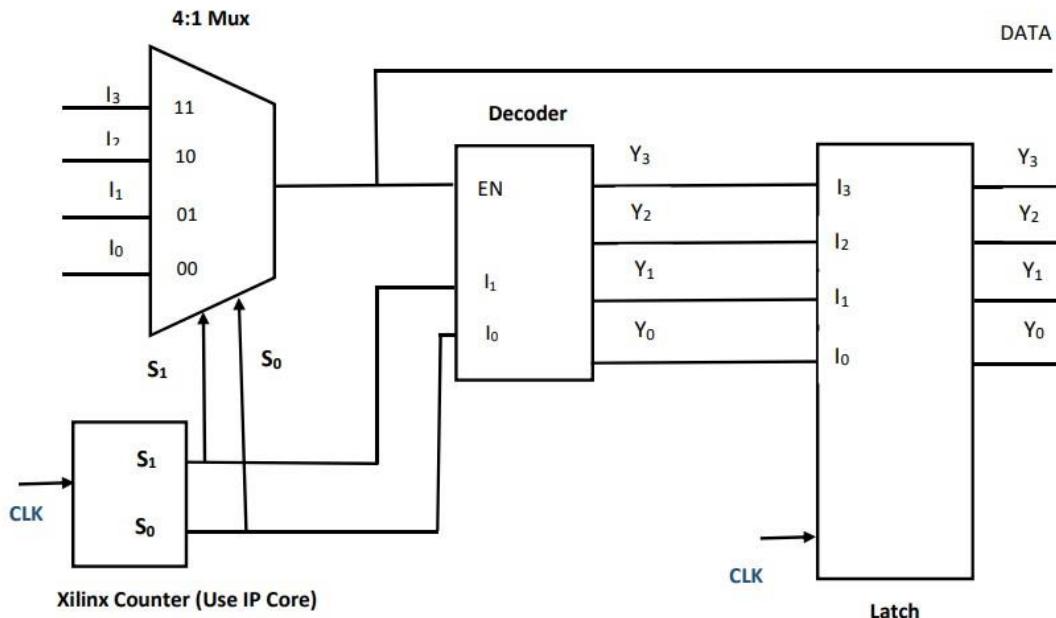


FPGA Workshop Miniproject

-Narrar Hemanth Reddy

Aim: Code the below design in Verilog HDL and implement the same on the FPGA Kit allotted



I have been allotted XC7A100T-1CSG324C and my board is F66A.

Verilog Code:

Note: Commented the outputs of vio to avoid multiple drivers.

```
module miniproject_nielit(input user_clk,
    //input user_rst,
    //input a3,
    //input a2,
    //input a1,
    //input a0,
    output data,
        output [3:0]x,
        output wire [3:0]y);
    wire [1:0]t;
    counter_binary_0 your_instance_name (
        .CLK(user_clk), // input wire CLK
        .SCLR(user_rst), // input wire SCLR
```

```

.Q(t)      // output wire [1 : 0] Q
);

ila_0 instance_name (
    .clk(user_clk), // input wire clk
    .probe0(x), // input wire [3:0] probe0
    .probe1(y), // input wire [3:0] probe1
    .probe2(data), // input wire [0:0] probe2
    .probe3(user_rst), // input wire [0:0] probe3
    .probe4(a3), // input wire [0:0] probe4
    .probe5(a2), // input wire [0:0] probe5
    .probe6(a1), // input wire [0:0] probe6
    .probe7(a0) // input wire [0:0] probe7
);

vio_0 your_instance (
    .clk(user_clk),           // input wire clk
    .probe_out0(user_rst), // output wire [0 : 0] probe_out0
    .probe_out1(a1), // output wire [0 : 0] probe_out1
    .probe_out2(a2), // output wire [0 : 0] probe_out2
    .probe_out3(a3), // output wire [0 : 0] probe_out3
    .probe_out4(a0) // output wire [0 : 0] probe_out4
);

mux4_1 multiplex4_1(.a3(a3),.a2(a2),.a1(a1),.a0(a0),.t(t),.mux_out(data));
dec2_4 decoder2_4(.b(t),.en(data),.w(x));
d_latch latch(.d(x),.user_clk(user_clk),.q(y));

endmodule

module mux4_1(input a3,input a2,input a1,input a0,input [1:0]t,output mux_out);
    assign mux_out=t[1]? (t[0] ? a3:a2):(t[0]? a1:a0);
endmodule

module dec2_4(input [1:0] b,input en,output [3:0]w);
    always @(b or en)
        begin
            if(!en) w=4'b0000;
            else
                begin

