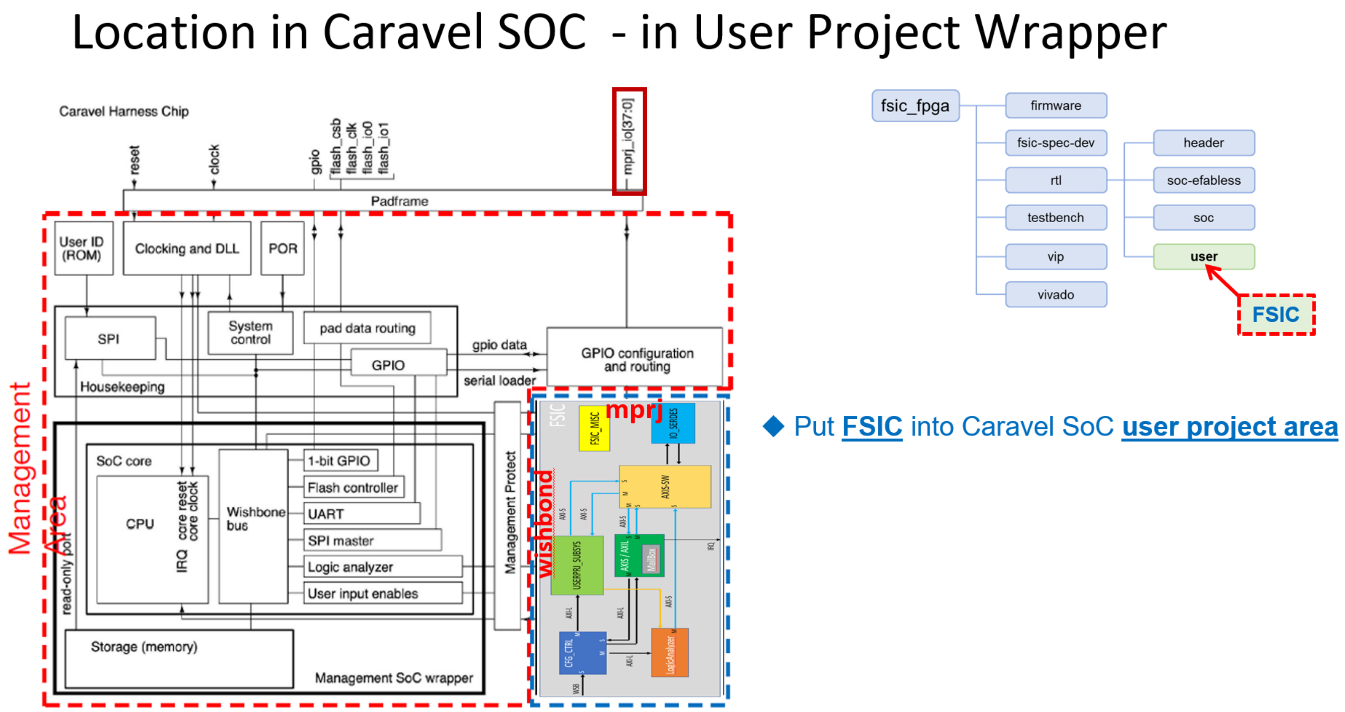
Lab01-fsic-sim report

作者:110020015 劉祐瑋 (GitHub link: https://github.com/nthuyouwei/asoclab)

一、 實驗簡介

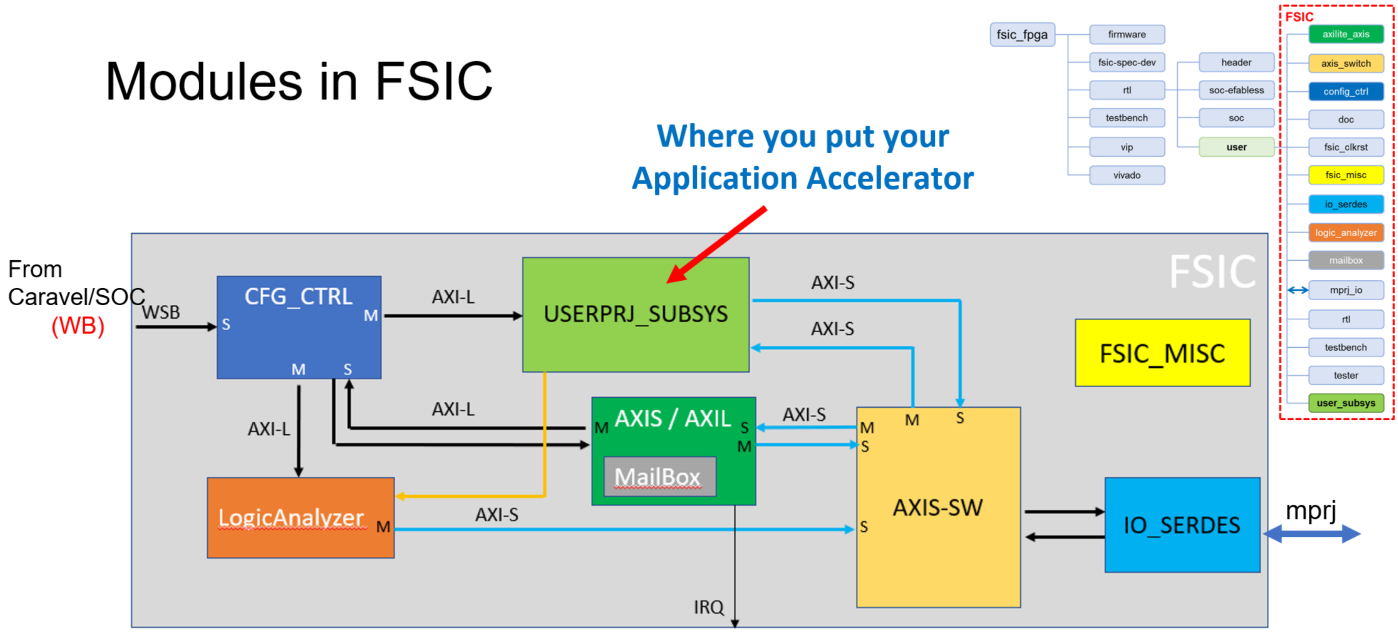
目的:Integrate FIR into FSIC environment.

架構介紹如下圖:


Location in Caravel SOC - in User Project Wrapper

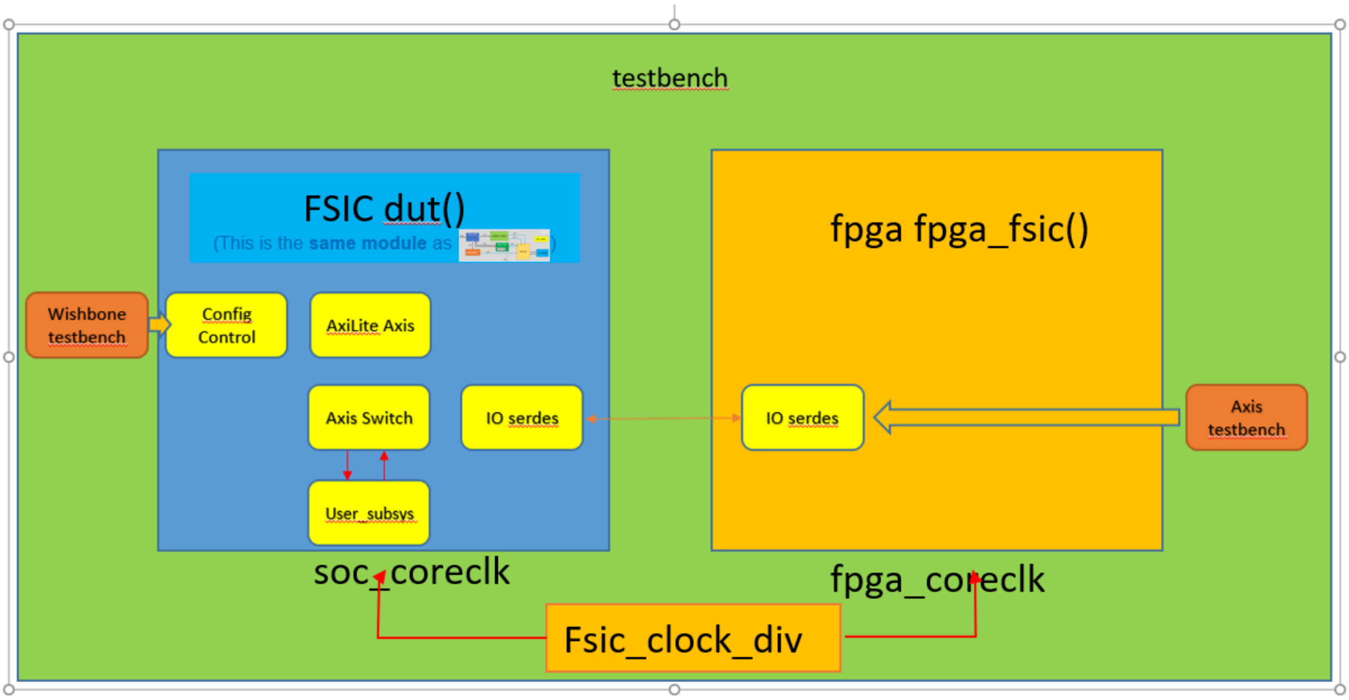
◆ Put **FSIC** into Caravel SoC **user project area**

(圖 1)


Modules in FSIC

Where you put your Application Accelerator

(圖 2)



