SOC Design

Lab 4-2 Caravel FIR

Group no: 5

Members:

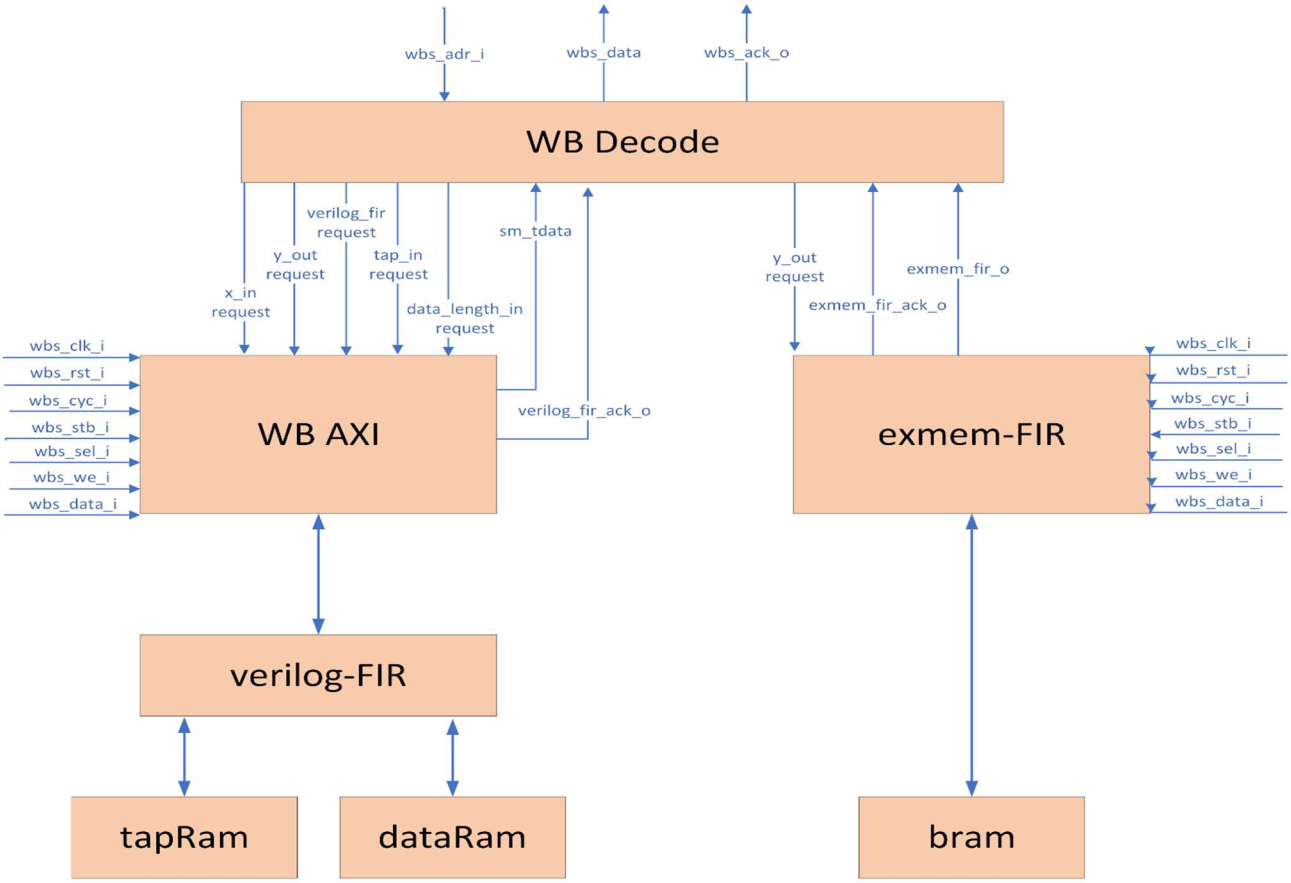
M11207415 陳謝鎧

M11207002 陳泊佑

M11107426 廖千慧

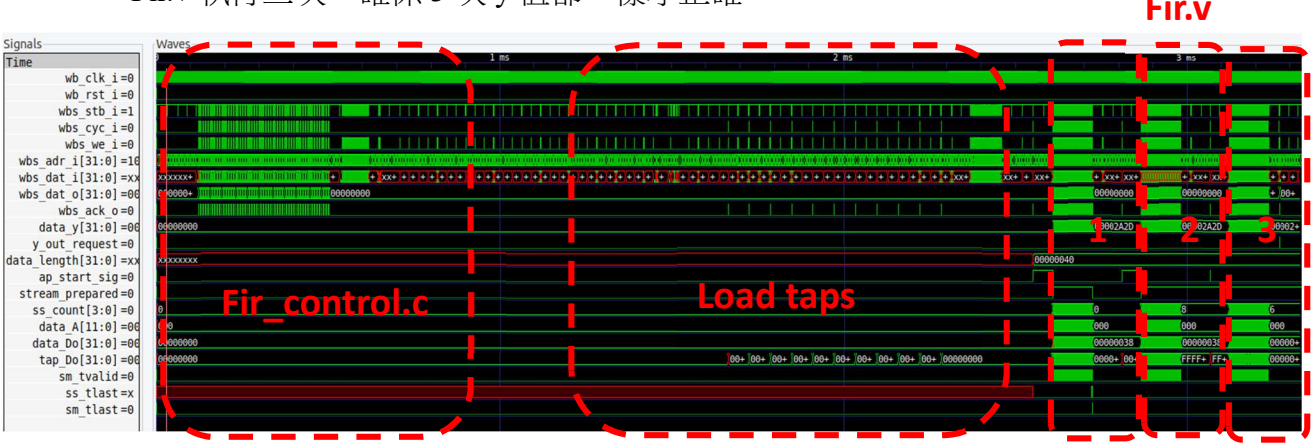
M11207328 吳奕帆

Block diagram:

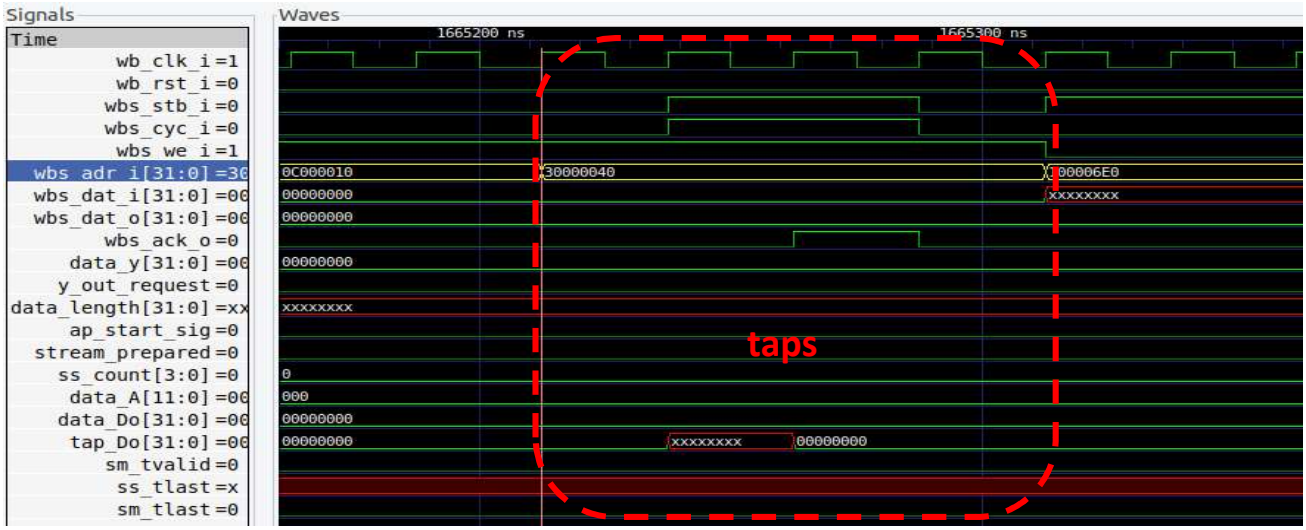


- The interface protocol between firmware, user project and testbench.
  a. The interface protocol between firmware and user project are wishbone.
  b. The interface protocol between testbench communicates with firmware and user projects through mprj pin.