```

```

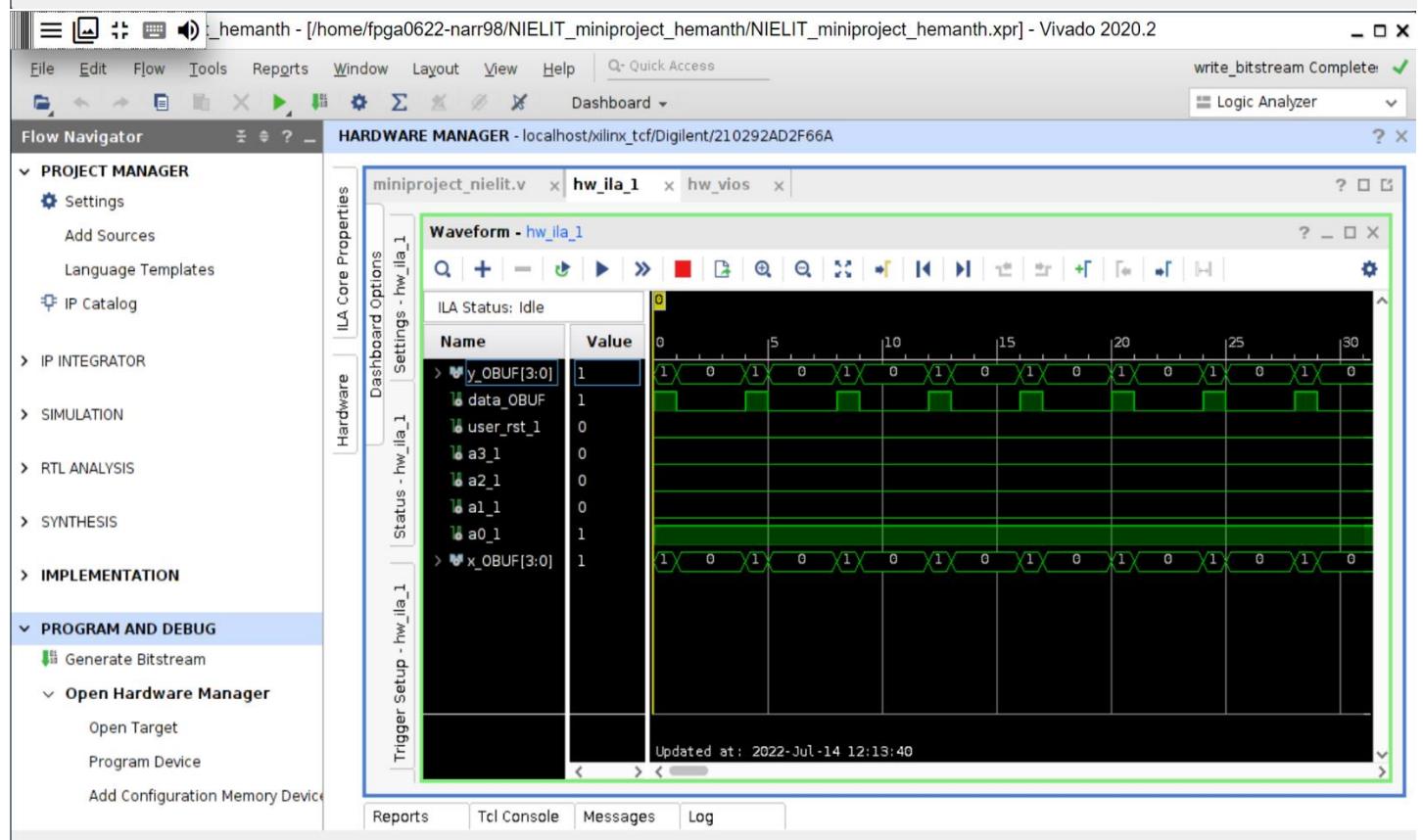
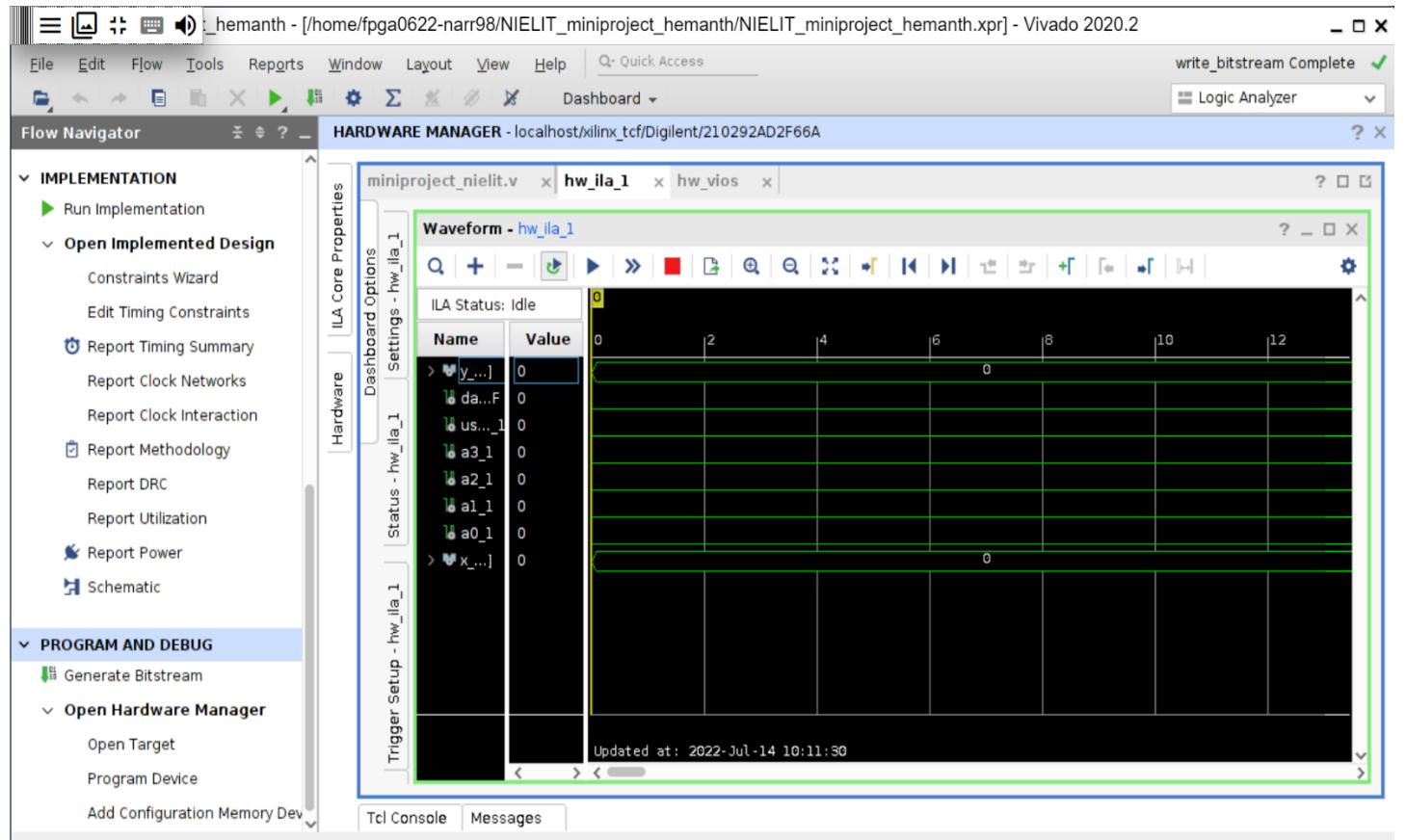
case(b)
2'b00: w=4'b0001;
2'b01: w=4'b0010;
2'b10: w=4'b0100;
2'b11: w=4'b1000;
default: w=4'bzzzz;
endcase
end
end

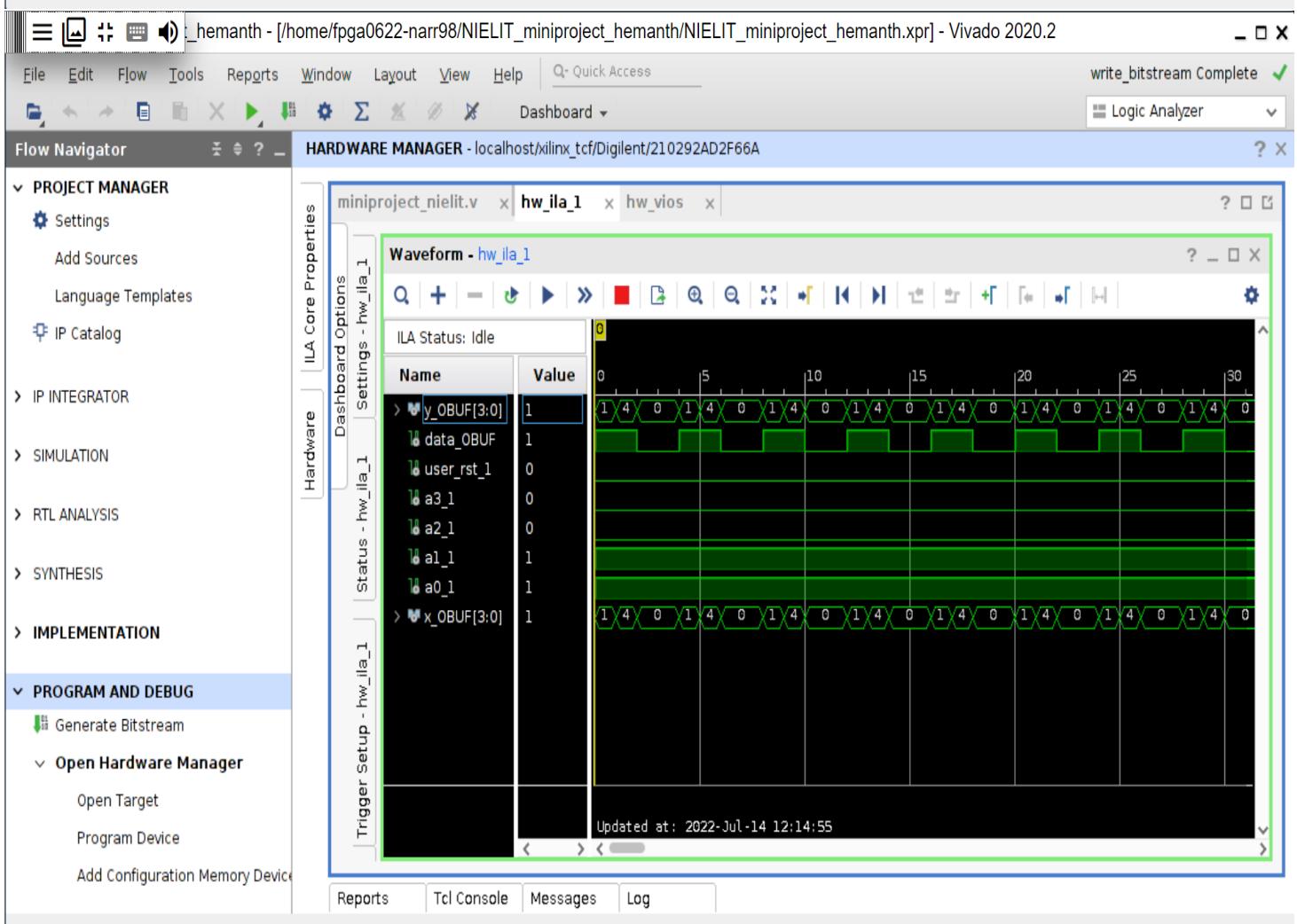
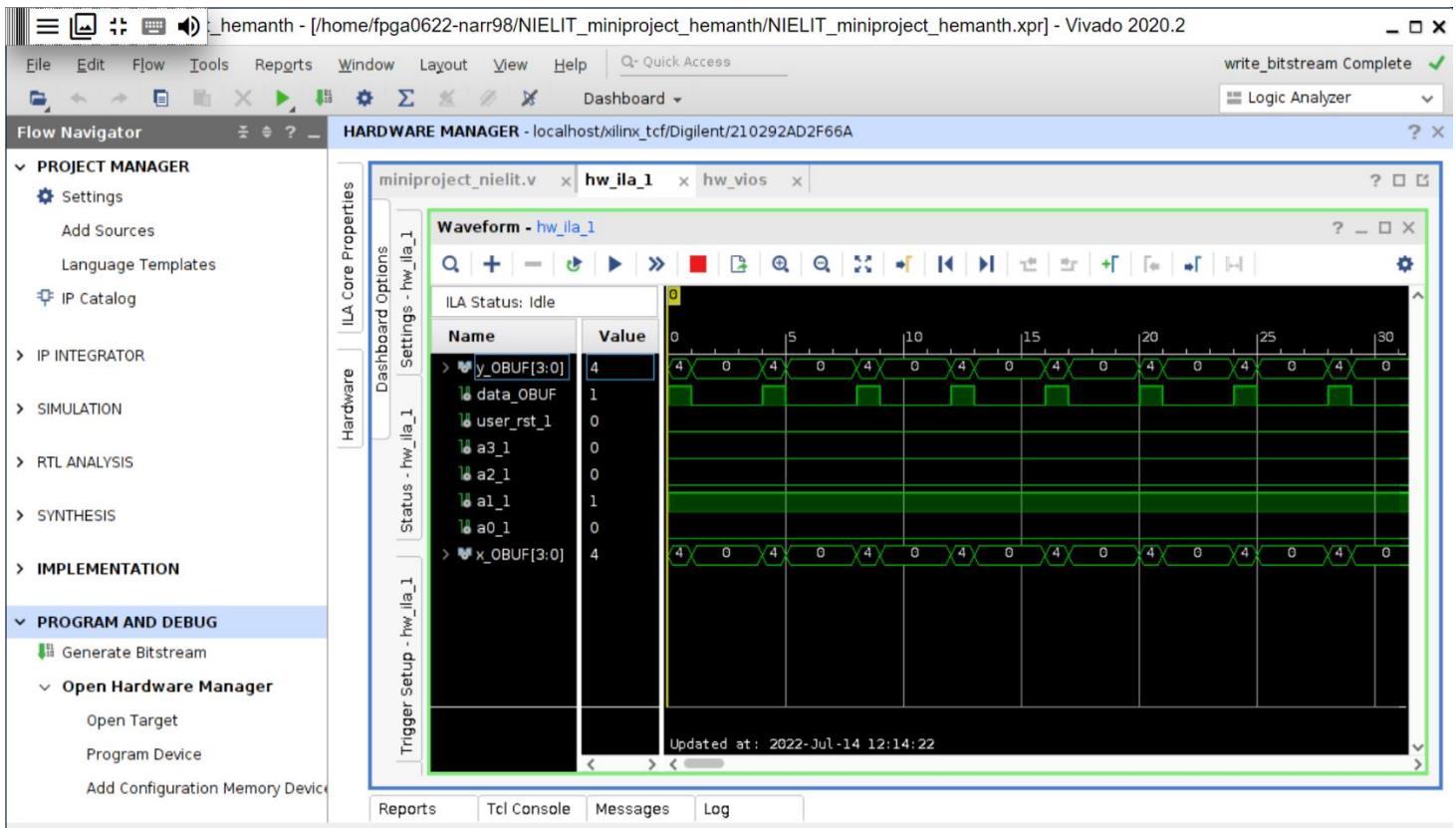
endmodule

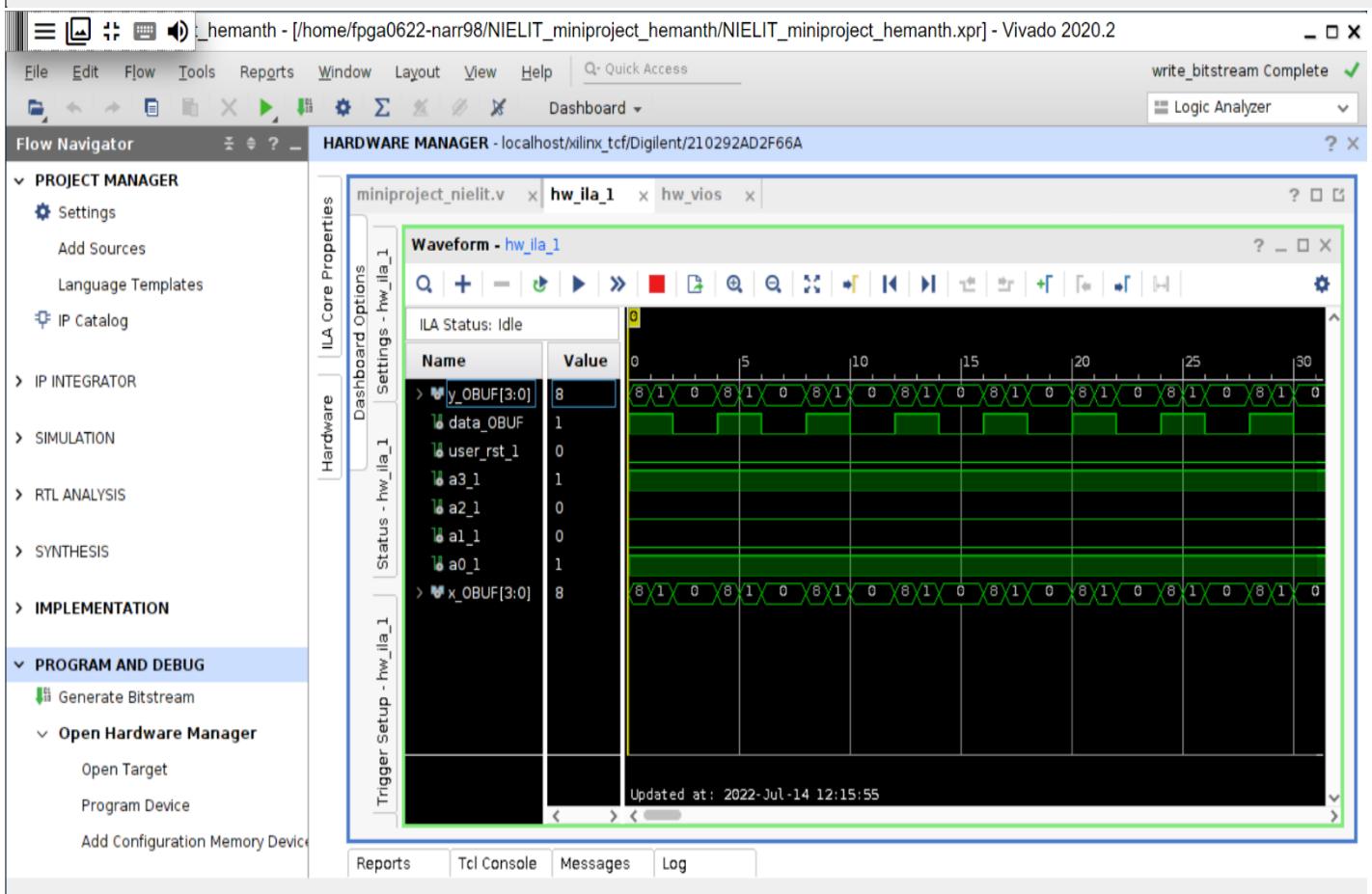
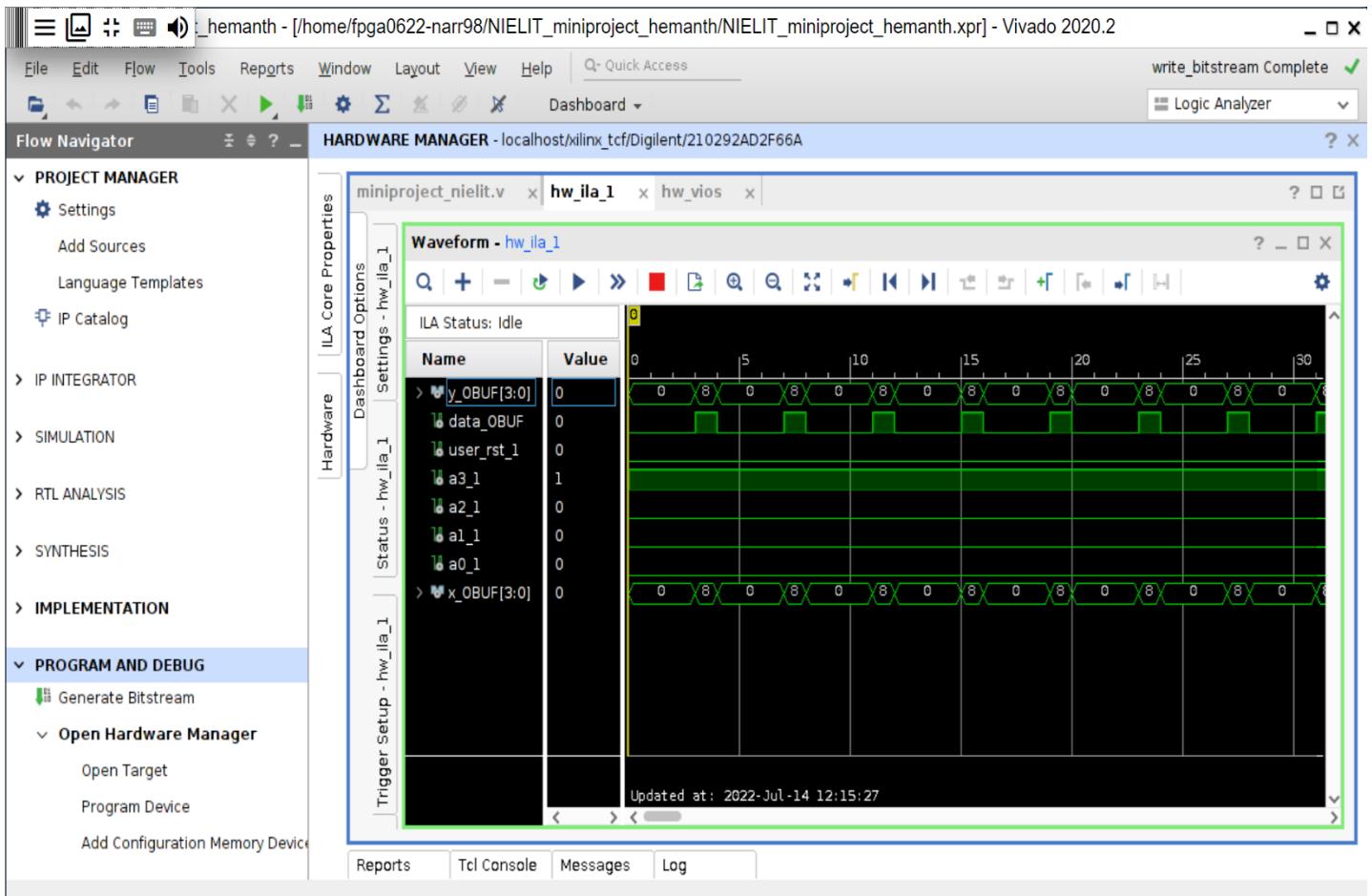
module d_latch(input [3:0]d,input user_clk,output reg [3:0]q);
always @(user_clk or d)
begin
  if(user_clk)
    q<=d;
  else if(!user_clk)
    q<=q;
  else
    q<=4'bzzzz;
end
endmodule

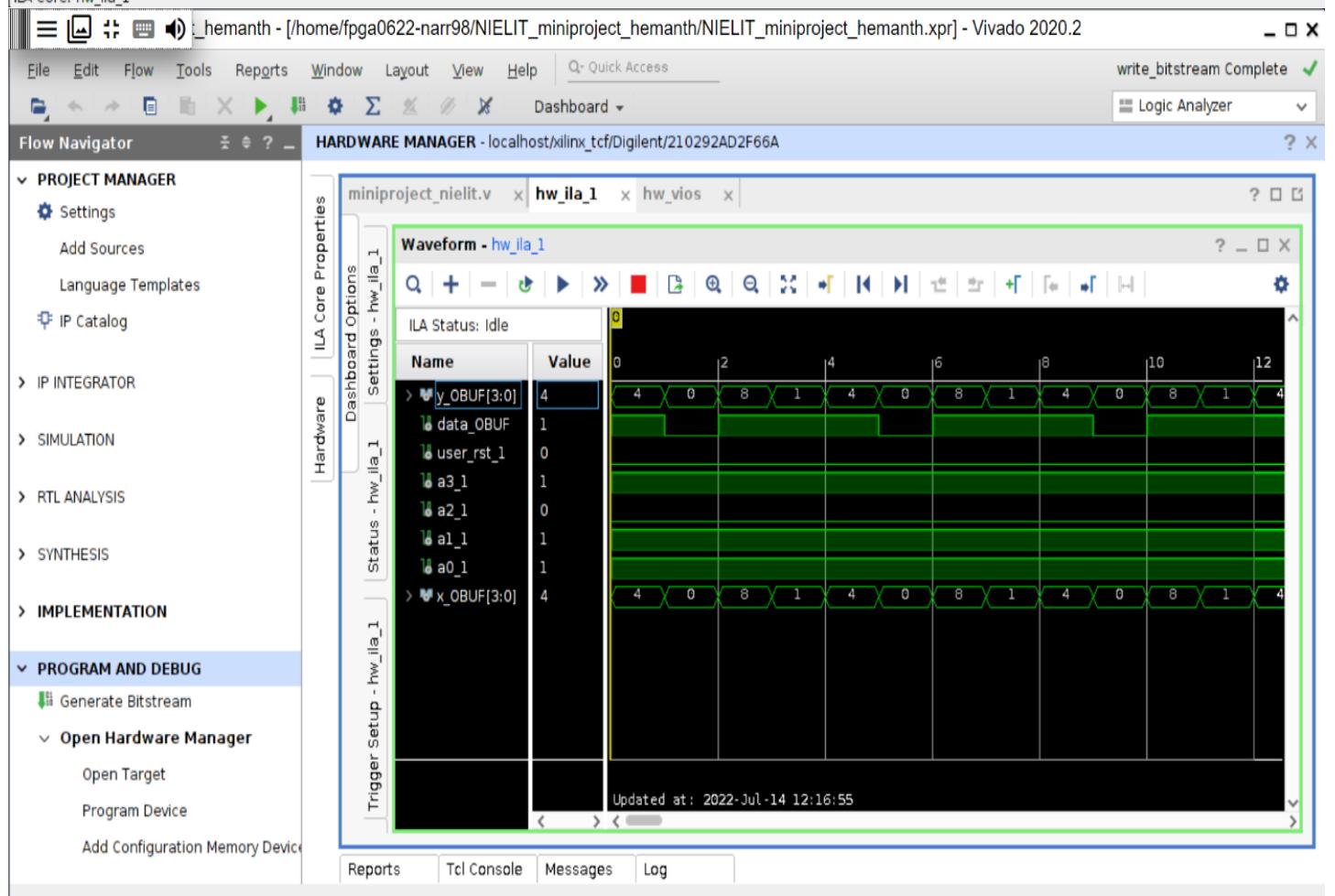
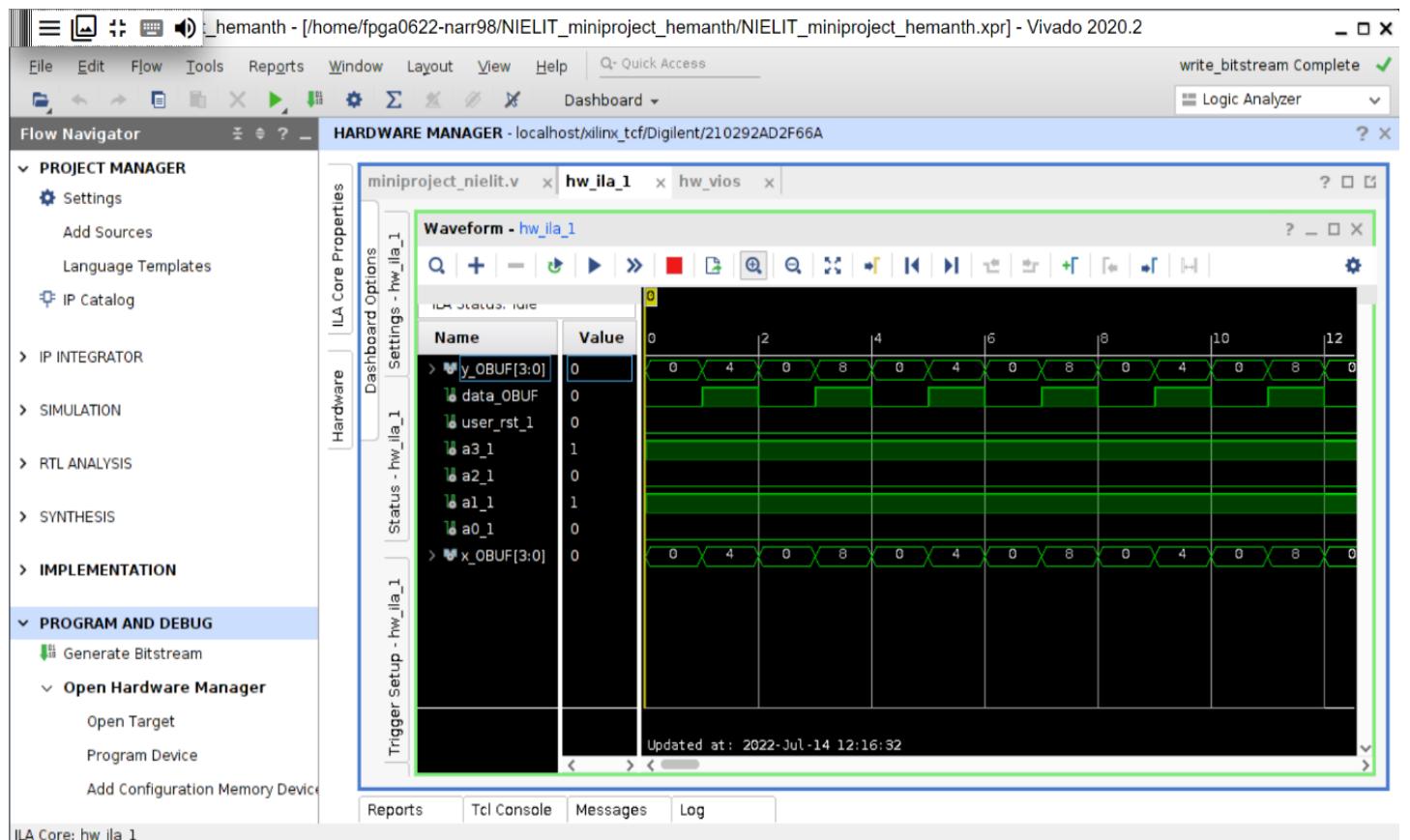
```