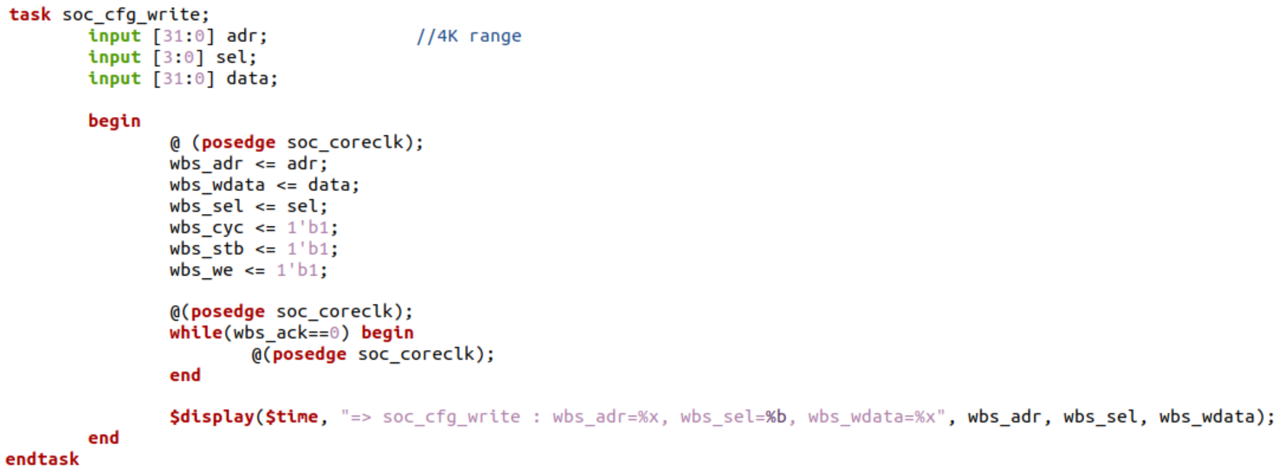
(圖 3)

二、 問題和解答

1. show the code that you use to program configuration address [‘h3000_5000].

Ans:

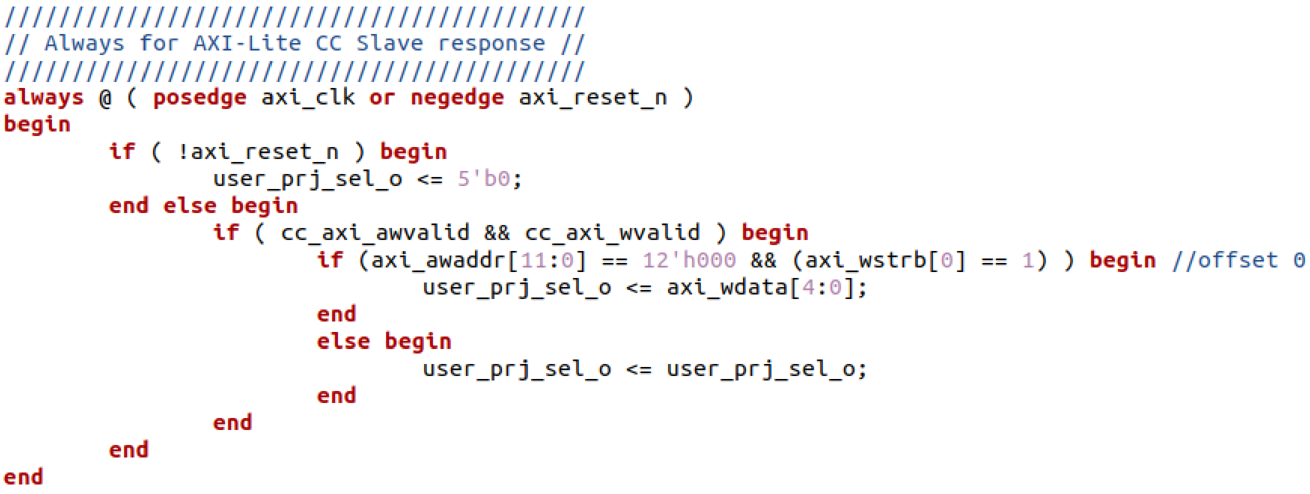
```
soc_cfg_write(32'h3000_5000, 4'b0001, 1);//start uspj1
```



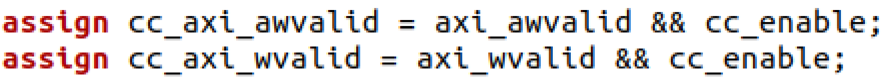
```
task soc_cfg_write;
        input [31:0] adr;               //4K range
        input [3:0] sel;
        input [31:0] data;

        begin
                @ (posedge soc_coreclk);
                wbs_adr <= adr;
                wbs_wdata <= data;
                wbs_sel <= sel;
                wbs_cyc <= 1'b1;
                wbs_stb <= 1'b1;
                wbs_we <= 1'b1;

                @(posedge soc_coreclk);
                while(wbs_ack==0) begin
                        @(posedge soc_coreclk);
                end

                $display($time, "=> soc_cfg_write : wbs_adr=%x, wbs_sel=%b, wbs_wdata=%x", wbs_adr, wbs_sel, wbs_wdata);
        end
endtask
```

2. Explain why “by programming configuration address [‘h3000_5000], signal user_prj_sel[4:0] will change accordingly”?

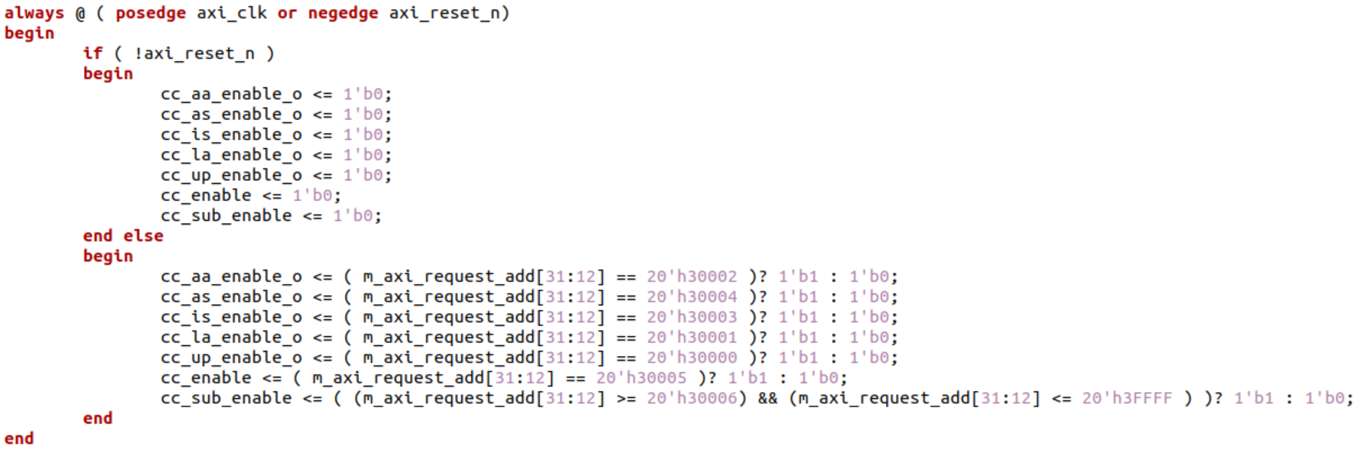
Ans:



```
///////////////////////////////////////////
// Always for AXI-Lite CC Slave response //
///////////////////////////////////////////
always @ ( posedge axi_clk or negedge axi_reset_n )
begin
        if ( !axi_reset_n ) begin
                user_prj_sel_o <= 5'b0;
        end else begin
                if ( cc_axi_awvalid && cc_axi_wvalid ) begin
                        if (axi_awaddr[11:0] == 12'h000 && (axi_wstrb[0] == 1) ) begin //offset 0
                                user_prj_sel_o <= axi_wdata[4:0];
                        end
                        else begin
                                user_prj_sel_o <= user_prj_sel_o;
                        end
                end
        end
end
```



```
assign cc_axi_awvalid = axi_awvalid && cc_enable;
assign cc_axi_wvalid = axi_wvalid && cc_enable;
```

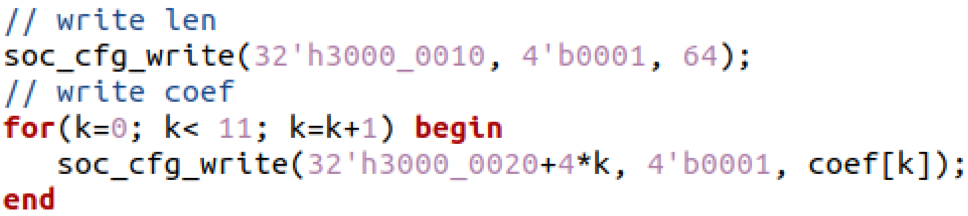


```
always @ ( posedge axi_clk or negedge axi_reset_n)
begin
        if ( !axi_reset_n )
        begin
                cc_aa_enable_o <= 1'b0;
                cc_as_enable_o <= 1'b0;
                cc_is_enable_o <= 1'b0;
                cc_la_enable_o <= 1'b0;
                cc_up_enable_o <= 1'b0;
                cc_enable <= 1'b0;
                cc_sub_enable <= 1'b0;
        end else
        begin
                cc_aa_enable_o <= ( m_axi_request_add[31:12] == 20'h30002 )? 1'b1 : 1'b0;
                cc_as_enable_o <= ( m_axi_request_add[31:12] == 20'h30004 )? 1'b1 : 1'b0;
                cc_is_enable_o <= ( m_axi_request_add[31:12] == 20'h30003 )? 1'b1 : 1'b0;
                cc_la_enable_o <= ( m_axi_request_add[31:12] == 20'h30001 )? 1'b1 : 1'b0;
                cc_up_enable_o <= ( m_axi_request_add[31:12] == 20'h30000 )? 1'b1 : 1'b0;
                cc_enable <= ( m_axi_request_add[31:12] == 20'h30005 )? 1'b1 : 1'b0;
                cc_sub_enable <= ( (m_axi_request_add[31:12] >= 20'h30006) && (m_axi_request_add[31:12] <= 20'h3FFFF ) )? 1'b1 : 1'b0;
        end
end
```

