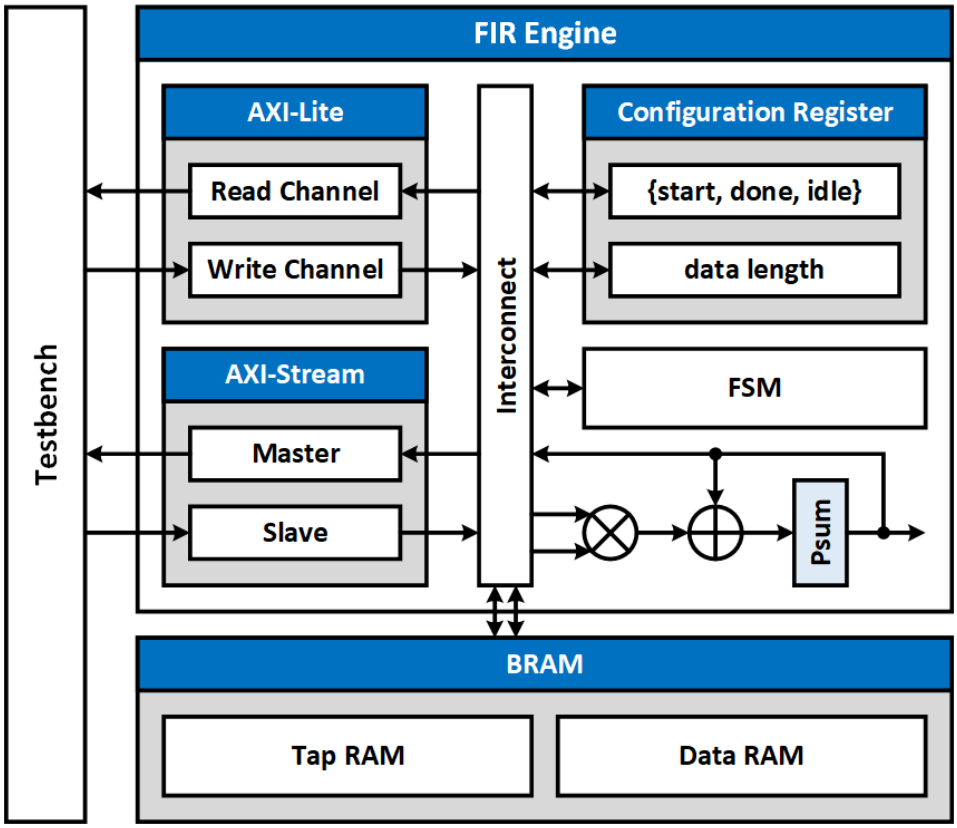
SoC Design Lab 3 Report

R11943006 張力元

1. Block diagram



Configuration register address map:

Address	Bit	Description
0x00	0	ap_start: set to 1 to start the FIR engine
	1	ap_done: assert when FIR engine processes and transfers all the data
	2	ap_idle: indicate whether FIR engine is actively processing data
0x10 – 0x14	31:0	data length
0x20 – 0xFF	31:0	tap coefficients

2. System description

(1) AXI-Lite control

Read/write configuration

- ✓ Write channel is enabled only when FIR engine is at the idle state
- ✓ Finite state machine for reading configuration:
  - ■ S_AXI_AR: receive read address request
  - ■ S_AXI_R: read data from configuration registers or tap RAM
  - ■ S_AXI_WAIT: wait for `rready` asserted by the testbench (handshake)
- ✓ `ap_start` is set by the testbench and reset by the FIR engine
- ✓ When receiving both read and write requests simultaneously, prioritize the read channel to

prevent resource competition (e.g., configuration registers and tap RAM)

Receive tap parameters and place into SRAM

✓ If `awaddr ≥ 0x20`, then write `wdata` to the corresponding address of the tap RAM

(2) AXI-Stream control

Receive data and place into SRAM

```
S_LOAD: begin
    if (ss_tvalid && (~ss_tready_r)) begin
        ss_tready_w = 1;
        data_WE     = 4'hf;
        data_Di     = ss_tdata;
        data_A      = data_last_id_r * 4;  // write 4-byte data
        state_w     = S_PROC;
```

Write results to testbench

```
S_OUTPUT: begin
    if (sm_tready && ((~last_flag_r) || (axil_rstate_r == S_AXI_AR))) begin
        sm_tvalid_w = 1;
        sm_tdata_w  = psum_r;
        sm_tlast_w  = last_flag_r;
```