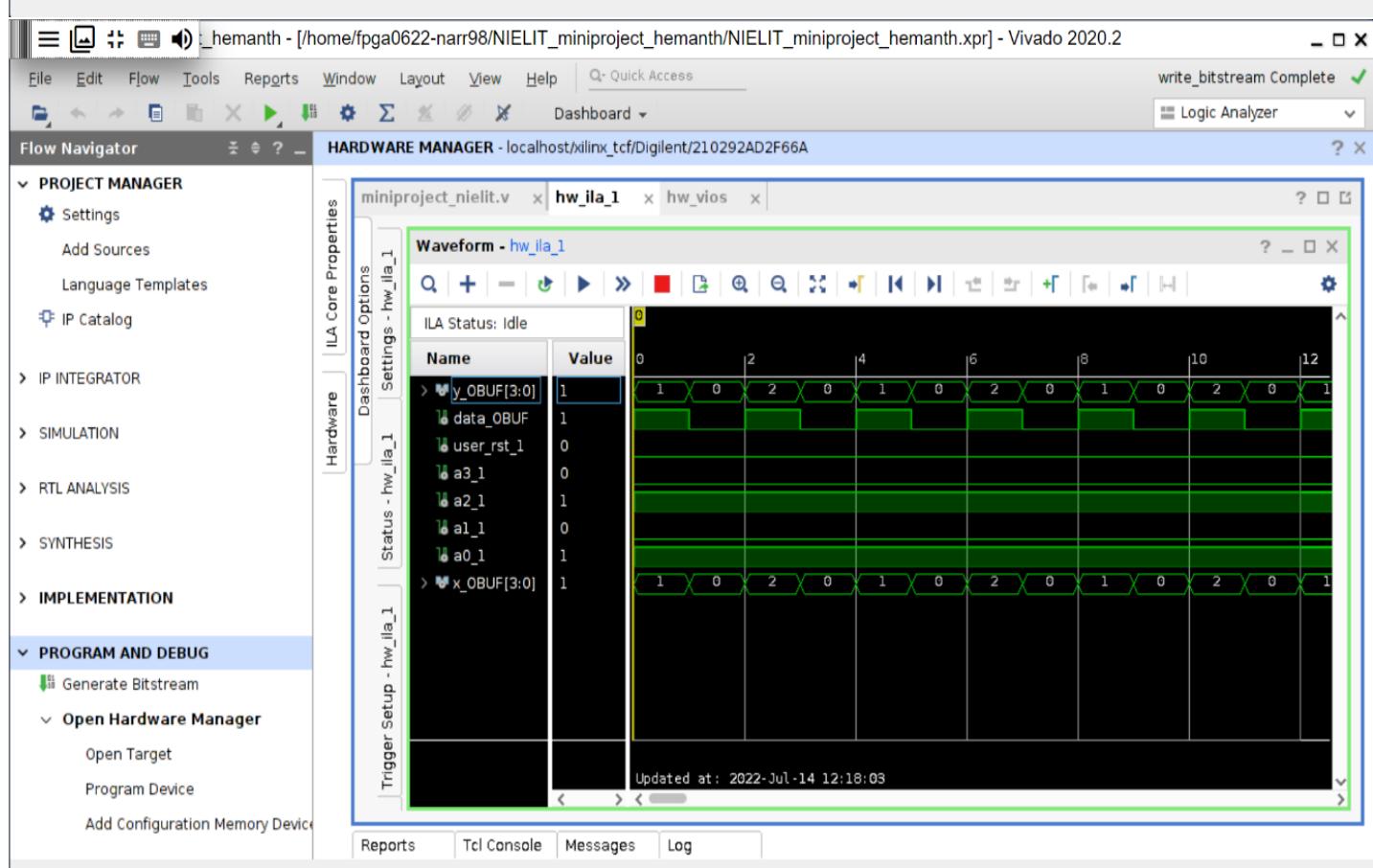
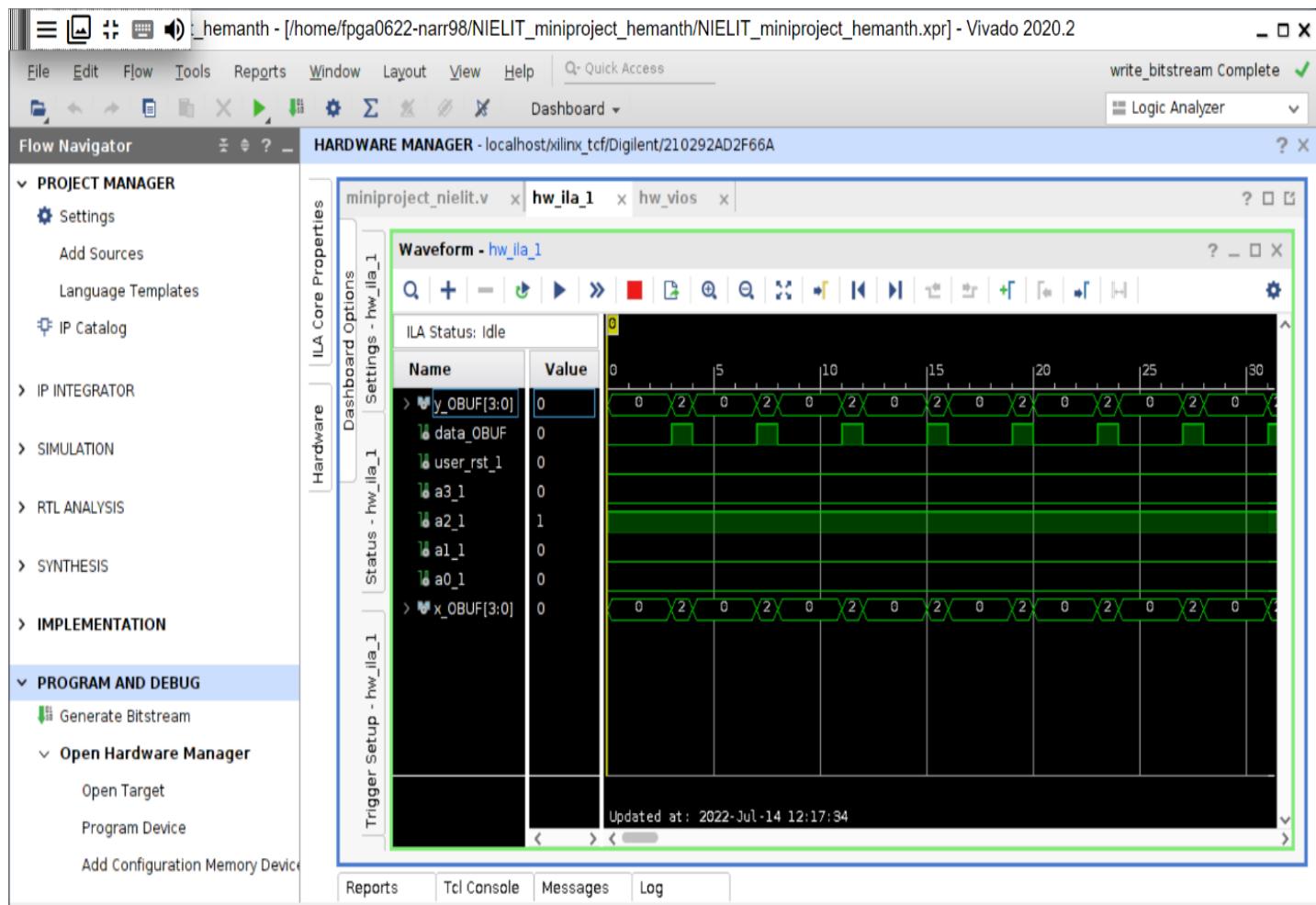
Waveforms and project image:

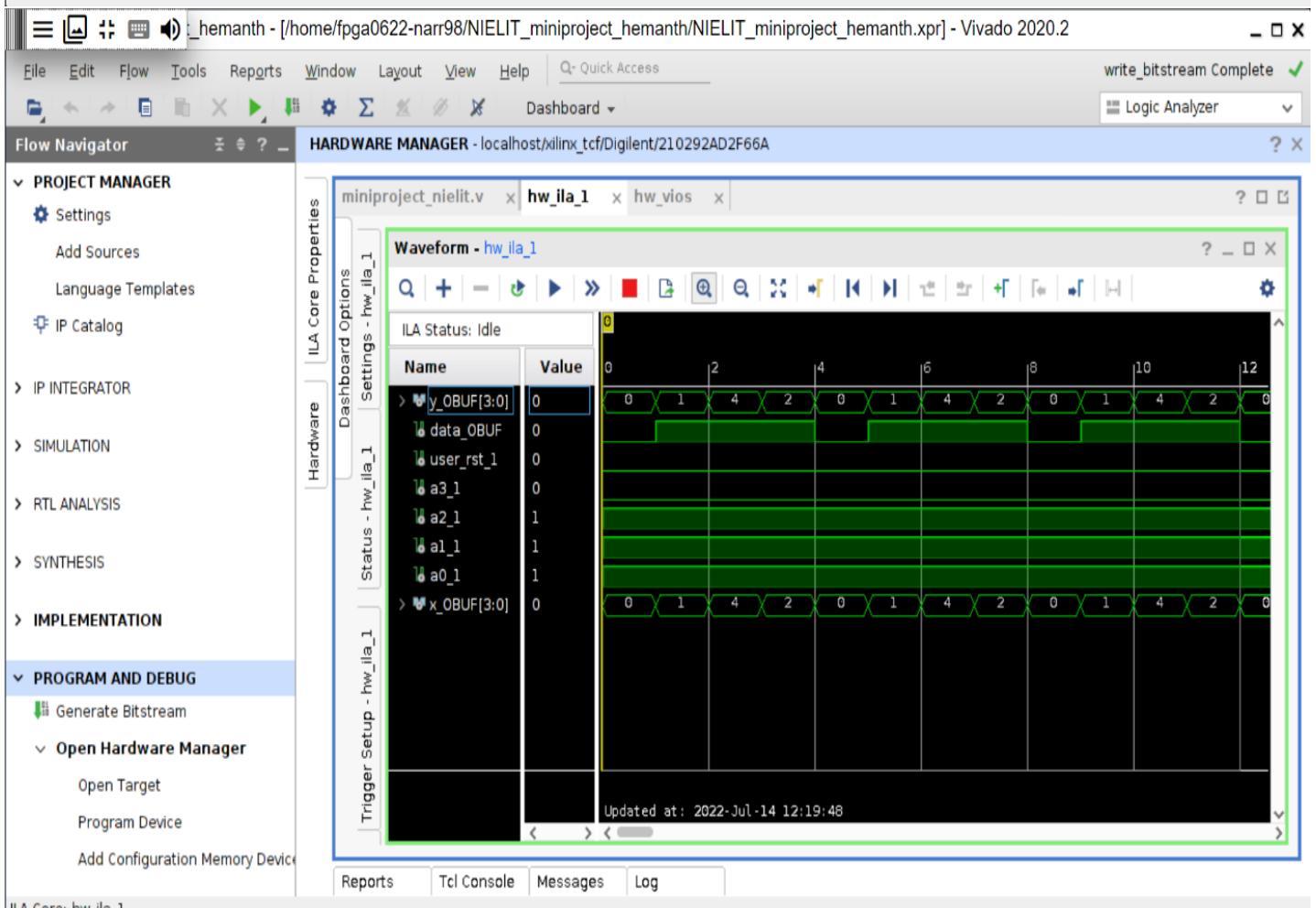
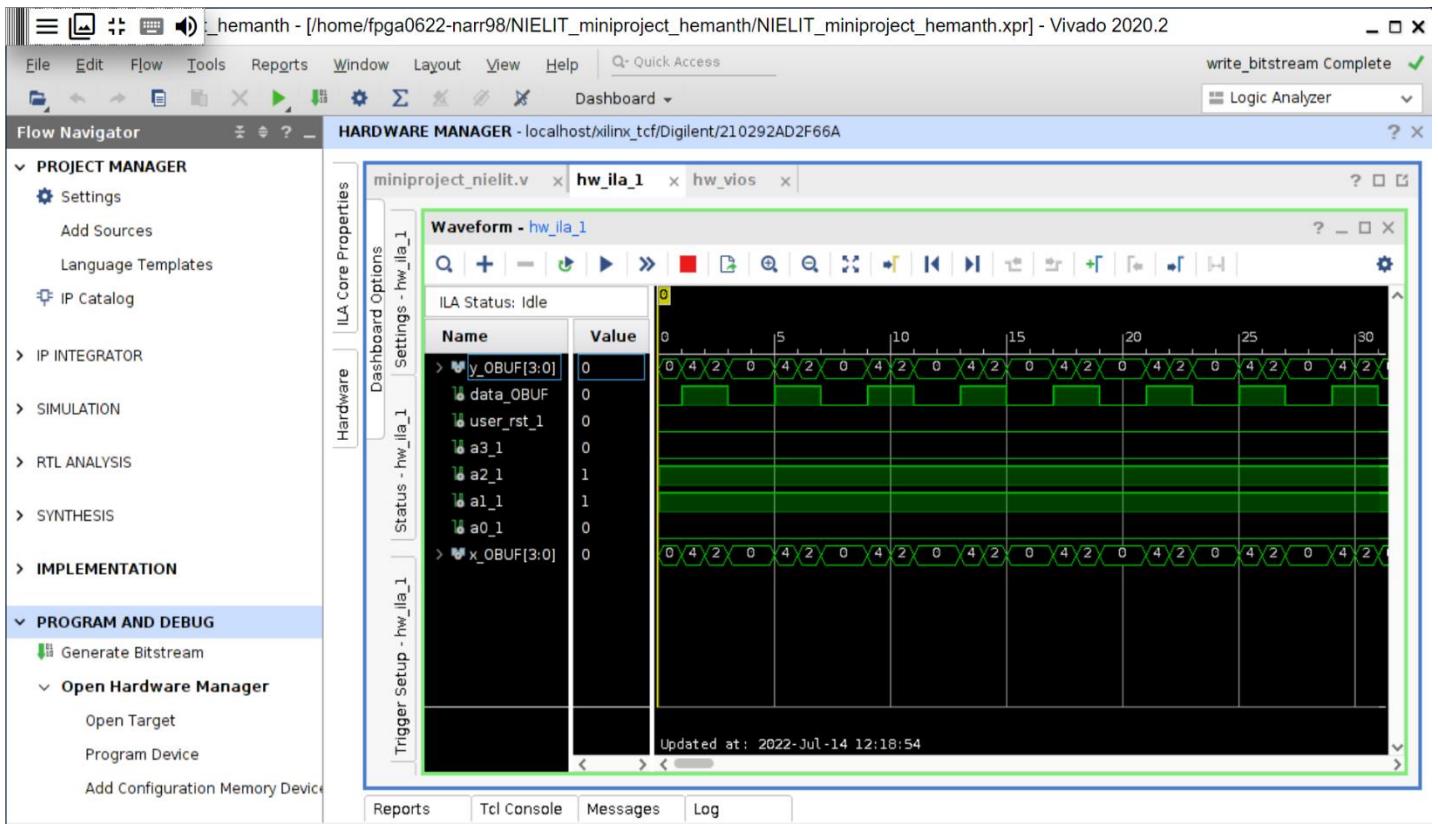




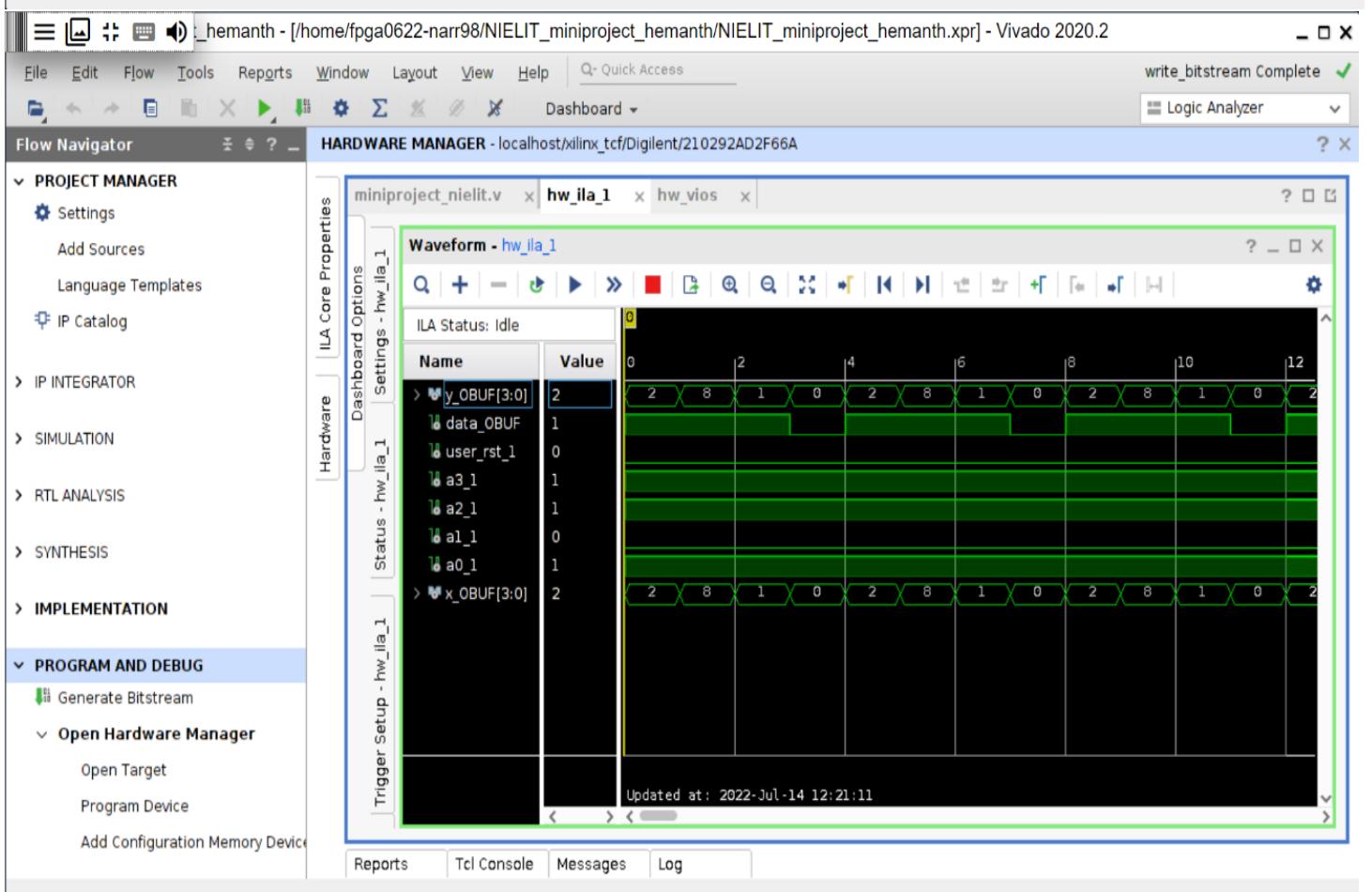
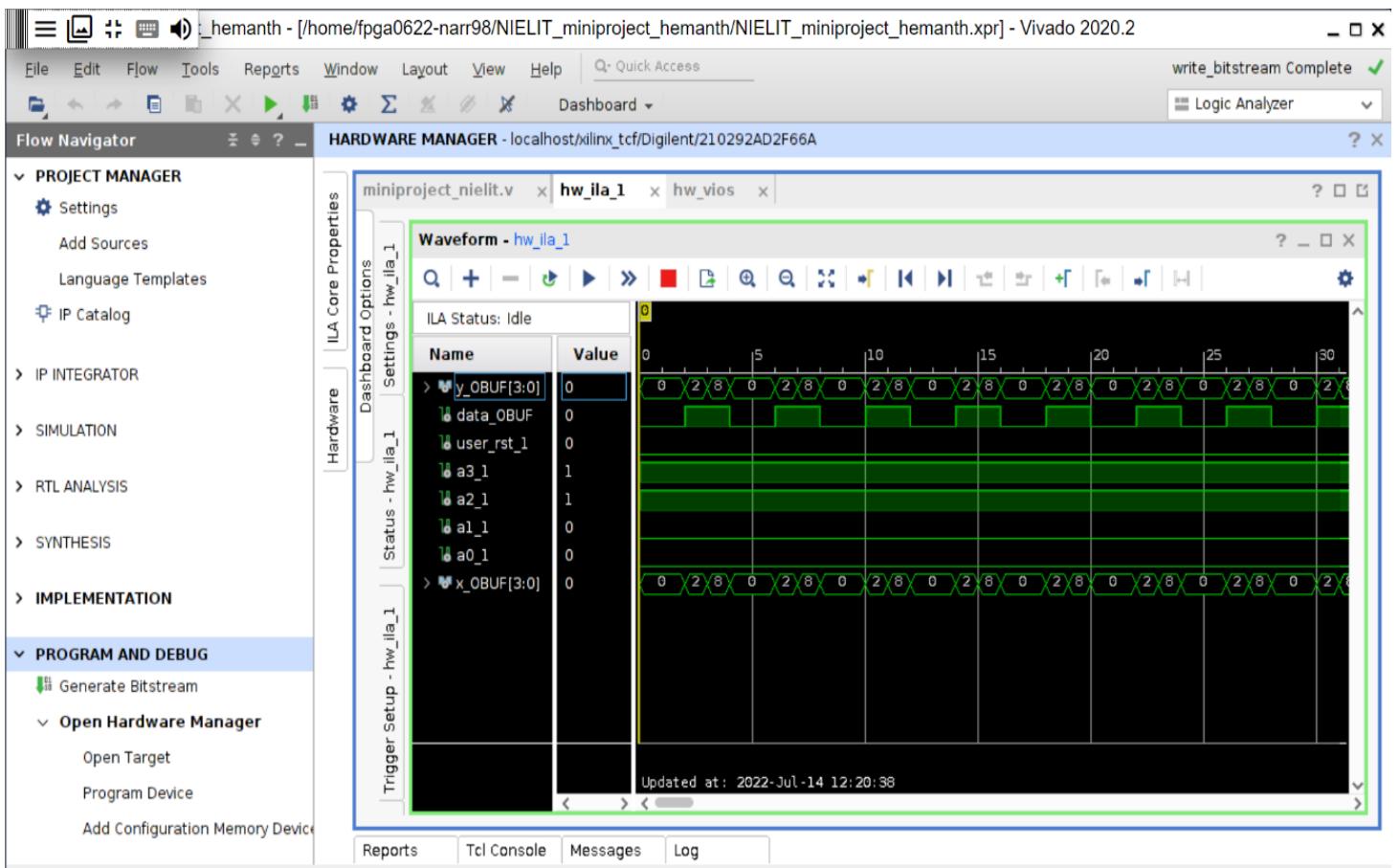


