Advanced SoC HW1

學號: 111061605　　姓名:周聖平

一、 Show the code that you use to program configuration address [‘h3000_5000].

透過 wishbone 寫值到 CC_BASE = [‘h3000_5000]

```
task soc_cc_cfg_prj_sel_write;
    input [4:0] data;
    begin
        $display($time,,"======= soc_cc_cfg_prj_sel_write =======");
        @ (posedge soc_coreclk);
        wbs_adr <= CC_BASE;

        wbs_wdata <= {27'd0, data};
        wbs_sel <= 4'hf;
        wbs_cyc <= 1'b1;
        wbs_stb <= 1'b1;
        wbs_we <= 1'b1;

        $display($time,,"soc_cc_cfg_prj_sel_write: waiting wbs_ack");
        @(posedge soc_coreclk);
        while(wbs_ack==0) begin
            @(posedge soc_coreclk);
        end

        // release ctrl
        @(posedge soc_coreclk);
        wbs_cyc <= 1'b0;
        wbs_stb <= 1'b0;

    end
endtask
```

二、 Explain why “By programming configuration address [‘h3000_5000], signal user_prj_sel[4:0] will change accordingly”?

前面 20 位去決定資料的目的地(AA、AS、IS、LA、UP、CC)。

CC_xx_enable 決定了要與哪個元件進行溝通，不是 wstrb。

```
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
```

當目的地為 CC 時，將相關的控制信號拉起來。

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

Addr[11:0]為 12’ h000 時，將資料賦值到 user_prj_sel。

```
////////////////////////////////////////////
// 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
```

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

首先，透過範例 code 的 soc_up_cfg_write 傳送 ap_start=1、data_length、tap 給 FIR(user_prj_1)。

接下來，再由 soc_aa_cfg_write 傳送 1 到 Mailbox 去通知 FIR 可以開始測試。

```
task fir_init_by_soc;
    integer i;
    begin
        $display($time,,"======= fir_init_by_soc =======");
        // send parameter to UP
        // ap_start & tap & data_length
        $display($time,,"fir_init_by_soc: sending data_len");
        soc_up_cfg_write(DATA_LEN_ADDR_OFFSET, 4'b1111, FIR_TEST_LEN);

        for(i=0; i<11; i=i+1) begin
            $display($time,,"fir_init_by_soc: sending tap[%2d]=%2d", i, taps[i]);
            soc_up_cfg_write(TAP_ADDR_OFFSET+4*i, 4'b1111, taps[i]);
        end

        $display($time,,"fir_init_by_soc: sending ap_start");
        soc_up_cfg_write(AP_STATUS_ADDR_OFFSET, 4'b1111, 32'd1);

        // send start signal to mailbox FPGA
        $display($time,,"fir_init_by_soc: sending start to mailbox to tell FPGA");
        soc_aa_cfg_write(AA_MailBox_Reg_Offset, 4'b1111, 32'd1);
    end
endtask
```

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

透過範例 code 的 fpga_axilite_write 傳送 ap_start=1、data_length、tap 給 FIR(user_prj_1)。

```
task fir_init_by_fpga;
    integer i;
    begin
        $display($time,,"======= fir_init_by_fpga =======");
        // send parameter to UP
        // ap_start & tap & data_length
        $display($time,,"fir_init_by_fpga: sending data_len");
        fpga_axilite_write(FPGA_to_SOC_UP_BASE+DATA_LEN_ADDR_OFFSET, 4'b1111, FIR_TEST_LEN);

        for(i=0; i<11; i=i+1) begin
            $display($time,,"fir_init_by_fpga: sending tap[%2d] = %2d", i, taps[i]);
            fpga_axilite_write(FPGA_to_SOC_UP_BASE+TAP_ADDR_OFFSET+4*i, 4'b1111, taps[i]);
        end

        $display($time,,"fir_init_by_fpga: sending ap_start");
        fpga_axilite_write(FPGA_to_SOC_UP_BASE+AP_STATUS_ADDR_OFFSET, 4'b1111, 32'd1);
    end
endtask
```

fpga_axilite_write 會先寫入到 AA 中，再由 AA 根據目的地轉傳到 UP

**** 此時，資料會透過 axilite 傳送到 UP。如果 FPGA 要透過 axis 傳送到 UP 的話，要參考範例 code 中的 fpga_axis_req。

**** TUSER_AXILITE_WRITE 第一 T 是傳送目的地跟 BE，第二 T 才是傳送資料。

```
task fpga_axilite_write;
    input [27:0] address;
    input [3:0] BE;
    input [31:0] data;
    begin
        fpga_as_is_tdata <= (BE<<28) + address; //for axilite write address phase
        //$strobe($time,, "=> fpga_as_is_tdata in address phase = %x", 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
        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

        fpga_as_is_tdata <=  data; //for axilite write data phase
        `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
        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
        fpga_as_is_tvalid <= 0;

    end
endtask
```