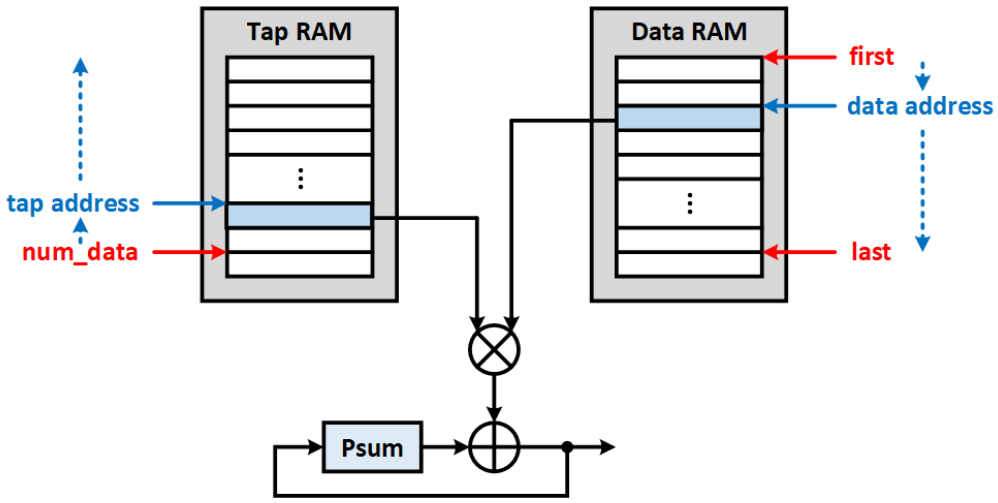
(3) How to access data RAM / tap RAM to do computation and how ap_done is generated

✓ Part of the finite state machine:

- S_PROC: utilize one multiplier and one adder to accumulate the partial sum
- S_OUTPUT: transfer the computation result to the testbench
- S_DONE: assert `ap_done` when the FIR engine finishes processing the last data and transfers the last result to the testbench
- S_DONE_WAIT: reset `ap_done` and transition to the idle state after `ap_done` (i.e. address 0x00) is read by the testbench

✓ Designing counters and address pointers for tap RAM and data RAM access

$$\mathbf{y[n]} = \sum_{i=0}^{10} \boldsymbol{h[i]} \times \boldsymbol{x[n-i]}$$



R11943006 張力元

3. Resource usage

Utilization report including FF, LUT and BRAM:

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

+-------------------------+------+-------+------------+-----------+-------+
|        Site Type        | Used | Fixed | Prohibited | Available | Util% |
+-------------------------+------+-------+------------+-----------+-------+
| Slice LUTs*             |  279 |     0 |          0 |     53200 |  0.52 |
|   LUT as Logic          |  279 |     0 |          0 |     53200 |  0.52 |
|   LUT as Memory         |    0 |     0 |          0 |     17400 |  0.00 |
| Slice Registers         |  222 |     0 |          0 |    106400 |  0.21 |
|   Register as Flip Flop |  222 |     0 |          0 |    106400 |  0.21 |
|   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 |
+-------------------------+------+-------+------------+-----------+-------+
* Warning! The Final LUT count, after physical optimizations and full implementation, is typically lower. Run opt_design after
synthesis, if not already completed, for a more realistic count.
```

```
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 |
+----------------+------+-------+------------+-----------+-------+
* Note: Each Block RAM Tile only has one FIFO logic available and therefore can accommodate only one FIFO36E1 or one FIFO18E1.
However, if a FIFO18E1 occupies a Block RAM Tile, that tile can still accommodate a RAMB18E1
```

4. Timing report

Clock constraint for synthesis:

```
## Clock Signal
set cycle 10.0;
create_clock -name axis_clk -period $cycle [get_ports axis_clk]
```

Setup time and hold time slack:

Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	4.912 ns	Worst Hold Slack (WHS):	0.144 ns	Worst Pulse Width Slack (WPWS):	4.500 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:	344	Total Number of Endpoints:	344	Total Number of Endpoints:	223

All user specified timing constraints are met.

Critical path:

Summary

Name	Path 1
Slack	4.912ns
Source	FSM_sequential_state_r_reg[0]/C (rising edge-triggered cell FDCE clocked by sys_clk {rise@0.000ns fall@5.000ns period=10.000ns})
Destination	data_len_r_reg[0]/CE (rising edge-triggered cell FDCE clocked by sys_clk {rise@0.000ns fall@5.000ns period=10.000ns})
Path Group	sys_clk
Path Type	Setup (Max at Slow Process Corner)
Requirement	10.000ns (sys_clk rise@10.000ns - sys_clk rise@0.000ns)
Data Path Delay	4.706ns (logic 1.145ns (24.331%) route 3.561ns (75.669%))
Logic Levels	4 (LUT4=1 LUT5=1 LUT6=2)
Clock Path Skew	-0.145ns
Clock Un...rtainty	0.035ns

