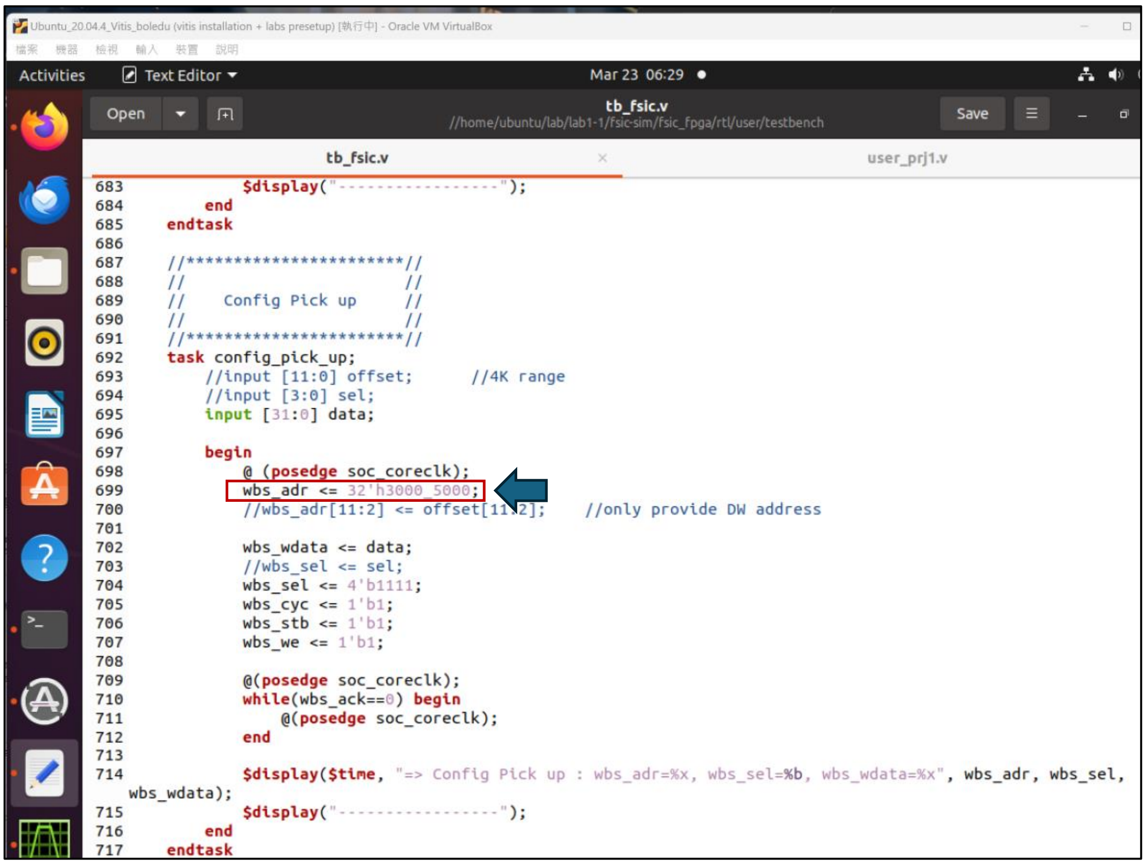
Advanced Soc Lab01 – fsic simulation

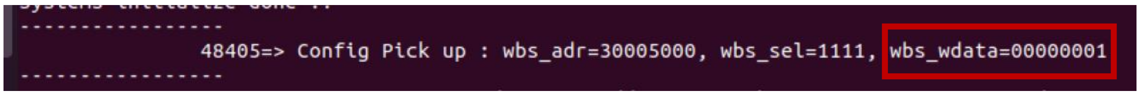
112061622 王允杰

Question 1. **Show the code** that you use to program configuration address [‘h3000_5000]


Ubuntu_20.04.4_Vitis_boledu (vitis installation + labs presetup) [執行中] - Oracle VM VirtualBox
Activities
Text Editor
Mar 23 06:29
Open
tb_fsic.v
//home/ubuntu/lab/lab1-1/fsic-sim/fsic_fpga/rtl/user/testbench
Save
tb_fsic.v
user_prj1.v

```verilog
            $display("-----------------");
        end
    endtask

    //***********************//
    //                       //
    //    Config Pick up     //
    //                       //
    //***********************//
    task config_pick_up;
        //input [11:0] offset;      //4K range
        //input [3:0] sel;
        input [31:0] data;

        begin
            @ (posedge soc_coreclk);
            wbs_adr <= 32'h3000_5000;
            //wbs_adr[11:2] <= offset[11:2];     //only provide DW address

            wbs_wdata <= data;
            //wbs_sel <= sel;
            wbs_sel <= 4'b1111;
            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, "=> Config Pick up : wbs_adr=%x, wbs_sel=%b, wbs_wdata=%x", wbs_adr, wbs_sel,
wbs_wdata);
            $display("-----------------");
        end
    endtask
```

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



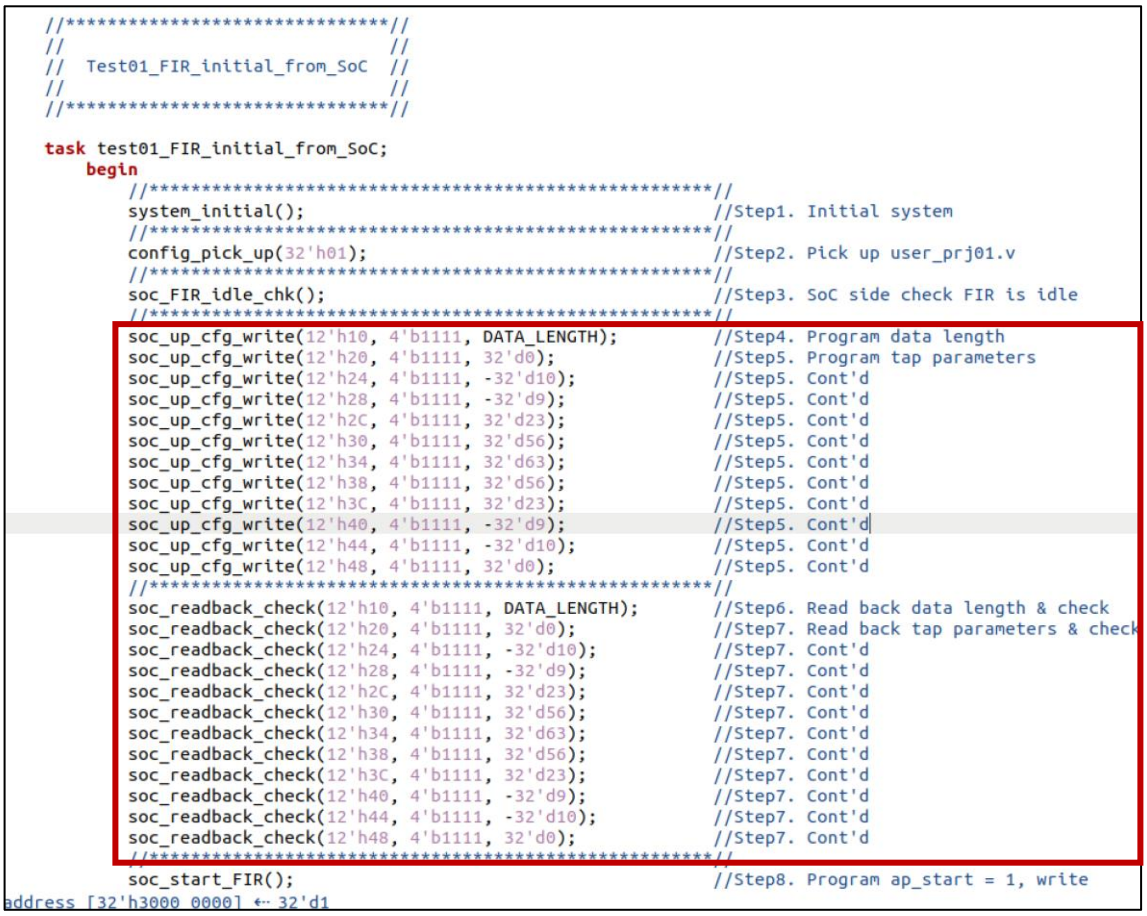
```
-----------------
                 48405=> Config Pick up : wbs_adr=30005000, wbs_sel=1111, wbs_wdata=00000001
-----------------
```