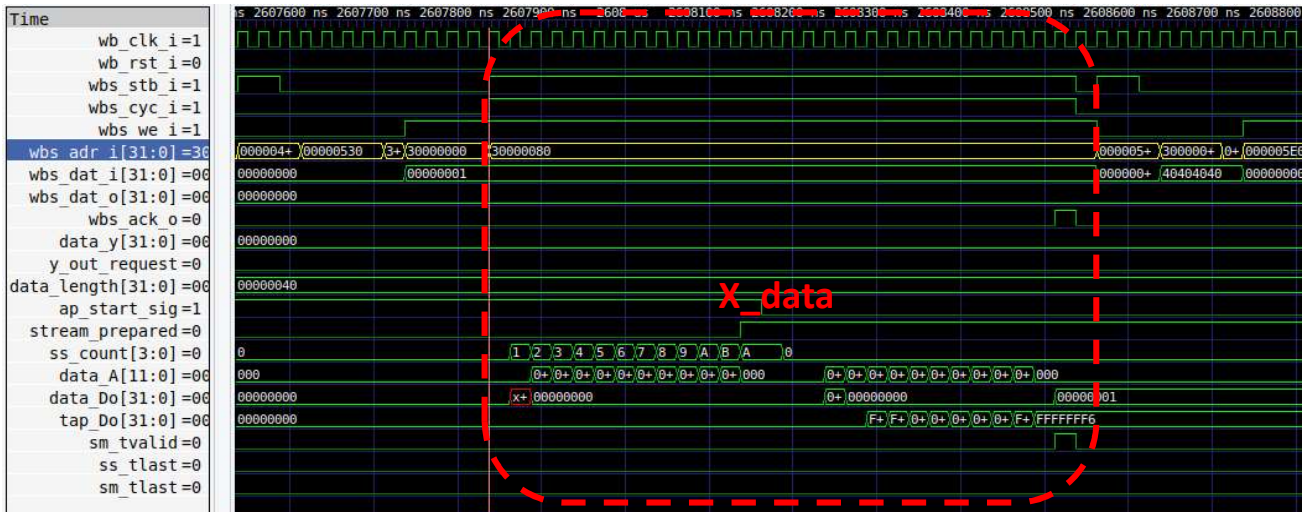
- Waveform and analysis of the hardware/software behavior.
  Fir.v 執行三次，確保 3 次 y 值都一樣才正確。