```
Max Delay Paths
--------------------------------------------------------------------------------------
Slack (MET) :             4.912ns  (required time - arrival time)
  Source:                 FSM_sequential_state_r_reg[0]/C
                            (rising edge-triggered cell FDCE clocked by axis_clk  {rise@0.000ns fall@5.000ns period=10.000ns})
  Destination:            data_len_r_reg[0]/CE
                            (rising edge-triggered cell FDCE clocked by axis_clk  {rise@0.000ns fall@5.000ns period=10.000ns})
  Path Group:             axis_clk
  Path Type:              Setup (Max at Slow Process Corner)
  Requirement:            10.000ns  (axis_clk rise@10.000ns - axis_clk rise@0.000ns)
  Data Path Delay:        4.706ns  (logic 1.145ns (24.331%)  route 3.561ns (75.669%))
  Logic Levels:           4  (LUT4=1 LUT5=1 LUT6=2)
  Clock Path Skew:        -0.145ns (DCD - SCD + CPR)
    Destination Clock Delay (DCD):    2.128ns = ( 12.128 - 10.000 )
    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=222, unplaced)         0.584     2.456    axis_clk_IBUF_BUFG
                     FDCE                                           r  FSM_sequential_state_r_reg[0]/C
-------------------------------------------------------------------    -------------------
                     FDCE (Prop_fdce_C_Q)           0.478     2.934 r  FSM_sequential_state_r_reg[0]/Q
                     net (fo=124, unplaced)         0.911     3.845    state_r[0]
                                                                   r  arready_r_i_3/I0
                     LUT5 (Prop_lut5_I0_O)          0.295     4.140 r  arready_r_i_3/O
                     net (fo=2, unplaced)           1.122     5.262    arready_r_i_3_n_0
                                                                   r  arready_r_i_1/I1
                     LUT6 (Prop_lut6_I1_O)          0.124     5.386 f  arready_r_i_1/O
                     net (fo=16, unplaced)          0.503     5.889    arready_r_i_1_n_0
                                                                   f  ap_ctrl_r[31]_i_2/I0
                     LUT4 (Prop_lut4_I0_O)          0.124     6.013 f  ap_ctrl_r[31]_i_2/O
                     net (fo=6, unplaced)           0.481     6.494    ap_ctrl_r[31]_i_2_n_0
                                                                   f  data_len_r[31]_i_1/I0
                     LUT6 (Prop_lut6_I0_O)          0.124     6.618 r  data_len_r[31]_i_1/O
                     net (fo=32, unplaced)          0.544     7.162    data_len_r[31]_i_1_n_0
                     FDCE                                           r  data_len_r_reg[0]/CE
-------------------------------------------------------------------    -------------------