由 wbs_wdata 輸入不同 data，來改變 signal user_prj_sel。

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

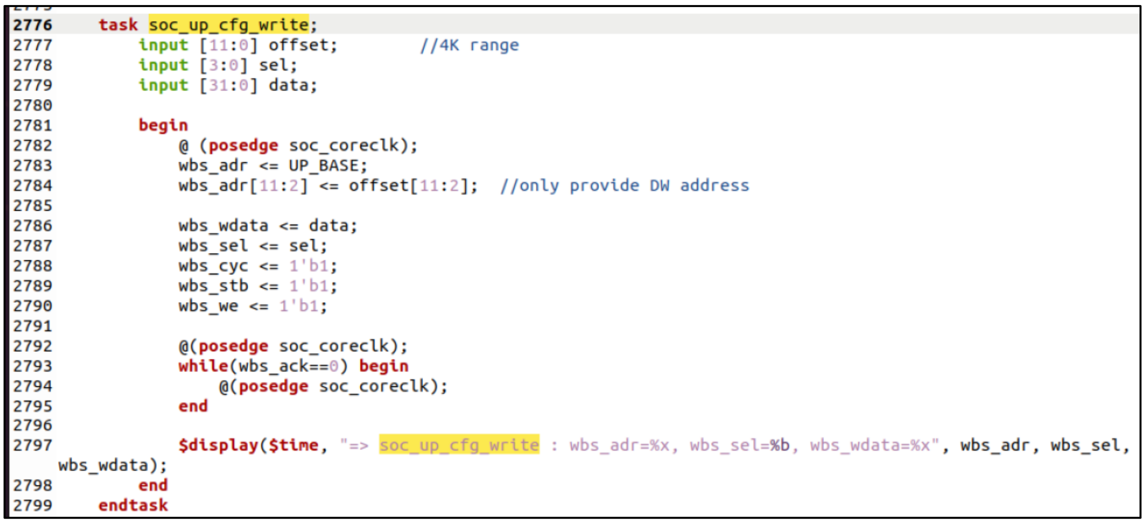
基本上，就是利用 soc_up_cfg_write 將(tap parameter, length)寫入後，再進行(tap parameter, length)讀取後與 golden value 比對。

Code：

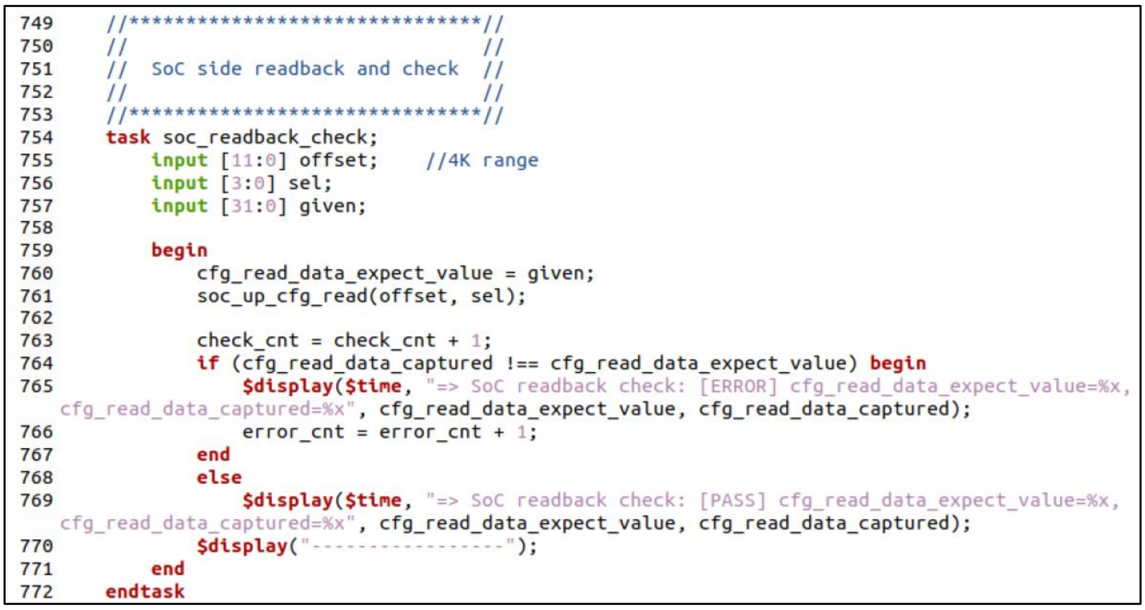


```
//*******************************//
//                               //
//  Test01_FIR_initial_from_SoC  //
//                               //
//*******************************//

task test01_FIR_initial_from_SoC;
    begin
        //********************************************************//
        system_initial();                                          //Step1. Initial system
        //********************************************************//
        config_pick_up(32'h01);                                    //Step2. Pick up user_prj01.v
        //********************************************************//
        soc_FIR_idle_chk();                                        //Step3. SoC side check FIR is idle
        //********************************************************//
        soc_up_cfg_write(12'h10, 4'b1111, DATA_LENGTH);            //Step4. Program data length
        soc_up_cfg_write(12'h20, 4'b1111, 32'd0);                  //Step5. Program tap parameters
        soc_up_cfg_write(12'h24, 4'b1111, -32'd10);                //Step5. Cont'd
        soc_up_cfg_write(12'h28, 4'b1111, -32'd9);                 //Step5. Cont'd
        soc_up_cfg_write(12'h2C, 4'b1111, 32'd23);                 //Step5. Cont'd
        soc_up_cfg_write(12'h30, 4'b1111, 32'd56);                 //Step5. Cont'd
        soc_up_cfg_write(12'h34, 4'b1111, 32'd63);                 //Step5. Cont'd
        soc_up_cfg_write(12'h38, 4'b1111, 32'd56);                 //Step5. Cont'd
        soc_up_cfg_write(12'h3C, 4'b1111, 32'd23);                 //Step5. Cont'd
        soc_up_cfg_write(12'h40, 4'b1111, -32'd9);                 //Step5. Cont'd
        soc_up_cfg_write(12'h44, 4'b1111, -32'd10);                //Step5. Cont'd
        soc_up_cfg_write(12'h48, 4'b1111, 32'd0);                  //Step5. Cont'd
        //********************************************************//
        soc_readback_check(12'h10, 4'b1111, DATA_LENGTH);          //Step6. Read back data length & check
        soc_readback_check(12'h20, 4'b1111, 32'd0);                //Step7. Read back tap parameters & check
        soc_readback_check(12'h24, 4'b1111, -32'd10);              //Step7. Cont'd
        soc_readback_check(12'h28, 4'b1111, -32'd9);               //Step7. Cont'd
        soc_readback_check(12'h2C, 4'b1111, 32'd23);               //Step7. Cont'd
        soc_readback_check(12'h30, 4'b1111, 32'd56);               //Step7. Cont'd
        soc_readback_check(12'h34, 4'b1111, 32'd63);               //Step7. Cont'd
        soc_readback_check(12'h38, 4'b1111, 32'd56);               //Step7. Cont'd
        soc_readback_check(12'h3C, 4'b1111, 32'd23);               //Step7. Cont'd
        soc_readback_check(12'h40, 4'b1111, -32'd9);               //Step7. Cont'd
        soc_readback_check(12'h44, 4'b1111, -32'd10);              //Step7. Cont'd
        soc_readback_check(12'h48, 4'b1111, 32'd0);                //Step7. Cont'd
        //********************************************************//
        soc_start_FIR();                                           //Step8. Program ap_start = 1, write
address [32'h3000_0000] <- 32'd1
```