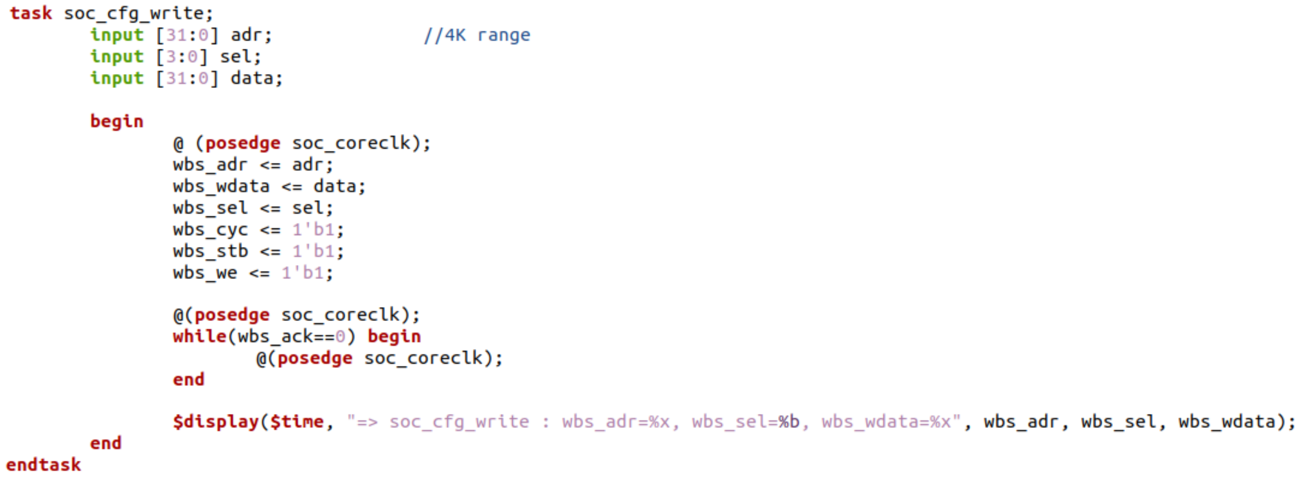
由上面三張圖可知當 addr 為 32’h3000_5000 時 ，因為 addr[31:12]==20’h30005 會先讓 cc_enable 變為 1，這時 cc_axi_awvalid 和 cc_axi_wvalid 也會拉為 1，最後由 addr[11:0]=12’h00 讓 user_prj_sel_o 改變而改變 user_prj_sel。

3. Briefly describe how you do FIR initialization (tap parameter, length) from SOC side (Test#1)?

Ans:

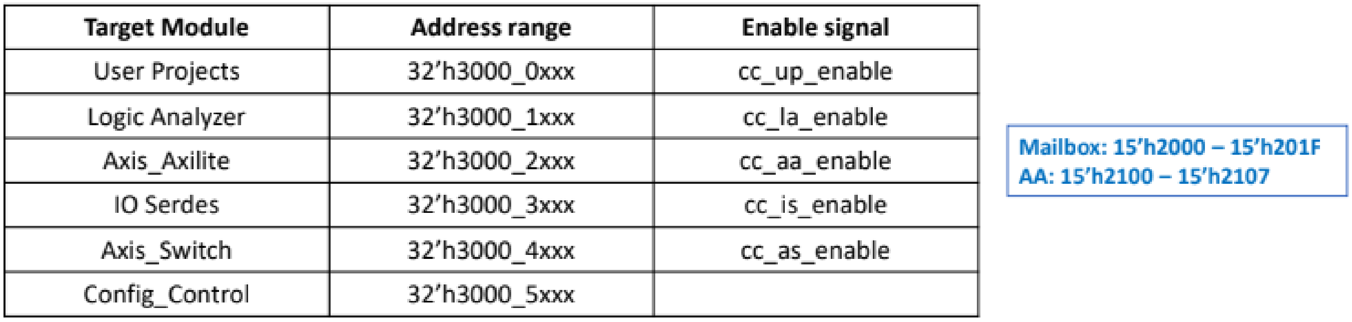


```
// write len
soc_cfg_write(32'h3000_0010, 4'b0001, 64);
// write coef
for(k=0; k< 11; k=k+1) begin
   soc_cfg_write(32'h3000_0020+4*k, 4'b0001, coef[k]);
end
```



```
task soc_cfg_write;
        input [31:0] adr;               //4K range
        input [3:0] sel;
        input [31:0] data;

        begin
                @ (posedge soc_coreclk);
                wbs_adr <= adr;
                wbs_wdata <= data;
                wbs_sel <= sel;
                wbs_cyc <= 1'b1;
                wbs_stb <= 1'b1;
                wbs_we <= 1'b1;

                @(posedge soc_coreclk);
                while(wbs_ack==0) begin
                        @(posedge soc_coreclk);
                end

                $display($time, "=> soc_cfg_write : wbs_adr=%x, wbs_sel=%b, wbs_wdata=%x", wbs_adr, wbs_sel, wbs_wdata);
        end
endtask
```

跟上學期 lab4_2 一樣直接寫到對應的位置就好，不過要注意 Address map 定義(如下圖)。



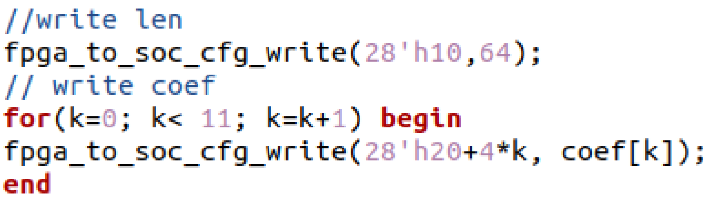
Target Module	Address range	Enable signal
User Projects	32’h3000_0xxx	cc_up_enable
Logic Analyzer	32’h3000_1xxx	cc_la_enable
Axis_Axilite	32’h3000_2xxx	cc_aa_enable
IO Serdes	32’h3000_3xxx	cc_is_enable
Axis_Switch	32’h3000_4xxx	cc_as_enable
Config_Control	32’h3000_5xxx	

Mailbox: 15’h2000 – 15’h201F
AA: 15’h2100 – 15’h2107

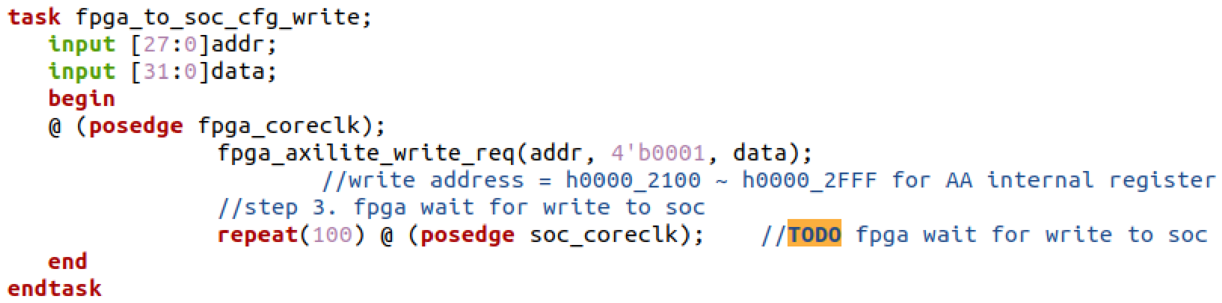
(圖 4)

4. Briefly describe how you do FIR initialization (tap parameter, length) from FPGA side (Test#2)?

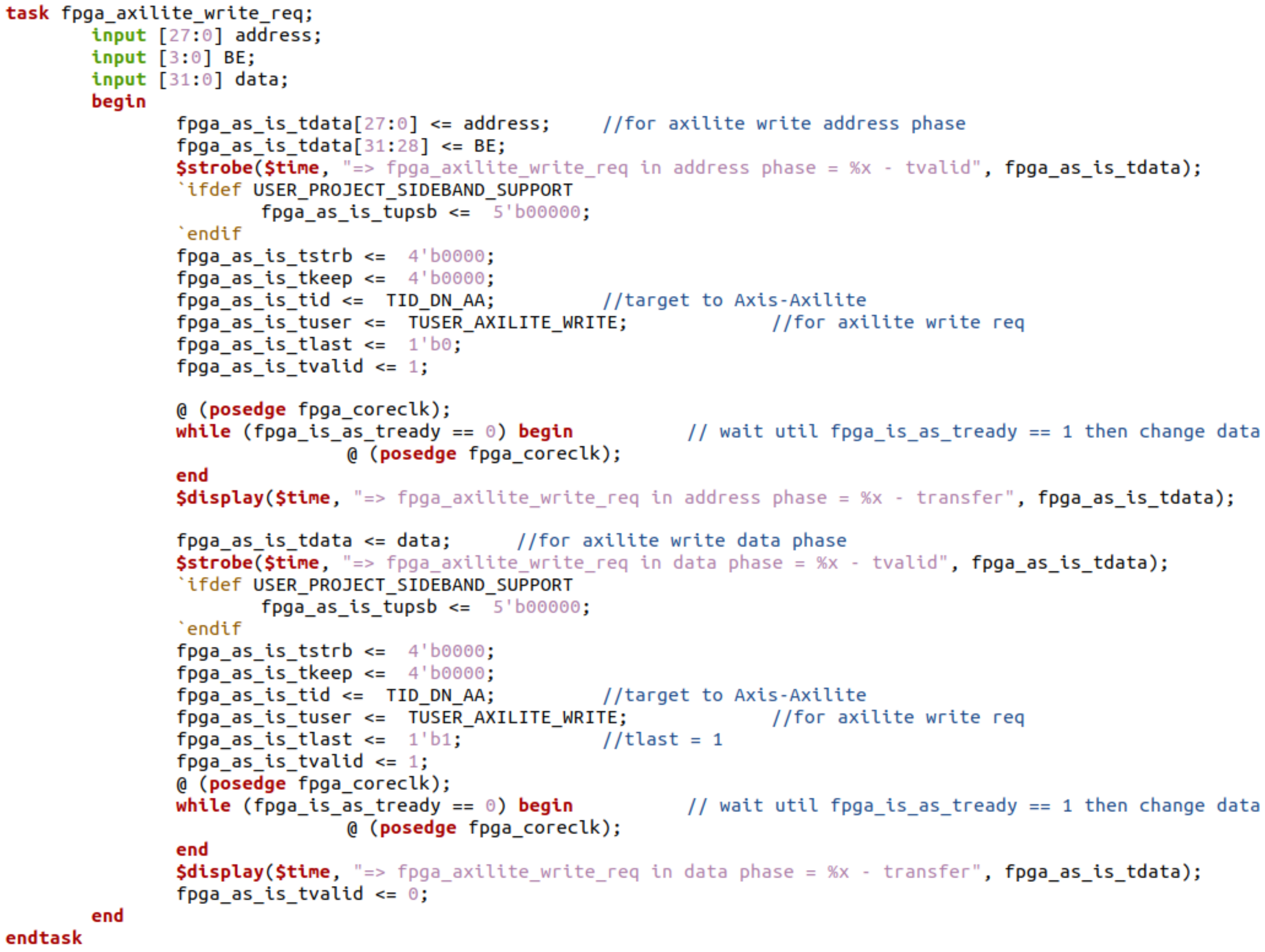
Ans:



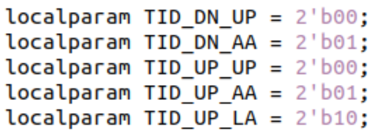
```
//write len
fpga_to_soc_cfg_write(28'h10,64);
// write coef
for(k=0; k< 11; k=k+1) begin
fpga_to_soc_cfg_write(28'h20+4*k, coef[k]);
end
```



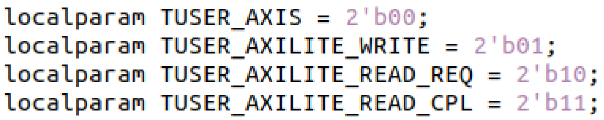
```verilog
task fpga_to_soc_cfg_write;
   input [27:0]addr;
   input [31:0]data;
   begin
   @ (posedge fpga_coreclk);
             fpga_axilite_write_req(addr, 4'b0001, data);
                     //write address = h0000_2100 ~ h0000_2FFF for AA internal register
             //step 3. fpga wait for write to soc
             repeat(100) @ (posedge soc_coreclk);    //TODO fpga wait for write to soc
   end
endtask
```



```verilog
task fpga_axilite_write_req;
        input [27:0] address;
        input [3:0] BE;
        input [31:0] data;
        begin
                fpga_as_is_tdata[27:0] <= address;     //for axilite write address phase
                fpga_as_is_tdata[31:28] <= BE;
                $strobe($time, "=> fpga_axilite_write_req in address phase = %x - tvalid", fpga_as_is_tdata);
                `ifdef USER_PROJECT_SIDEBAND_SUPPORT
                        fpga_as_is_tupsb <=  5'b00000;
                `endif
                fpga_as_is_tstrb <=  4'b0000;
                fpga_as_is_tkeep <=  4'b0000;
                fpga_as_is_tid <=  TID_DN_AA;          //target to Axis-Axilite
                fpga_as_is_tuser <=  TUSER_AXILITE_WRITE;              //for axilite write req
                fpga_as_is_tlast <=  1'b0;
                fpga_as_is_tvalid <= 1;

                @ (posedge fpga_coreclk);
                while (fpga_is_as_tready == 0) begin           // wait util fpga_is_as_tready == 1 then change data
                                @ (posedge fpga_coreclk);
                end
                $display($time, "=> fpga_axilite_write_req in address phase = %x - transfer", fpga_as_is_tdata);

                fpga_as_is_tdata <= data;      //for axilite write data phase
                $strobe($time, "=> fpga_axilite_write_req in data phase = %x - tvalid", fpga_as_is_tdata);
                `ifdef USER_PROJECT_SIDEBAND_SUPPORT
                        fpga_as_is_tupsb <=  5'b00000;
                `endif
                fpga_as_is_tstrb <=  4'b0000;
                fpga_as_is_tkeep <=  4'b0000;
                fpga_as_is_tid <=  TID_DN_AA;          //target to Axis-Axilite
                fpga_as_is_tuser <=  TUSER_AXILITE_WRITE;              //for axilite write req
                fpga_as_is_tlast <=  1'b1;             //tlast = 1
                fpga_as_is_tvalid <= 1;
                @ (posedge fpga_coreclk);
                while (fpga_is_as_tready == 0) begin           // wait util fpga_is_as_tready == 1 then change data
                                @ (posedge fpga_coreclk);
                end
                $display($time, "=> fpga_axilite_write_req in data phase = %x - transfer", fpga_as_is_tdata);
                fpga_as_is_tvalid <= 0;
        end
endtask
```

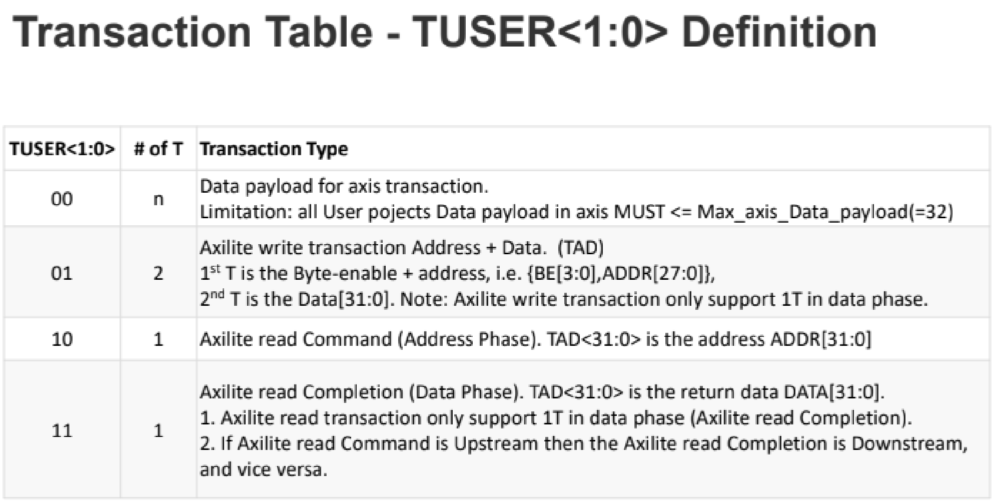


```verilog
localparam TID_DN_UP = 2'b00;
localparam TID_DN_AA = 2'b01;
localparam TID_UP_UP = 2'b00;
localparam TID_UP_AA = 2'b01;
localparam TID_UP_LA = 2'b10;
```



```verilog
localparam TUSER_AXIS = 2'b00;
localparam TUSER_AXILITE_WRITE = 2'b01;
localparam TUSER_AXILITE_READ_REQ = 2'b10;
localparam TUSER_AXILITE_READ_CPL = 2'b11;
```

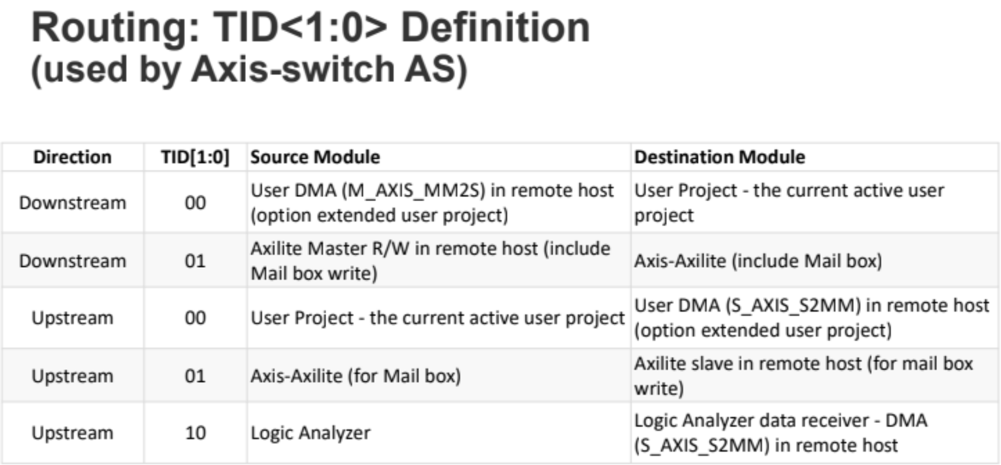
首先，FPGA side 只有 axi-stream 可以用，但這個系統中有設計 AXIS/AXIL module 可以把 axi-stream 轉成 axi-lite 來使用。我們先設定 TUSER[1:0]成 axilite write 模式，再利用 TID[1:0]來指向目標位置(user_prj)。 Axilite write 模式會傳兩筆資料，第一筆是 axi-lite 的 addr，第二筆為 axi-lite 的 data。下圖為 TUSER 和 TID 的定義。



Transaction Table - TUSER<1:0> Definition

TUSER<1:0>	# of T	Transaction Type
00	n	Data payload for axis transaction. Limitation: all User pojects Data payload in axis MUST <= Max_axis_Data_payload(=32)
01	2	Axilite write transaction Address + Data. (TAD) 1st T is the Byte-enable + address, i.e. {BE[3:0],ADDR[27:0]}, 2nd T is the Data[31:0]. Note: Axilite write transaction only support 1T in data phase.
10	1	Axilite read Command (Address Phase). TAD<31:0> is the address ADDR[31:0]
11	1	Axilite read Completion (Data Phase). TAD<31:0> is the return data DATA[31:0]. 1. Axilite read transaction only support 1T in data phase (Axilite read Completion). 2. If Axilite read Command is Upstream then the Axilite read Completion is Downstream, and vice versa.