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

在每輪測試時，會有一個 Driver 專門去傳送 X 到 FPGA。

```
task testFIR;
    input init_by;
    begin
        $display($time,,"======= testFIR =======");
        testFIR_fsic_init();

        // let CC chosen user_prj_1
        soc_cc_cfg_prj_sel_write(5'h1);

        // initial fir parameter
        case(init_by)
            INIT_BY_FPGA: fir_init_by_fpga();
            INIT_BY_SOC:  begin
                fir_init_by_soc();
                fir_fpga_wait_start();
            end
        endcase

        // fir start to feed X & recv Y
        -> start_fir_driver;
        -> start_fir_receiver;

        // wait whole test finish
        @(finish_fir_test);
        // read ap_done
        fir_read_ap_done();
    end
endtask
```

Drive 會去傳送設定好的資料(x=1~64) 到 FIR。

```
integer driver_i;
initial begin
    while(1) begin
        @(start_fir_driver);
        // feed test data
        for(driver_i=1; driver_i<=FIR_TEST_LEN; driver_i=driver_i+1) begin
            $display($time,,"fpga_send_x_to_fir: sending x = %2d to FIR", i);
            fpga_send_x_to_fir(driver_i, (driver_i == FIR_TEST_LEN));
        end
    end
end
```

透過 FPGA Side 的 AXIS 傳送資料到 User Prj。

TID_DN_UP、TUSER_AXIS 代表資料會透過 AXIS 的端口傳送到 UP

```
task fpga_send_x_to_fir;
    input [31:0] data;
    input last;
    begin
        fpga_as_is_tdata <= data;
        `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_UP;
        fpga_as_is_tuser <= TUSER_AXIS;
        fpga_as_is_tlast <= last;
        fpga_as_is_tvalid <= 1;

        @ (posedge fpga_coreclk);
        $display($time,,"fpga_to_UP_axis_write: waiting fpga_is_as_tready");
        while (fpga_is_as_tready == 0) begin        // wait util fpga_is_as_tready == 1 then change data
            @ (posedge fpga_coreclk);
        end
        $display($time,,"fpga_to_UP_axis_write: recv fpga_is_as_tready");
    end
endtask
```

(參考範例 code: fpga_axis_req)

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

當 receiver 被觸發時，就會去等待 FIR 傳送 Y 到 FPGA 的資料。接收到資料 Y 後，Receiver 會將 Y 與 golden_data 去做比對，並將 ERROR 記錄下來。

```
integer receiver_i;
initial begin
    while(1) begin
        receiver_i = 0;
        fir_error = 0;
        @(start_fir_receiver);
        while( receiver_i < FIR_TEST_LEN ) begin
            $display($time,,"fpga_recv_y_from_fir: waiting fpga_recv_soc_data_event");
            @(fpga_recv_soc_data_event);
            recv_y = fpga_is_as_tdata;
            if(recv_y !== golden_list[receiver_i]) begin
                $display($time,,"testFIR: [ERROR] round = %2d, recv_y = %2d, gold_data = %2d", receiver_i, recv_y, golden_list[receiver_i]);
                fir_error = fir_error + 1;
            end
            else begin
                $display($time,,"testFIR: [PASS] round = %2d, recv_y = %2d, gold_data = %2d", receiver_i, recv_y, golden_list[receiver_i]);
            end

            receiver_i = receiver_i + 1;
        end

        scoreboard_display_result(fir_error);
        ->finish_fir_test;
    end
end
```

當 FPGA 收到從 UP 傳來的資料時，就會觸發事件 fpga_recv_soc_data_event。

```
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
                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 bl
                #0 -> fpga_recv_soc_data_event;
            end
        `else
            if (fpga_is_as_tvalid == 1 && fpga_is_as_tid == TID_UP_UP && fpga_is_as_tuser == TUSER_AXIS) begin
                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
                #0 -> fpga_recv_soc_data_event;
            end
        `endif
    end
end
```

七、 Screenshot simulation results printed on screen, to show that your Test#1 & Test#2 complete successfully

Test #1 : 64 筆測資通過 (x=1~64)