Sub function：



```
task soc_up_cfg_write;
    input [11:0] offset;        //4K range
    input [3:0] sel;
    input [31:0] data;

    begin
        @ (posedge soc_coreclk);
        wbs_adr <= UP_BASE;
        wbs_adr[11:2] <= offset[11:2];  //only provide DW address

        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_up_cfg_write : wbs_adr=%x, wbs_sel=%b, wbs_wdata=%x", wbs_adr, wbs_sel,
wbs_wdata);
    end
endtask
```

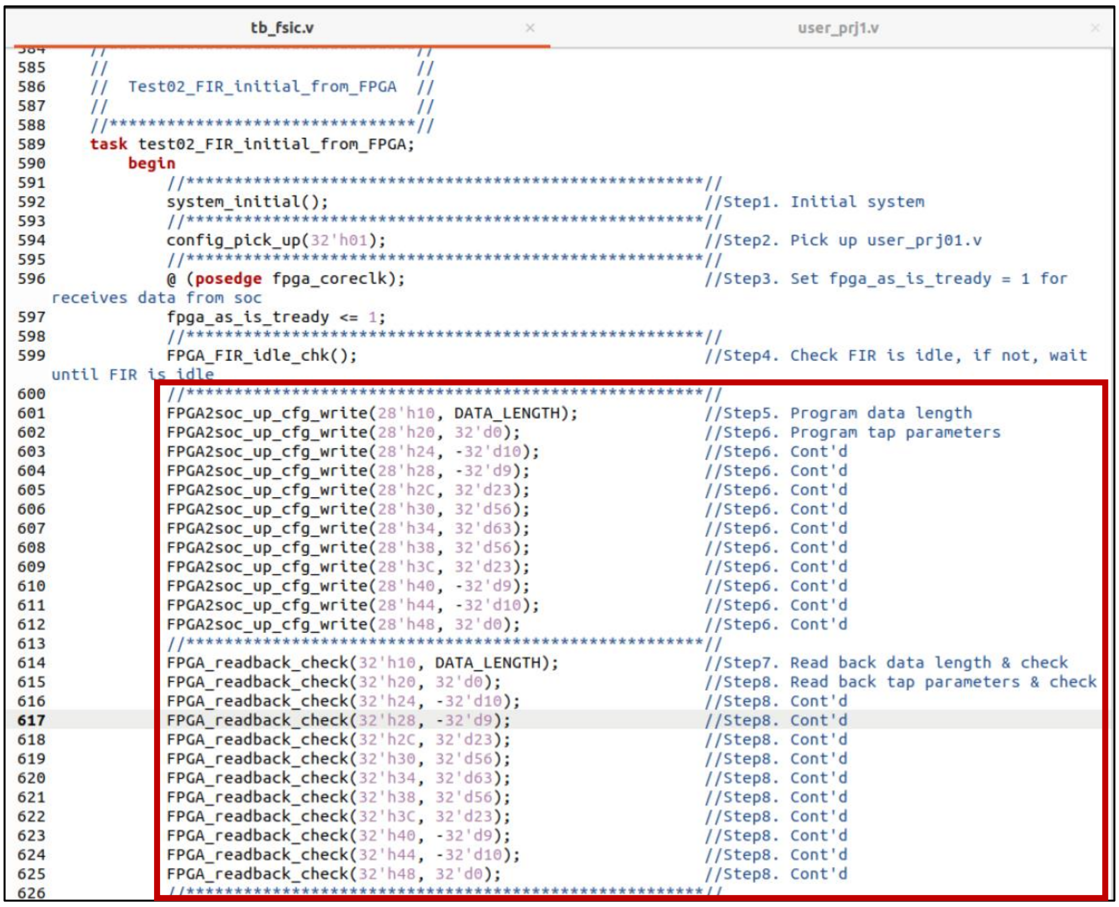


```
//********************************//
//                                //
//  SoC side readback and check   //
//                                //
//********************************//
task soc_readback_check;
    input [11:0] offset;    //4K range
    input [3:0] sel;
    input [31:0] given;

    begin
        cfg_read_data_expect_value = given;
        soc_up_cfg_read(offset, sel);

        check_cnt = check_cnt + 1;
        if (cfg_read_data_captured !== cfg_read_data_expect_value) begin
            $display($time, "=> SoC readback check: [ERROR] cfg_read_data_expect_value=%x,
cfg_read_data_captured=%x", cfg_read_data_expect_value, cfg_read_data_captured);
            error_cnt = error_cnt + 1;
        end
        else
            $display($time, "=> SoC readback check: [PASS] cfg_read_data_expect_value=%x,
cfg_read_data_captured=%x", cfg_read_data_expect_value, cfg_read_data_captured);
        $display("-----------------");
    end
endtask
```

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

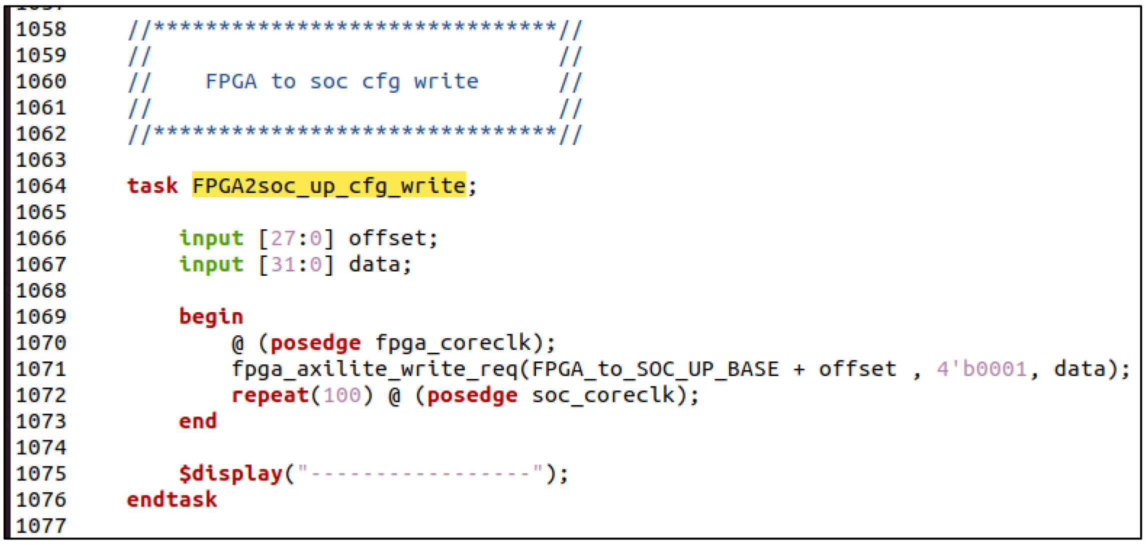
基本上，方式與 **Question 3.**方法雷同，就是利用 FPGA2soc_up_cfg_write 將(tap parameter, length)寫入後，再進行(tap parameter, length)讀取後與 golden value 比對，此外是用提供的 task006()來做 modify。