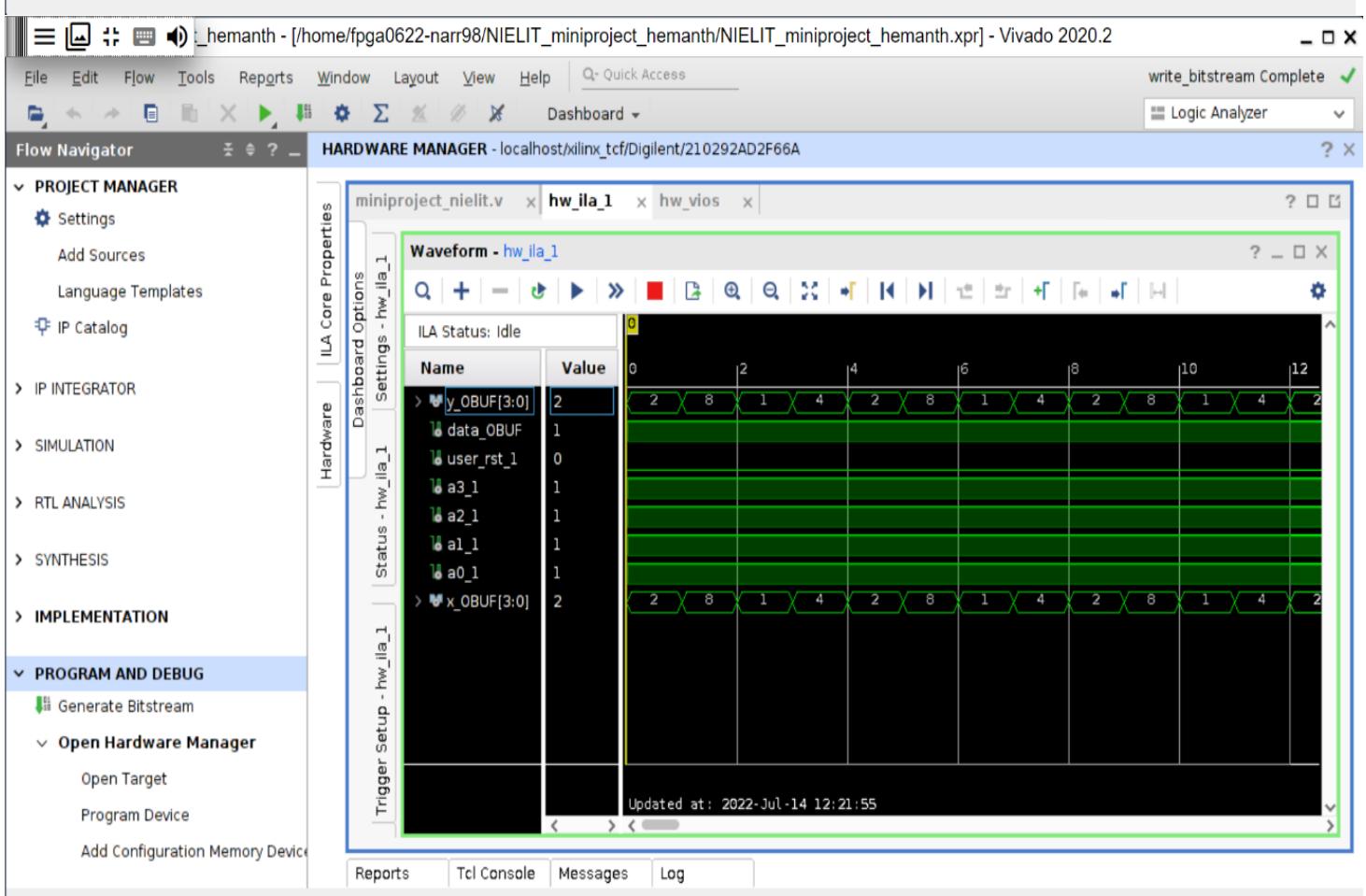
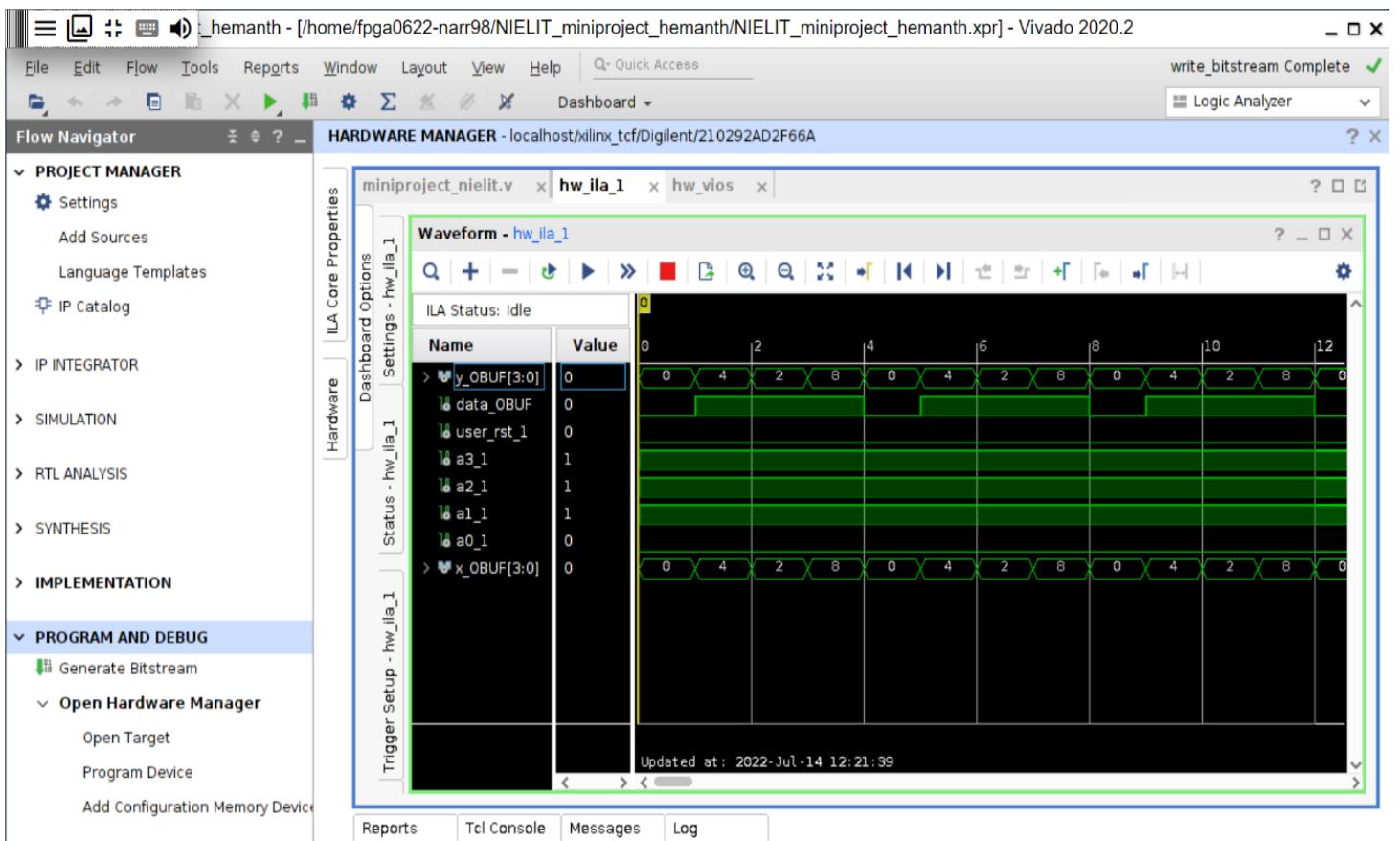


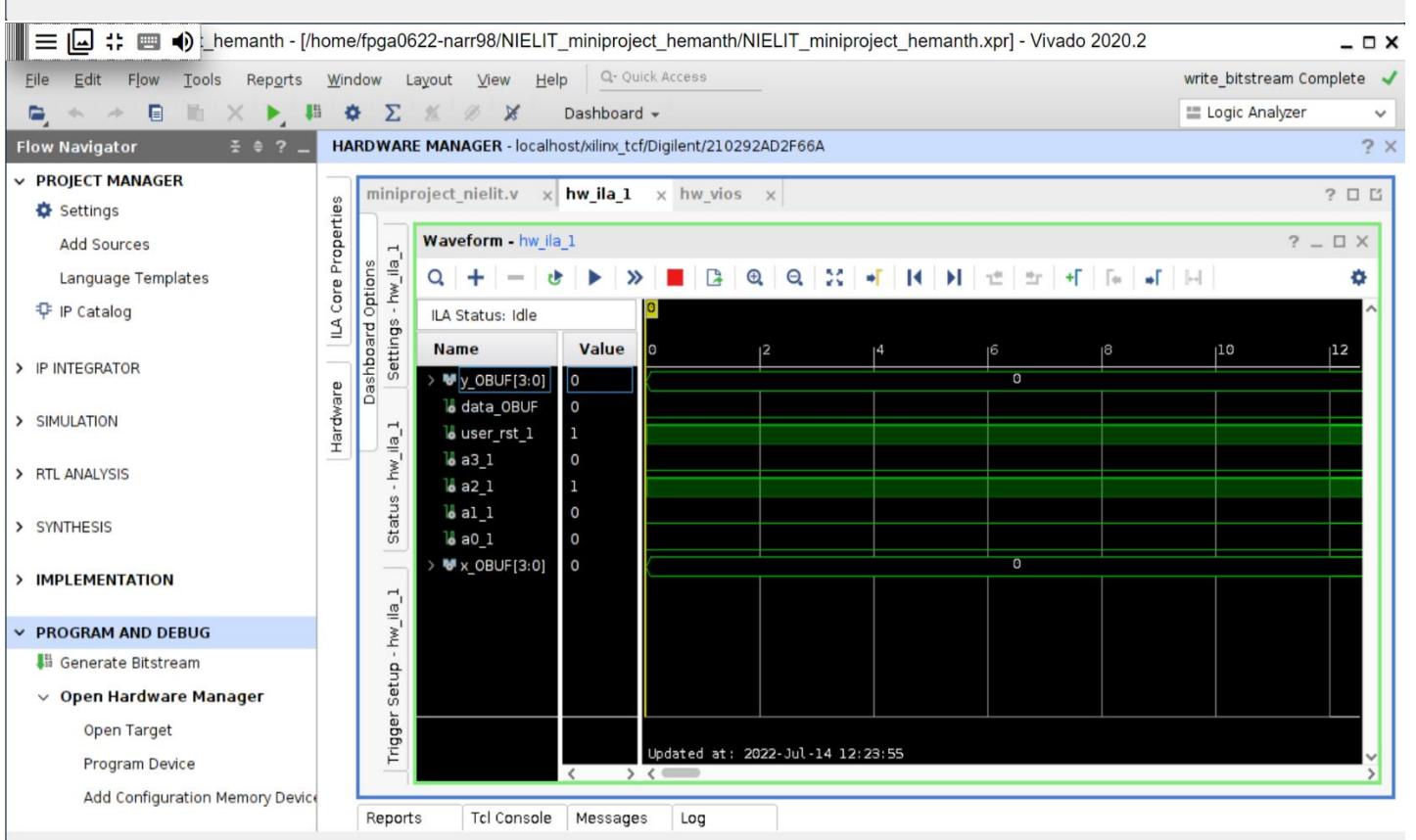
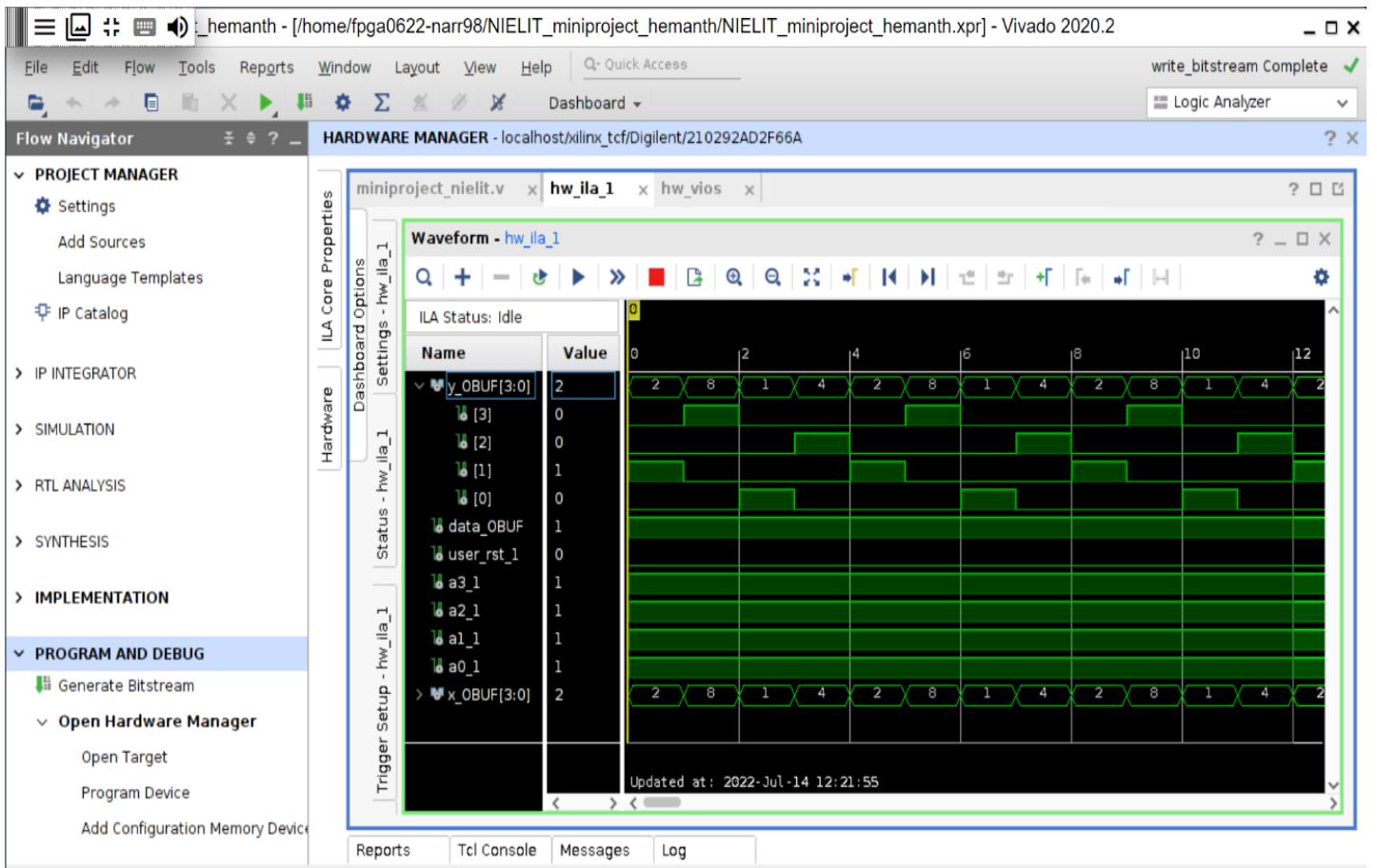