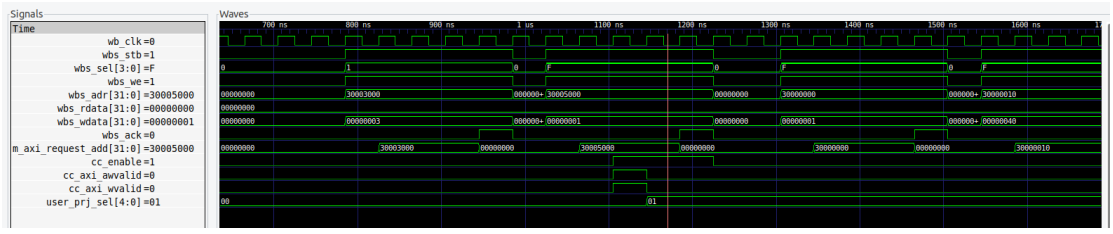
```
39945 testFIR: [PASS] round = 60, recv y = 10248, gold_data = 10248
39945 fpga_recv_y_from_fir: waiting fpga_recv_soc_data_event
40505 testFIR: [PASS] round = 61, recv y = 10431, gold_data = 10431
40505 fpga_recv_y_from_fir: waiting fpga_recv_soc_data_event
41065 testFIR: [PASS] round = 62, recv y = 10614, gold_data = 10614
41065 fpga_recv_y_from_fir: waiting fpga_recv_soc_data_event
41625 testFIR: [PASS] round = 63, recv y = 10797, gold_data = 10797
41625 ======= scoreboard =======
41625 ============================
41625      Congratulations !!!
41625 ============================
41945 => soc_up_cfg_read : wbs_adr=30000000, wbs_sel=1111
41945 => soc wishbone read data result : send soc_cfg_read_event
41945 => soc_up_cfg_read : got soc_cfg_read_event
41945 ============= FIR Done =============
```

Test #2 : 64 筆測資通過 (x=1~64)

```
82285 fpga_recv_y_from_fir: waiting fpga_recv_soc_data_event
82845 testFIR: [PASS] round = 58, recv y = 9882, gold_data = 9882
82845 fpga_recv_y_from_fir: waiting fpga_recv_soc_data_event
83405 testFIR: [PASS] round = 59, recv y = 10065, gold_data = 10065
83405 fpga_recv_y_from_fir: waiting fpga_recv_soc_data_event
83965 testFIR: [PASS] round = 60, recv y = 10248, gold_data = 10248
83965 fpga_recv_y_from_fir: waiting fpga_recv_soc_data_event
84525 testFIR: [PASS] round = 61, recv y = 10431, gold_data = 10431
84525 fpga_recv_y_from_fir: waiting fpga_recv_soc_data_event
85085 testFIR: [PASS] round = 62, recv y = 10614, gold_data = 10614
85085 fpga_recv_y_from_fir: waiting fpga_recv_soc_data_event
85645 testFIR: [PASS] round = 63, recv y = 10797, gold_data = 10797
85645 ======= scoreboard =======
85645 ============================
85645      Congratulations !!!
85645 ============================
86005 => soc_up_cfg_read : wbs_adr=30000000, wbs_sel=1111
86005 => soc wishbone read data result : send soc_cfg_read_event
86005 => soc_up_cfg_read : got soc_cfg_read_event
86005 ============= FIR Done =============
```

八、 Screenshot simulation waveform:

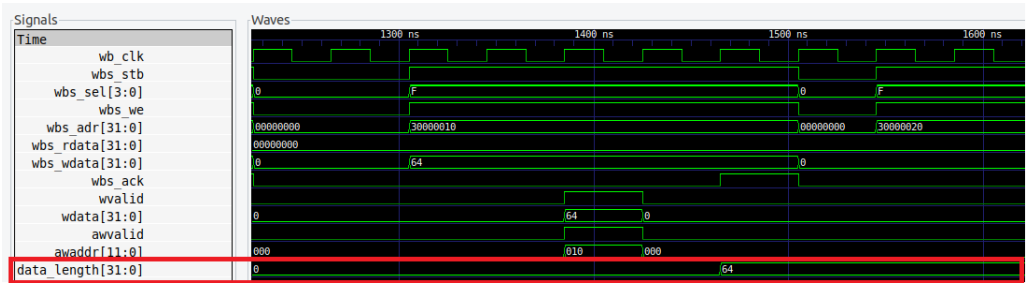
- Configuration cycle (when we program [‘h3000_5000] = 32’ h01, signal user_prj_sel changes accordingly)



