Bridge of Life Education

Advanced SOC Design
Lab 1 – FSIC-SIM

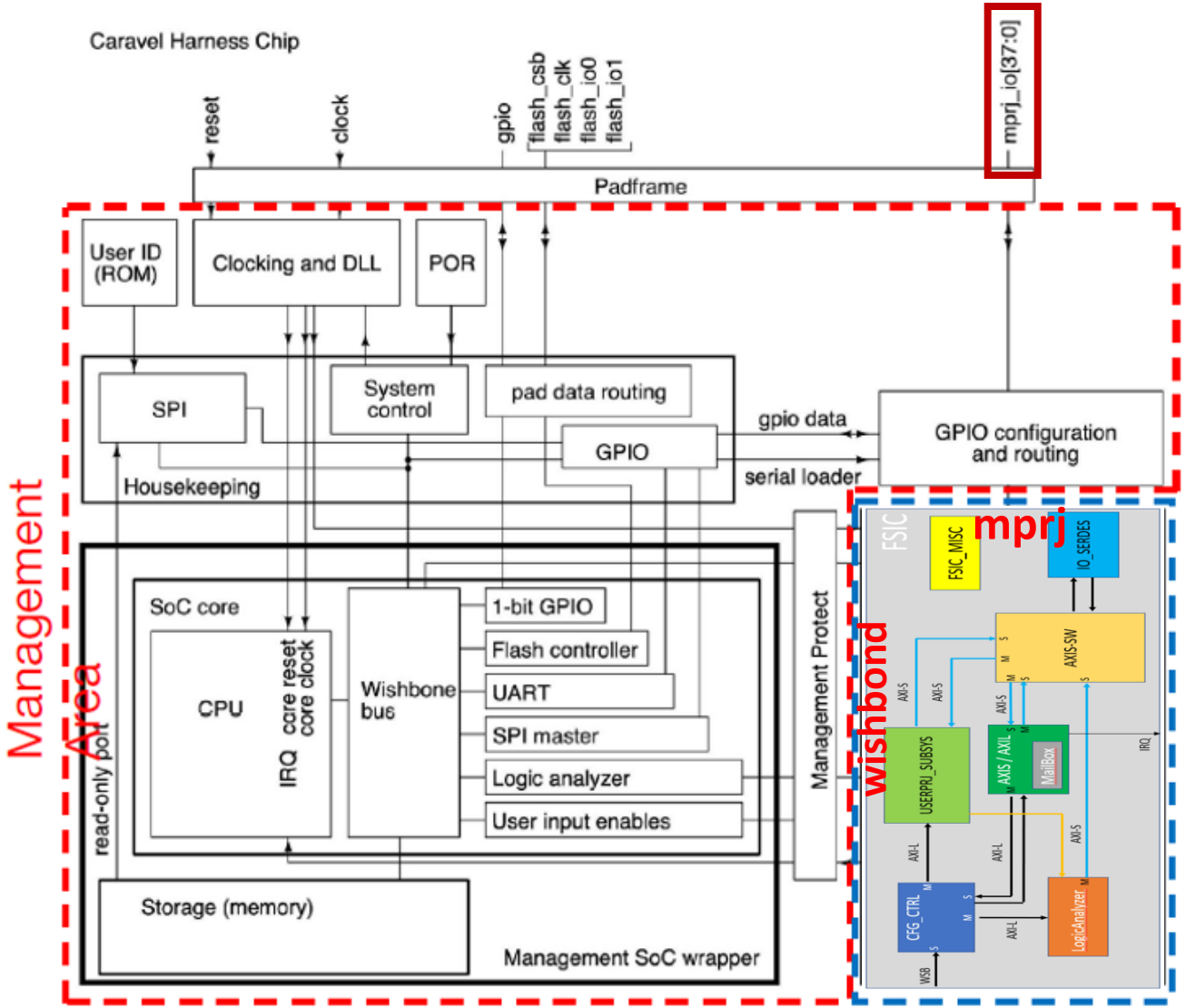
(**individual** work)

