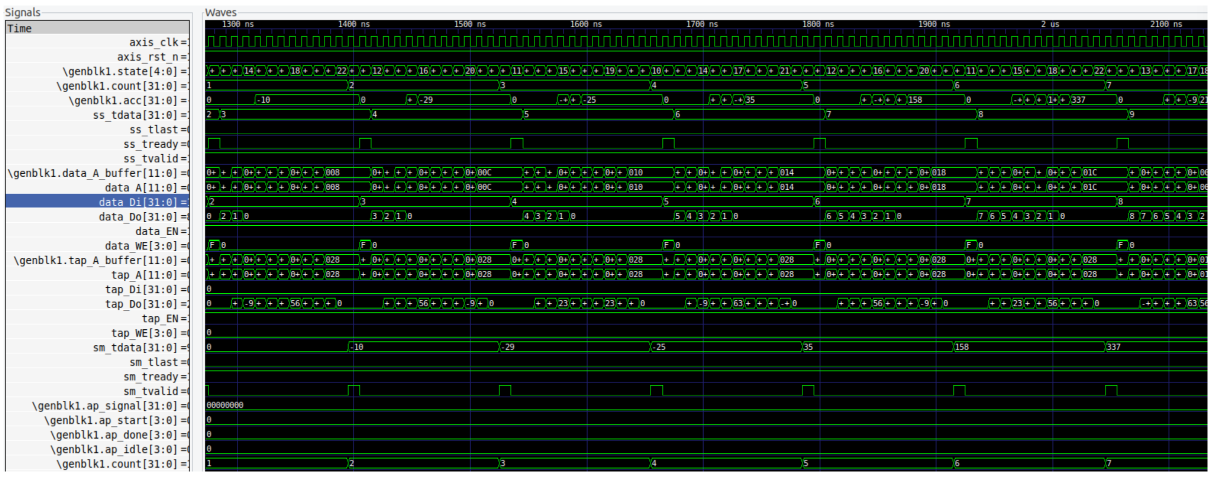


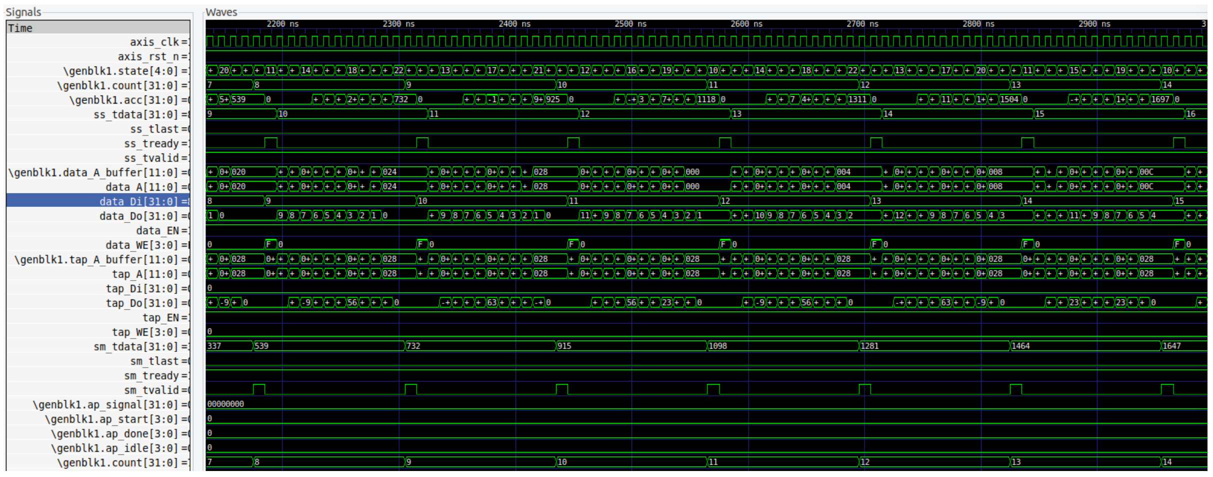
- **Data-in stream-in**





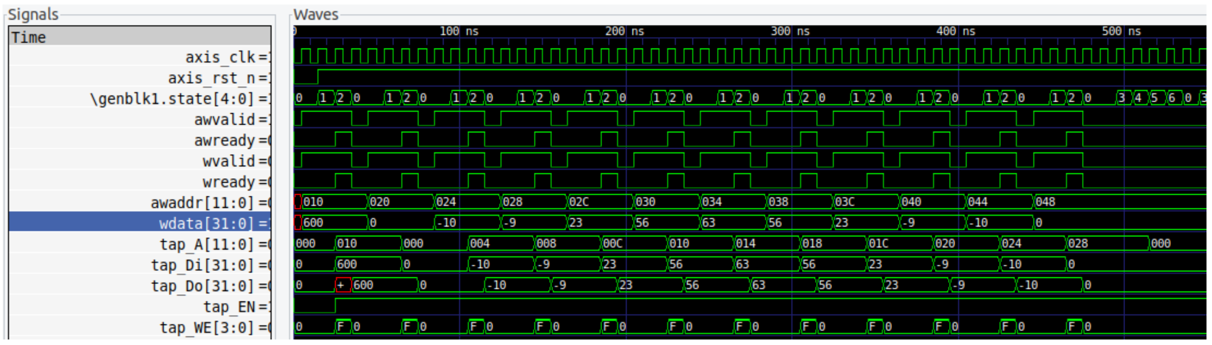
- **Data-out stream-out**



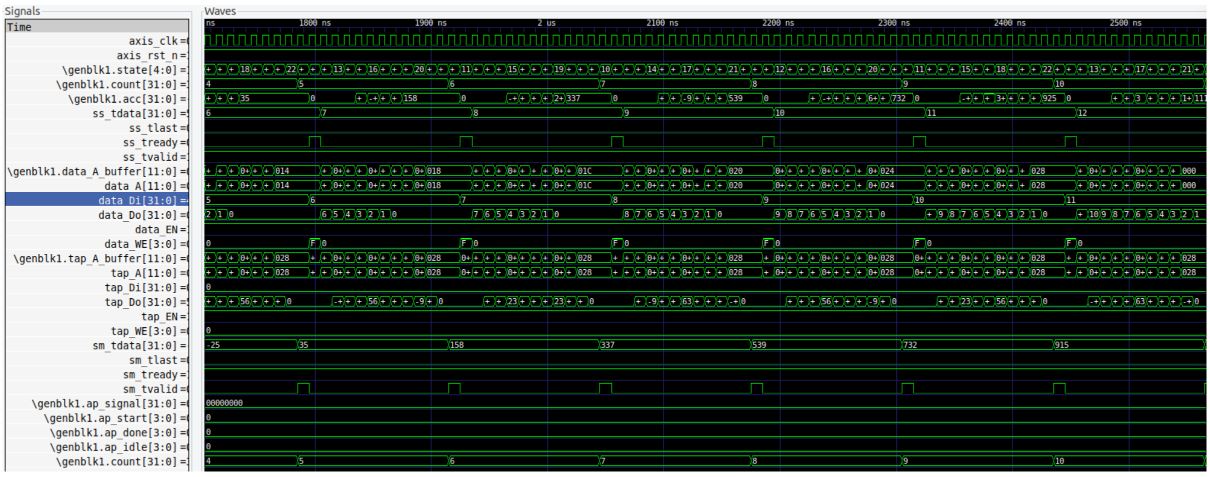


- **RAM access control**

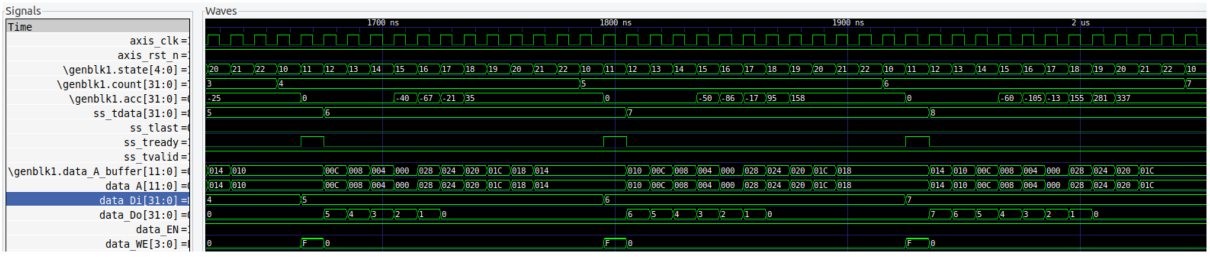
Tap RAM



Data RAM


Signals
Waves
Time
axis_clk
axis_rst_n
\genblk1.state[4:0]
\genblk1.count[31:0]
\genblk1.acc[31:0]
ss_tdata[31:0]
ss_tlast
ss_tready
ss_tvalid
\genblk1.data_A_buffer[11:0]
data_A[11:0]
data_Di[31:0]
data_Do[31:0]
data_EN
data_WE[3:0]
\genblk1.tap_A_buffer[11:0]
tap_A[11:0]
tap_Di[31:0]
tap_Do[31:0]
tap_EN
tap_WE[3:0]
sm_tdata[31:0]
sm_tlast
sm_tready
sm_tvalid
\genblk1.ap_signal[31:0]
\genblk1.ap_start[3:0]
\genblk1.ap_done[3:0]
\genblk1.ap_idle[3:0]
\genblk1.count[31:0]

- **FSM**


Signals
Waves
Time
axis_clk
axis_rst_n
\genblk1.state[4:0]
\genblk1.count[31:0]
\genblk1.acc[31:0]
ss_tdata[31:0]
ss_tlast
ss_tready
ss_tvalid
\genblk1.data_A_buffer[11:0]
data_A[11:0]
data_Di[31:0]
data_Do[31:0]
data_EN
data_WE[3:0]