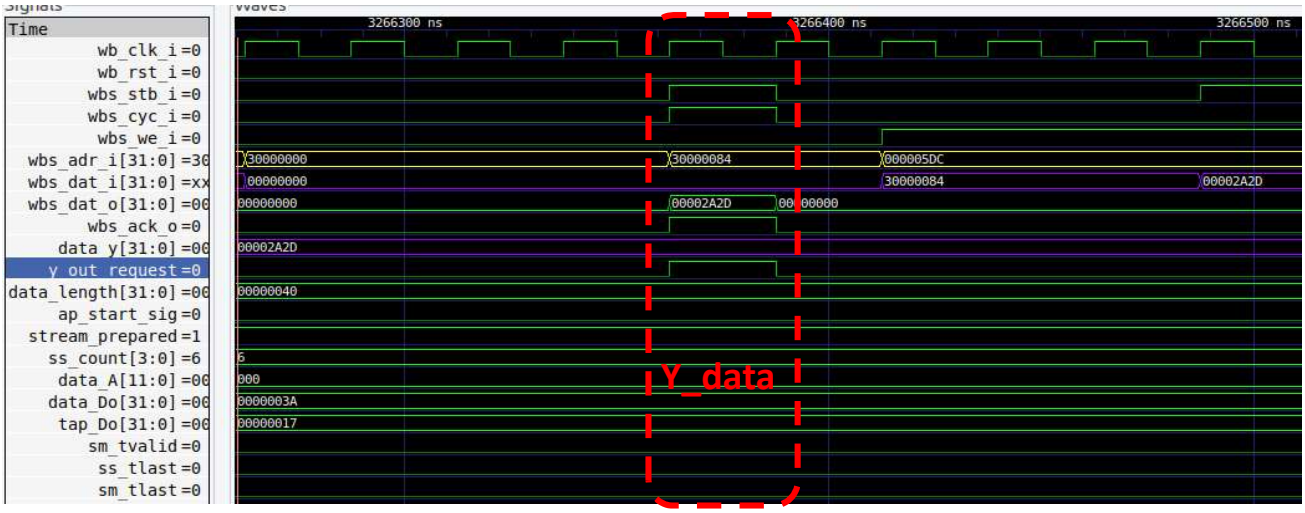
Tap 的位置在 0x30000040



X_data 的位置在 0x30000080



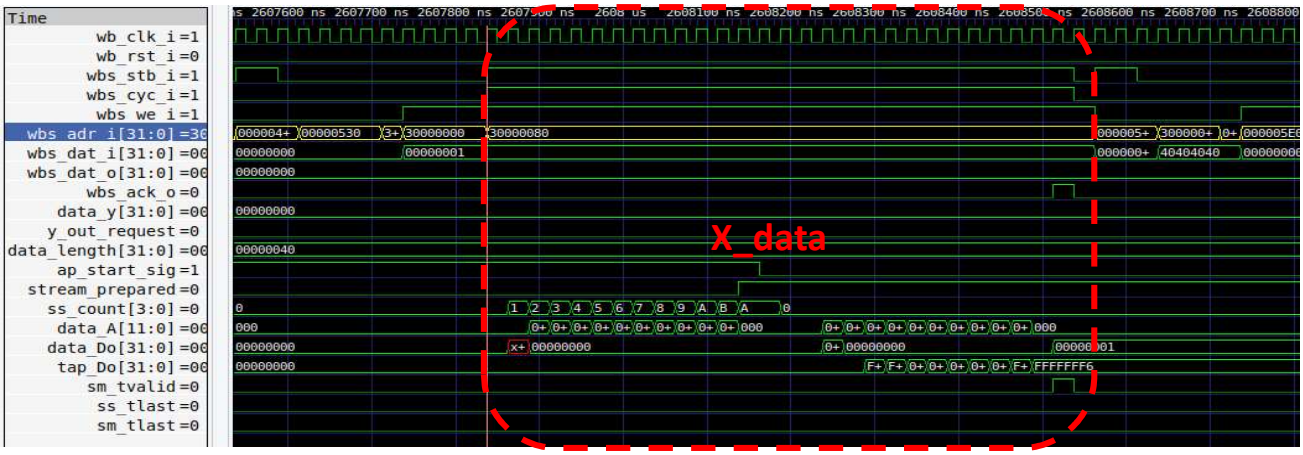
Fir.v 讀取輸出 Y_data 的位置在 0x30000084



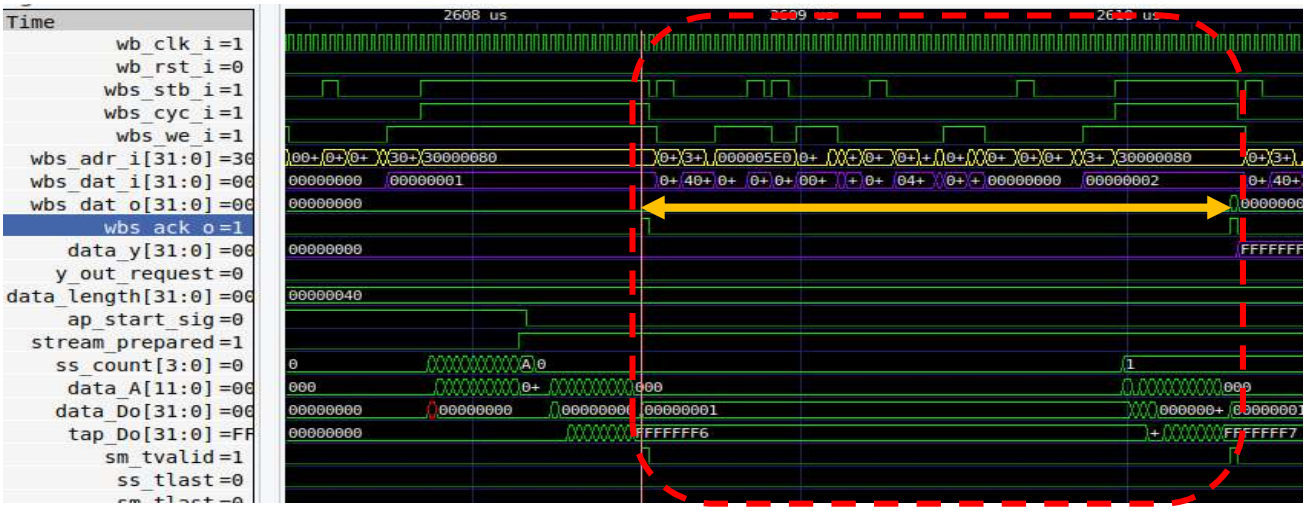
- What is the FIR engine theoretical throughput, i.e. data rate? Actually measured throughput?

  Clock period: 25ns

  theoretical throughput -> data rate: 15 clock

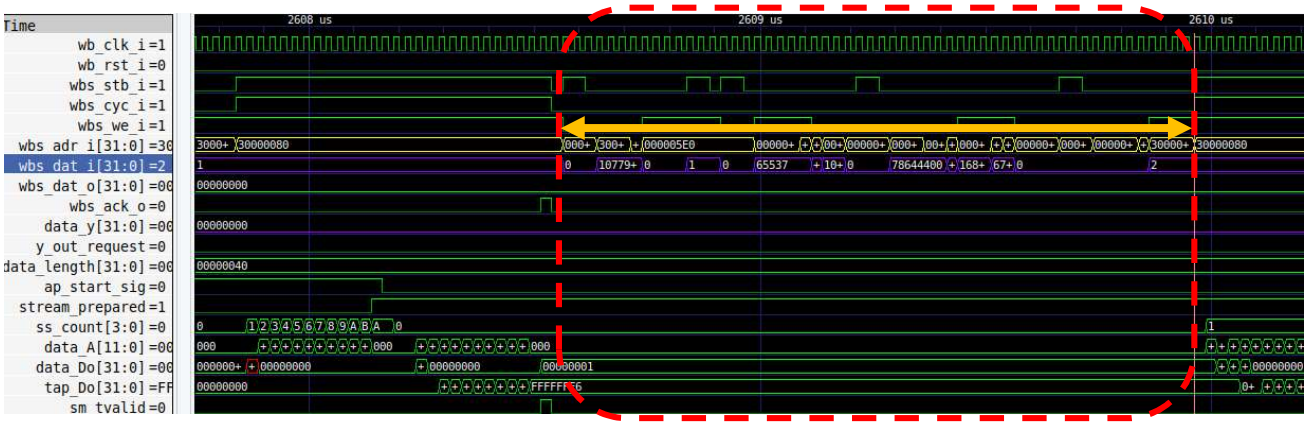


Actually measured throughput: 72 clock



- What is latency for firmware to feed data?

  Latency: 56 clocks



- What techniques used to improve the throughput?

Bram12 比 Bram11 能多存一筆資料，能夠將資料做 prefetch，加速後續的資料的存取，因此能夠增加 throuthput 量。透過增加資料輸入的 bandwidth 也能夠增加其 throughput。

Prepare firmware code & RTL:

- Firmware code (fir_control.c)

```c
// This include is relative to $CARAVEL_PATH (see Makefile)
#include <defs.h>
#include <stub.c>

#define N 11            // # of Tap
#define data_length 64 // X data length  64

// Define mmio registers
#define reg_fir_verilog (*(volatile uint32_t *)0x30000000)
#define reg_data_length (*(volatile uint32_t *)0x30000010) // 10 ~ 14
#define reg_fir_coeff (*(volatile uint32_t *)0x30000040)    // tap 40 ~ 7F
#define reg_fir_x (*(volatile uint32_t *)0x30000080)        // 80 ~ 83
#define reg_fir_y (*(volatile uint32_t *)0x30000084)        // 84 ~ 87
#define reg_fir_exmem (*(volatile uint32_t *)0x38000000)

void main()
{
    int j;

    reg_mprj_io_31 = GPIO_MODE_MGMT_STD_OUTPUT;
    reg_mprj_io_30 = GPIO_MODE_MGMT_STD_OUTPUT;
    reg_mprj_io_29 = GPIO_MODE_MGMT_STD_OUTPUT;
    reg_mprj_io_28 = GPIO_MODE_MGMT_STD_OUTPUT;
    reg_mprj_io_27 = GPIO_MODE_MGMT_STD_OUTPUT;
    reg_mprj_io_26 = GPIO_MODE_MGMT_STD_OUTPUT;
    reg_mprj_io_25 = GPIO_MODE_MGMT_STD_OUTPUT;
    reg_mprj_io_24 = GPIO_MODE_MGMT_STD_OUTPUT;
    reg_mprj_io_23 = GPIO_MODE_MGMT_STD_OUTPUT;
    reg_mprj_io_22 = GPIO_MODE_MGMT_STD_OUTPUT;
    reg_mprj_io_21 = GPIO_MODE_MGMT_STD_OUTPUT;
    reg_mprj_io_20 = GPIO_MODE_MGMT_STD_OUTPUT;
    reg_mprj_io_19 = GPIO_MODE_MGMT_STD_OUTPUT;
    reg_mprj_io_18 = GPIO_MODE_MGMT_STD_OUTPUT;
    reg_mprj_io_17 = GPIO_MODE_MGMT_STD_OUTPUT;
    reg_mprj_io_16 = GPIO_MODE_MGMT_STD_OUTPUT;

    reg_mprj_io_15 = GPIO_MODE_USER_STD_OUTPUT;
