

Team: System Solutions

Decimation and Figure-Of-Merit Calculation

Test 1

Alexis Wilson
3-8-2020

TABLE OF CONTENTS

TABLE OF CONTENTS	1
REVISION HISTORY	2
TEST PROCEDURE	3
PURPOSE OF TEST	3
REQUIREMENTS	3
TEST IMPLEMENTATION	3
<i>Setup</i>	3
<i>Procedure</i>	3
<i>Results</i>	4
CONCLUSION	5
APPENDIX	6
ADC_TO_DAC_WRAPPER.SV.....	7
DECIMATOR.SV	7
FIG_OF_MERIT.SV.....	10
TB.SV	11
RED_PITAYA.XDC.....	16

Revision History

Name	Date	Revision Made	Version Number
Alexis Wilson	03/08/2020	Creation of Document	1.0
Alexis Wilson	05/02/2020	Updated Result Pictures, updated code	2.0

Purpose of Test:

The purpose of this test is to verify the System Verilog code for decimating multiple inputs from an Analog-to-Digital Converter is functioning correctly as well as being able to calculate the Figure of Merit Calculation between two values. This will make it possible to know that the data being interpreted is being interpreted as desired.

Requirements:

To perform this test, one will need a working computer or laptop with a version of Vivado. The Vivado version that was used to run this test was version 2016.4; however, a newer version of Vivado should function very similarly to this. Also, to create the Vivado Project file, one will need to know the specifications of the board they are utilizing: the one xc7z010clg400-1 specific for the STEMLab 125-114 Red Pitaya boards is labelled xc7z010clg400-1.

Test Implementation:

Setup:

1. Open up Vivado version 2016.4
2. Create a new project specific for your Red Pitaya Board:
 - a. File -> new Project
 - b. Next
 - c. Title project: FPGA_Data_Acquisition_And_Control
 - d. Create File location
 - e. RTL Sources
 - f. Add/Create 4 source Files:
 - i. adc_to_dac_wrapper.sv
 - ii. decimator.sv
 - iii. fig_of_merit.sv
 - g. Add/Create a simulation file during source file creation
 - i. tb.sv
 - h. Add/Create 1 Constraint file:
 - i. RedPitaya.xdc
 - i. Search for specific board: xc7z010clg400-1
 - j. Finish: Let Vivado work its magic

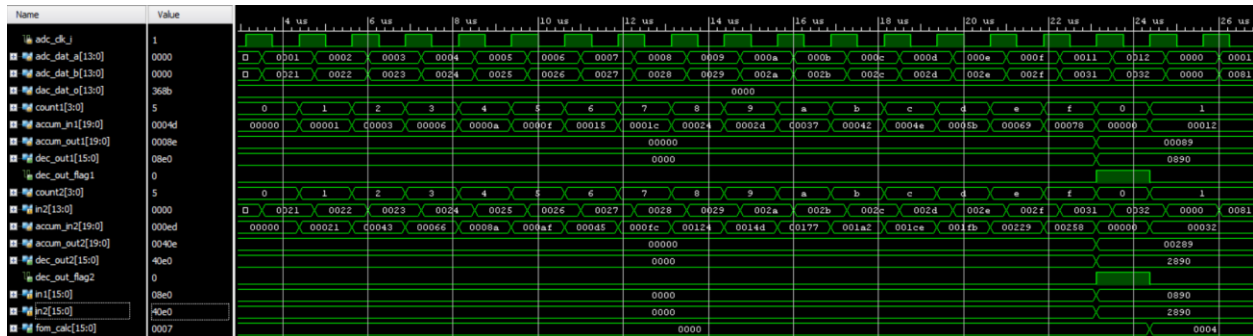
Procedure:

1. In adc_to_dac_wrapper.sv

- a. Copy code from Appendix under adc_to_dac_wrapper.sv
2. In decimator.sv
 - a. Copy code from Appendix under decimator.sv
3. In fig_of_merit.sv
 - a. Copy code from Appendix under fig_of_merit.sv
4. In tb.sv
 - a. Copy code from Appendix under tb.sv
5. In RedPitaya.xdc
 - a. Copy code from Appendix under RedPitaya.xdc
6. Run synthesis to check there is no errors
7. Once project synthesises with no critical warnings or errors:
 - a. Click Run Simulation
 - b. Behavioral Simulation
8. In behavioral Simulation:
 - a. Add all variables to waveform
9. Run waveform to completeness
 - a. The results of this test bench are shown in the screen-captures in the results section

Results:

1. Output from decimation of two inputs and a single FOM calculation



Conclusion:

The purpose of this test was to verify that all System Verilog code written for the Decimator and the Figure-Of-Merit Calculation functions as desired. The simulations created in this test procedure will help verify that the inputs from the system are being interpreted correctly once the bit file is generated and pushed to the board.