                     (clock axis_clk rise edge)
                                                   10.000    10.000 r
                                                    0.000    10.000 r  axis_clk (IN)
                     net (fo=0)                     0.000    10.000    axis_clk
                                                                   r  axis_clk_IBUF_inst/I
                     IBUF (Prop_ibuf_I_O)           0.838    10.838 r  axis_clk_IBUF_inst/O
                     net (fo=1, unplaced)           0.760    11.598    axis_clk_IBUF
                                                                   r  axis_clk_IBUF_BUFG_inst/I
                     BUFG (Prop_bufg_I_O)           0.091    11.689 r  axis_clk_IBUF_BUFG_inst/O
                     net (fo=222, unplaced)         0.439    12.128    axis_clk_IBUF_BUFG
                     FDCE                                           r  data_len_r_reg[0]/C
                     clock pessimism                0.184    12.311
                     clock uncertainty             -0.035    12.276
                     FDCE (Setup_fdce_C_CE)        -0.202    12.074    data_len_r_reg[0]
 -------------------------------------------------------------------
                     required time                          12.074
                     arrival time                           -7.162
 -------------------------------------------------------------------
                     slack                                   4.912
```

R11943006 張力元

5. Simulation waveform

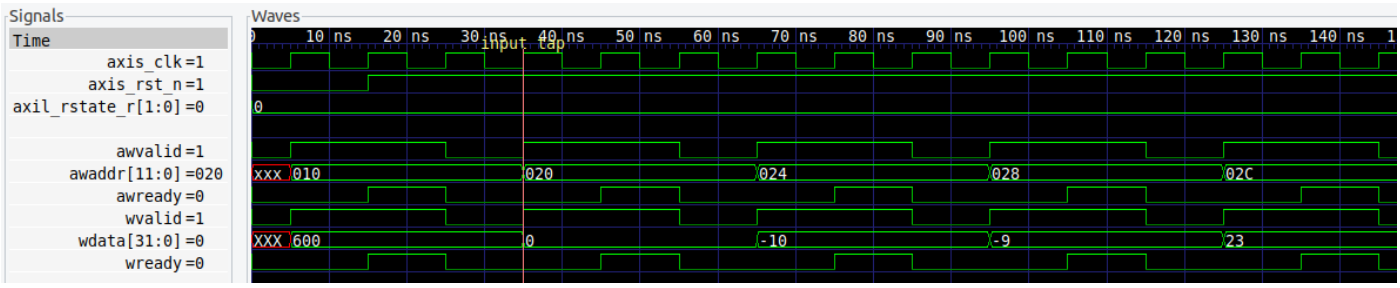
(1) Makefile for simulation