(圖 5)



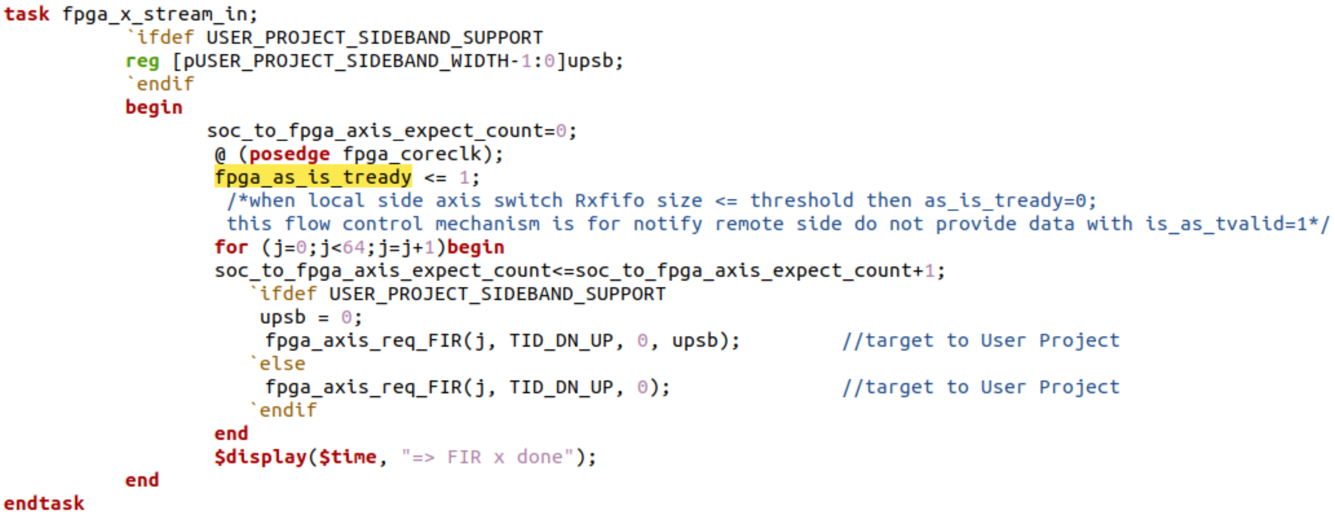
Routing: TID<1:0> Definition (used by Axis-switch AS)

Direction	TID[1:0]	Source Module	Destination Module
Downstream	00	User DMA (M_AXIS_MM2S) in remote host (option extended user project)	User Project - the current active user project
Downstream	01	Axilite Master R/W in remote host (include Mail box write)	Axis-Axilite (include Mail box)
Upstream	00	User Project - the current active user project	User DMA (S_AXIS_S2MM) in remote host (option extended user project)
Upstream	01	Axis-Axilite (for Mail box)	Axilite slave in remote host (for mail box write)
Upstream	10	Logic Analyzer	Logic Analyzer data receiver - DMA (S_AXIS_S2MM) in remote host

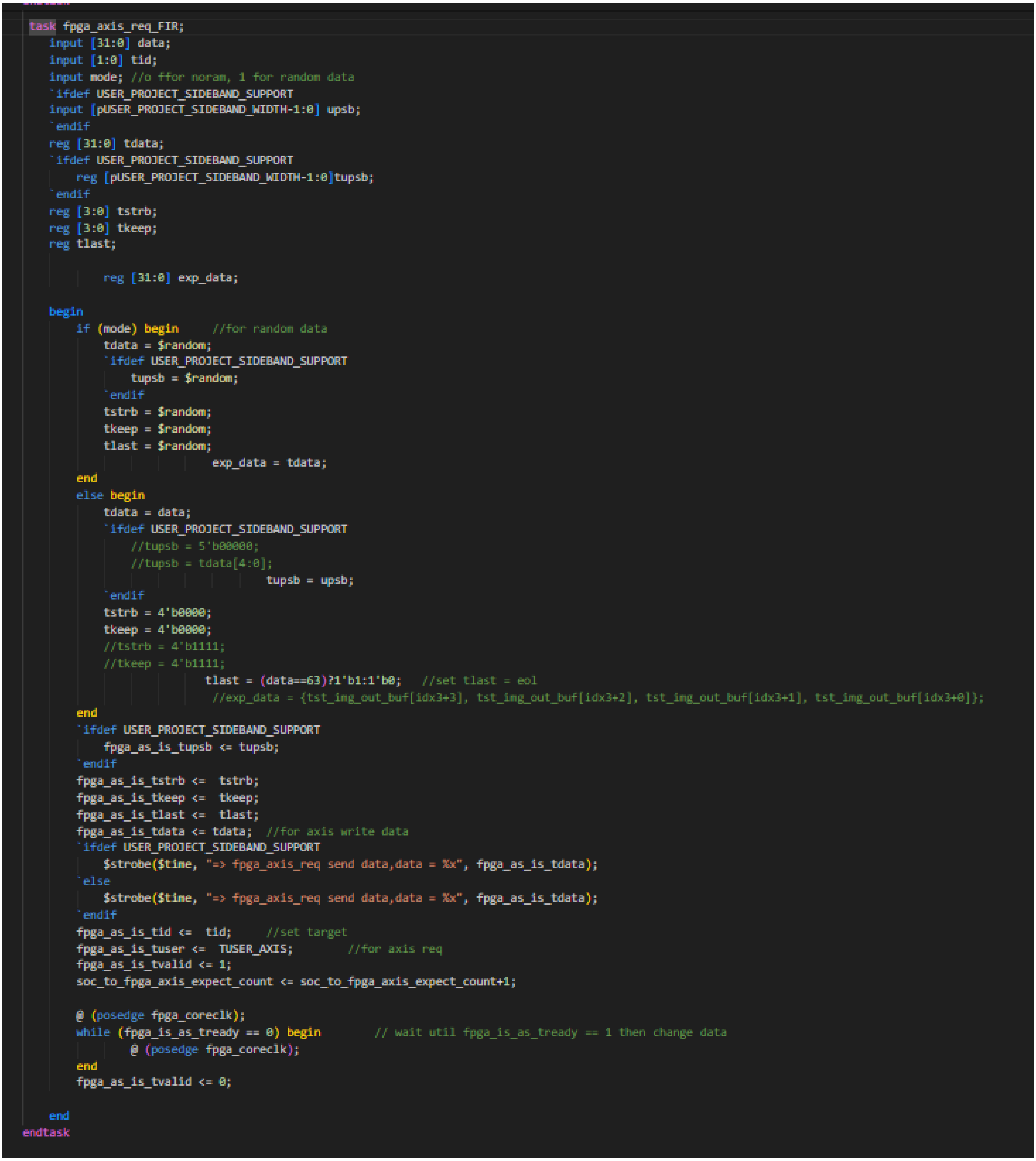
(圖 6)

5. Briefly describe how you feed in X data from FPGA side

Ans:



```
task fpga_x_stream_in;
         `ifdef USER_PROJECT_SIDEBAND_SUPPORT
        reg [pUSER_PROJECT_SIDEBAND_WIDTH-1:0]upsb;
         `endif
        begin
               soc_to_fpga_axis_expect_count=0;
                @ (posedge fpga_coreclk);
                fpga_as_is_tready <= 1;
                 /*when local side axis switch Rxfifo size <= threshold then as_is_tready=0;
                 this flow control mechanism is for notify remote side do not provide data with is_as_tvalid=1*/
                for (j=0;j<64;j=j+1)begin
                soc_to_fpga_axis_expect_count<=soc_to_fpga_axis_expect_count+1;
                    `ifdef USER_PROJECT_SIDEBAND_SUPPORT
                     upsb = 0;
                      fpga_axis_req_FIR(j, TID_DN_UP, 0, upsb);         //target to User Project
                    `else
                      fpga_axis_req_FIR(j, TID_DN_UP, 0);               //target to User Project
                    `endif
                end
                $display($time, "=> FIR x done");
        end
endtask
```