    reg_mprj_io_14 = GPIO_MODE_USER_STD_OUTPUT;
    reg_mprj_io_13 = GPIO_MODE_USER_STD_OUTPUT;
    reg_mprj_io_12 = GPIO_MODE_USER_STD_OUTPUT;
    reg_mprj_io_11 = GPIO_MODE_USER_STD_OUTPUT;
    reg_mprj_io_10 = GPIO_MODE_USER_STD_OUTPUT;
    reg_mprj_io_9 = GPIO_MODE_USER_STD_OUTPUT;
    reg_mprj_io_8 = GPIO_MODE_USER_STD_OUTPUT;
    reg_mprj_io_7 = GPIO_MODE_USER_STD_OUTPUT;
    reg_mprj_io_5 = GPIO_MODE_USER_STD_OUTPUT;
    reg_mprj_io_4 = GPIO_MODE_USER_STD_OUTPUT;
    reg_mprj_io_3 = GPIO_MODE_USER_STD_OUTPUT;
    reg_mprj_io_2 = GPIO_MODE_USER_STD_OUTPUT;
    reg_mprj_io_1 = GPIO_MODE_USER_STD_OUTPUT;
    reg_mprj_io_0 = GPIO_MODE_USER_STD_OUTPUT;

    reg_mprj_io_6 = GPIO_MODE_MGMT_STD_OUTPUT;

    // Set UART clock to 64 kbaud (enable before I/O configuration)
    // reg_uart_clkdiv = 625;
    reg_uart_enable = 1;

    // Now, apply the configuration
    reg_mprj_xfer = 1;
    while (reg_mprj_xfer == 1);

    // Configure LA probes [31:0], [127:64] as inputs to the cpu
    // Configure LA probes [63:32] as outputs from the cpu
    reg_la0_oenb = reg_la0_iena = 0x00000000; // [31:0]
    reg_la1_oenb = reg_la1_iena = 0xFFFFFFFF; // [63:32]
    reg_la2_oenb = reg_la2_iena = 0x00000000; // [95:64]
    reg_la3_oenb = reg_la3_iena = 0x00000000; // [127:96]

    // Flag start of the test
    reg_mprj_datal = 0x00A50000; // Start mark ('hA5)

    // Set Counter value to zero through LA probes [63:32]
    reg_la1_data = 0x00000000;

    // Configure LA probes from [63:32] as inputs to disable counter write
    reg_la1_oenb = reg_la1_iena = 0x00000000;

    // tap parameters
    int tap[N] = {0, -10, -9, 23, 56, 63, 56, 23, -9, -10, 0};

    for (int i = 0; i < N; i++){
        (*(volatile uint32_t *)(0x30000040 + (4 * i))) = *(tap + i);
    }

    // X data
    int X[data_length];
    for (int i = 0; i < data_length; i++){
        X[i] = i+1;
    }

    // repeat m times
    for (int m = 0; m < 3; m++){
        reg_data_length=data_length;
        reg_fir_verilog = 0x01;

        for (int i = 0; i < data_length; i++)
            reg_fir_x = *(X + i);
    }

    // Y data
    int Y = reg_fir_y;

    if (Y == 0x2A2D)          // Y[7:0]-> mprj[31:24], End mark= 'h5A
        reg_mprj_datal = (Y << 24) | 0x005A0000;

}
```