Code：

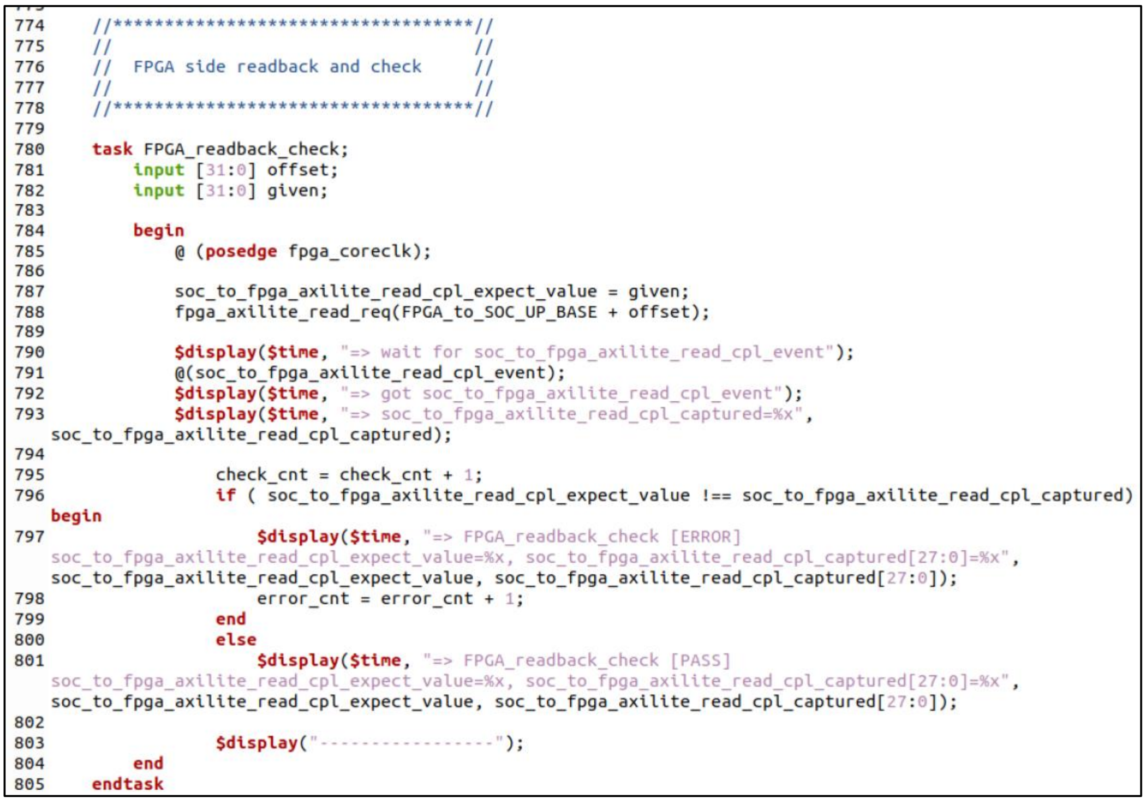


```
tb_fsic.v                                              user_prj1.v

    //                                  //
    //  Test02_FIR_initial_from_FPGA  //
    //                                  //
    //********************************//
    task test02_FIR_initial_from_FPGA;
        begin
            //*******************************************************//
            system_initial();                                        //Step1. Initial system
            //*******************************************************//
            config_pick_up(32'h01);                                  //Step2. Pick up user_prj01.v
            //*******************************************************//
            @ (posedge fpga_coreclk);                                //Step3. Set fpga_as_is_tready = 1 for
receives data from soc
            fpga_as_is_tready <= 1;
            //*******************************************************//
            FPGA_FIR_idle_chk();                                     //Step4. Check FIR is idle, if not, wait
until FIR is idle
            //*******************************************************//
            FPGA2soc_up_cfg_write(28'h10, DATA_LENGTH);              //Step5. Program data length
            FPGA2soc_up_cfg_write(28'h20, 32'd0);                    //Step6. Program tap parameters
            FPGA2soc_up_cfg_write(28'h24, -32'd10);                  //Step6. Cont'd
            FPGA2soc_up_cfg_write(28'h28, -32'd9);                   //Step6. Cont'd
            FPGA2soc_up_cfg_write(28'h2C, 32'd23);                   //Step6. Cont'd
            FPGA2soc_up_cfg_write(28'h30, 32'd56);                   //Step6. Cont'd
            FPGA2soc_up_cfg_write(28'h34, 32'd63);                   //Step6. Cont'd
            FPGA2soc_up_cfg_write(28'h38, 32'd56);                   //Step6. Cont'd
            FPGA2soc_up_cfg_write(28'h3C, 32'd23);                   //Step6. Cont'd
            FPGA2soc_up_cfg_write(28'h40, -32'd9);                   //Step6. Cont'd
            FPGA2soc_up_cfg_write(28'h44, -32'd10);                  //Step6. Cont'd
            FPGA2soc_up_cfg_write(28'h48, 32'd0);                    //Step6. Cont'd
            //*******************************************************//
            FPGA_readback_check(32'h10, DATA_LENGTH);                //Step7. Read back data length & check
            FPGA_readback_check(32'h20, 32'd0);                      //Step8. Read back tap parameters & check
            FPGA_readback_check(32'h24, -32'd10);                    //Step8. Cont'd
            FPGA_readback_check(32'h28, -32'd9);                     //Step8. Cont'd
            FPGA_readback_check(32'h2C, 32'd23);                     //Step8. Cont'd
            FPGA_readback_check(32'h30, 32'd56);                     //Step8. Cont'd
            FPGA_readback_check(32'h34, 32'd63);                     //Step8. Cont'd
            FPGA_readback_check(32'h38, 32'd56);                     //Step8. Cont'd
            FPGA_readback_check(32'h3C, 32'd23);                     //Step8. Cont'd
            FPGA_readback_check(32'h40, -32'd9);                     //Step8. Cont'd
            FPGA_readback_check(32'h44, -32'd10);                    //Step8. Cont'd
            FPGA_readback_check(32'h48, 32'd0);                      //Step8. Cont'd
            //*******************************************************//
```