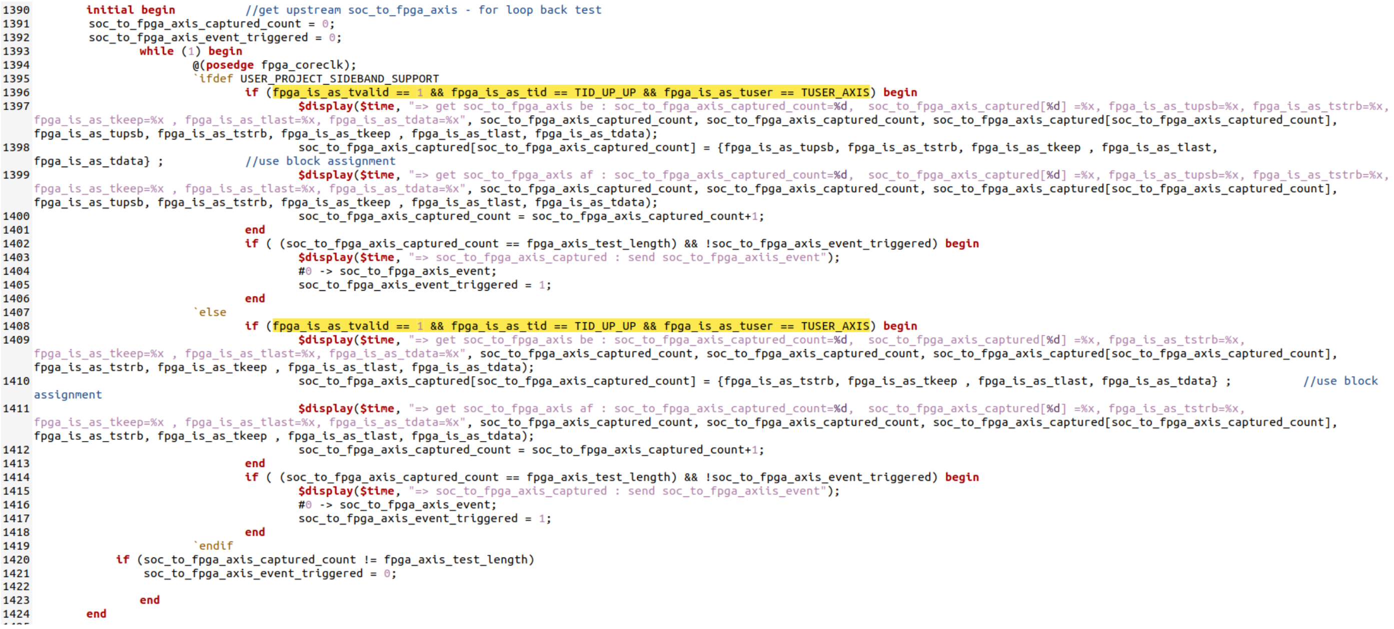


```
task fpga_axis_req_FIR;
    input [31:0] data;
    input [1:0] tid;
    input mode; //o ffor noram, 1 for random data
    `ifdef USER_PROJECT_SIDEBAND_SUPPORT
    input [pUSER_PROJECT_SIDEBAND_WIDTH-1:0] upsb;
    `endif
    reg [31:0] tdata;
    `ifdef USER_PROJECT_SIDEBAND_SUPPORT
        reg [pUSER_PROJECT_SIDEBAND_WIDTH-1:0]tupsb;
    `endif
    reg [3:0] tstrb;
    reg [3:0] tkeep;
    reg tlast;

            reg [31:0] exp_data;

    begin
        if (mode) begin     //for random data
            tdata = $random;
            `ifdef USER_PROJECT_SIDEBAND_SUPPORT
                tupsb = $random;
            `endif
            tstrb = $random;
            tkeep = $random;
            tlast = $random;
                            exp_data = tdata;
        end
        else begin
            tdata = data;
            `ifdef USER_PROJECT_SIDEBAND_SUPPORT
                //tupsb = 5'b00000;
                //tupsb = tdata[4:0];
                                tupsb = upsb;
            `endif
            tstrb = 4'b0000;
            tkeep = 4'b0000;
            //tstrb = 4'b1111;
            //tkeep = 4'b1111;
                           tlast = (data==63)?1'b1:1'b0;   //set tlast = eol
                            //exp_data = {tst_img_out_buf[idx3+3], tst_img_out_buf[idx3+2], tst_img_out_buf[idx3+1], tst_img_out_buf[idx3+0]};
        end
        `ifdef USER_PROJECT_SIDEBAND_SUPPORT
            fpga_as_is_tupsb <= tupsb;
        `endif
        fpga_as_is_tstrb <=  tstrb;
        fpga_as_is_tkeep <=  tkeep;
        fpga_as_is_tlast <=  tlast;
        fpga_as_is_tdata <= tdata;  //for axis write data
        `ifdef USER_PROJECT_SIDEBAND_SUPPORT
            $strobe($time, "=> fpga_axis_req send data,data = %x", fpga_as_is_tdata);
        `else
            $strobe($time, "=> fpga_axis_req send data,data = %x", fpga_as_is_tdata);
        `endif
        fpga_as_is_tid <=  tid;     //set target
        fpga_as_is_tuser <=  TUSER_AXIS;        //for axis req
        fpga_as_is_tvalid <= 1;
        soc_to_fpga_axis_expect_count <= soc_to_fpga_axis_expect_count+1;

        @ (posedge fpga_coreclk);
        while (fpga_is_as_tready == 0) begin        // wait util fpga_is_as_tready == 1 then change data
                @ (posedge fpga_coreclk);
        end
        fpga_as_is_tvalid <= 0;

    end
endtask
```

這裡我們一樣由 TUSER 和 TID 來設定我們 fpga 的 axi-stream，讓他可以直接送進去 user_prj 的 axi-stream in。

6. Briefly describe how you get output Y data in testbench, and how to do comparison with golden values?

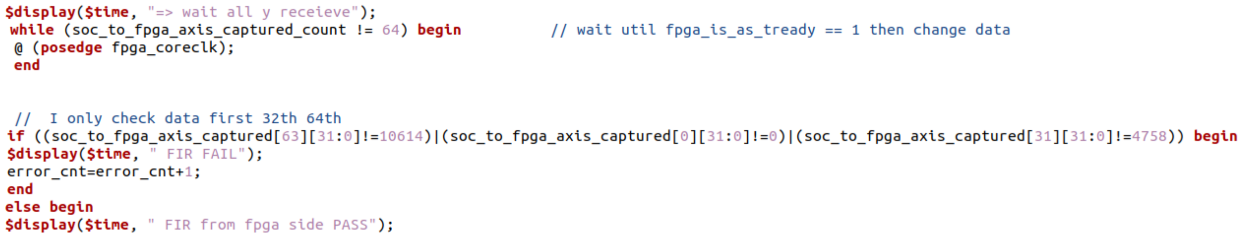
Ans (how you get output Y data in testbench):



```
        initial begin           //get upstream soc_to_fpga_axis - for loop back test
         soc_to_fpga_axis_captured_count = 0;
         soc_to_fpga_axis_event_triggered = 0;
                while (1) begin
                        @(posedge fpga_coreclk);
                        `ifdef USER_PROJECT_SIDEBAND_SUPPORT
                                if (fpga_is_as_tvalid == 1 && fpga_is_as_tid == TID_UP_UP && fpga_is_as_tuser == TUSER_AXIS) begin
                                        $display($time, "=> get soc_to_fpga_axis be : soc_to_fpga_axis_captured_count=%d,  soc_to_fpga_axis_captured[%d] =%x, fpga_is_as_tupsb=%x, fpga_is_as_tstrb=%x, fpga_is_as_tkeep=%x , fpga_is_as_tlast=%x, fpga_is_as_tdata=%x", soc_to_fpga_axis_captured_count, soc_to_fpga_axis_captured_count, soc_to_fpga_axis_captured[soc_to_fpga_axis_captured_count], fpga_is_as_tupsb, fpga_is_as_tstrb, fpga_is_as_tkeep , fpga_is_as_tlast, fpga_is_as_tdata);
                                        soc_to_fpga_axis_captured[soc_to_fpga_axis_captured_count] = {fpga_is_as_tupsb, fpga_is_as_tstrb, fpga_is_as_tkeep , fpga_is_as_tlast, fpga_is_as_tdata} ;             //use block assignment
                                        $display($time, "=> get soc_to_fpga_axis af : soc_to_fpga_axis_captured_count=%d,  soc_to_fpga_axis_captured[%d] =%x, fpga_is_as_tupsb=%x, fpga_is_as_tstrb=%x, fpga_is_as_tkeep=%x , fpga_is_as_tlast=%x, fpga_is_as_tdata=%x", soc_to_fpga_axis_captured_count, soc_to_fpga_axis_captured_count, soc_to_fpga_axis_captured[soc_to_fpga_axis_captured_count], fpga_is_as_tupsb, fpga_is_as_tstrb, fpga_is_as_tkeep , fpga_is_as_tlast, fpga_is_as_tdata);
                                        soc_to_fpga_axis_captured_count = soc_to_fpga_axis_captured_count+1;
                                end
                                if ( (soc_to_fpga_axis_captured_count == fpga_axis_test_length) && !soc_to_fpga_axis_event_triggered) begin
                                        $display($time, "=> soc_to_fpga_axis_captured : send soc_to_fpga_axiis_event");
                                        #0 -> soc_to_fpga_axis_event;
                                        soc_to_fpga_axis_event_triggered = 1;
                                end
                        `else
                                if (fpga_is_as_tvalid == 1 && fpga_is_as_tid == TID_UP_UP && fpga_is_as_tuser == TUSER_AXIS) begin
                                        $display($time, "=> get soc_to_fpga_axis be : soc_to_fpga_axis_captured_count=%d,  soc_to_fpga_axis_captured[%d] =%x, fpga_is_as_tstrb=%x, fpga_is_as_tkeep=%x , fpga_is_as_tlast=%x, fpga_is_as_tdata=%x", soc_to_fpga_axis_captured_count, soc_to_fpga_axis_captured_count, soc_to_fpga_axis_captured[soc_to_fpga_axis_captured_count], fpga_is_as_tstrb, fpga_is_as_tkeep , fpga_is_as_tlast, fpga_is_as_tdata);
                                        soc_to_fpga_axis_captured[soc_to_fpga_axis_captured_count] = {fpga_is_as_tstrb, fpga_is_as_tkeep , fpga_is_as_tlast, fpga_is_as_tdata} ;          //use block assignment
                                        $display($time, "=> get soc_to_fpga_axis af : soc_to_fpga_axis_captured_count=%d,  soc_to_fpga_axis_captured[%d] =%x, fpga_is_as_tstrb=%x, fpga_is_as_tkeep=%x , fpga_is_as_tlast=%x, fpga_is_as_tdata=%x", soc_to_fpga_axis_captured_count, soc_to_fpga_axis_captured_count, soc_to_fpga_axis_captured[soc_to_fpga_axis_captured_count], fpga_is_as_tstrb, fpga_is_as_tkeep , fpga_is_as_tlast, fpga_is_as_tdata);
                                        soc_to_fpga_axis_captured_count = soc_to_fpga_axis_captured_count+1;
                                end
                                if ( (soc_to_fpga_axis_captured_count == fpga_axis_test_length) && !soc_to_fpga_axis_event_triggered) begin
                                        $display($time, "=> soc_to_fpga_axis_captured : send soc_to_fpga_axiis_event");
                                        #0 -> soc_to_fpga_axis_event;
                                        soc_to_fpga_axis_event_triggered = 1;
                                end
                        `endif
            if (soc_to_fpga_axis_captured_count != fpga_axis_test_length)
                soc_to_fpga_axis_event_triggered = 0;

                end
        end
```