➢ RTL code (user_proj_example.counter.v)

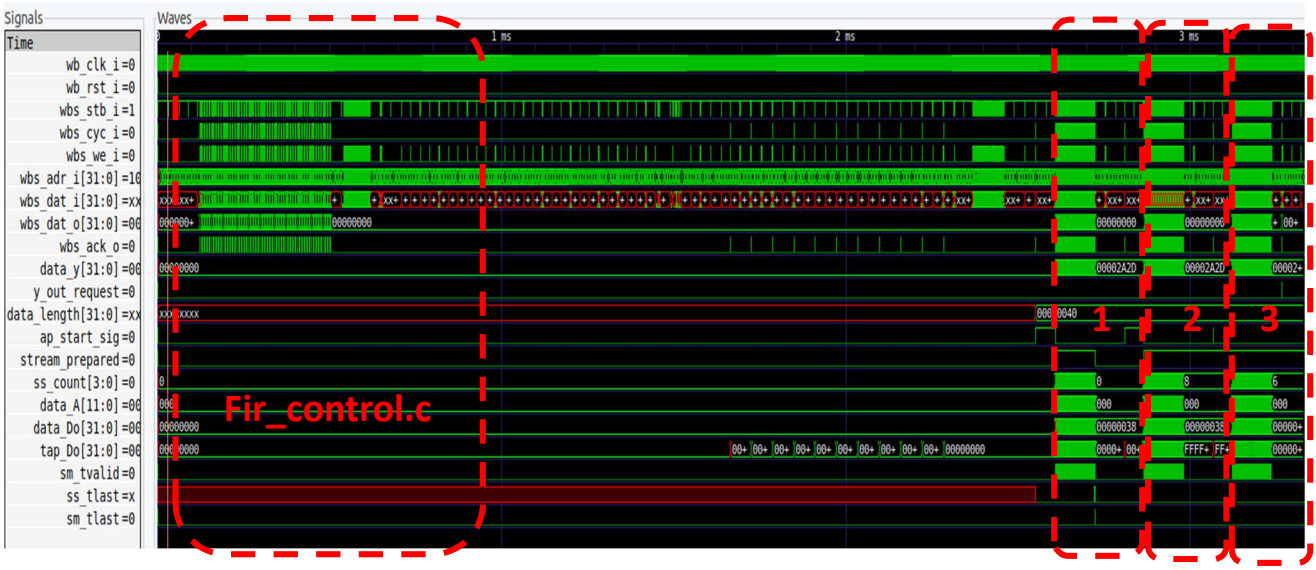
```
module user_proj_example #(
    parameter BITS = 32,
    parameter DELAYS=10,
    parameter pADDR_WIDTH = 12,
    parameter pDATA_WIDTH = 32
)(
`ifdef USE_POWER_PINS
    inout vccd1,    // User area 1 1.8V supply
    inout vssd1,    // User area 1 digital ground
`endif

    // Wishbone Slave ports (WB MI A)
    input wb_clk_i,
    input wb_rst_i,
    input wbs_stb_i,
    input wbs_cyc_i,
    input wbs_we_i,
    input [3:0] wbs_sel_i,
    input [31:0] wbs_dat_i,
    input [31:0] wbs_adr_i,
    output wbs_ack_o,
    output [31:0] wbs_dat_o,

    // Logic Analyzer Signals
    input  [127:0] la_data_in,
    output [127:0] la_data_out,
    input  [127:0] la_oenb,

    // IOs
    input  wire  [`MPRJ_IO_PADS-1:0] io_in,
    output wire  [`MPRJ_IO_PADS-1:0] io_out,
    output wire[`MPRJ_IO_PADS-1:0] io_oeb,

    // IRQ
    output [2:0] irq
);

    wire clk;
    wire rst;
    assign clk = wb_clk_i;
    assign rst = wb_rst_i;

    // wire [`MPRJ_IO_PADS-1:0] io_in;
    // wire [`MPRJ_IO_PADS-1:0] io_out;
    // wire [`MPRJ_IO_PADS-1:0] io_oeb;

    wire [1:0] decode;
    assign decode = (wbs_adr_i[31:16]==16'h3800)? 2'd2:    // exmem_fir
                (wbs_adr_i[31:16]==16'h3000)? 2'd1: 2'd0;   // veilog_fir


    wire exmem_fir_request, verilog_fir_request;    // 3800 , 3000
    assign exmem_fir_request = wbs_cyc_i & wbs_stb_i & (decode == 2'd2);
    assign verilog_fir_request = wbs_cyc_i & wbs_stb_i & (decode == 2'd1);

    // write data to on_chip ram only when request_sig assert
    wire [31:0] ram_adr;
    wire [31:0] ram_data;
    wire [3:0] ram_we;
    wire ram_en;
    assign ram_adr = (exmem_fir_request==1'b1)? wbs_adr_i : 32'b0;
    assign ram_data = (exmem_fir_request==1'b1)? wbs_dat_i : 32'b0;
    assign ram_we = (exmem_fir_request==1'b1)?  ({4{wbs_we_i}} & wbs_sel_i) : 4'b0;
    assign ram_en = (exmem_fir_request==1'b1)? (wbs_cyc_i & wbs_stb_i): 1'b0;


    wire                     awready;       // o
    wire                     wready;        // o
    wire                     awvalid;       // i
    wire [(pADDR_WIDTH-1):0] awaddr;        // i
    wire                     wvalid;        // i
    wire [(pDATA_WIDTH-1):0] wdata;         // i
    wire                     arready;       // o
    wire                     rready;        // i
    wire                     arvalid;       // i
    wire [(pADDR_WIDTH-1):0] araddr;        // i
    wire                     rvalid;        // o
    wire [(pDATA_WIDTH-1):0] rdata;         // o
    wire                     ss_tvalid;     // i
    wire [(pDATA_WIDTH-1):0] ss_tdata;      // i
    wire                     ss_tlast;      // i
    wire                     ss_tready;     // o
    wire                     sm_tready;     // i
    wire                     sm_tvalid;     // o
    wire [(pDATA_WIDTH-1):0] sm_tdata;      // o
    wire                     sm_tlast;      // o
    wire                     axis_clk;
    wire                     axis_rst_n;

    // bram for tap RAM
    wire           [3:0]     tap_WE;
    wire                     tap_EN;
    wire [(pDATA_WIDTH-1):0] tap_Di;
    wire [(pADDR_WIDTH-1):0] tap_A;
    wire [(pDATA_WIDTH-1):0] tap_Do;     // i

    // bram for data RAM
    wire             [3:0]    data_WE;
    wire                      data_EN;
    wire [(pDATA_WIDTH-1):0] data_Di;
    wire [(pADDR_WIDTH-1):0] data_A;
    wire [(pDATA_WIDTH-1):0] data_Do;   // i

    wire tap_in_request, data_length_in_request, ctrl_in_request;
    wire x_in_request, y_out_request;
    reg [(pDATA_WIDTH-1):0] data_length, data_length_count;

    assign axis_clk = wb_clk_i;
    assign axis_rst_n = !wb_rst_i;

    assign tap_in_request = (verilog_fir_request && wbs_adr_i[7:0] >= 8'h40 && wbs_adr_i[7:0] <= 8'h7F)? 1'b1:1'b0;
    assign x_in_request = (verilog_fir_request && wbs_adr_i[7:0] >= 8'h80 && wbs_adr_i[7:0] <= 8'h83)? 1'b1:1'b0;
    assign y_out_request = (verilog_fir_request && wbs_adr_i[7:0] >= 8'h84 && wbs_adr_i[7:0] <= 8'h87)? 1'b1:1'b0;
    assign data_length_in_request = (verilog_fir_request && wbs_adr_i[7:0] >= 8'h10 && wbs_adr_i[7:0] <= 8'h13)? 1'b1:1'b0;
    assign ctrl_in_request = (verilog_fir_request && wbs_adr_i[7:0] == 8'h00 )? 1'b1:1'b0;

    assign awvalid   = ((ctrl_in_request || tap_in_request ||  data_length_in_request) & wbs_we_i)? 1'b1 : 1'b0;
    assign awaddr    = ((ctrl_in_request || tap_in_request ||  data_length_in_request) & wbs_we_i)? wbs_adr_i : 0;

    assign wvalid    = ((ctrl_in_request || tap_in_request ||  data_length_in_request)  & wbs_we_i)? 1'b1 : 1'b0;
    assign wdata     = ((ctrl_in_request || tap_in_request ||  data_length_in_request)  & wbs_we_i) ? wbs_dat_i : 0;

    assign rready    = (!wbs_we_i & tap_in_request ) ? 1'b1 : 1'b0;
    assign arvalid   = (!wbs_we_i & tap_in_request ) ? 1'b1 : 1'b0;
    assign araddr    = (!wbs_we_i & tap_in_request ) ? wbs_adr_i : 0;

    assign ss_tlast  = (data_length_count==(data_length-1)) ? 1'b1 : 1'b0;
    assign ss_tvalid = (wbs_we_i & x_in_request) ? 1'b1 : 1'b0;
    assign ss_tdata  = (wbs_we_i & x_in_request) ? wbs_dat_i : 0;
    assign sm_tready = (verilog_fir_request) ? 1'b1 : 1'b0;
```

```
wire verlog_fir_ack_o;
    reg exmem_fir_ack_o;
    wire [31:0] exmem_fir_o;

    assign verlog_fir_ack_o = (awready && wready) ? 1'b1 :
        (rvalid) ? 1'b1 :
        (data_length_in_request && awready) ? 1'b1 :
        (ctrl_in_request && awready) ? 1'b1 :
        (sm_tvalid) ? 1'b1 :
        (y_out_request) ? 1'b1 : 1'b0;

    assign wbs_ack_o =  verlog_fir_ack_o | exmem_fir_ack_o;

    reg [(pDATA_WIDTH-1):0] data_y;

    assign wbs_dat_o = (exmem_fir_ack_o)? exmem_fir_o :
                       (y_out_request)? data_y :
                       (verlog_fir_ack_o)? sm_tdata : 1'b0 ;

    always @ (posedge wb_clk_i) begin
        if (wb_rst_i)
            data_y <= 0;
        else if (sm_tvalid)
            data_y <= sm_tdata;

    end

    always@(posedge wb_clk_i)begin
        if(wb_rst_i) begin
            data_length_count <= 0;
        end
        else if(data_length_in_request)
            data_length <= wbs_dat_i;

        else if(sm_tvalid==1)
            data_length_count <= data_length_count + 1;

        else if (sm_tlast)
            data_length_count <= 0;
    end

    reg [3:0] delay_cnt;   // delay = 10 (DELAYS) < 2^4
    always @ (posedge clk) begin
        if (rst) begin
            exmem_fir_ack_o <= 0;
            delay_cnt <= 0;
        end
        else if (exmem_fir_request == 1'b1) begin
            if (delay_cnt == DELAYS) begin
                exmem_fir_ack_o <= 1'b1;
                delay_cnt <= 0;
            end
            else begin
                exmem_fir_ack_o <= 1'b0;
                delay_cnt <= delay_cnt + 1 ;
            end
        end
        else begin
            delay_cnt <= 0;
            exmem_fir_ack_o <= 1'b0;
        end
    end

wire [1:0] state;
wire ap_start_sig. ss_write_valid;
wire ctrl_tap_ready, ctrl_tap_valid;

fir user_fir(
        .awready(awready),          // o
        .wready(wready),            // o
        .awvalid(awvalid),          // i
        .awaddr(awaddr),            // i
        .wvalid(wvalid),            // i
        .wdata(wdata),              // i
        .arready(arready),          // o
        .rready(rready),            // i
        .arvalid(arvalid),          // i
        .araddr(araddr),            // i
        .rvalid(rvalid),            // o
        .rdata(rdata),              // o
        .ss_tvalid(ss_tvalid),      // i
        .ss_tdata(ss_tdata),        // i
        .ss_tlast(ss_tlast),        // i
        .ss_tready(ss_tready),      // o
        .sm_tready(sm_tready),      // i
        .sm_tvalid(sm_tvalid),      // o
        .sm_tdata(sm_tdata),        // o
        .sm_tlast(sm_tlast),        // o

        // ram for tap
        .tap_WE(tap_WE),            // o
        .tap_EN(tap_EN),            // o
        .tap_Di(tap_Di),            // o
        .tap_Do(tap_Do),            // i
        .tap_A(tap_A),              // o

        // ram for data
        .data_WE(data_WE),          // o
        .data_EN(data_EN),          // o
        .data_Di(data_Di),          // o
        .data_Do(data_Do),          // i
    .data_A(data_A),            // o

        .axis_clk(axis_clk),
        .axis_rst_n(axis_rst_n),
        .state(state),
        .ap_start_sig(ap_start_sig),
        .ctrl_tap_valid(ctrl_tap_valid),
        .ctrl_tap_ready(ctrl_tap_ready),
        .ss_write_valid(ss_write_valid)
    );

bram exmem_bram(
    .CLK(clk),
    .WE0(ram_we),
    .EN0(ram_en),
    .Di0(wbs_dat_i),
    .Do0(exmem_fir_o),
    .A0(wbs_adr_i)
);

// RAM for tap
bram11 tap_RAM (
            .CLK(axis_clk),
            .WE(tap_WE),
            .EN(tap_EN),
            .Di(tap_Di),
            .Do(tap_Do),
            .A(tap_A)
        );

// RAM for data: choose bram11 or bram12
bram11 data_RAM(
            .CLK(axis_clk),
            .WE(data_WE),
            .EN(data_EN),
            .Di(data_Di),
            .Do(data_Do),
            .A(data_A)
        );


endmodule


`default_nettype wire
```