Sub function：

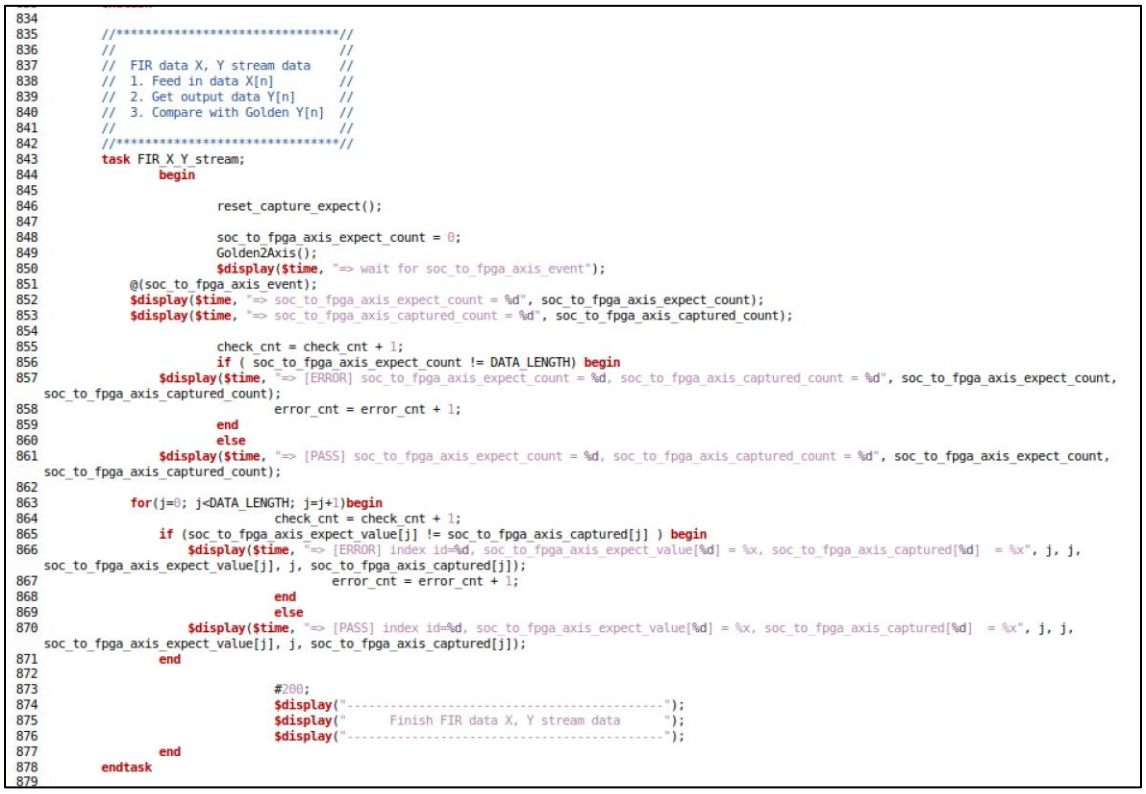


```
    //********************************//
    //                                  //
    //    FPGA to soc cfg write         //
    //                                  //
    //********************************//

    task FPGA2soc_up_cfg_write;

        input [27:0] offset;
        input [31:0] data;

        begin
            @ (posedge fpga_coreclk);
            fpga_axilite_write_req(FPGA_to_SOC_UP_BASE + offset , 4'b0001, data);
            repeat(100) @ (posedge soc_coreclk);
        end

        $display("-----------------");
    endtask

```



```verilog
    //************************************//
    //                                    //
    //  FPGA side readback and check      //
    //                                    //
    //************************************//

    task FPGA_readback_check;
        input [31:0] offset;
        input [31:0] given;

        begin
            @ (posedge fpga_coreclk);

            soc_to_fpga_axilite_read_cpl_expect_value = given;
            fpga_axilite_read_req(FPGA_to_SOC_UP_BASE + offset);

            $display($time, "=> wait for soc_to_fpga_axilite_read_cpl_event");
            @(soc_to_fpga_axilite_read_cpl_event);
            $display($time, "=> got soc_to_fpga_axilite_read_cpl_event");
            $display($time, "=> soc_to_fpga_axilite_read_cpl_captured=%x", soc_to_fpga_axilite_read_cpl_captured);

                check_cnt = check_cnt + 1;
                if ( soc_to_fpga_axilite_read_cpl_expect_value !== soc_to_fpga_axilite_read_cpl_captured) begin
                    $display($time, "=> FPGA_readback_check [ERROR] soc_to_fpga_axilite_read_cpl_expect_value=%x, soc_to_fpga_axilite_read_cpl_captured[27:0]=%x", soc_to_fpga_axilite_read_cpl_expect_value, soc_to_fpga_axilite_read_cpl_captured[27:0]);
                    error_cnt = error_cnt + 1;
                end
                else
                    $display($time, "=> FPGA_readback_check [PASS] soc_to_fpga_axilite_read_cpl_expect_value=%x, soc_to_fpga_axilite_read_cpl_captured[27:0]=%x", soc_to_fpga_axilite_read_cpl_expect_value, soc_to_fpga_axilite_read_cpl_captured[27:0]);

                $display("-----------------");
        end
    endtask
```

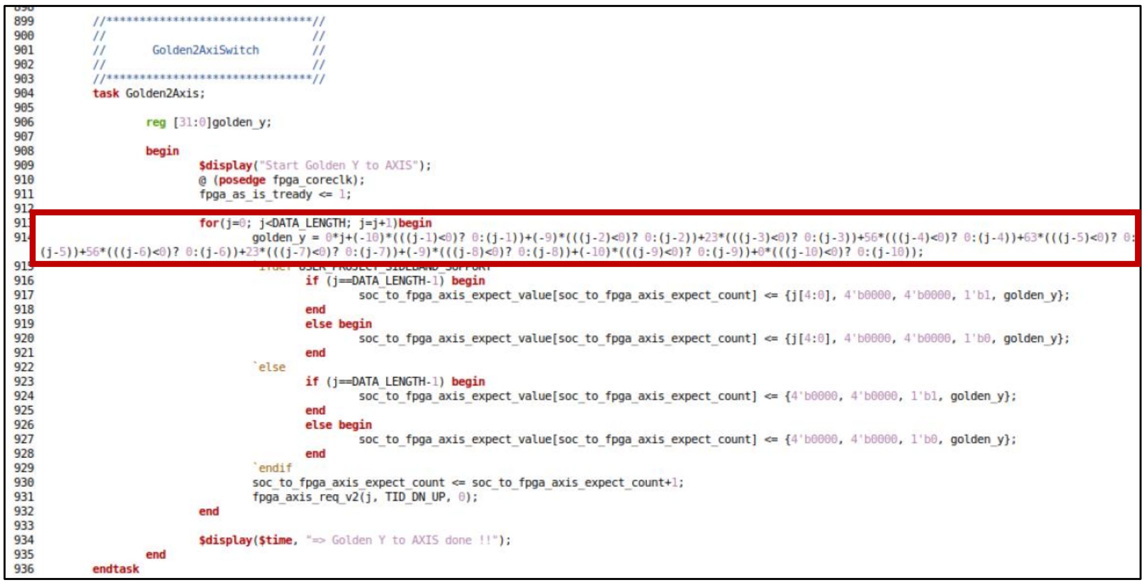
Question 5. **Briefly describe** how you feed in X data from FPGA side.



```verilog
        //********************************//
        //                                //
        //  FIR data X, Y stream data     //
        //  1. Feed in data X[n]          //
        //  2. Get output data Y[n]       //
        //  3. Compare with Golden Y[n]   //
        //                                //
        //********************************//
        task FIR_X_Y_stream;
                begin

                        reset_capture_expect();

                        soc_to_fpga_axis_expect_count = 0;
                        Golden2Axis();
                        $display($time, "=> wait for soc_to_fpga_axis_event");
                @(soc_to_fpga_axis_event);
                $display($time, "=> soc_to_fpga_axis_expect_count = %d", soc_to_fpga_axis_expect_count);
                $display($time, "=> soc_to_fpga_axis_captured_count = %d", soc_to_fpga_axis_captured_count);

                        check_cnt = check_cnt + 1;
                        if ( soc_to_fpga_axis_expect_count != DATA_LENGTH) begin
                    $display($time, "=> [ERROR] soc_to_fpga_axis_expect_count = %d, soc_to_fpga_axis_captured_count = %d", soc_to_fpga_axis_expect_count, soc_to_fpga_axis_captured_count);
                                error_cnt = error_cnt + 1;
                        end
                        else
                    $display($time, "=> [PASS] soc_to_fpga_axis_expect_count = %d, soc_to_fpga_axis_captured_count = %d", soc_to_fpga_axis_expect_count, soc_to_fpga_axis_captured_count);

                for(j=0; j<DATA_LENGTH; j=j+1)begin
                                check_cnt = check_cnt + 1;
                    if (soc_to_fpga_axis_expect_value[j] != soc_to_fpga_axis_captured[j] ) begin
                        $display($time, "=> [ERROR] index id=%d, soc_to_fpga_axis_expect_value[%d] = %x, soc_to_fpga_axis_captured[%d]  = %x", j, j, soc_to_fpga_axis_expect_value[j], j, soc_to_fpga_axis_captured[j]);
                                        error_cnt = error_cnt + 1;
                                end
                                else
                        $display($time, "=> [PASS] index id=%d, soc_to_fpga_axis_expect_value[%d] = %x, soc_to_fpga_axis_captured[%d]  = %x", j, j, soc_to_fpga_axis_expect_value[j], j, soc_to_fpga_axis_captured[j]);
                    end

                                #200;
                                $display("-------------------------------------------");
                                $display("      Finish FIR data X, Y stream data      ");
                                $display("-------------------------------------------");
                    end
        endtask
```