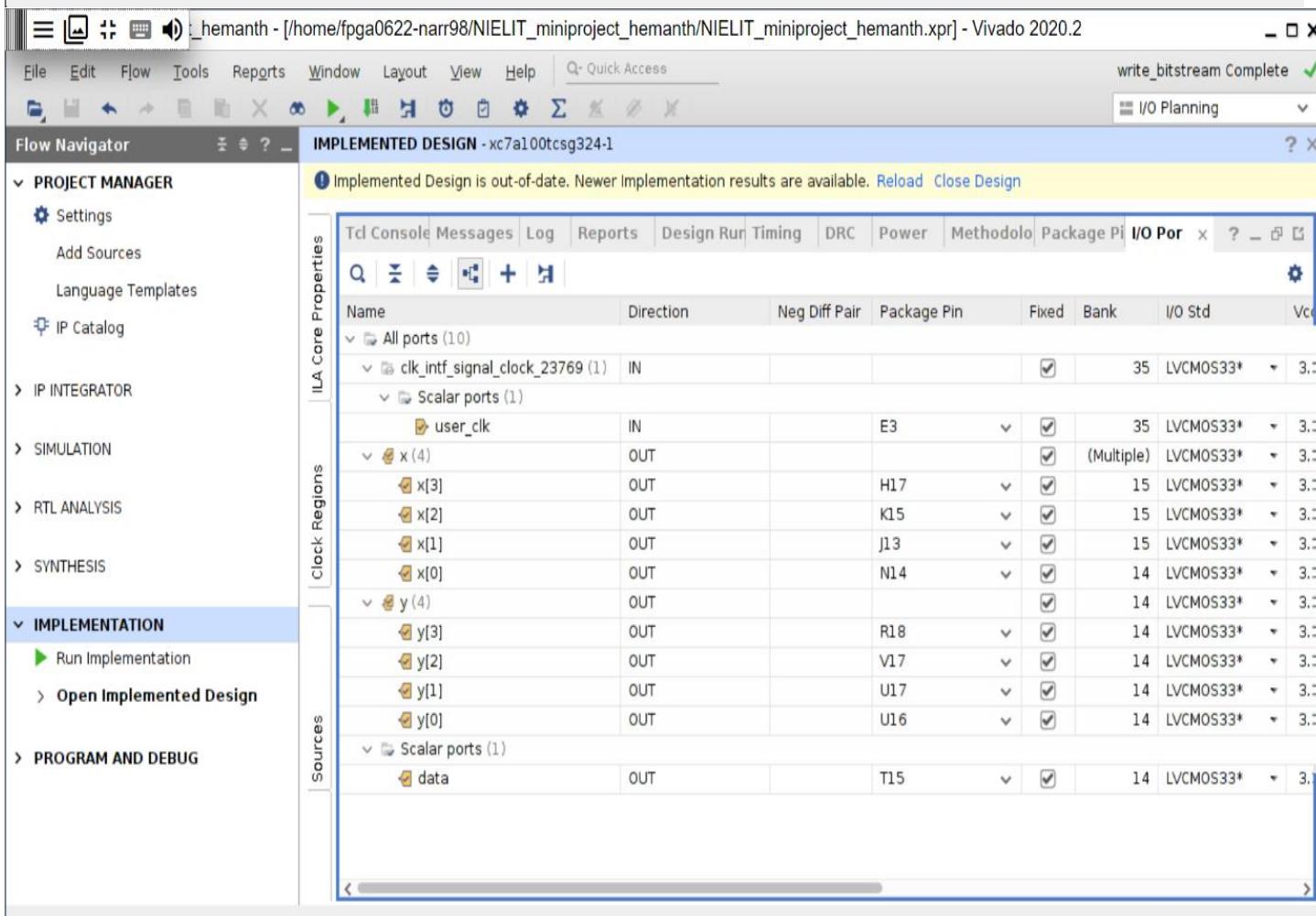
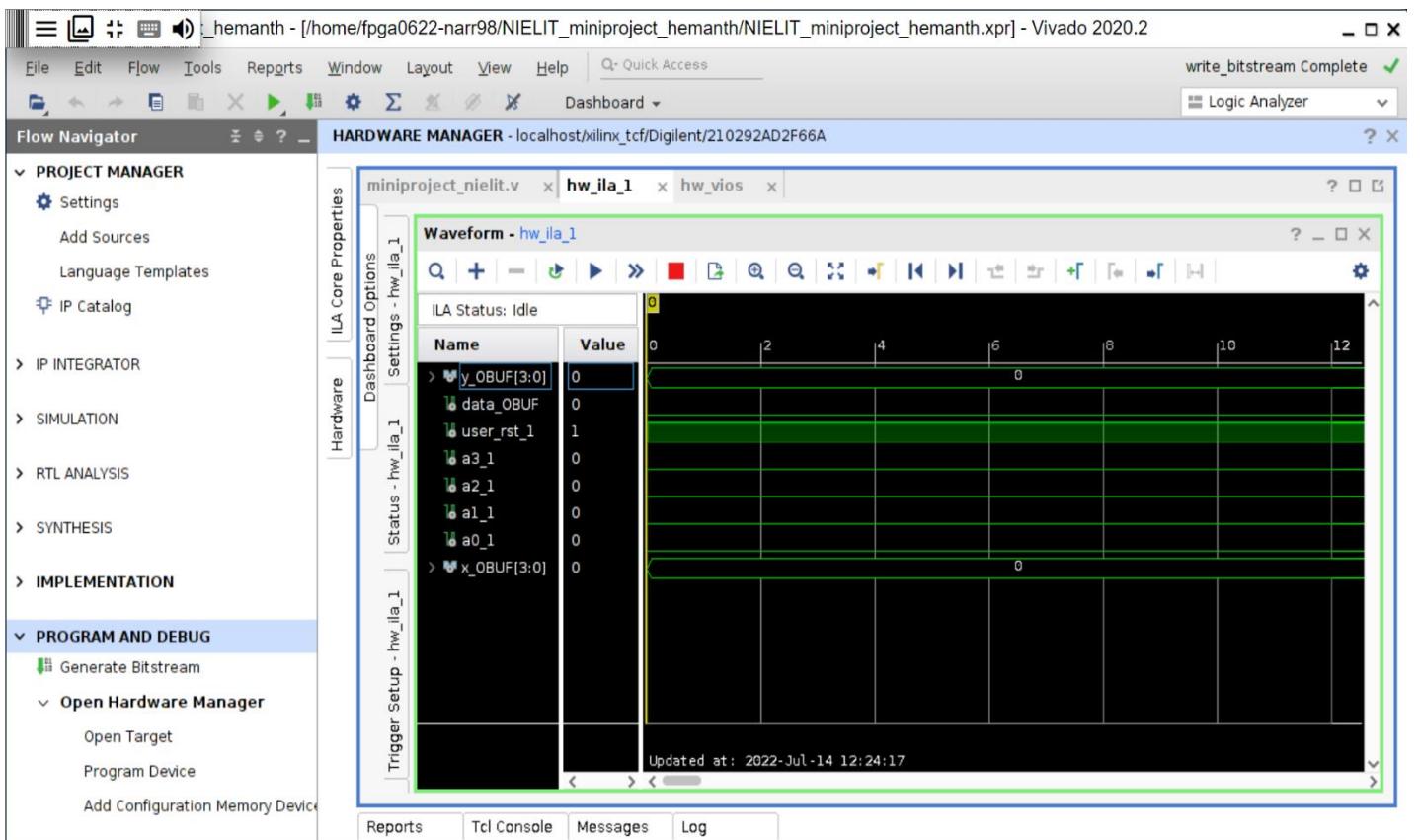


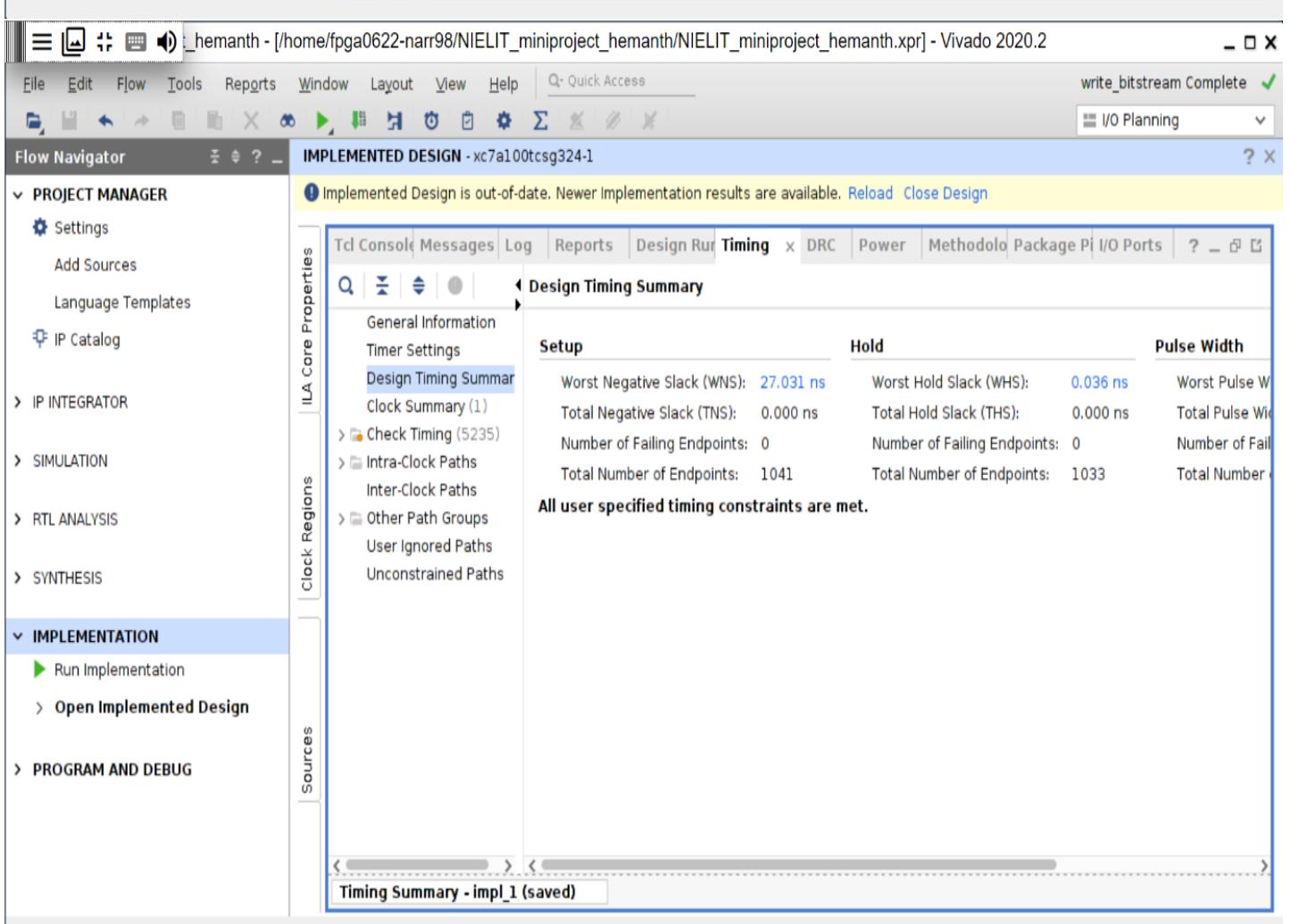
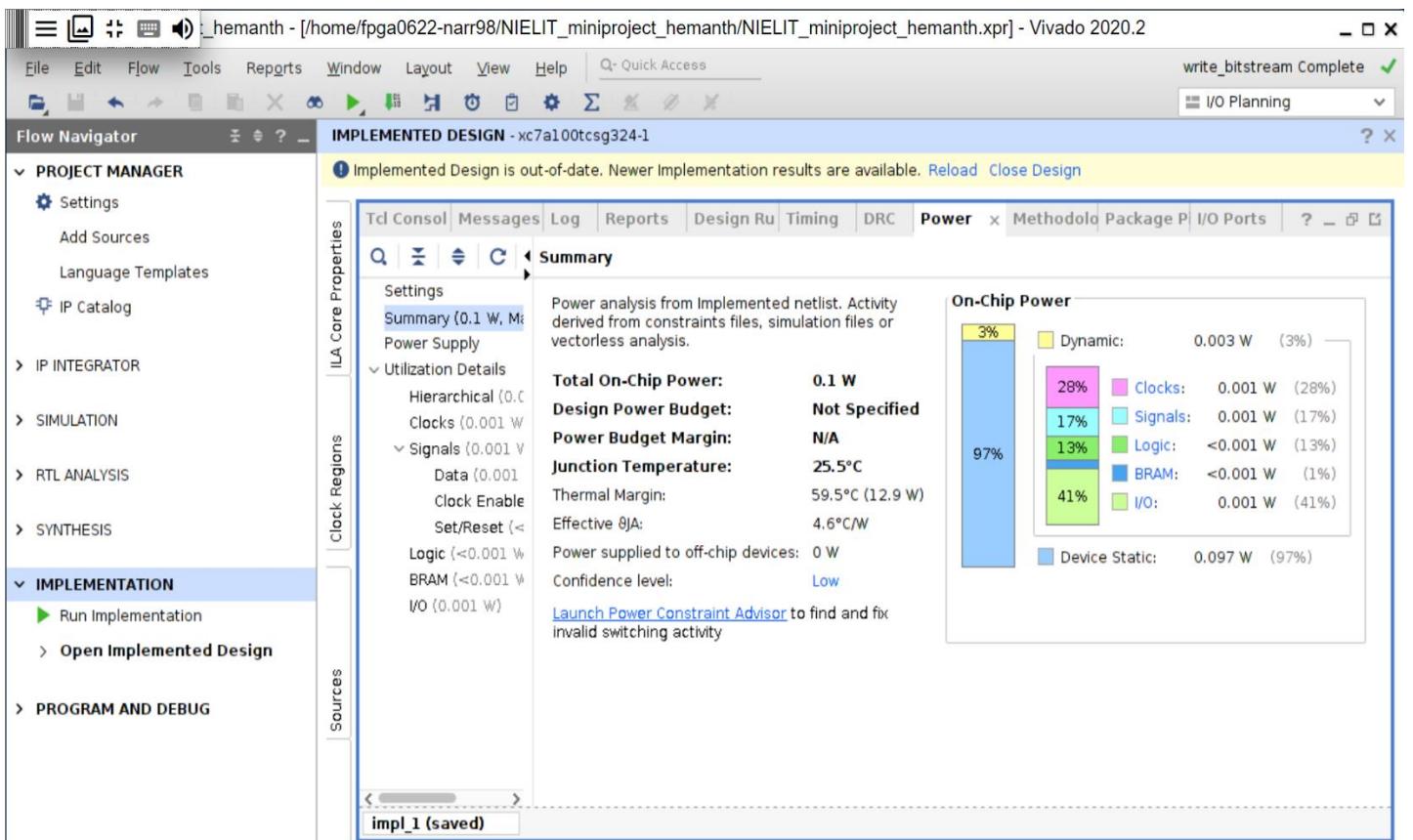
ILA Core: hw_ilia_1