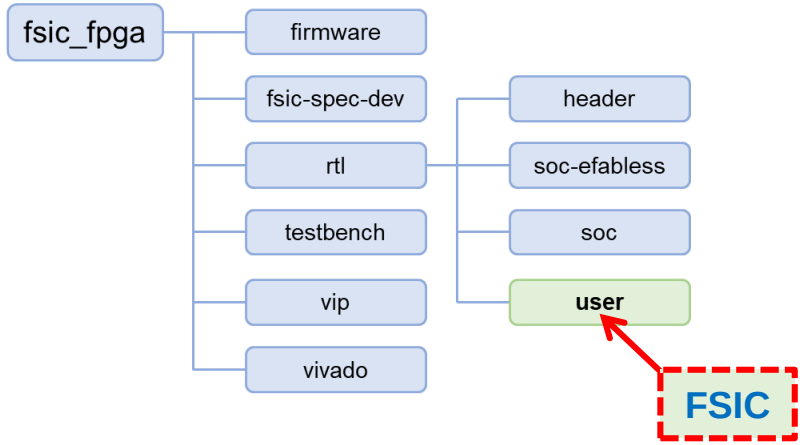
Jiin Lai

FSIC-SIM - FSIC Simulation

A simulation environment dedicated for FSIC design verification

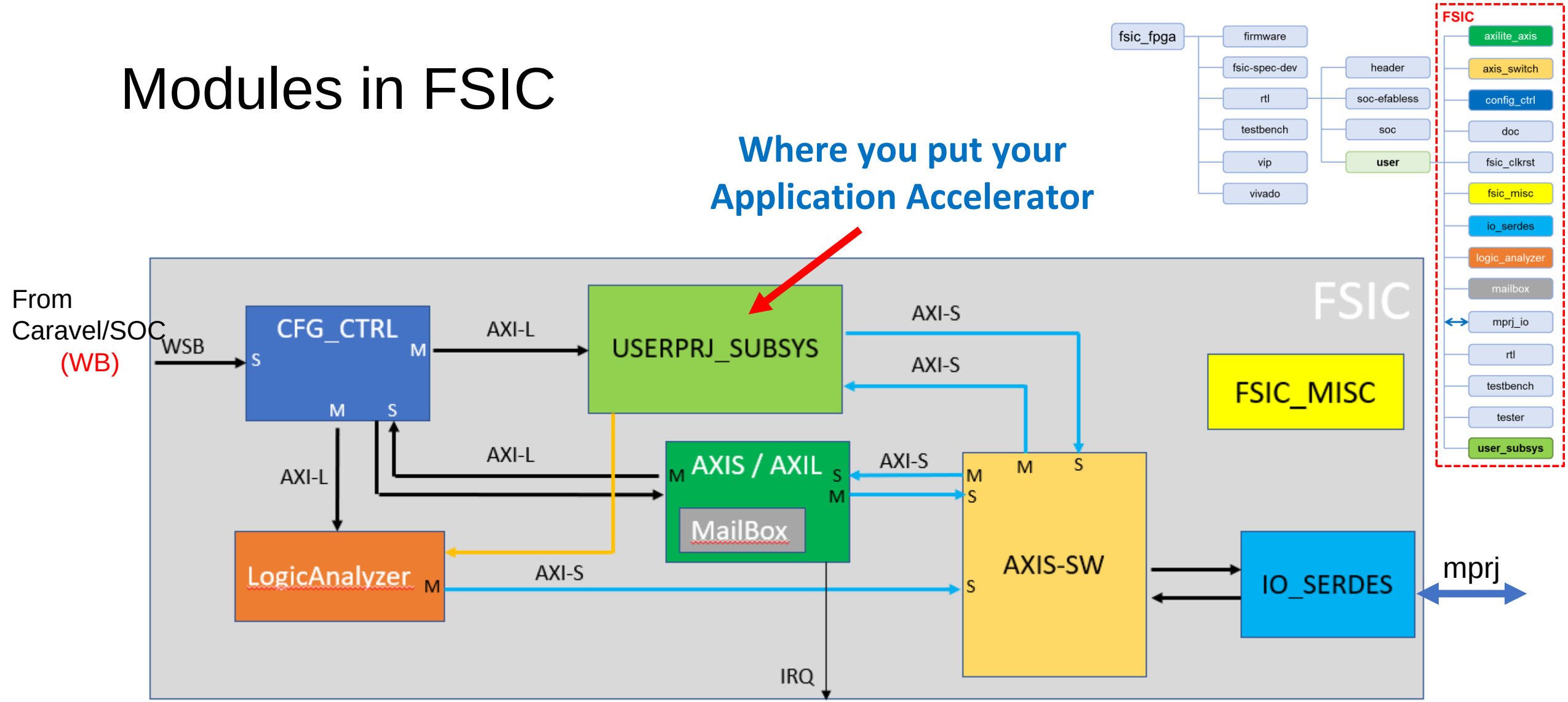
Location in Caravel SOC - in User Project Wrapper





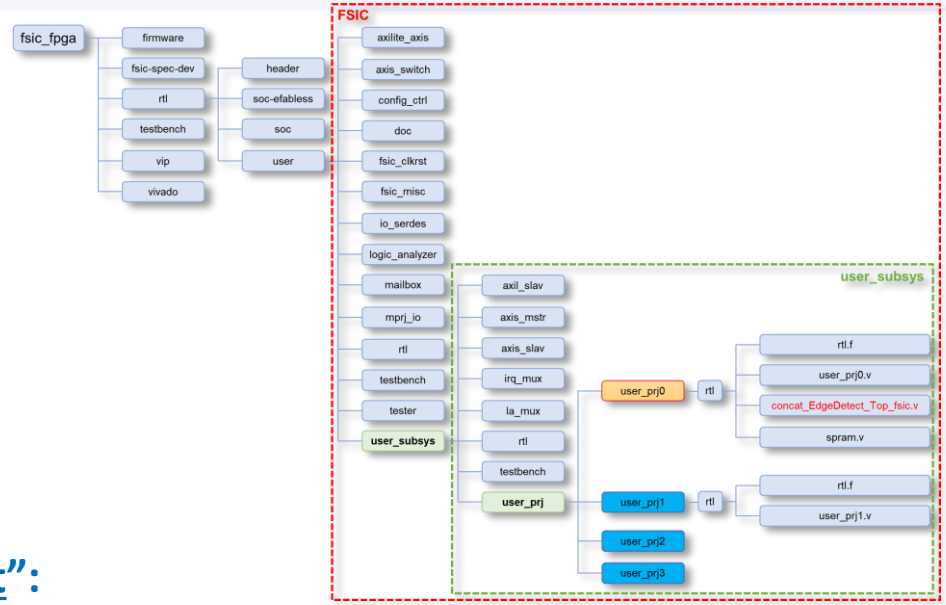
- Put **FSIC** into Caravel SoC **user project area**

Modules in FSIC



in User_Subsys:

[4: 0] user_prj_sel
(defined in config_ctrl module, **32'h3000_5000**)



user_subsys

Mux out the selected user_proj

- axil_slav
- axis_mstr
- axis_slav
- irq_mux
- a_mux

user_prj0

user_prj0.v
concat_EdgeDetect_Top_fsic.v
spram.v

- user_prj1
- user_prj2
- user_prj3

in "filelist":

user_subsys

../../user_subsys/rtl/user_subsys.v
../../user_subsys/axil_slav/rtl/axil_slav.v
../../user_subsys/axis_mstr/rtl/axis_mstr.v
../../user_subsys/axis_slav/rtl/axis_slav.v
../../user_subsys/irq_mux/rtl/irq_mux.v
../../user_subsys/la_mux/rtl/la_mux.v
../../user_subsys/user_prj/user_prj0/rtl/user_prj0.v
../../user_subsys/user_prj/user_prj0/rtl/concat_EdgeDetect_Top_fsic.v
../../user_subsys/user_prj/user_prj0/rtl/spram.v
../../user_subsys/user_prj/user_prj1/rtl/user_prj1.v
../../user_subsys/user_prj/user_prj2/rtl/user_prj2.v
../../user_subsys/user_prj/user_prj3/rtl/user_prj3.v