這個 task 將啟動 FIR 的串流以開始數據 X 傳輸，它將數據從 0 到 63 通過 AXI-Stream 進行傳輸。

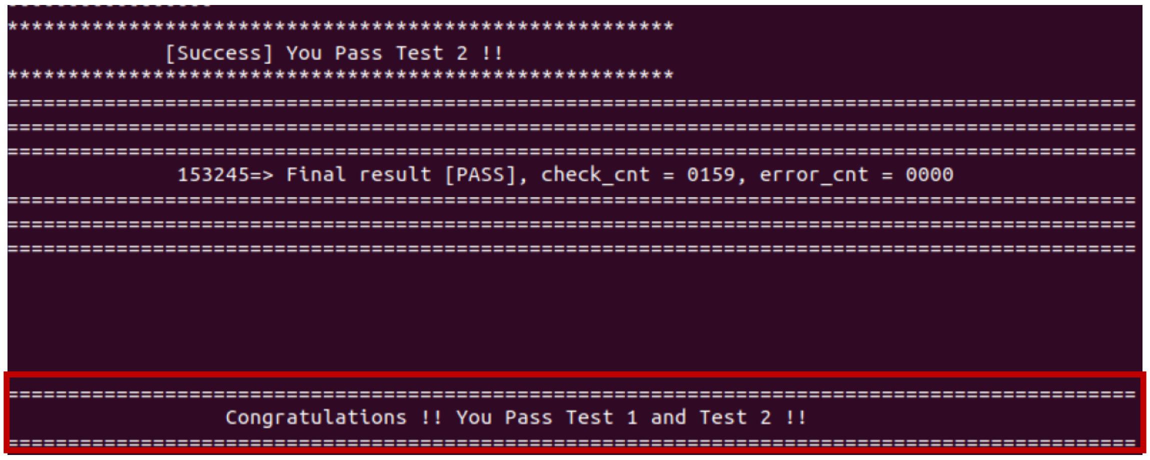
Question 6. Briefly describe how you get output Y data in testbench, and how to do comparison with golden values.



```verilog
    //********************************//
    //                                //
    //       Golden2AxiSwitch         //
    //                                //
    //********************************//
    task Golden2Axis;

            reg [31:0]golden_y;

            begin
                    $display("Start Golden Y to AXIS");
                    @ (posedge fpga_coreclk);
                    fpga_as_is_tready <= 1;

                    for(j=0; j<DATA_LENGTH; j=j+1)begin
                            golden_y = 0*j+(-10)*(((j-1)<0)? 0:(j-1))+(-9)*(((j-2)<0)? 0:(j-2))+23*(((j-3)<0)? 0:(j-3))+56*(((j-4)<0)? 0:(j-4))+63*(((j-5)<0)? 0:(j-5))+56*(((j-6)<0)? 0:(j-6))+23*(((j-7)<0)? 0:(j-7))+(-9)*(((j-8)<0)? 0:(j-8))+(-10)*(((j-9)<0)? 0:(j-9))+0*(((j-10)<0)? 0:(j-10));
                            `ifdef USER_PROJECT_SIDEBAND_SUPPORT
                                    if (j==DATA_LENGTH-1) begin
                                            soc_to_fpga_axis_expect_value[soc_to_fpga_axis_expect_count] <= {j[4:0], 4'b0000, 4'b0000, 1'b1, golden_y};
                                    end
                                    else begin
                                            soc_to_fpga_axis_expect_value[soc_to_fpga_axis_expect_count] <= {j[4:0], 4'b0000, 4'b0000, 1'b0, golden_y};
                                    end
                            `else
                                    if (j==DATA_LENGTH-1) begin
                                            soc_to_fpga_axis_expect_value[soc_to_fpga_axis_expect_count] <= {4'b0000, 4'b0000, 1'b1, golden_y};
                                    end
                                    else begin
                                            soc_to_fpga_axis_expect_value[soc_to_fpga_axis_expect_count] <= {4'b0000, 4'b0000, 1'b0, golden_y};
                                    end
                            `endif
                            soc_to_fpga_axis_expect_count <= soc_to_fpga_axis_expect_count+1;
                            fpga_axis_req_v2(j, TID_DN_UP, 0);
                    end

                    $display($time, "=> Golden Y to AXIS done !!");
            end
    endtask
```

Y[n]的傳輸方式與 **Question 5.**雷同，都是透過 AXI-Stream 進行傳輸，差異是會增加一個重設步驟，以確保資料從 0 到 63 而不是從 64 開始，最後再與框起來的部分計算出的 golden 進行比較。

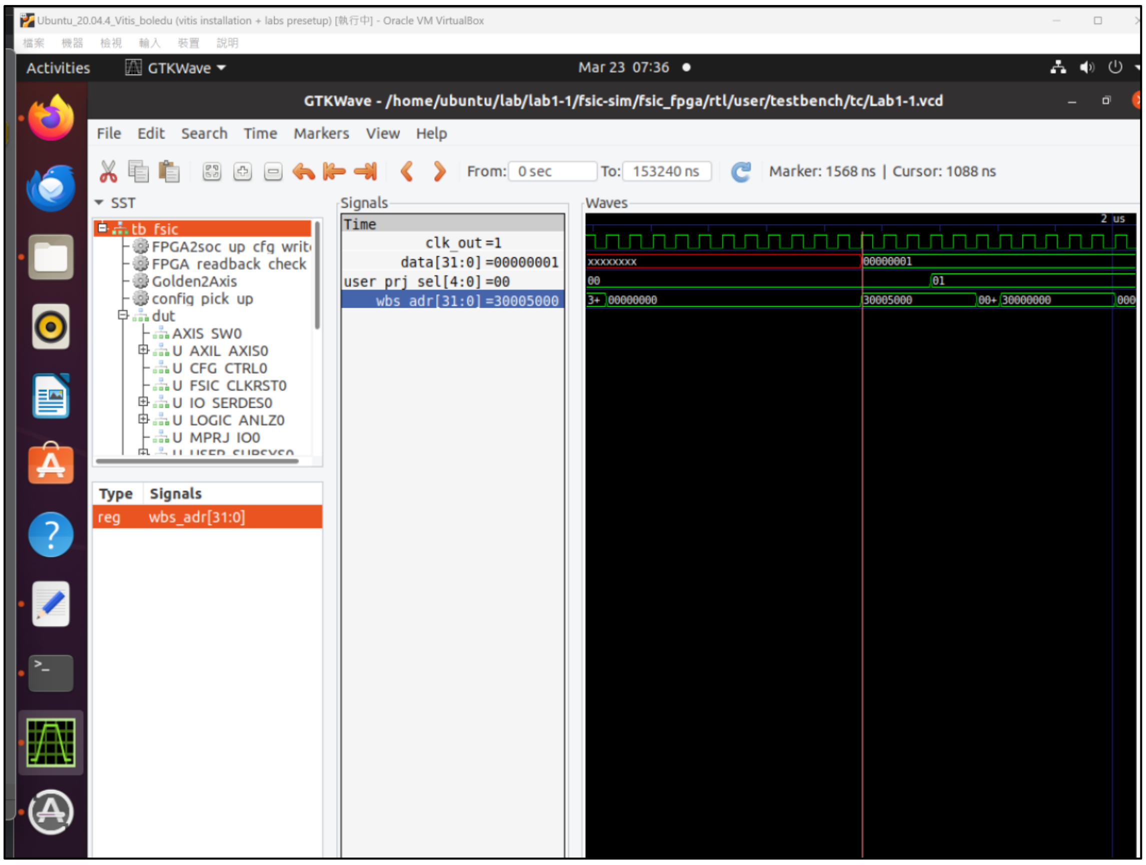
Question 7. Screenshot simulation results printed on screen, to show that your Test#1 & Test#2 complete successfully



```
*******************************************************
             [Success] You Pass Test 2 !!
*******************************************************
==========================================================================================
==========================================================================================
==========================================================================================
              153245=> Final result [PASS], check_cnt = 0159, error_cnt = 0000
==========================================================================================
==========================================================================================
==========================================================================================



==========================================================================================
                  Congratulations !! You Pass Test 1 and Test 2 !!
==========================================================================================
```