➢ Compilation

Synthesis & Verification:

➢ Waveform



➢ Timing report

```
------------------------------------------------------------------------------------------------
Timer Settings
--------------
------------------------------------------------------------------------------------------------

Enable Multi Corner Analysis                :  Yes
Enable Pessimism Removal                    :  Yes
Pessimism Removal Resolution                :  Nearest Common Node
Enable Input Delay Default Clock            :  No
Enable Preset / Clear Arcs                  :  No
Disable Flight Delays                       :  No
Ignore I/O Paths                            :  No
Timing Early Launch at Borrowing Latches    :  No
Borrow Time for Max Delay Exceptions        :  Yes
Merge Timing Exceptions                     :  Yes

Corner  Analyze    Analyze
Name    Max Paths  Min Paths
------  ---------  ---------
Slow    Yes        Yes
Fast    Yes        Yes
```

```
------------------------------------------------------------------------------------------------
| Design Timing Summary
| ---------------------
------------------------------------------------------------------------------------------------

    WNS(ns)      TNS(ns)  TNS Failing Endpoints  TNS Total Endpoints      WHS(ns)      THS(ns)  THS Failing Endpoints  THS Total
Endpoints     WPWS(ns)     TPWS(ns)  TPWS Failing Endpoints  TPWS Total Endpoints
    -------      -------  ---------------------  -------------------      -------      -------  ---------------------  -------------
----     --------     --------  ---------------------  -------------------
      2.880        0.000                      0                 1229        0.072        0.000                      0
1229        6.750        0.000                      0                  402


All user specified timing constraints are met.


------------------------------------------------------------------------------------------------
| Clock Summary
| -------------
------------------------------------------------------------------------------------------------

Clock     Waveform(ns)       Period(ns)      Frequency(MHz)
-----     ------------       ----------      --------------
wb_clk_i  {0.000 8.000}      16.000          62.500


------------------------------------------------------------------------------------------------
| Intra Clock Table
| -----------------
------------------------------------------------------------------------------------------------

Clock             WNS(ns)      TNS(ns)  TNS Failing Endpoints  TNS Total Endpoints      WHS(ns)      THS(ns)  THS Failing Endpoints
THS Total Endpoints     WPWS(ns)     TPWS(ns)  TPWS Failing Endpoints  TPWS Total Endpoints
-----             -------      -------  ---------------------  -------------------      -------      -------  ---------------------
-----------------     --------     --------  ---------------------  -------------------
------------------------------------------------------------------------------------------------
From Clock:  wb_clk_i
  To Clock:  wb_clk_i

Setup :            0  Failing Endpoints,  Worst Slack        2.880ns,  Total Violation        0.000ns
Hold  :            0  Failing Endpoints,  Worst Slack        0.072ns,  Total Violation        0.000ns
PW    :            0  Failing Endpoints,  Worst Slack        6.750ns,  Total Violation        0.000ns
------------------------------------------------------------------------------------------------
```

```
Max Delay Paths
--------------------------------------------------------------------------------------
Slack (MET) :             2.880ns  (required time - arrival time)
  Source:                 data_RAM/r_A_reg_rep_bsel[2]/C
                            (rising edge-triggered cell FDRE clocked by wb_clk_i  {rise@0.000ns fall@8.000ns period=16.000ns})
  Destination:            user_fir/sm_tdata_reg_reg[29]/D
                            (rising edge-triggered cell FDRE clocked by wb_clk_i  {rise@0.000ns fall@8.000ns period=16.000ns})
  Path Group:             wb_clk_i
  Path Type:              Setup (Max at Slow Process Corner)
  Requirement:            16.000ns  (wb_clk_i rise@16.000ns - wb_clk_i rise@0.000ns)
  Data Path Delay:        13.015ns  (logic 8.576ns (65.891%)  route 4.439ns (34.109%))
  Logic Levels:           11  (CARRY4=5 DSP48E1=2 LUT2=2 LUT5=1 RAMD32=1)
  Clock Path Skew:        -0.145ns (DCD - SCD + CPR)
    Destination Clock Delay (DCD):    2.128ns = ( 18.128 - 16.000 )
    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 wb_clk_i rise edge)
                                                    0.000     0.000 r
                                                    0.000     0.000 r  wb_clk_i (IN)
                         net (fo=0)                 0.000     0.000    wb_clk_i
                                                                 r  wb_clk_i_IBUF_inst/I
                         IBUF (Prop_ibuf_I_O)       0.972     0.972 r  wb_clk_i_IBUF_inst/O
                         net (fo=1, unplaced)       0.800     1.771    wb_clk_i_IBUF
                                                                 r  wb_clk_i_IBUF_BUFG_inst/I
                         BUFG (Prop_bufg_I_O)       0.101     1.872 r  wb_clk_i_IBUF_BUFG_inst/O
                         net (fo=401, unplaced)     0.584     2.456    data_RAM/wb_clk_i_IBUF_BUFG
                         FDRE                                        r  data_RAM/r_A_reg_rep_bsel[2]/C
  -------------------------------------------------------------------    -------------------
                         FDRE (Prop_fdre_C_Q)       0.478     2.934 r  data_RAM/r_A_reg_rep_bsel[2]/Q
                         net (fo=32, unplaced)      1.036     3.970    data_RAM/RAM_reg_0_15_16_16/DPRA0
                                                                 r  data_RAM/RAM_reg_0_15_16_16/DP/RADR0
                         RAMD32 (Prop_ramd32_RADR0_O)
                                                    0.295     4.265 r  data_RAM/RAM_reg_0_15_16_16/DP/O
                         net (fo=2, unplaced)       1.122     5.387    user_fir/old_ram_data_reg_reg[31]_0[16]
                                                                 r  user_fir/sm_tdata_reg1__0_i_1/I0
                         LUT5 (Prop_lut5_I0_O)      0.124     5.511 r  user_fir/sm_tdata_reg1__0_i_1/O
                         net (fo=1, unplaced)       0.800     6.311    user_fir/o_ram_data[16]
                                                                 r  user_fir/sm_tdata_reg1__0/A[16]
                         DSP48E1 (Prop_dsp48e1_A[16]_PCOUT[47])
                                                    4.036    10.347 r  user_fir/sm_tdata_reg1__0/PCOUT[47]
                         net (fo=1, unplaced)       0.055    10.402    user_fir/sm_tdata_reg1__0_n_106
                                                                 r  user_fir/sm_tdata_reg1__1/PCIN[47]
                         DSP48E1 (Prop_dsp48e1_PCIN[47]_P[0])
                                                    1.518    11.920 r  user_fir/sm_tdata_reg1__1/P[0]
                         net (fo=2, unplaced)       0.800    12.720    user_fir/sm_tdata_reg1__1_n_105
                                                                 r  user_fir/sm_tdata_reg1_carry_i_3/I0
                         LUT2 (Prop_lut2_I0_O)      0.124    12.844 r  user_fir/sm_tdata_reg1_carry_i_3/O
                         net (fo=1, unplaced)       0.000    12.844    user_fir/sm_tdata_reg1_carry_i_3_n_0
                                                                 r  user_fir/sm_tdata_reg1_carry/S[1]
                         CARRY4 (Prop_carry4_S[1]_CO[3])
                                                    0.533    13.377 r  user_fir/sm_tdata_reg1_carry/CO[3]
                         net (fo=1, unplaced)       0.009    13.386    user_fir/sm_tdata_reg1_carry_n_0
                                                                 r  user_fir/sm_tdata_reg1_carry__0/CI
                         CARRY4 (Prop_carry4_CI_CO[3])
                                                    0.117    13.503 r  user_fir/sm_tdata_reg1_carry__0/CO[3]
                         net (fo=1, unplaced)       0.000    13.503    user_fir/sm_tdata_reg1_carry__0_n_0
                                                                 r  user_fir/sm_tdata_reg1_carry__1/CI
                         CARRY4 (Prop_carry4_CI_O[3])
                                                    0.331    13.834 r  user_fir/sm_tdata_reg1_carry__1/O[3]
                         net (fo=1, unplaced)       0.618    14.452    user_fir/sm_tdata_reg1_carry__1_n_4
                                                                 r  user_fir/sm_tdata_reg0_carry__5_i_1/I1
                         LUT2 (Prop_lut2_I1_O)      0.307    14.759 r  user_fir/sm_tdata_reg0_carry__5_i_1/O
                         net (fo=1, unplaced)       0.000    14.759    user_fir/sm_tdata_reg0_carry__5_i_1_n_0
                                                                 r  user_fir/sm_tdata_reg0_carry__5/S[3]
                         CARRY4 (Prop_carry4_S[3]_CO[3])
                                                    0.376    15.135 r  user_fir/sm_tdata_reg0_carry__5/CO[3]
                         net (fo=1, unplaced)       0.000    15.135    user_fir/sm_tdata_reg0_carry__5_n_0
                                                                 r  user_fir/sm_tdata_reg0_carry__6/CI
                         CARRY4 (Prop_carry4_CI_O[1])
                                                    0.337    15.472 r  user_fir/sm_tdata_reg0_carry__6/O[1]
                         net (fo=1, unplaced)       0.000    15.472    user_fir/sm_tdata_reg0_carry__6_n_6
                         FDRE                                        r  user_fir/sm_tdata_reg_reg[29]/D
  -------------------------------------------------------------------    -------------------

                         (clock wb_clk_i rise edge)
                                                   16.000    16.000 r
                                                    0.000    16.000 r  wb_clk_i (IN)
                         net (fo=0)                 0.000    16.000    wb_clk_i
                                                                 r  wb_clk_i_IBUF_inst/I
                         IBUF (Prop_ibuf_I_O)       0.838    16.838 r  wb_clk_i_IBUF_inst/O
                         net (fo=1, unplaced)       0.760    17.598    wb_clk_i_IBUF
                                                                 r  wb_clk_i_IBUF_BUFG_inst/I
                         BUFG (Prop_bufg_I_O)       0.091    17.689 r  wb_clk_i_IBUF_BUFG_inst/O
                         net (fo=401, unplaced)     0.439    18.128    user_fir/wb_clk_i_IBUF_BUFG
                         FDRE                                        r  user_fir/sm_tdata_reg_reg[29]/C
                         clock pessimism            0.184    18.311
                         clock uncertainty         -0.035    18.276
                         FDRE (Setup_fdre_C_D)      0.076    18.352    user_fir/sm_tdata_reg_reg[29]
  -------------------------------------------------------------------
                         required time                       18.352
                         arrival time                       -15.472
  -------------------------------------------------------------------
                         slack                               2.880
```