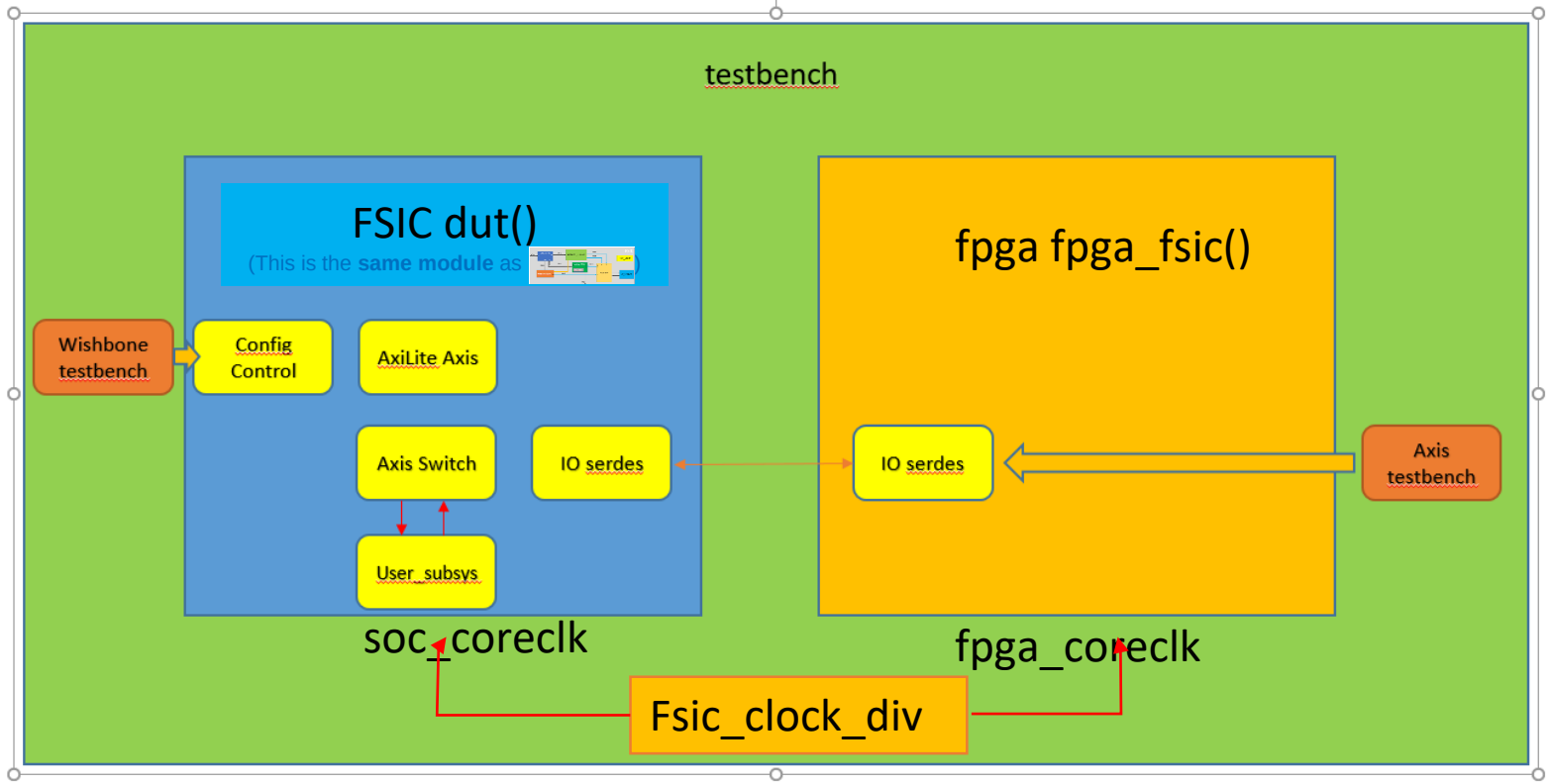
Currently, we only have user_prj0 (EdgeDetect IP generated from Catapult HLS)

Testbench

run_xsim :

```
xvlog -sv -i ../ ../tb_fsic.v -f filelist
xelab tb_fsic -debug typical --snapshot tb_fsic_xelab --timescale 1ns/1ns
xsim tb_fsic_xelab --tclbatch log_wave.tcl
```

FSIC module testbench in fsic_tony



in "filelist":

fsic

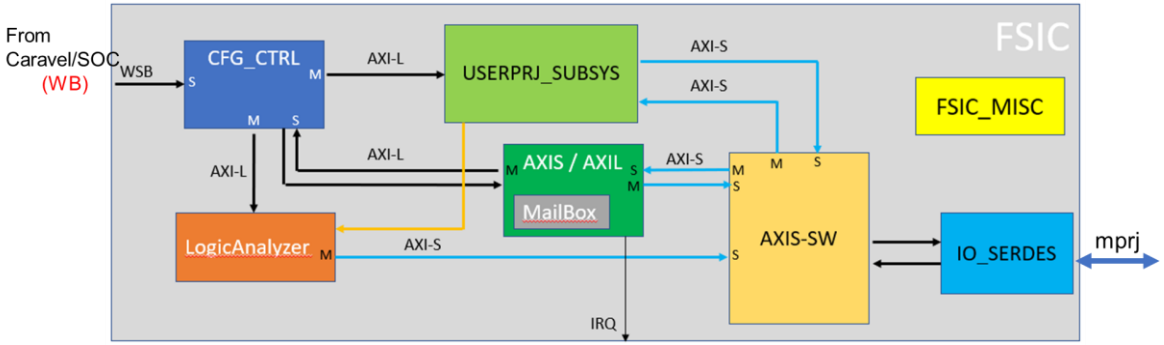
```
../fpga.v
../fsic_clock.v
../../rtl/fsic.v
../../axilite_axis/rtl/axi_ctrl_logic.sv
../../axilite_axis/rtl/axil_axis.sv
../../axilite_axis/rtl/axilite_master.sv
../../axilite_axis/rtl/axilite_slave.sv
../../axilite_axis/rtl/axis_master.sv
../../axilite_axis/rtl/axis_slave.sv
../../axis_switch/rtl/sw_caravel.v
../../config_ctrl/rtl/config_ctrl.v
../../fsic_clkrst/rtl/fsic_clkrst.v
../../io_serdes/rtl/fsic_coreclk_phase_cnt.v
../../io_serdes/rtl/fsic_io_serdes_rx.v
../../io_serdes/rtl/io_serdes.v
../../logic_analyzer/rtl/LogicAnalyzer.v
../../logic_analyzer/rtl/Sram.v
../../mprj_io/rtl/fsic_mprj_io.v
```

user_subsys

```
../../user_subsys/rtl/user_subsys.v
../../user_subsys/axil_slav/rtl/axil_slav.v
../../user_subsys/axis_mstr/rtl/axis_mstr.v
../../user_subsys/axis_slav/rtl/axis_slav.v
../../user_subsys/irq_mux/rtl/irq_mux.v
../../user_subsys/la_mux/rtl/la_mux.v
../../user_subsys/user_prj/user_prj0/rtl/user_prj0.v
../../user_subsys/user_prj/user_prj0/rtl/concat_EdgeDetect_Top_fsic.v
../../user_subsys/user_prj/user_prj0/rtl/spram.v
../../user_subsys/user_prj/user_prj1/rtl/user_prj1.v
../../user_subsys/user_prj/user_prj2/rtl/user_prj2.v
../../user_subsys/user_prj/user_prj3/rtl/user_prj3.v
```