這裡 while 迴圈會一直偵測是否有 stream-out 出來(黃色標示判斷)，如果有會把資料存進去 reg [(4+4+1+32-1):0]soc_to_fpga_axis_captured[127:0]。

Ans (how to do comparison with golden values):



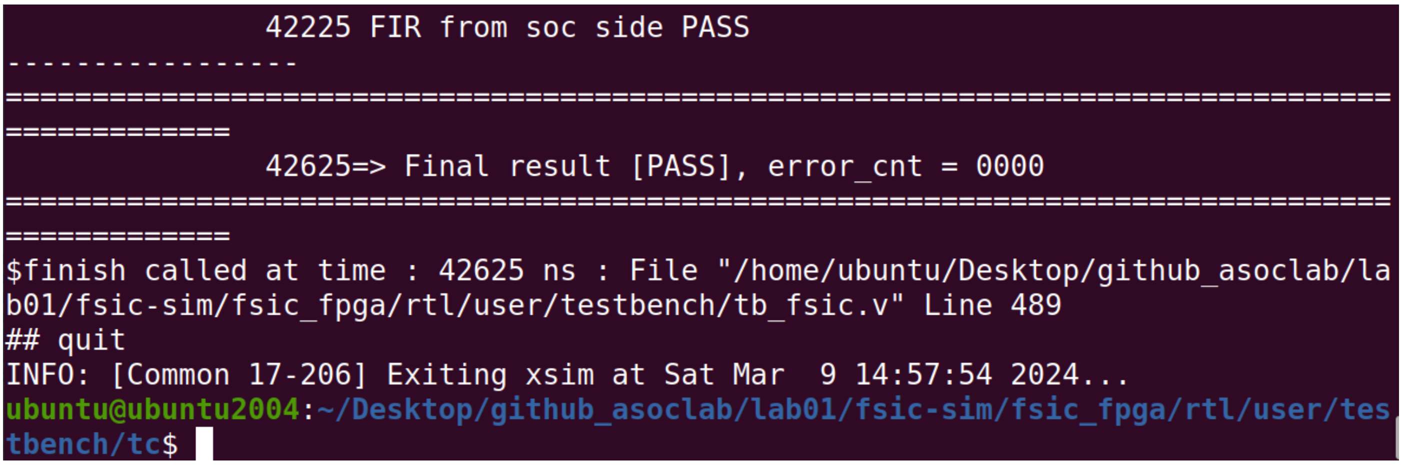
```
$display($time, "=> wait all y receieve");
 while (soc_to_fpga_axis_captured_count != 64) begin          // wait util fpga_is_as_tready == 1 then change data
 @ (posedge fpga_coreclk);
 end


 //  I only check data first 32th 64th
if ((soc_to_fpga_axis_captured[63][31:0]!=10614)|(soc_to_fpga_axis_captured[0][31:0]!=0)|(soc_to_fpga_axis_captured[31][31:0]!=4758)) begin
$display($time, " FIR FAIL");
error_cnt=error_cnt+1;
end
else begin
$display($time, " FIR from fpga side PASS");
```

我只有比較第 1、32 、64 筆是否正確(開始、中間、結束)，注意到程式碼有一個 while 是為了讓 y 完整收完才進行下一步驟。

7. screenshot simulation results printed on screen, to show that your Test#1 & Test#2 complete successfully

Ans:

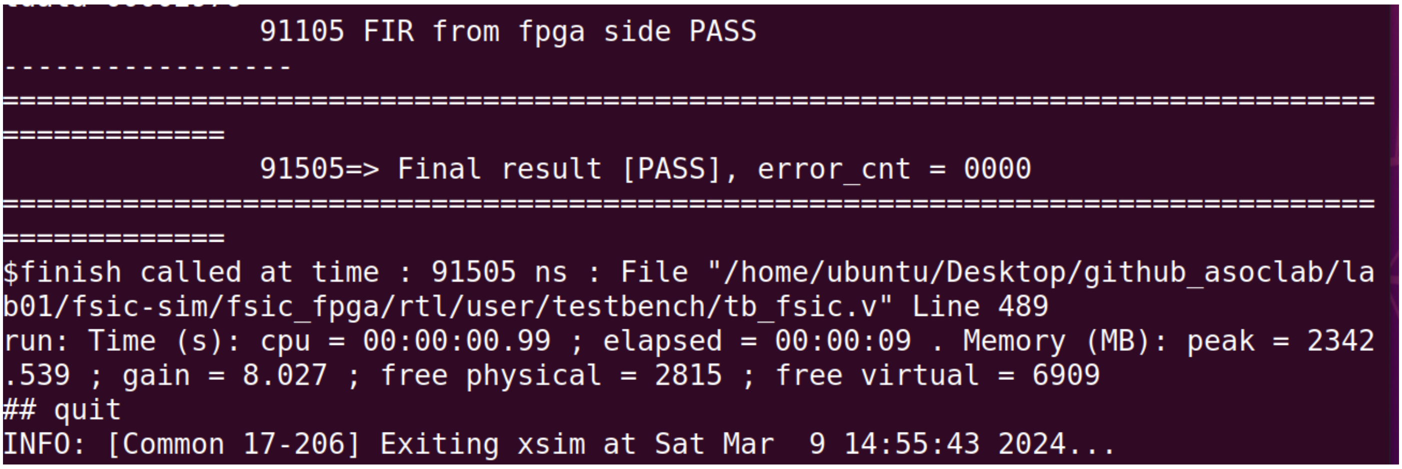
Test 1:



```
                  42225 FIR from soc side PASS
-----------------
=================================================================================
=============
                  42625=> Final result [PASS], error_cnt = 0000
=================================================================================
=============
$finish called at time : 42625 ns : File "/home/ubuntu/Desktop/github_asoclab/la
b01/fsic-sim/fsic_fpga/rtl/user/testbench/tb_fsic.v" Line 489
## quit
INFO: [Common 17-206] Exiting xsim at Sat Mar  9 14:57:54 2024...
ubuntu@ubuntu2004:~/Desktop/github_asoclab/lab01/fsic-sim/fsic_fpga/rtl/user/tes
tbench/tc$ 
```

(圖 7)

Test 2:



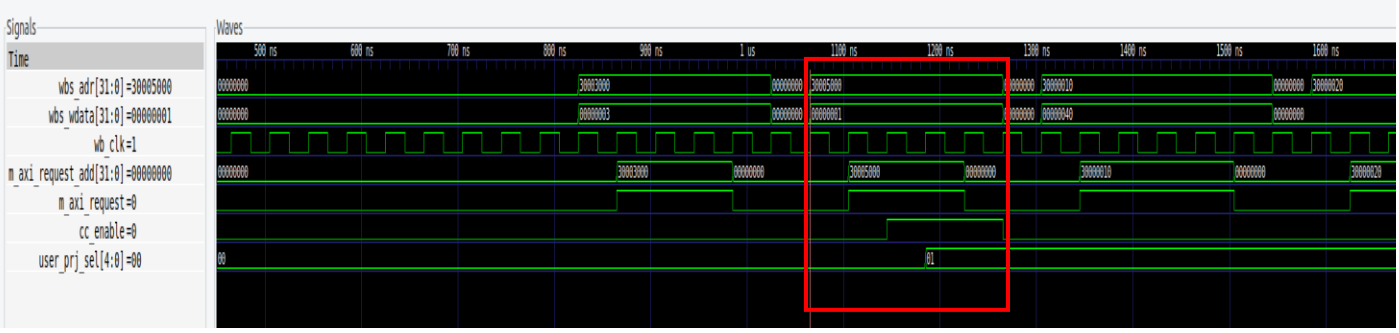
```
               91105 FIR from fpga side PASS
-----------------
================================================================================
=============
               91505=> Final result [PASS], error_cnt = 0000
================================================================================
=============
$finish called at time : 91505 ns : File "/home/ubuntu/Desktop/github_asoclab/la
b01/fsic-sim/fsic_fpga/rtl/user/testbench/tb_fsic.v" Line 489
run: Time (s): cpu = 00:00:00.99 ; elapsed = 00:00:09 . Memory (MB): peak = 2342
.539 ; gain = 8.027 ; free physical = 2815 ; free virtual = 6909
## quit
INFO: [Common 17-206] Exiting xsim at Sat Mar  9 14:55:43 2024...
```

(圖 8)

8. screenshot simulation waveform

Ans:

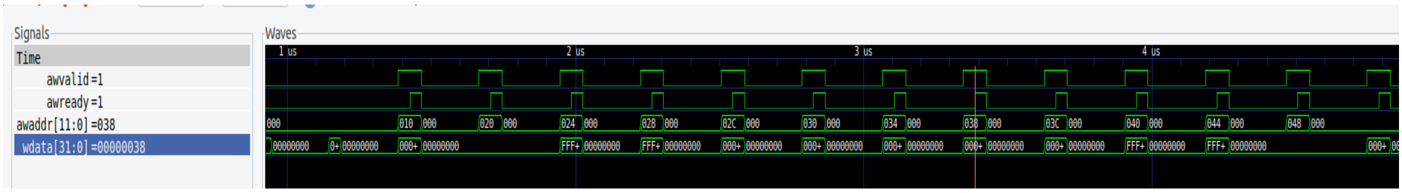
Configuration cycle:



(圖 9)