➢ Synthesis report

```
Utilization Design Information

Table of Contents
-----------------
1. Slice Logic
1.1 Summary of Registers by Type
2. Memory
3. DSP
4. IO and GT Specific
5. Clocking
6. Specific Feature
7. Primitives
8. Black Boxes
9. Instantiated Netlists

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

+----------------------------+------+-------+------------+-----------+-------+
|          Site Type         | Used | Fixed | Prohibited | Available | Util% |
+----------------------------+------+-------+------------+-----------+-------+
| Slice LUTs*                |  530 |     0 |          0 |     53200 |  1.00 |
|   LUT as Logic             |  402 |     0 |          0 |     53200 |  0.76 |
|   LUT as Memory            |  128 |     0 |          0 |     17400 |  0.74 |
|     LUT as Distributed RAM |  128 |     0 |            |           |       |
|     LUT as Shift Register  |    0 |     0 |            |           |       |
| Slice Registers            |  265 |     0 |          0 |    106400 |  0.25 |
|   Register as Flip Flop    |  265 |     0 |          0 |    106400 |  0.25 |
|   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.


1.1 Summary of Registers by Type
--------------------------------

+-------+--------------+-------------+--------------+
| Total | Clock Enable | Synchronous | Asynchronous |
+-------+--------------+-------------+--------------+
| 0     |            _ |           - |            - |
| 0     |            _ |           - |          Set |
| 0     |            _ |           - |        Reset |
| 0     |            _ |         Set |            - |
| 0     |            _ |       Reset |            - |
| 0     |          Yes |           - |            - |
| 0     |          Yes |           - |          Set |
| 0     |          Yes |           - |        Reset |
| 5     |          Yes |         Set |            - |
| 260   |          Yes |       Reset |            - |
+-------+--------------+-------------+--------------+


2. Memory
---------

+-------------------+------+-------+------------+-----------+-------+
|     Site Type     | Used | Fixed | Prohibited | Available | Util% |
+-------------------+------+-------+------------+-----------+-------+
| Block RAM Tile    |    4 |     0 |          0 |       140 |  2.86 |
|   RAMB36/FIFO*    |    4 |     0 |          0 |       140 |  2.86 |
|     RAMB36E1 only |    4 |       |            |           |       |
|   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

3. DSP
------

+----------------+------+-------+------------+-----------+-------+
|    Site Type   | Used | Fixed | Prohibited | Available | Util% |
+----------------+------+-------+------------+-----------+-------+
| DSPs           |    3 |     0 |          0 |       220 |  1.36 |
|   DSP48E1 only |    3 |       |            |           |       |
+----------------+------+-------+------------+-----------+-------+


4. IO and GT Specific
---------------------

+-----------------------------+------+-------+------------+-----------+--------+
|          Site Type          | Used | Fixed | Prohibited | Available |  Util% |
+-----------------------------+------+-------+------------+-----------+--------+
| Bonded IOB                  |  309 |     0 |          0 |       125 | 247.20 |
| Bonded IPADs                |    0 |     0 |          0 |         2 |   0.00 |
| Bonded IOPADs               |    0 |     0 |          0 |       130 |   0.00 |
| PHY_CONTROL                 |    0 |     0 |          0 |         4 |   0.00 |
| PHASER_REF                  |    0 |     0 |          0 |         4 |   0.00 |
| OUT_FIFO                    |    0 |     0 |          0 |        16 |   0.00 |
| IN_FIFO                     |    0 |     0 |          0 |        16 |   0.00 |
| IDELAYCTRL                  |    0 |     0 |          0 |         4 |   0.00 |
| IBUFDS                      |    0 |     0 |          0 |       121 |   0.00 |
| PHASER_OUT/PHASER_OUT_PHY   |    0 |     0 |          0 |        16 |   0.00 |
| PHASER_IN/PHASER_IN_PHY     |    0 |     0 |          0 |        16 |   0.00 |
| IDELAYE2/IDELAYE2_FINEDELAY |    0 |     0 |          0 |       200 |   0.00 |
| ILOGIC                      |    0 |     0 |          0 |       125 |   0.00 |
| OLOGIC                      |    0 |     0 |          0 |       125 |   0.00 |
+-----------------------------+------+-------+------------+-----------+--------+


5. Clocking
-----------

+------------+------+-------+------------+-----------+-------+
|  Site Type | Used | Fixed | Prohibited | Available | Util% |
+------------+------+-------+------------+-----------+-------+
| BUFGCTRL   |    1 |     0 |          0 |        32 |  3.13 |
| BUFIO      |    0 |     0 |          0 |        16 |  0.00 |
| MMCME2_ADV |    0 |     0 |          0 |         4 |  0.00 |
| PLLE2_ADV  |    0 |     0 |          0 |         4 |  0.00 |
| BUFMRCE    |    0 |     0 |          0 |         8 |  0.00 |
| BUFHCE     |    0 |     0 |          0 |        72 |  0.00 |
| BUFR       |    0 |     0 |          0 |        16 |  0.00 |
+------------+------+-------+------------+-----------+-------+


6. Specific Feature
-------------------

+-------------+------+-------+------------+-----------+-------+
|  Site Type  | Used | Fixed | Prohibited | Available | Util% |
+-------------+------+-------+------------+-----------+-------+
| BSCANE2     |    0 |     0 |          0 |         4 |  0.00 |
| CAPTUREE2   |    0 |     0 |          0 |         1 |  0.00 |
| DNA_PORT    |    0 |     0 |          0 |         1 |  0.00 |
| EFUSE_USR   |    0 |     0 |          0 |         1 |  0.00 |
| FRAME_ECCE2 |    0 |     0 |          0 |         1 |  0.00 |
| ICAPE2      |    0 |     0 |          0 |         2 |  0.00 |
| STARTUPE2   |    0 |     0 |          0 |         1 |  0.00 |
| XADC        |    0 |     0 |          0 |         1 |  0.00 |
+-------------+------+-------+------------+-----------+-------+


7. Primitives
-------------

+----------+------+----------------------+
| Ref Name | Used | Functional Category |
+----------+------+----------------------+
| FDRE     |  260 |         Flop & Latch |
| OBUFT    |  207 |                   IO |
| RAMD32   |  128 |  Distributed Memory |
| LUT2     |  116 |                  LUT |
| LUT4     |   88 |                  LUT |
| LUT6     |   85 |                  LUT |
| IBUF     |   69 |                   IO |
| LUT5     |   63 |                  LUT |
| LUT1     |   47 |                  LUT |
| CARRY4   |   40 |           CarryLogic |
| OBUF     |   33 |                   IO |
| LUT3     |   23 |                  LUT |
| FDSE     |    5 |         Flop & Latch |
| RAMB36E1 |    4 |         Block Memory |
| DSP48E1  |    3 |    Block Arithmetic |
| BUFG     |    1 |                Clock |
+----------+------+----------------------+
```