HackMD:
https://hackmd.io/@TonyHo/rk6Siw0k6
https://github.com/TonyHo722/fsic_tony

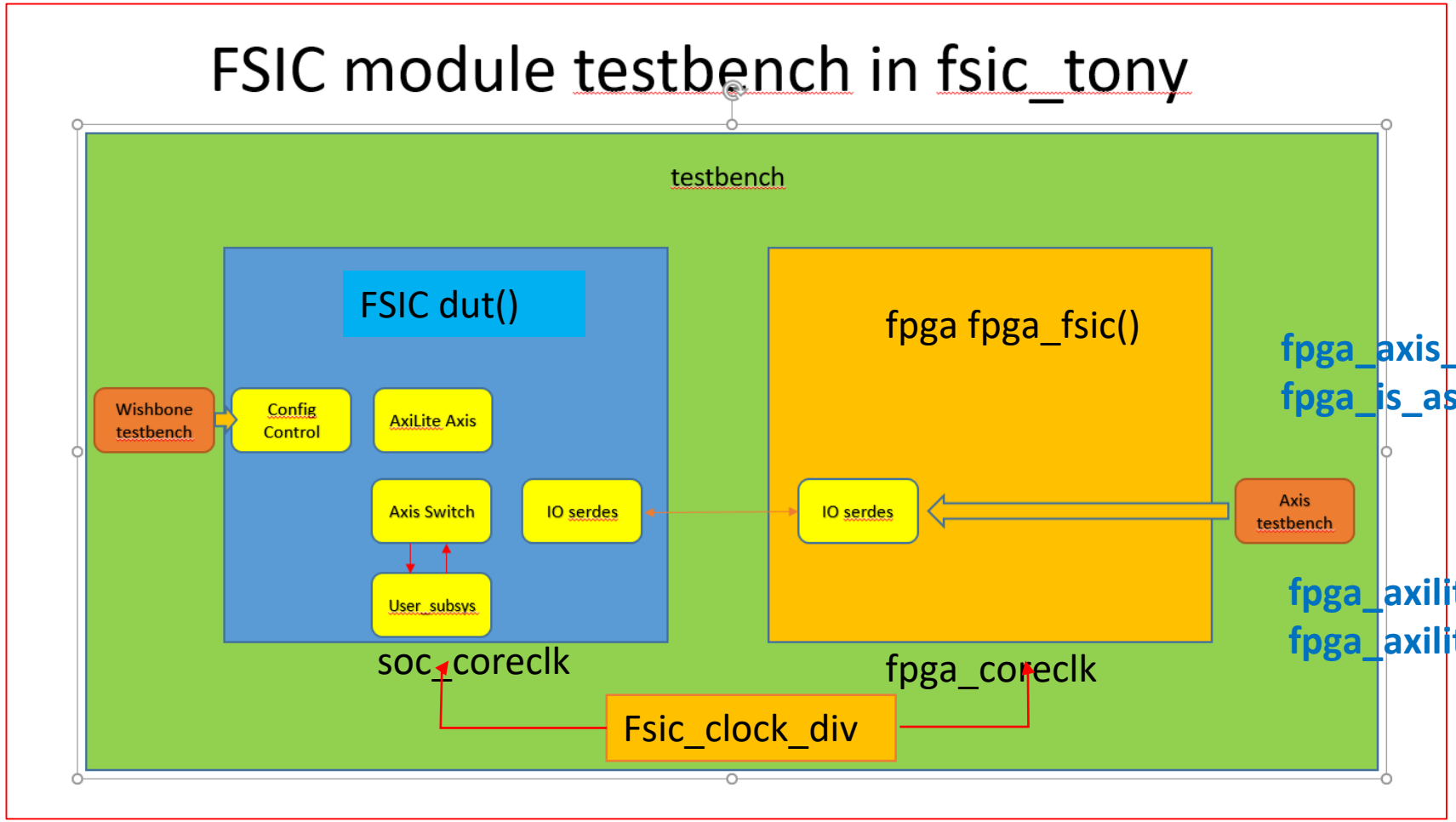
Test Items in tb_fsic.v



- test001(); //soc cfg write/read test
- test002(); //test002_fpga_axis_req
- test003(); //test003_fpga_to_soc_cfg_read
- test004(); //test004_fpga_to_soc_mail_box_write
- test005(); //test005_aa_mailbox_soc_cfg
- test006(); //test006_fpga_to_soc_cfg_write
- test007(); //test007_mailbox_interrupt test

Testbench — Task Definition

fsic_system_initial()



soc_cfg_write()
soc_cfg_read()

fpga_axis_req()
fpga_is_as_data_valid()

fpga_axilite_write_req()
fpga_axilite_read_req()

HackMD:
https://hackmd.io/@TonyHo/rk6Siw0k6
https://github.com/TonyHo722/fsic_tony