Question 8. Screenshot simulation waveform:

• Configuration cycle (when we program [‘h3000_5000] = 32’h01, signal user_prj_sel changes accordingly)

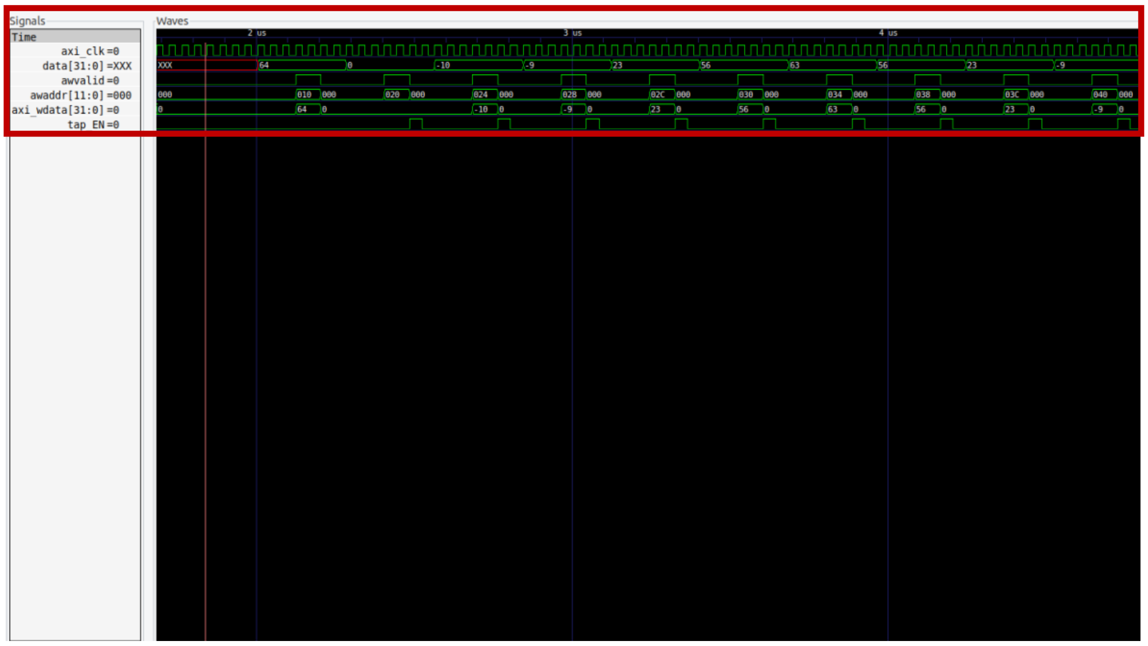


```
GTKWave - /home/ubuntu/lab/lab1-1/fsic-sim/fsic_fpga/rtl/user/testbench/tc/Lab1-1.vcd
From: 0 sec   To: 153240 ns   Marker: 1568 ns | Cursor: 1088 ns

Time
clk_out =1
data[31:0] =00000001          xxxxxxxx | 00000001
user_prj_sel[4:0] =00         00 | 01
wbs_adr[31:0] =30005000       3+ | 00000000 | 30005000 | 00+ | 30000000 | 000
```

program [‘h3000_5000] = 32’h01，user_prj_sel 過幾個 cycle 會

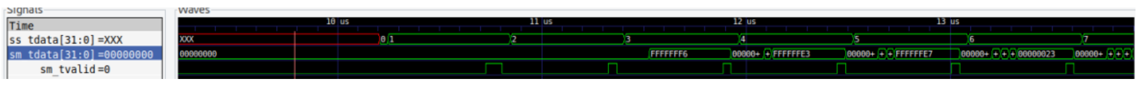
拉到 1。

- AXI-Lite transaction cycles (feed in tap parameters, data_length)