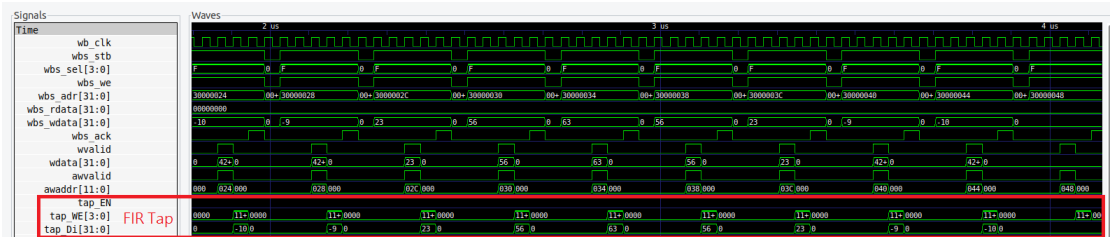
透過 Wishbone 寫入 CC 中的 3000_5000 (user_prj_sel)

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

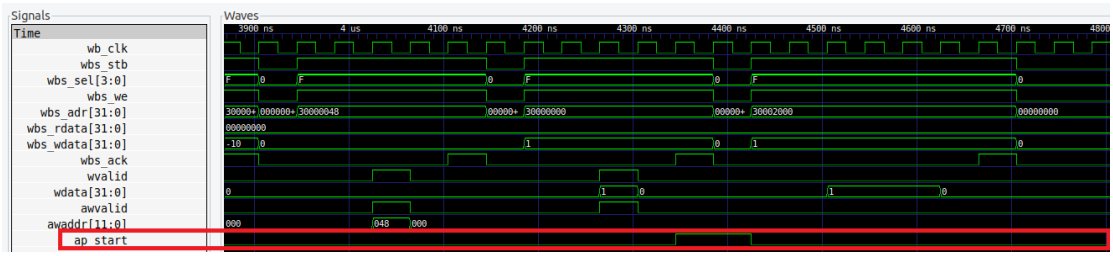
  Data_length



  Tap

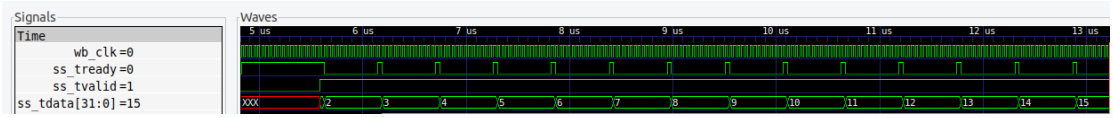


  ap_start

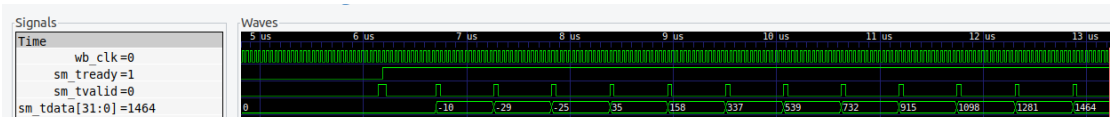


- Stream-in, Stream-out

  Stream-in



  Stream-out



九、 Debug experience

Bug:

Golden_data.dat 的路徑寫錯，導致 golden_data 的值都為 x，而檢查的時候因為是用 != 無法判斷 unknown 狀態，因此判斷也跟著出錯，導致最後出來的結果為正確，而實際上是錯誤的。

Fix:

```
// GET GOLDEN DATA
initial begin
    golden = $fopen("/home/ubuntu/advanced_soc/fsic-sim/fsic_fpga/rtl/user/testbench/out_gold.dat","r");
    if(golden == 0) begin
        $display($time,, "FAIL to Open out_gold");
        $finish;
    end

    $display($time,, "================== GET GOLDEN DATA ================");
    for(m=0; m<(FIR_TEST_LEN*3-1); m=m+1) begin
        if( !$fscanf(golden,"%d", golden_list[m])) begin
            $display($time,, "FAIL to Get golden data");
            $finish;
        end
        else begin
            $display($time,, "Golden_data[%2d] = %2d", m, golden_list[m]);
        end
    end
end
```

當 fopen 開啟失敗時，就中斷測試並 print 出來。

```
if(recv_y !== golden_list[receiver_i]) begin
    $display($time,,"testFIR: [ERROR] round = %2d, recv y = %2d, gold_data = %2d", receiver_i, recv_y, golden_list[receiver_i]);
    fir_error = fir_error + 1;
end
else begin
    $display($time,,"testFIR: [PASS] round = %2d, recv y = %2d, gold_data = %2d", receiver_i, recv_y, golden_list[receiver_i]);
end
```

在檢查結果時，應該使用 ‘!==’ or ‘===’ 。