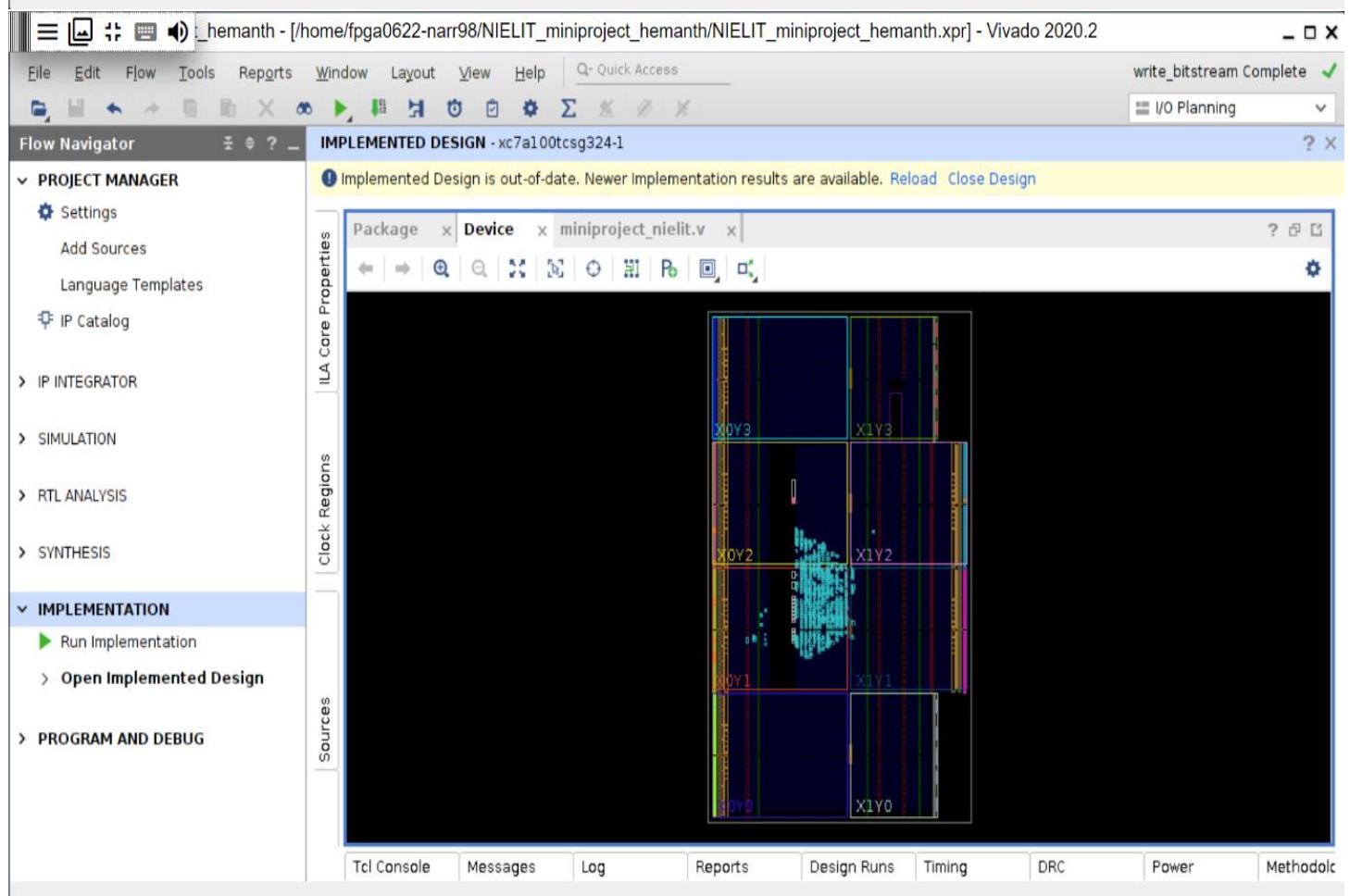
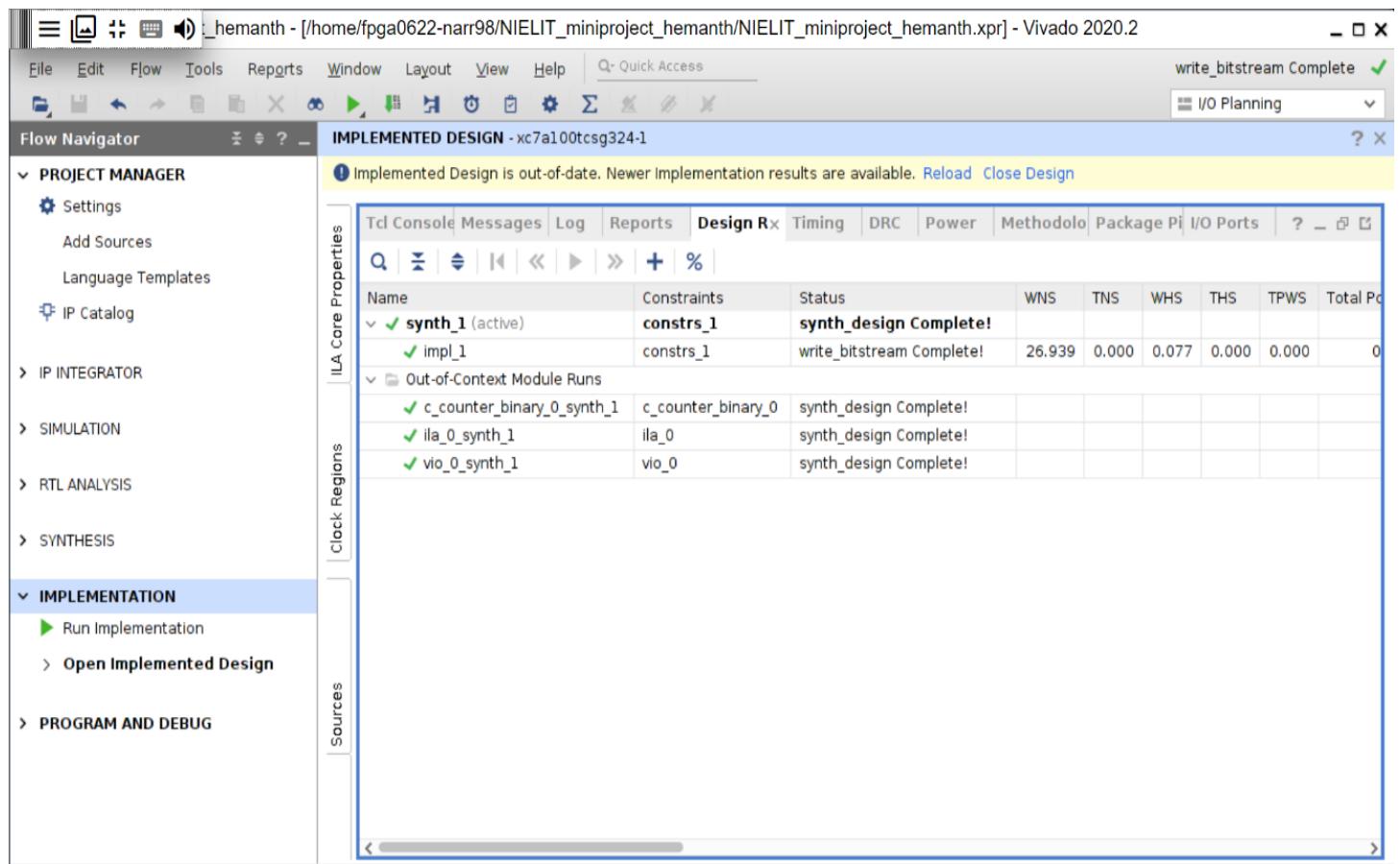


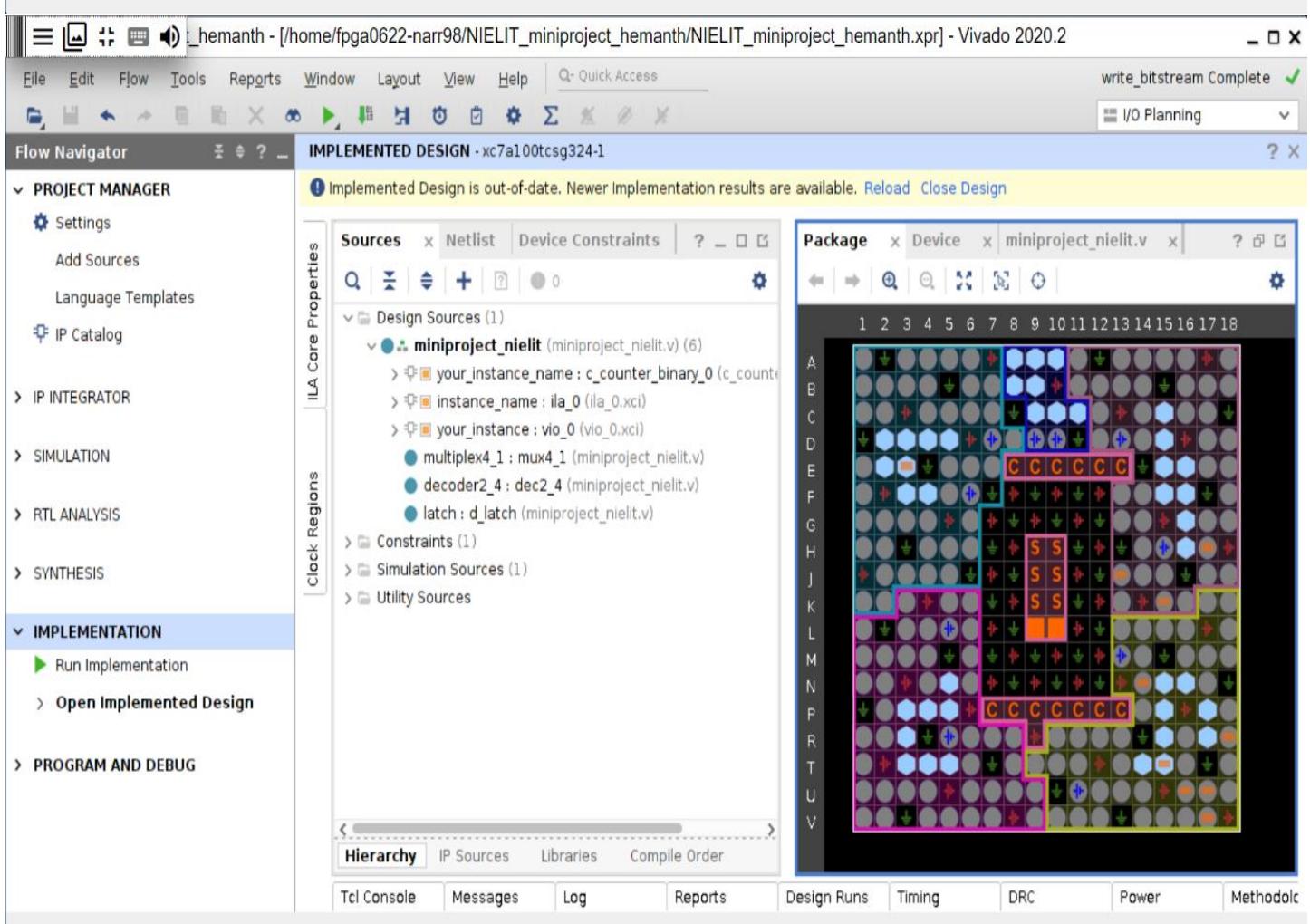
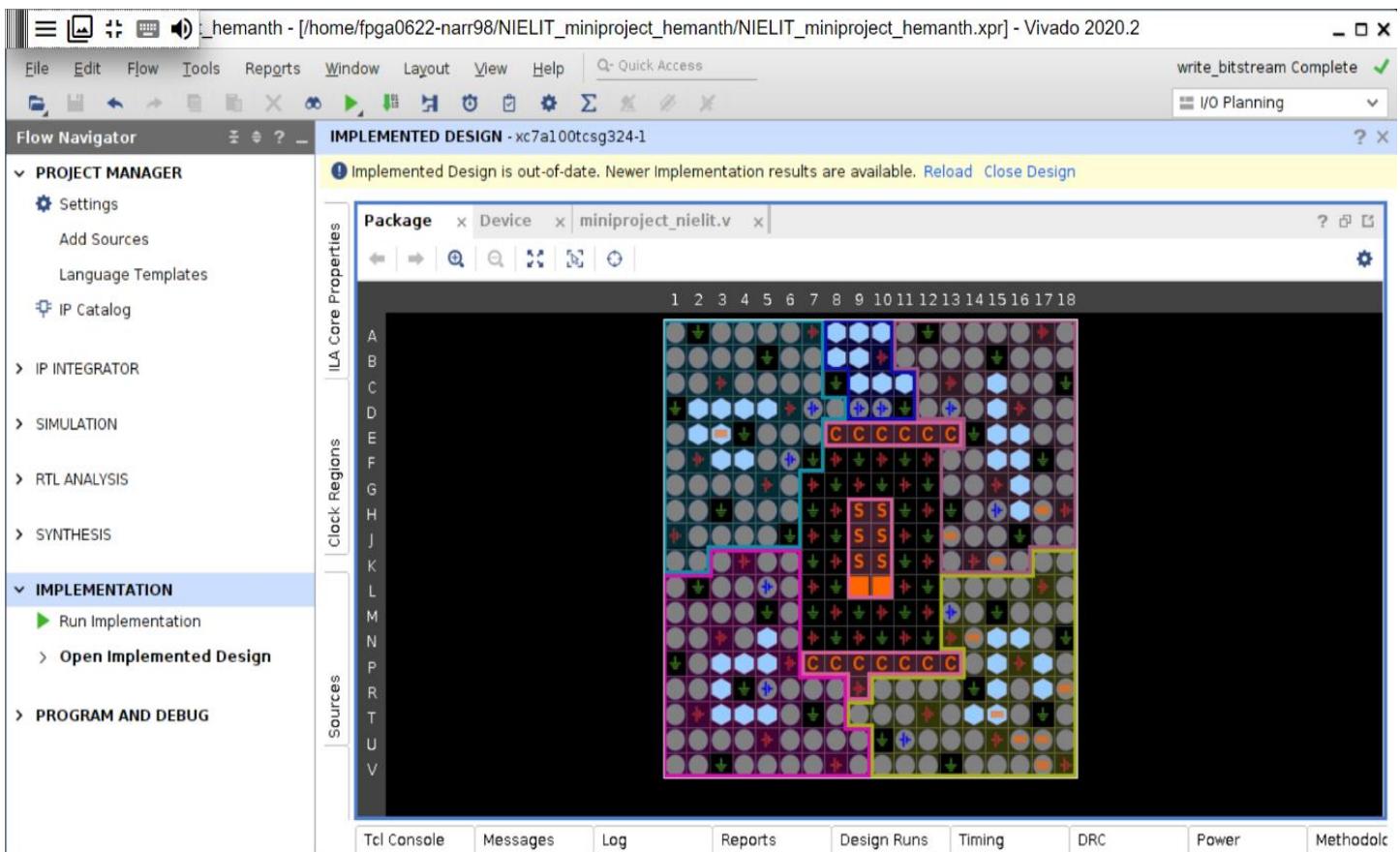