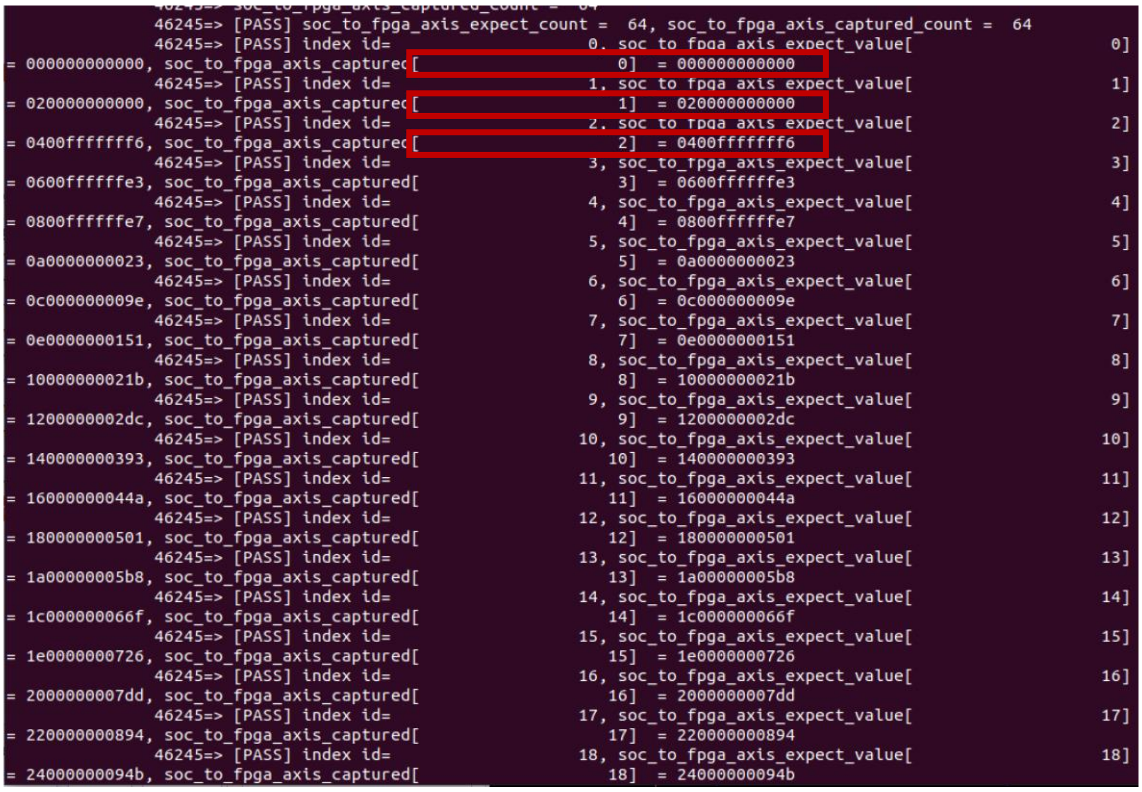


當 awvalid 拉到 1，當下的 data 會被寫入指定的 address 中，由 axi_wdata 可知道當下寫入的 data 符合預期。

- Stream-in, Stream-out



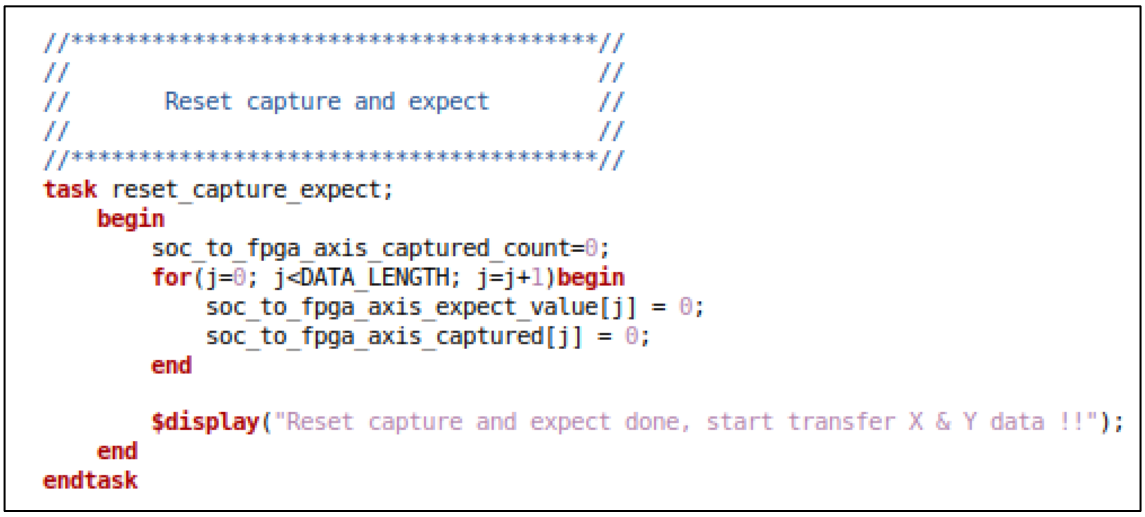
ss_tdata 表示 index:0~63 即 stream-in，當 sm_tvalid 拉到 1 時，會將當前的值作為 stream-out。以下為 golden 與 stream-out 值得比較：



```
              46245=> [PASS] soc_to_fpga_axis_expect_count =  64, soc_to_fpga_axis_captured_count =  64
              46245=> [PASS] index id=                   0, soc_to_fpga_axis_expect_value[          0]
= 000000000000, soc_to_fpga_axis_captured[                 0]  = 000000000000
              46245=> [PASS] index id=                   1, soc_to_fpga_axis_expect_value[          1]
= 020000000000, soc_to_fpga_axis_captured[                 1]  = 020000000000
              46245=> [PASS] index id=                   2, soc_to_fpga_axis_expect_value[          2]
= 0400fffffff6, soc_to_fpga_axis_captured[                 2]  = 0400fffffff6
              46245=> [PASS] index id=                   3, soc_to_fpga_axis_expect_value[          3]
= 0600ffffffe3, soc_to_fpga_axis_captured[                 3]  = 0600ffffffe3
              46245=> [PASS] index id=                   4, soc_to_fpga_axis_expect_value[          4]
= 0800ffffffe7, soc_to_fpga_axis_captured[                 4]  = 0800ffffffe7
              46245=> [PASS] index id=                   5, soc_to_fpga_axis_expect_value[          5]
= 0a0000000023, soc_to_fpga_axis_captured[                 5]  = 0a0000000023
              46245=> [PASS] index id=                   6, soc_to_fpga_axis_expect_value[          6]
= 0c000000009e, soc_to_fpga_axis_captured[                 6]  = 0c000000009e
              46245=> [PASS] index id=                   7, soc_to_fpga_axis_expect_value[          7]
= 0e0000000151, soc_to_fpga_axis_captured[                 7]  = 0e0000000151
              46245=> [PASS] index id=                   8, soc_to_fpga_axis_expect_value[          8]
= 10000000021b, soc_to_fpga_axis_captured[                 8]  = 10000000021b
              46245=> [PASS] index id=                   9, soc_to_fpga_axis_expect_value[          9]
= 1200000002dc, soc_to_fpga_axis_captured[                 9]  = 1200000002dc
              46245=> [PASS] index id=                  10, soc_to_fpga_axis_expect_value[         10]
= 140000000393, soc_to_fpga_axis_captured[                10]  = 140000000393
              46245=> [PASS] index id=                  11, soc_to_fpga_axis_expect_value[         11]
= 16000000044a, soc_to_fpga_axis_captured[                11]  = 16000000044a
              46245=> [PASS] index id=                  12, soc_to_fpga_axis_expect_value[         12]
= 180000000501, soc_to_fpga_axis_captured[                12]  = 180000000501
              46245=> [PASS] index id=                  13, soc_to_fpga_axis_expect_value[         13]
= 1a00000005b8, soc_to_fpga_axis_captured[                13]  = 1a00000005b8
              46245=> [PASS] index id=                  14, soc_to_fpga_axis_expect_value[         14]
= 1c000000066f, soc_to_fpga_axis_captured[                14]  = 1c000000066f
              46245=> [PASS] index id=                  15, soc_to_fpga_axis_expect_value[         15]
= 1e0000000726, soc_to_fpga_axis_captured[                15]  = 1e0000000726
              46245=> [PASS] index id=                  16, soc_to_fpga_axis_expect_value[         16]
= 2000000007dd, soc_to_fpga_axis_captured[                16]  = 2000000007dd
              46245=> [PASS] index id=                  17, soc_to_fpga_axis_expect_value[         17]
= 220000000894, soc_to_fpga_axis_captured[                17]  = 220000000894
              46245=> [PASS] index id=                  18, soc_to_fpga_axis_expect_value[         18]
= 24000000094b, soc_to_fpga_axis_captured[                18]  = 24000000094b
```

Question 9. Debug experience (bug found, and how to fix it)

由於 Testbench 較為複雜，有遇到一個狀況就是當 test01 與 test02 依序執行時，Stream-in, Stream-out 遇到 testing failure，主要的原因是 stream data 的 reg 並未 reset。當下多寫個 reset register function 就解決了。



```
//*****************************************//
//                                         //
//        Reset capture and expect         //
//                                         //
//*****************************************//
task reset_capture_expect;
    begin
        soc_to_fpga_axis_captured_count=0;
        for(j=0; j<DATA_LENGTH; j=j+1)begin
            soc_to_fpga_axis_expect_value[j] = 0;
            soc_to_fpga_axis_captured[j] = 0;
        end

        $display("Reset capture and expect done, start transfer X & Y data !!");
    end
endtask
```