Task Definition: fsic_system_initial() — Explained

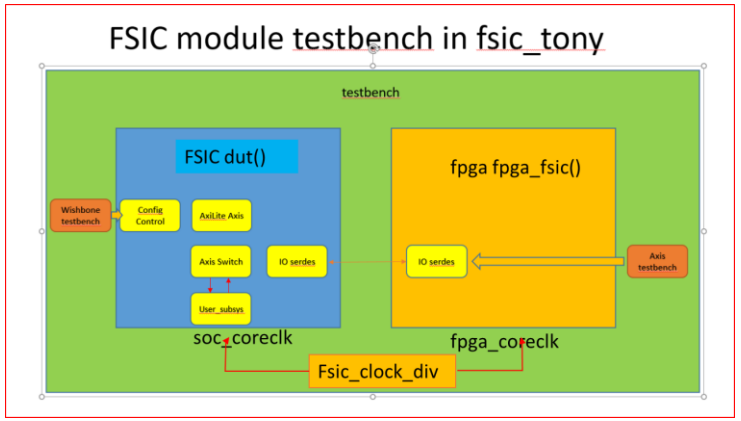
```
// SOC & FPGA Reset
fork
    soc_apply_reset(40, 40);      //change coreclk phase in soc
    fpga_apply_reset(40,40);                  //fix coreclk phase in fpga
join
#40;
fpga_as_to_is_init();
//soc_cc_is_enable=1;
fpga_cc_is_enable=1;
// Enable RX on SOC and FPGA side
fork
   soc_is_cfg_write(0, 4'b0001, 1);          //ioserdes rxen
   fpga_cfg_write(0,1,1,0);
join
// Enable TX on SOC and FPGA side
#400;
fork
   soc_is_cfg_write(0, 4'b0001, 3);          //ioserdes txen
   fpga_cfg_write(0,3,1,0);
join
```

soc rxen_ctl=1

fpga rxen_ctl=1

soc txen_ctl=1

fpga txen_ctl=1




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

        begin
                @ (posedge soc_coreclk);
                wbs_adr <= IS_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_is_cfg_write : wbs_adr=%x, wbs_sel=%b, wbs_wdata=%x", wbs_adr, wbs_sel, wbs_wdata);
        end
endtask
```

localparam IS_BASE=32'h3000_3000;

Task Definition – SOC Side Configuration Tasks

- You may **use or modify** these tasks in tb_fsic.v to achieve your goal in this lab!

(including soc_**is**_cfg_write/soc_**aa**_cfg_write/soc_**up**_cfg_write)

```
task soc_cfg_write            // soc side configuration write - using wb
   input [3:0]  target;       // 4 bit for AA, IS, CC,  register range
   input [11:0] offset;       //4K range
   input [3:0]  sel;          // byte enable
   input [31:0] data;
```

IS: io-serdes
AA: AXIS-AXIL
UP: User Project

(including soc_**is**_cfg_read/soc_**aa**_cfg_read/soc_**up**_cfg_read)

```
task soc_cfg_read             // with auto check value
   input [3:0]    target;     // 4 bit for AA, IS, CC
   input [11:0]   offset;     //4K range
   input [31:0]   expected;   // expected_value
   input [31:0]   mask;       // bit set to 1 for expected value comparison
```

Task Definition – FPGA Side Configuration Cycle (AxiLite)

- You may **use or modify** these tasks in tb_fsic.v to achieve your goal in this lab!

```
task fpga_axilite_write_req;    // axilite-> axis conversion to io_serdes
   input [27:0] address;      // address
   input [3:0] BE;           // byte-enable
   input [31:0] data;         // data
```

```
task fpga_axilite_read_req;     // axilite read -> axis conversion to io_serdes
   input [31:0] address;
   input [31:0] expect;       // expected value
   input [31:0] mask;         // bit set to 1 for exepected value comparision
```

Task Definition – FPGA Side Data Transfer (AXI Stream)

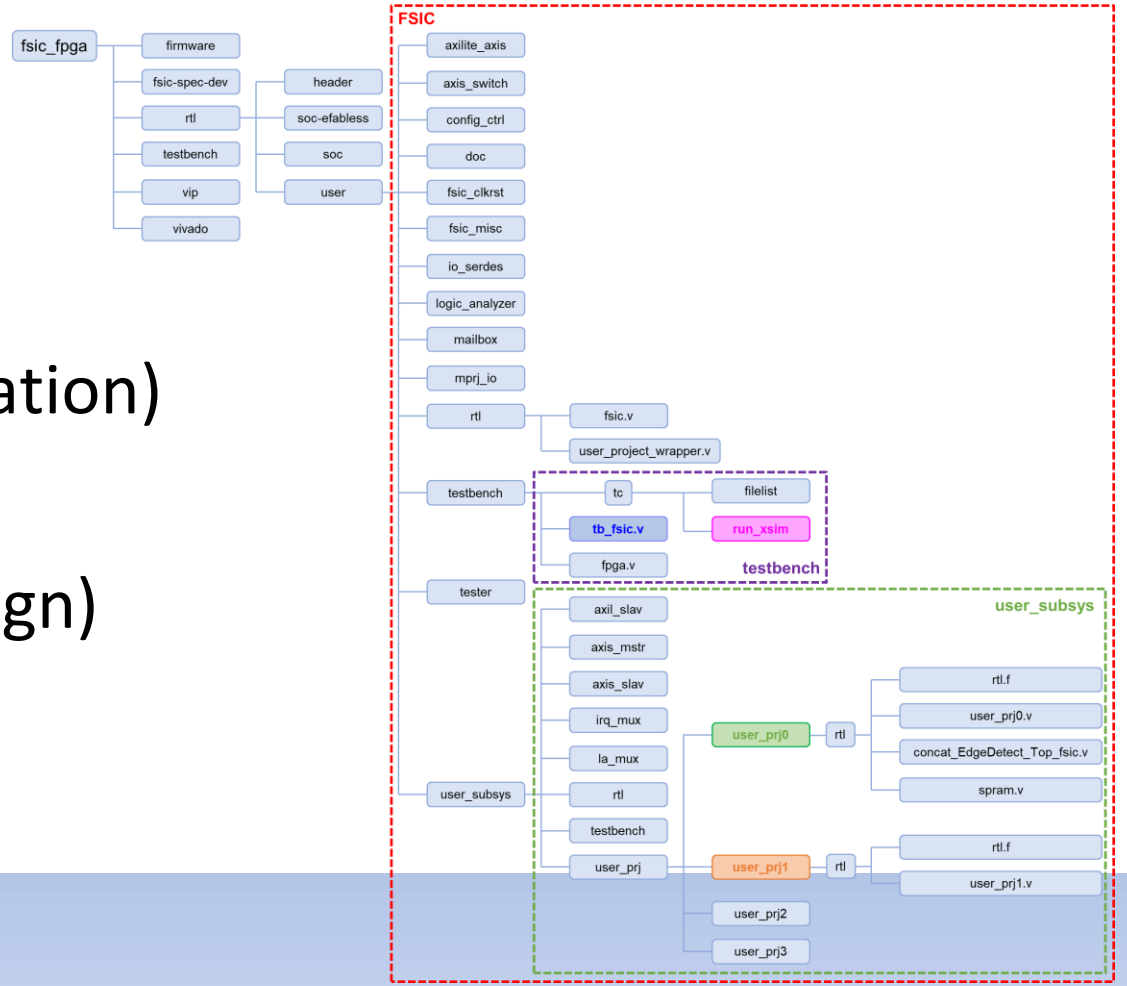
- You may **use or modify** these tasks in tb_fsic.v to achieve your goal in this lab!