```
fir:
	rm -rf xsim.dir/ *.log *.pb *.jou *.wdb
	xvlog -f ./include.rtl.list.xsim ./tb/fir_tb.v
	xelab -top fir_tb -snapshot fir_tb_elab -relax
	xsim fir_tb_elab -R
```

(2) Coefficient program and read back

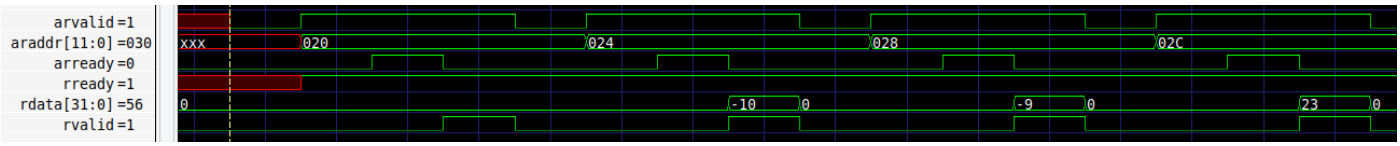
Write data length and tap coefficients to the FIR engine:

Activate `awready` to receive the request, write data to the designated location, and then trigger `wready`.



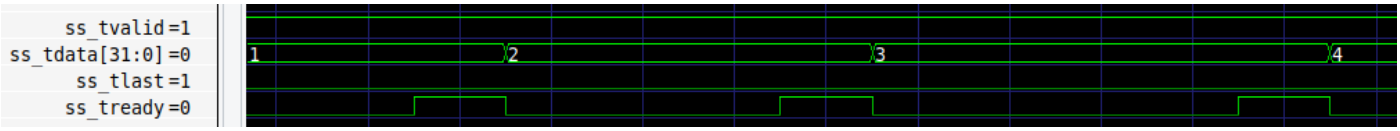
Read tap coefficients from FIR engine:

Assert `arready` to receive the read request, and output valid data in the subsequent cycle.



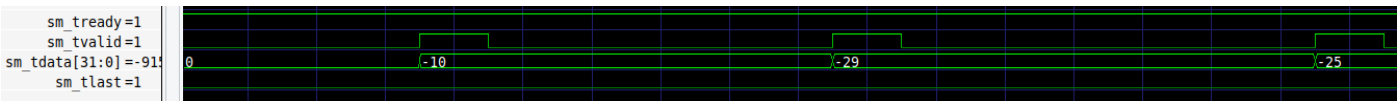
(3) Data stream in

Raise `ss_tready` to read the input data and store it in the data RAM.

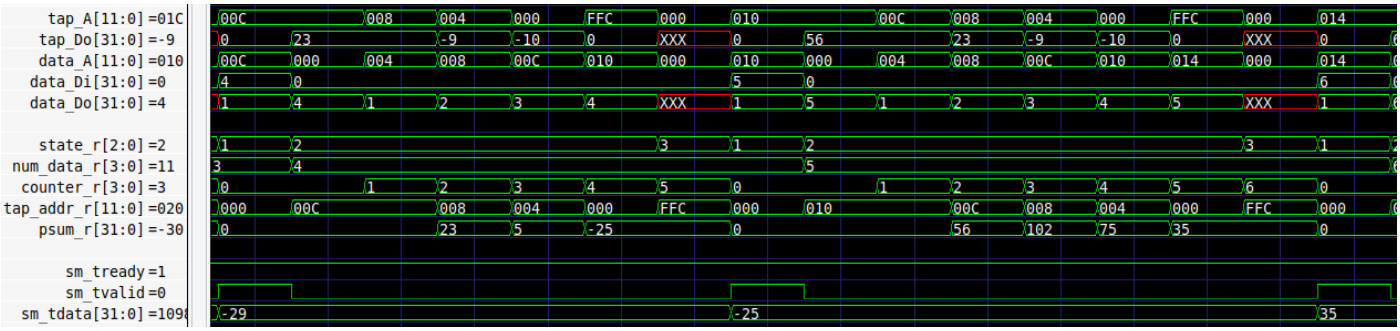


(4) Data stream out

When `sm_tready` is active, assert `sm_tvalid` and output the computation result.



(5) Bram access control and FSM



R11943006 張力元

6. Problems and solutions

(1) A compile error arises during the execution of the Makefile: “Module fir doesn't have a timescale but at least one module in design has a timescale.”

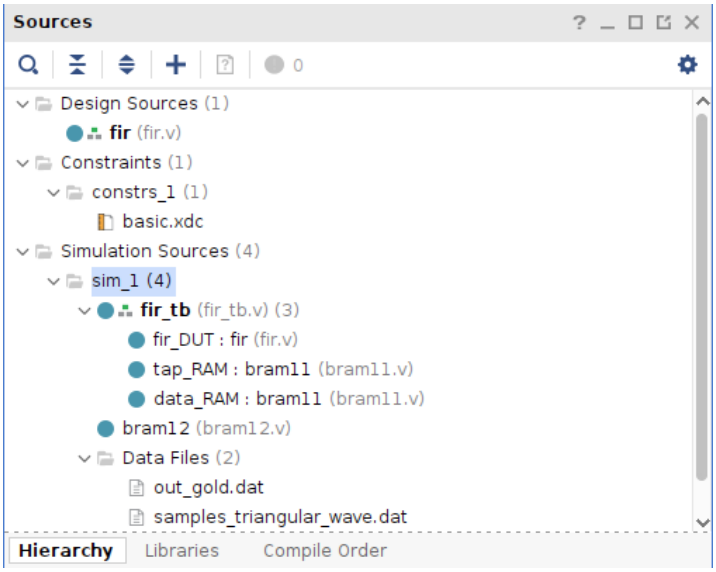
The error message is not received when launching XSIM through the Vivado GUI. This is because the “-relax” switch is used during elaboration by default, which reduces this error message to a warning. An alternative workaround would be to add “-relax” to the xelab command in the Makefile.

[Ref] https://support.xilinx.com/s/article/64388?language=en_US

(2) In the testbench, set the read/write address for tap coefficients to (0x20 + 4*k) due to each coefficient having a 4-byte data width

```
initial begin
    error_coef = 0;
    $display("----Start the coefficient input(AXI-lite)----");
    config_write(12'h10, data_length);
    for (k = 0; k < Tape_Num; k = k + 1) begin
        config_write(12'h20 + 4*k, coef[k]);
    end
    awvalid <= 0; wvalid <= 0;
    // read-back and check
    $display(" Check Coefficient ...");
    for (k = 0; k < Tape_Num; k = k + 1) begin
        config_read_check(12'h20 + 4*k, coef[k], 32'hffffffff);
    end
    arvalid <= 0;
    $display(" Tape programming done ...");
    $display(" Start FIR");
    @(posedge axis_clk) config_write(12'h00, 32'h0000_0001);    // ap_start = 1
    $display("----End the coefficient input(AXI-lite)----");
end
```

(3) Vivado sources



(4) Modify the font size in the waveform of GTKWave

Add `fontname_waves Monospace 10` to the file `~/.gtkwaverc`

7. GitHub link

https://github.com/liyuan-chang/SoC-Lab-FIR