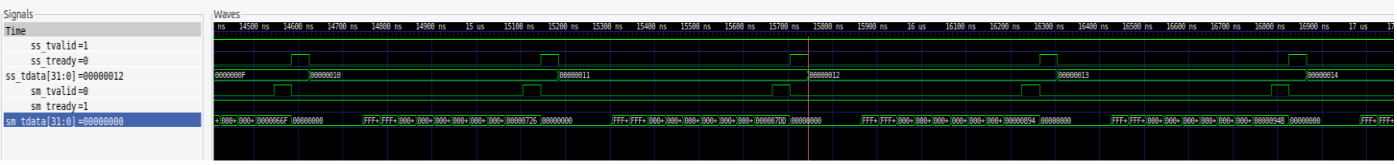
過程如 Q1 一樣，花了 3T。

Axi-lite transaction cycles:



(圖 10)

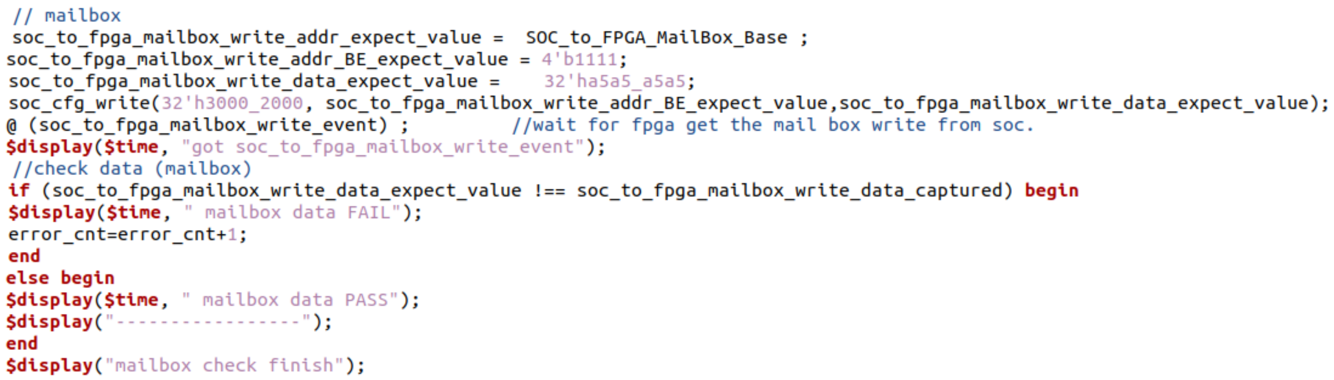
Stream-in, Stream-out:



(圖 11)

9.(option) mailbox design

Ans:

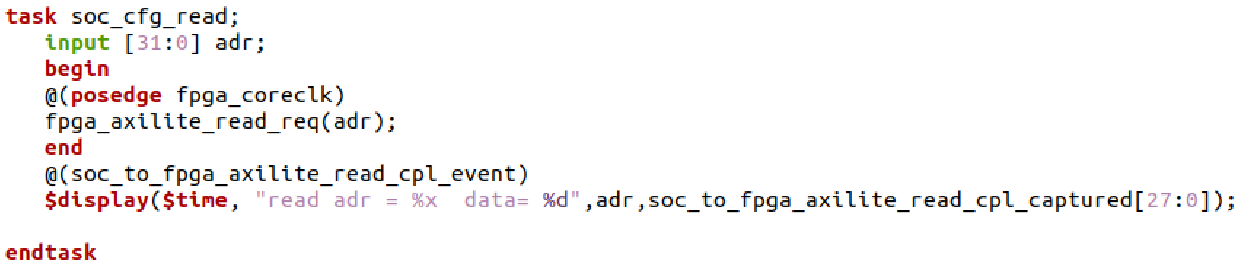


```
 // mailbox
 soc_to_fpga_mailbox_write_addr_expect_value =  SOC_to_FPGA_MailBox_Base ;
soc_to_fpga_mailbox_write_addr_BE_expect_value = 4'b1111;
soc_to_fpga_mailbox_write_data_expect_value =    32'ha5a5_a5a5;
soc_cfg_write(32'h3000_2000, soc_to_fpga_mailbox_write_addr_BE_expect_value,soc_to_fpga_mailbox_write_data_expect_value);
@ (soc_to_fpga_mailbox_write_event) ;          //wait for fpga get the mail box write from soc.
$display($time, "got soc_to_fpga_mailbox_write_event");
 //check data (mailbox)
if (soc_to_fpga_mailbox_write_data_expect_value !== soc_to_fpga_mailbox_write_data_captured) begin
$display($time, " mailbox data FAIL");
error_cnt=error_cnt+1;
end
else begin
$display($time, " mailbox data PASS");
$display("-----------------");
end
$display("mailbox check finish");
```

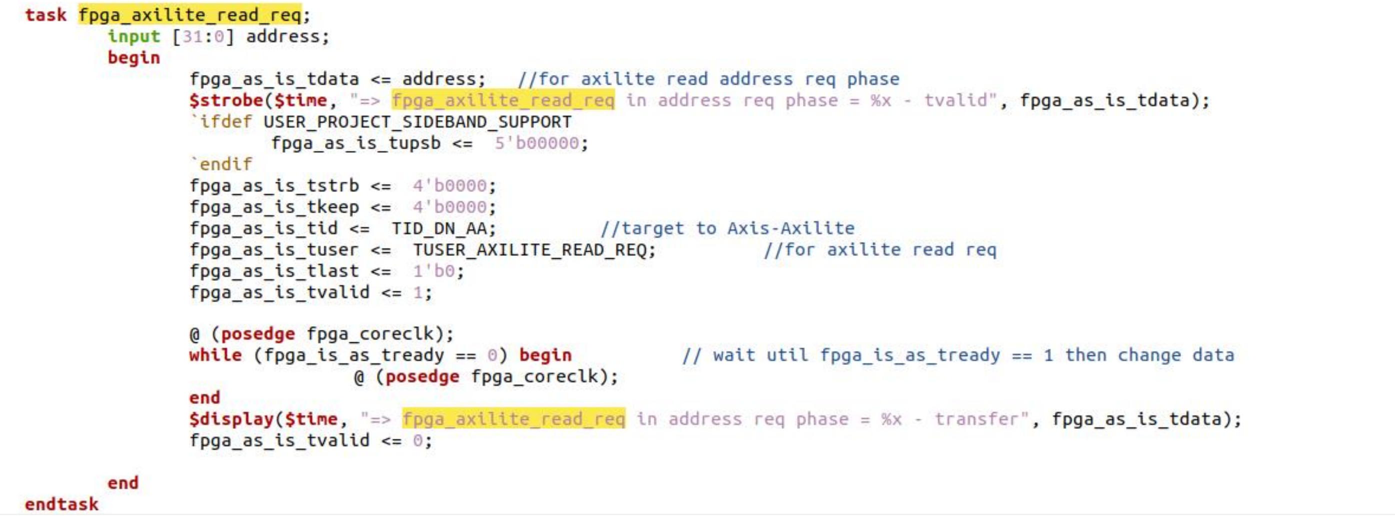
從 soc side 利用 soc_fpg_write 寫到 mailbox 的位置 32’h3000_2000，再等待 soc_to_fpga_mailbox_write_event 發生。最後應證是否 FPGA side 收到的資料跟 SOC side 傳的一樣。

10.(option)read configuration

如果我們要測試是否有正確寫入 coef，我們可以利用下面程式碼來讀取。



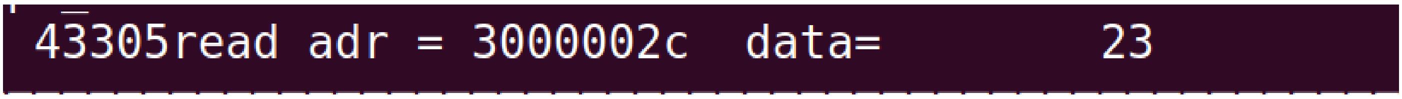
```
task soc_cfg_read;
   input [31:0] adr;
   begin
   @(posedge fpga_coreclk)
   fpga_axilite_read_req(adr);
   end
   @(soc_to_fpga_axilite_read_cpl_event)
   $display($time, "read adr = %x  data= %d",adr,soc_to_fpga_axilite_read_cpl_captured[27:0]);

endtask
```



```
task fpga_axilite_read_req;
        input [31:0] address;
        begin
                fpga_as_is_tdata <= address;   //for axilite read address req phase
                $strobe($time, "=> fpga_axilite_read_req in address req phase = %x - tvalid", fpga_as_is_tdata);
                `ifdef USER_PROJECT_SIDEBAND_SUPPORT
                        fpga_as_is_tupsb <=  5'b00000;
                `endif
                fpga_as_is_tstrb <=  4'b0000;
                fpga_as_is_tkeep <=  4'b0000;
                fpga_as_is_tid <=  TID_DN_AA;          //target to Axis-Axilite
                fpga_as_is_tuser <=  TUSER_AXILITE_READ_REQ;          //for axilite read req
                fpga_as_is_tlast <=  1'b0;
                fpga_as_is_tvalid <= 1;

                @ (posedge fpga_coreclk);
                while (fpga_is_as_tready == 0) begin          // wait util fpga_is_as_tready == 1 then change data
                                @ (posedge fpga_coreclk);
                end
                $display($time, "=> fpga_axilite_read_req in address req phase = %x - transfer", fpga_as_is_tdata);
                fpga_as_is_tvalid <= 0;

        end
endtask
```

說明，他其實是利用 fpga side 去讀取，利用上面所提我們可以設定對應的 tid tuser 來判斷我們 axi_stream 要做甚麼事情。

結果如下圖，我可以讀取 3000_002c 這個位置來看我 soc_write 是否有正確寫入。



```
 43305read adr = 3000002c  data=        23
```