- Utilization

Reports | Design Runs | **Utilization** | Timing

Hierarchy

Name	Slice LUTs (53200)	Slice Registers (106400)	Block RAM Tile (140)	DSPs (220)	Bonded IOB (125)	BUFGCTRL (32)
user_proj_example	530	265	4	3	309	1
data_RAM (bram11)	64	4	0	0	0	0
exmem_bram (bram)	17	0	4	0	0	0
tap_RAM (bram11_0)	100	4	0	0	0	0
user_fir (fir)	309	156	0	3	0	0

- Timing Report

Period: 16ns

Design Runs | Timing

Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	2.880 ns	Worst Hold Slack (WHS):	0.072 ns	Worst Pulse Width Slack (WPWS):	6.750 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:	1229	Total Number of Endpoints:	1229	Total Number of Endpoints:	402

All user specified timing constraints are met.

Design Runs | Timing

Intra-Clock Paths - wb_clk_i - Setup

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clo
Path 1	2.880	11	12	32	data_RAM/r_A...ep_bsel[2]/C	user_fir/sm_t...eg_reg[29]/D	13.015	8.576	4.439	16.0	wb_clk_i
Path 2	2.886	11	12	32	data_RAM/r_A...ep_bsel[2]/C	user_fir/sm_t...eg_reg[31]/D	13.009	8.570	4.439	16.0	wb_clk_i
Path 3	2.961	11	12	32	data_RAM/r_A...ep_bsel[2]/C	user_fir/sm_t...eg_reg[30]/D	12.934	8.495	4.439	16.0	wb_clk_i
Path 4	2.985	11	12	32	data_RAM/r_A...ep_bsel[2]/C	user_fir/sm_t...eg_reg[28]/D	12.910	8.471	4.439	16.0	wb_clk_i
Path 5	2.997	10	11	32	data_RAM/r_A...ep_bsel[2]/C	user_fir/sm_t...eg_reg[25]/D	12.898	8.459	4.439	16.0	wb_clk_i
Path 6	3.003	10	11	32	data_RAM/r_A...ep_bsel[2]/C	user_fir/sm_t...eg_reg[27]/D	12.892	8.453	4.439	16.0	wb_clk_i
Path 7	3.078	10	11	32	data_RAM/r_A...ep_bsel[2]/C	user_fir/sm_t...eg_reg[26]/D	12.817	8.378	4.439	16.0	wb_clk_i
Path 8	3.102	10	11	32	data_RAM/r_A...ep_bsel[2]/C	user_fir/sm_t...eg_reg[24]/D	12.793	8.354	4.439	16.0	wb_clk_i
Path 9	3.227	9	10	32	data_RAM/r_A...ep_bsel[2]/C	user_fir/sm_t...eg_reg[21]/D	12.668	8.238	4.430	16.0	wb_clk_i
Path 10	3.233	9	10	32	data_RAM/r_A...ep_bsel[2]/C	user_fir/sm_t...eg_reg[23]/D	12.662	8.232	4.430	16.0	wb_clk_i

- Metrics: #-of-clock (latency-timer) * clock-period * gate-resource(LUT+FF)
  ➔ 72 * 25 * (530+265) = 1,431,000

心得:

在 lab4-2 中需要搭配前 2 次的 lab，Lab3 和 Lab4-1 分別為 fir.v 和 firmware(fir_control.c) 要做出 Caravel FIR 須將他們整合，透過 wishbone 以及專換成 AXI 能進行溝通，在 testbench 中再透過 mprj pin 給出訊號來聯絡。經過這次的練習讓我們對 Caravel SOC 的 protocal 更加的熟悉了。在做 Lab4-2 過程中在 synthesis 的 timing settime 是有通過的，但 holdtime 卻會 failed，有試著增加 clock period 來使其能夠符合，但仍然為 failed。後來嘗試將板子由 KV2600 改成 PNYQ 板子即可成功的 timing。