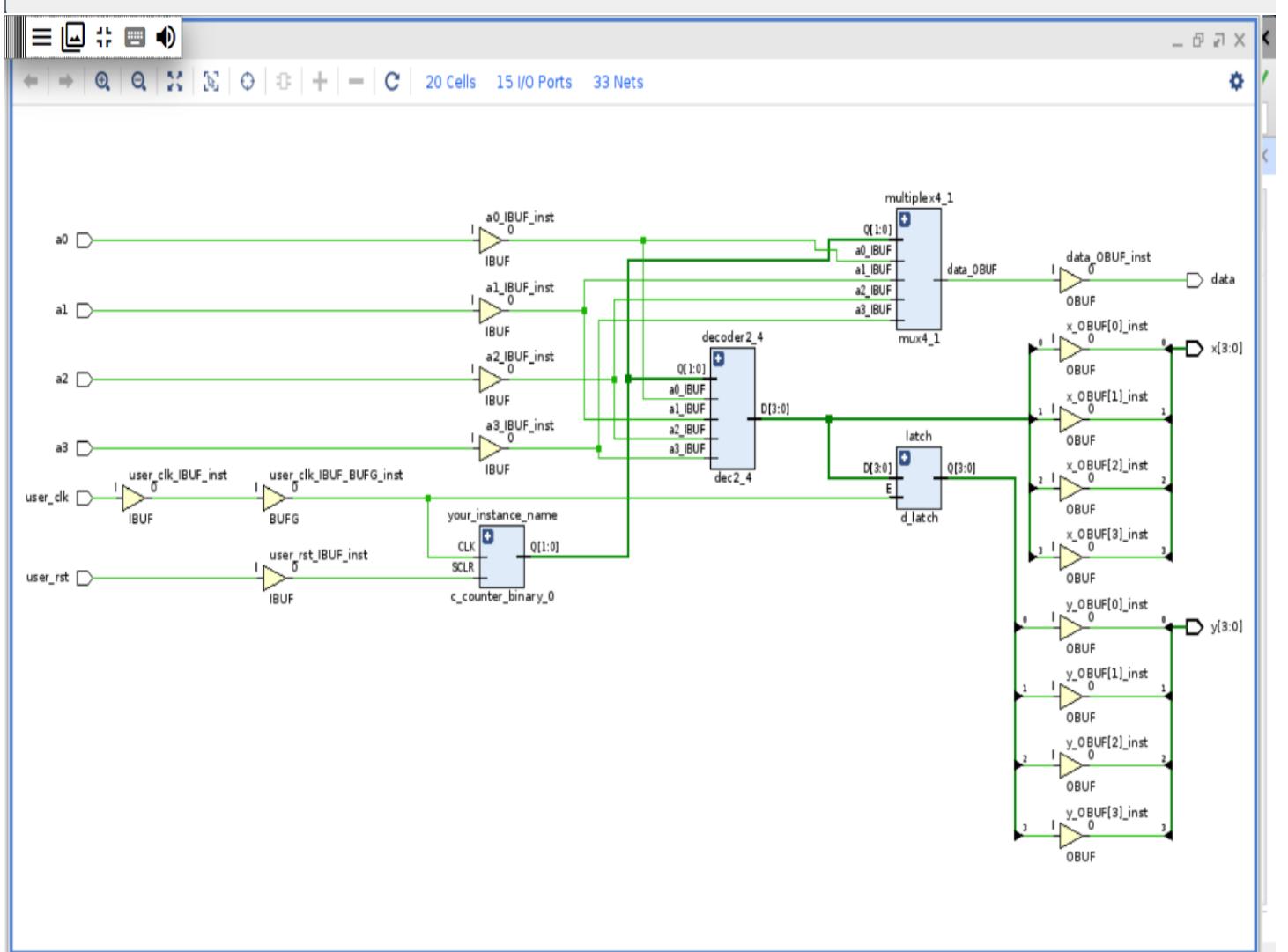
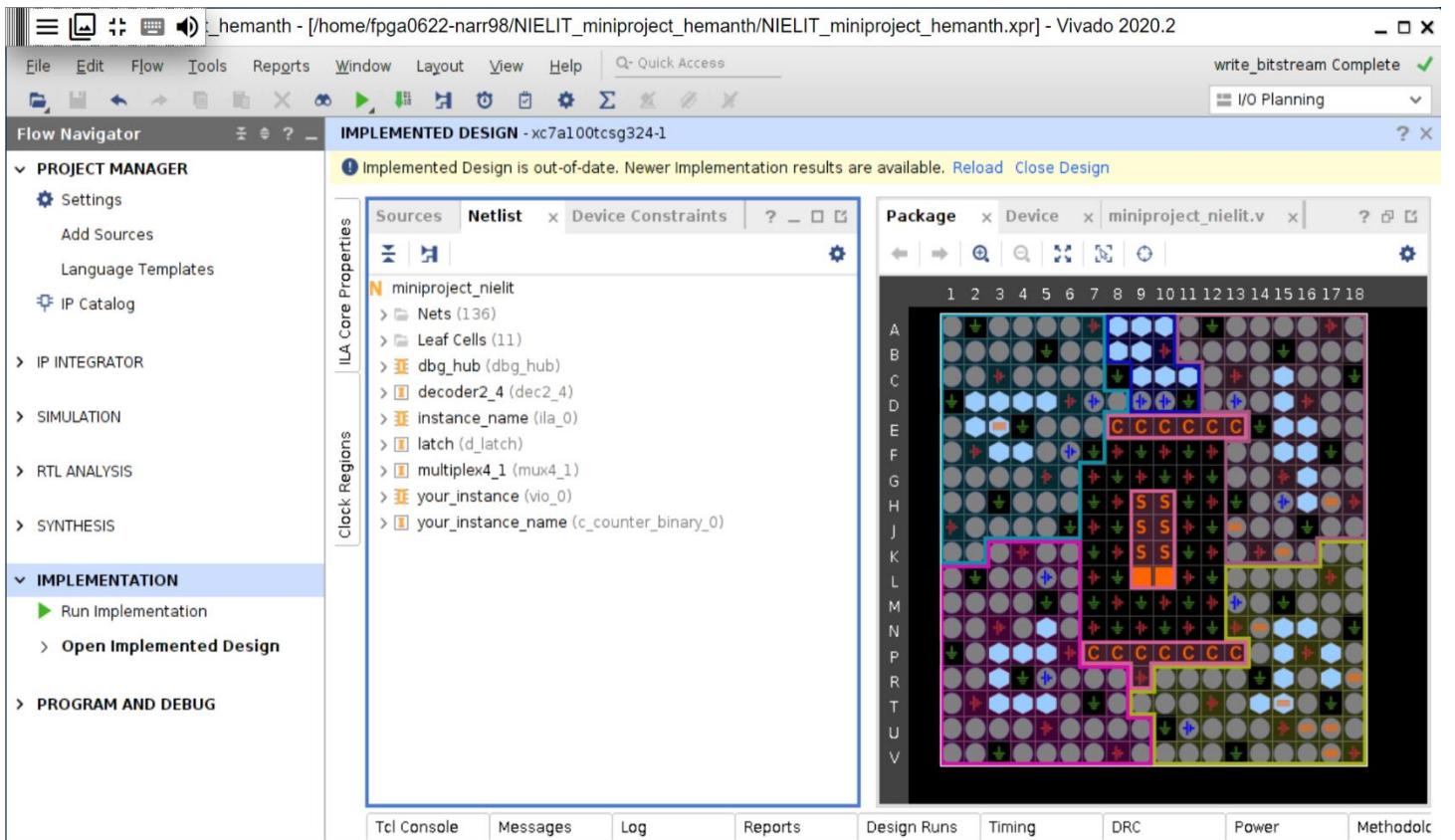


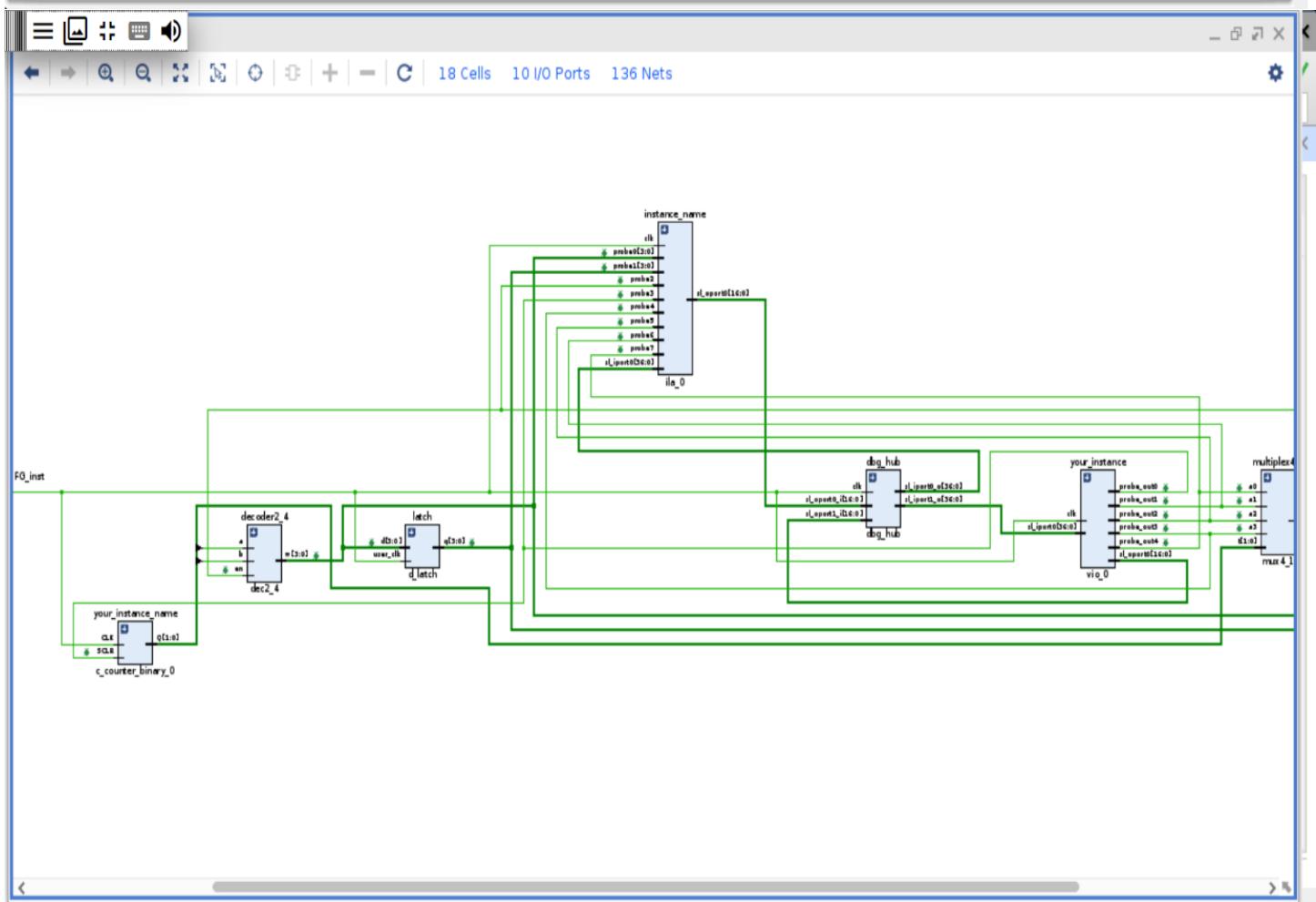
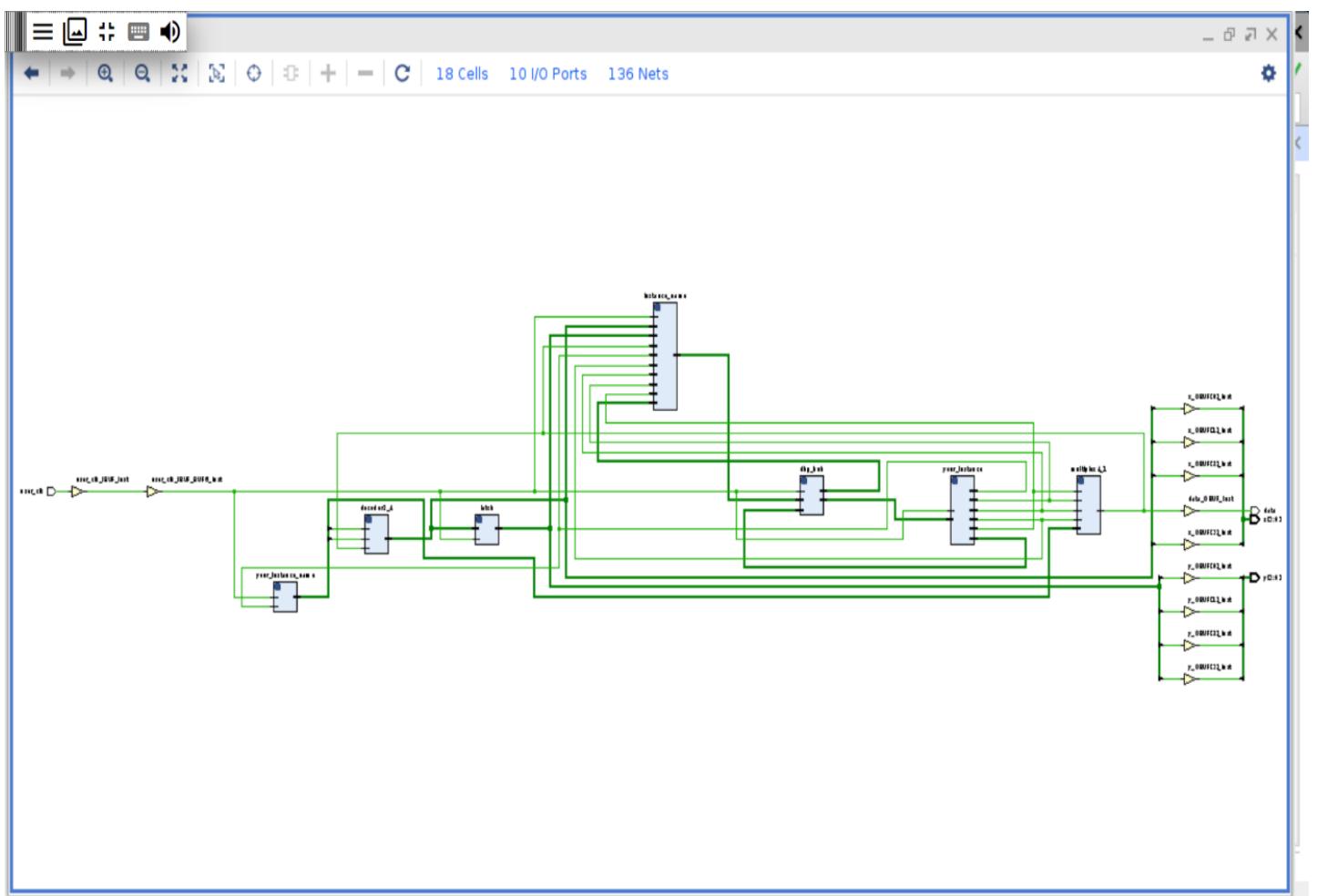












Result :

All the required waveforms have been attached and the supporting images are attached.