```
task fpga_axis_req;          // switch to io_serdes  : downstream transfer
   input [31:0] data;
   input [1:0] tid;
   input mode; //0 for noram, 1 for random data
   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;
```

```
task fpga_is_as_data_valid;    // io serdes to switch : upstream transfer
   input [31:0] expect;       // expected value to check
   input [31:0] mask;         // bit set to 1 for expected value comparison
```

FSIC Simulation Environment — Github Folder (1/2)

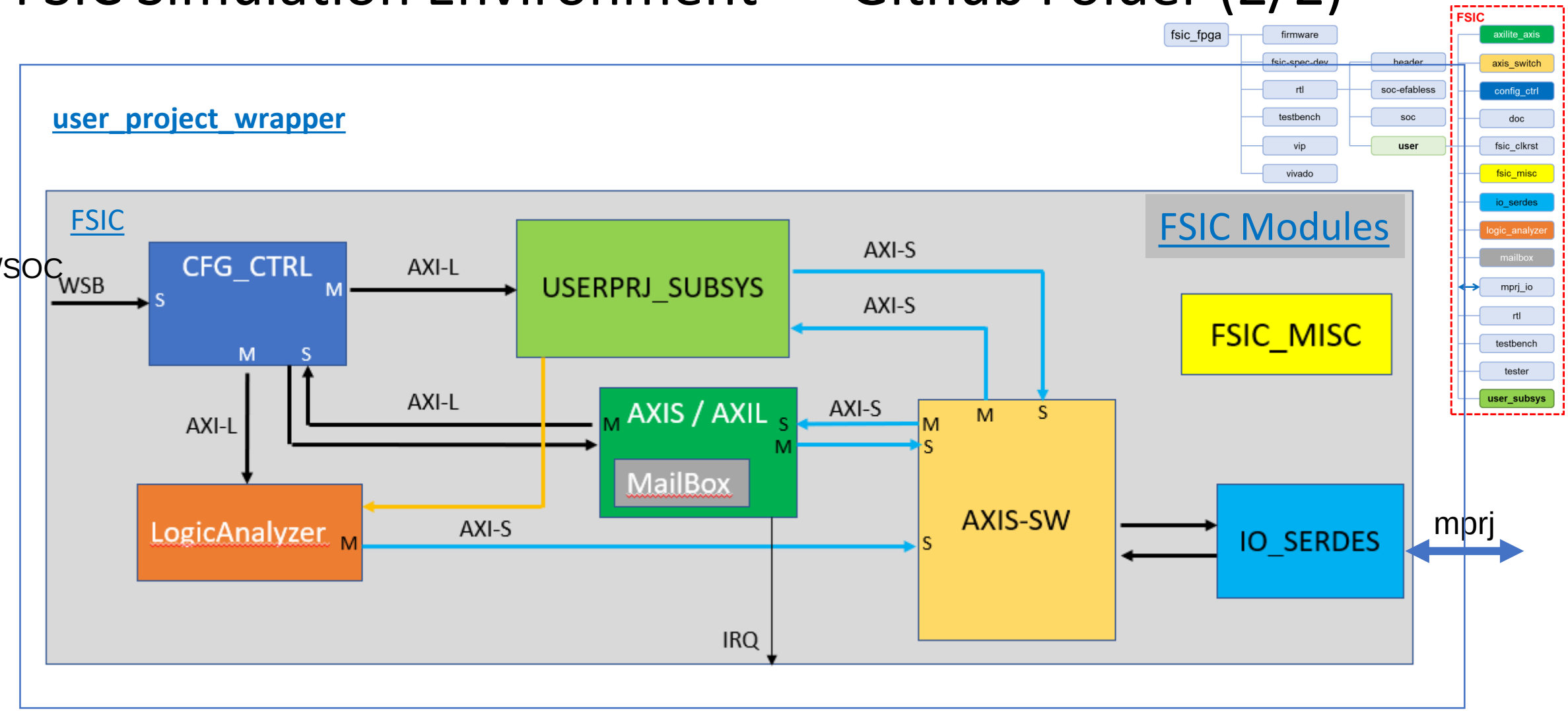
- **Github of this lab (FSIC Design)**
  - https://github.com/bol-edu/caravel-soc_fpga-lab/tree/main/fsic-sim



- run_xsim (Script for Running Simulation)
- user_prj0 (for your reference)
- user_prj1 (where you put your design)
- Testbench (tb_fsic.v)

FSIC Simulation Environment — Github Folder (2/2)

From Caravel/SOC



Lab work

Lab-fsic-sim : Lab Work

- Goal:

1. Setup FSIC simulation environment
2. Integrate your FIR into FSIC
3. Write a testbench to feed FIR input data & take FIR output data

- Detailed implementation will be described in the next page.
- You don’t need to do synthesis in this lab, just simulation!

Lab-fsic-sim : Lab Work (detail) (1/3)

- Implementation:

→ **1. Setup FSIC simulation environment**

1. Use the following command:

```
$ git clone https://github.com/bol-edu/caravel-soc_fpga-lab.git
$ cd caravel-soc_fpga-lab/fsic-sim
$ cd fsic_fpga/rtl/user/testbench/tc
$ ./run_xsim
```

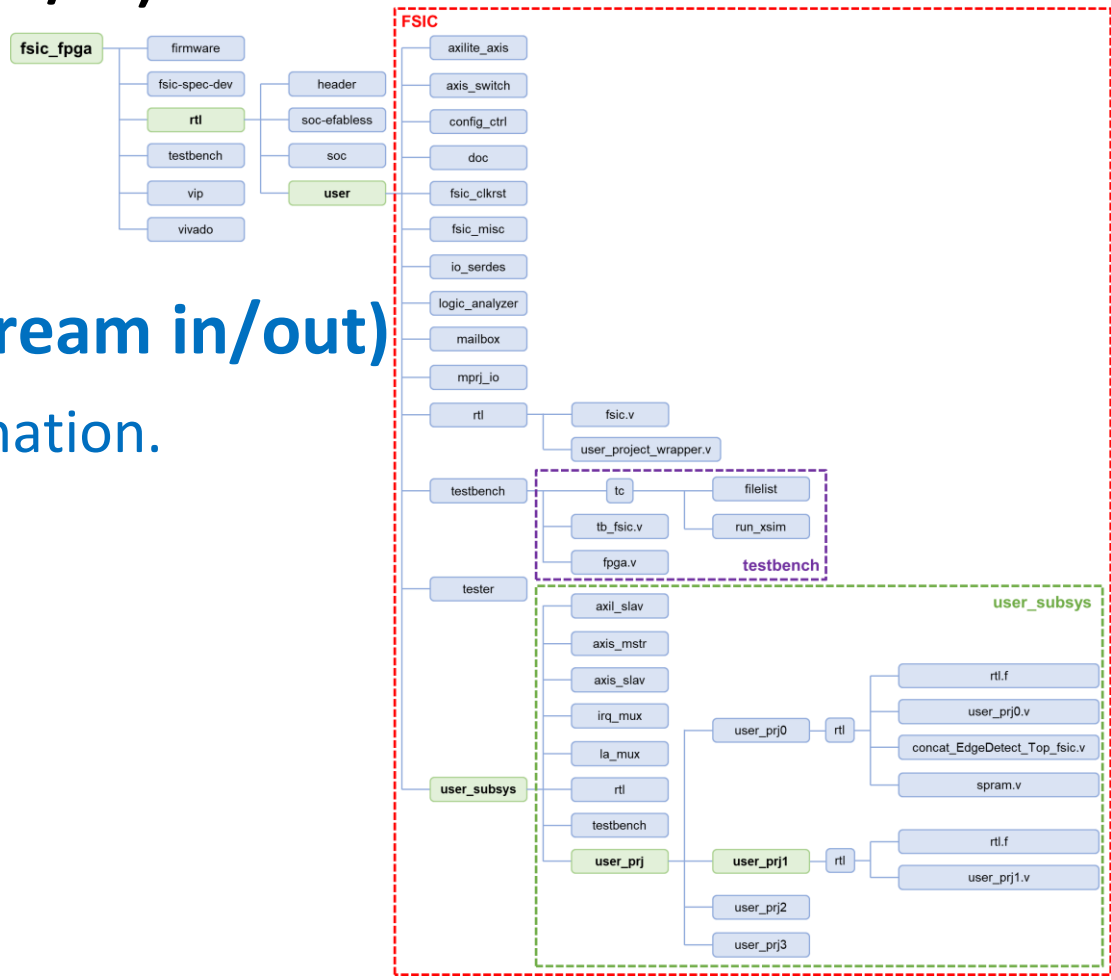
2. Wait for a few minutes to finish simulation.
3. Check whether there exists any simulation error. The information printed on screen should be almost the same as https://github.com/bol-edu/caravel-soc_fpga-lab/blob/main/fsic-sim/fsic_fpga/rtl/user/testbench/tc/log/xsim.log

2. Integrate FIR into PRJ1 (axilite, axi-stream in/out)
3. TestBench (modify fsic_tb.v)

Lab-fsic-sim : Lab Work (detail) (2/3)

- Implementation:

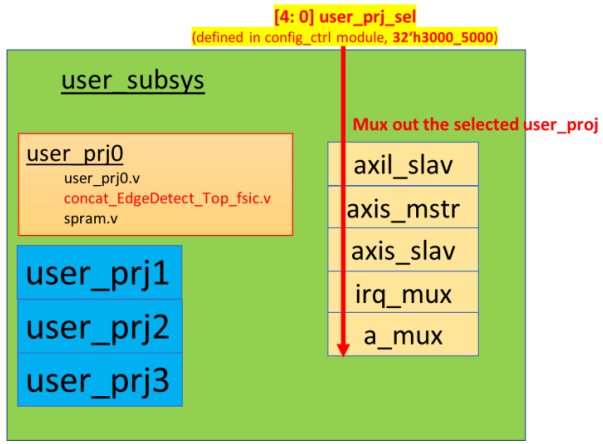
1. Setup FSIC simulation environment
2. **Integrate FIR into PRJ1 (axilite, axi-stream in/out)**
   - Please see the next page for more information.
3. TestBench (modify fsic_tb.v)



How to integrate your design in PRJ1

- Use fir.v that you have designed in SoC-design course (last semester); for those who did not take last semester's course, we have a reference fir.v in https://github.com/bol-edu/caravel-soc_fpga-lab/tree/main/fsic-sim/reference_FIR

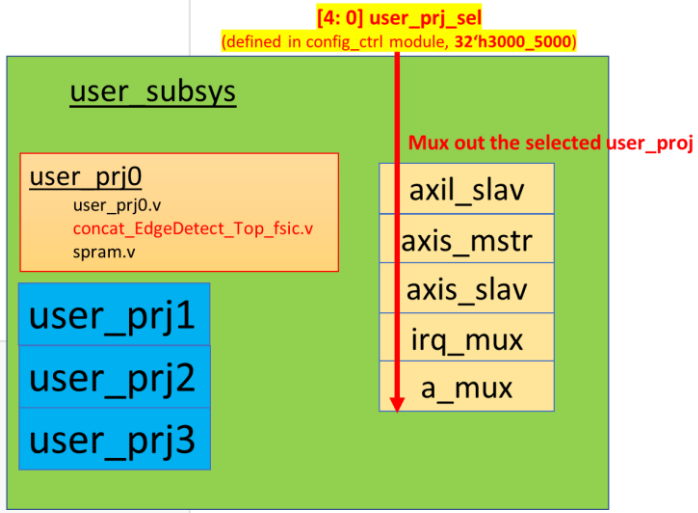
- user_prj1 folder
  - https://github.com/bol-edu/caravel-soc_fpga-lab/tree/main/fsic-sim/fsic_fpga/rtl/user/user_subsys/user_prj/user_prj1/rtl
- Use user_prj1.v as your module top interface
- Define all your submodule in **rtl.f** & **filelist**
- Enable PRJ1 in Caravel SOC, in firmware code (Run in Caravel SOC)
  - Program [‘h3000_5000] = **32’h01**;
  - This will direct all



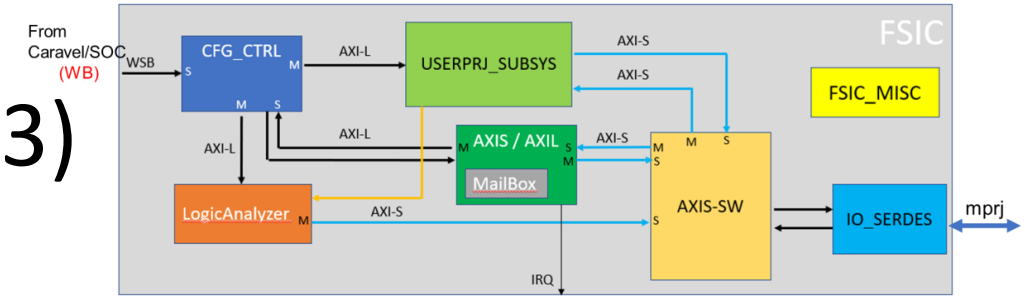
CC – Configuration Register

Configuration Control Group: 32'h3000_5000~32'h30000_5FFF

RegisterName	Offset Address	Description
User Project Selction Control	12'h000	User Project Selection Control Register Definition This 5bits register is used for User Project selection. The selection mapping is defined as following: [4:0] 5'h0: User Project 0 enabled (Default) 5'h1: User Project 1 enabled 5'h2: User Project 2 enabled 5'h3: User Project 3 enabled ... [31:5] 27'hxxxxxxx: Reserved
Reserved	12'h004 ~ 12'hFFC	Reserved



Lab-fsic-sim : Lab Work (detail) (3/3)



- Implementation:

1. Setup FSIC simulation environment
2. Integrate FIR into PRJ1 (axilite, axi-stream in/out)
3. **TestBench (modify tb_fsic.v)**
   1. Test#1 – FIR initialization (tap parameter, length) from SOC side
      - FIR initialization (tap parameter, length) **from SOC side**
      - **Use Mailbox to notify FPGA side to start X, Y stream transfer**
      - FIR data X, Y stream data from FPGA side
      - Check if output data Y are correct
   2. Test#2 – FIR initialization from FPGA side
      - FIR initialization **from FPGA side**
      - FIR data X, Y stream data from FPGA side
      - Check if output data Y are correct

You can refer to:

```
test001();    //soc cfg write/read test
test002();  //test002_fpga_axis_req
test003();  //test003_fpga_to_soc_cfg_read
test004();  //test004_fpga_to_soc_mail_box_write
test005();  //test005_aa_mailbox_soc_cfg
test006();  //test006_fpga_to_soc_cfg_write
test007();  //test007_mailbox_interrupt test
task soc_{is/aa/up}_cfg_write;
task soc_{is/aa/up}_cfg_read;
task fpga_axilite_write_req;
task fpga_axilite_read_req;
task fpga_axis_req;
task fpga_is_as_data_valid;
```

IS: io-serdes
AA: AXIS-AXIL
UP: User Project

Modify them to finish Test#1 and Test#2 !!

Don't forget to call **task fsic_system_initial()** at the beginning of each Test !!!

Submission

- Please submit the following files to {NTHU eeclass / NTU COOL / NYCU E3} before **2024/3/14** :
  1. **report_StudentID.pdf**
  2. **Github_link.txt**

- Please check the next pages for more detailed information.

What should be included in **report_StudentID.pdf** (1/2)

1. **Show the code** that you use to program configuration address [‘h3000_5000].
2. Explain **why** “By programming configuration address [‘h3000_5000], signal user_prj_sel[4:0] will change accordingly”? (Hint: trace code in config_ctrl.v)

3. Briefly describe how you do FIR initialization (tap parameter, length) **from SOC side** (Test#1).
4. Briefly describe how you do FIR initialization (tap parameter, length) **from FPGA side** (Test#2).

5. Briefly describe how you **feed in X data** from FPGA side.
6. Briefly describe how you **get output Y data** in testbench, and **how to do comparison** with golden values.

What should be included in **report_StudentID.pdf** (2/2)

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

8. **Screenshot** simulation waveform:
   - Configuration cycle (when we program [‘h**3000_5000**] = **32’h01**, signal **user_prj_sel** changes accordingly)
   - AXI-Lite transaction cycles (feed in tap parameters, data_length)
   - Stream-in, Stream-out

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

What should be uploaded to Github

1. Design_sources
   - **All files** in “/fsic_fpga/rtl/user/user_subsys/user_prj/user_prj1/rtl/”

2. Testbench
   - filelist (located at /fsic_fpga/rtl/user/testbench/tc/filelist)
   - tb_fsic.v (located at /fsic_fpga/rtl/user/testbench/tb_fsic.v)
   - xsim.log (after integrating your FIR) (located at /fsic_fpga/rtl/user/testbench/tc/xsim.log)

Supplement

How to generate .vcd wave file for debugging?

- If you'd like to use GTKWave to see waveform, you can follow the following steps:

1. Add these codes into tb_fsic.v:

```
initial begin
    $dumpfile("FSIC_FIR.vcd");
    $dumpvars(0, tb_fsic);
end
```

2. After simulation, it will generate a waveform file named "FSIC_FIR.vcd", which can be read by GTKWave.

Review: tb_fsic.v

- Address mapping table: SOC -> FPGA, FPGA -> SOC
- Explain macro definition
  - USE_EDGEDETECT_IP
  - USER_PROJECT_SIDEBAND_SUPPORT
  - `USE_POWER_PINS
- Clocking scheme:
    - loclk_source -> soc_coreclk, fpga_coreclk
- Interface FSIC + FPGA
- FPGA Internal block Diagram
- Test Items

https://github.com/bol-edu/caravel-soc_fpga-lab/blob/main/fsic-sim/fsic_fpga/rtl/user/testbench/tb_fsic.v

What is this for?

```
task fpga_cfg_write          // confguration write from fpga side
   input [pADDR_WIDTH-1:0] awaddr;
   input [pDATA_WIDTH-1:0] wdata;
   input [3:0] wstrb;
   //   input [7:0] valid_delay;   - removed
```