Appendix:

adc_to_dac_wrapper.sv:

```
module adc_to_dac_wrapper(  
    // adc inputs  
    input adc_clk_i,  
    input [13:0] adc_dat_a,  
    input [13:0] adc_dat_b,  
    // dac output  
    output [13:0] dac_dat_o,  
  
    // AXI interface inputs and outputs  
    input s_axis_clk, // need axi clock for data transfer  
    input s_axis_resestn, // using axi reset  
    input s_axis_tready1,  
  
    output s_axis_tvalid1,  
    output [15:0] s_axis_tdata1  
);  
  
logic dec_out_flag1 = 0;  
logic dec_out_flag2 = 0;  
logic [15:0] dec_out1;  
logic [15:0] dec_out2;  
  
// decimator instantiation  
decimator decimator1 (  
    .clk(adc_clk_i),  
    .in1(adc_dat_a),  
    .in2(adc_dat_b),  
    .dec_out_flag1(dec_out_flag1),  
    .dec_out_flag2(dec_out_flag2),
```

```

        .dec_out1(dec_out1),
        .dec_out2(dec_out2)
    );

    logic [15:0] fom_calc;
    logic fom_calc_flag = 0;
    // FOM calculation instantiation
    fig_of_merit fig_of_merit1(
        .clk(adc_clk_i),
        .in1(dec_out1),
        .in2(dec_out2),
        .dec_out_flag1(dec_out_flag1),
        .dec_out_flag2(dec_out_flag2),
        .fom_calc_flag(fom_calc_flag),
        .fom_calc(fom_calc)
    );

    assign dac_dat_o = fom_calc;

endmodule

```

decimator.sv:

```

module decimator(
    input clk,
    input [13:0] in1,
    input [13:0] in2,
    output dec_out_flag1,
    output dec_out_flag2,
    output [15:0] dec_out1,
    output [15:0] dec_out2
);

```

```

logic [3:0] count1 = 0;
logic [19:0] accum_in1 = 0;
logic [19:0] accum_out1 = 0;
logic [15:0] dec_calc1 = 0;
logic flag1 = 0;

// Accumulator for Input 1
always_ff @(posedge clk)
begin
    // waits for an initial value to come into the system
    if(count1 == 0 && in1 != 0)
    begin
        accum_in1 = in1;
        count1++;
        flag1 = 0;
    end
    // As long as only 15 values have gone in aka countonly reaches 14 it enters this loop
    else if (count1 != 15 && in1 != 0)
    begin
        accum_in1 = accum_in1 + in1;
        count1++;
        flag1 = 0;
    end
    /* means this is the 16th value to be inputted
    assign the new accumulated value to output of accumulator
    and then reset */
    else if(count1 == 15)
    begin
        accum_out1 = accum_in1 + in1;
        count1++;
        accum_in1 = 0;
    end
end

```



```

        flag1 = 1;
    end
    else
    begin
        flag1 = 0;
    end
end

//Decimation for Input 1
always_ff @(posedge clk)
begin
    if(flag1 == 1)
    begin
        dec_calc1 = accum_out1 <<< 4;
    end
end

// how dec_out is assigned until decimation is written
assign dec_out1 = dec_calc1;
assign dec_out_flag1 = flag1;

logic [3:0] count2 = 0;
logic [19:0] accum_in2 = 0;
logic [19:0] accum_out2 = 0;
logic [15:0] dec_calc2 = 0;
logic flag2 = 0;

// Accumulator for Input 2
always_ff @(posedge clk)
begin
    // waits for an initial value to come into the system
    if(count2 == 0 && in2 != 0)
    begin

```

```

    accum_in2 = in2;

    count2++;

    flag2 = 0;
end

// As long as only 15 values have gone in aka countonly reaches 14 it enters this loop
else if (count2 != 15 && in2 != 0)
begin
    accum_in2 = accum_in2 + in2;

    count2++;

    flag2 = 0;
end

/* means this is the 16th value to be inputted
   assign the new accumulated value to output of accumulator
   and then reset */
else if(count2 == 15)
begin
    accum_out2 = accum_in2 + in2;

    count2++;

    accum_in2 = 0;

    flag2 = 1;
end

else
begin
    flag2 = 0;
end

end

// Decimation for Input 2
always_ff @(posedge clk)
begin
    if(flag2 == 1)
begin

```

```

        dec_calc2 = accum_out2 <<< 4;
    end
end
// how dec_out is assigned until decimation is written
assign dec_out2 = dec_calc2;
assign dec_out_flag2 = flag2;

endmodule

```

fig_of_merit.sv:

```

module fig_of_merit(
    input clk,
    input [15:0] in1,
    input [15:0] in2,
    input dec_out_flag1,
    input dec_out_flag2,
    output fom_calc_flag,
    output [15:0] fom_calc
    //output [13:0] dac_dat_o
);

//sets some base values to save calculation to
logic fom_flag;
logic [15:0] fom = 0;

always_ff @(posedge clk)
begin
    // check there is an input and that there was no previous FOM calculaion
    if(dec_out_flag1 == 1 && dec_out_flag2 == 1)
    begin
        fom = in2/in1;
    end
end

```

```

        fom_flag = 1;
    end
    else
    begin
        fom_flag = 0;
    end
end

assign fom_calc = fom;
assign fom_calc_flag = fom_flag;

endmodule

```

tb.sv:

```

module tb;

    // Inputs
    logic adc_clk_i;
    logic [13:0] adc_dat_a;
    logic [13:0] adc_dat_b;

    // Outputs
    logic [13:0] dac_dat_o; // not used in this section

    // Instantiate the Unit Under Test (UUT)
    adc_to_dac_wrapper #(.SIM_ONLY(1)) uut (
        .adc_clk_i(adc_clk_i),
        .adc_dat_a(adc_dat_a),
        .adc_dat_b(adc_dat_b),
        .dac_dat_o(dac_dat_o)
    );

    parameter CLK_PRD = 1250; // 125 MHz clock

```

```

parameter HOLD_TIME = (CLK_PRD*0.3);
parameter MAX_SIM_TIME = (CLK_PRD*30);

initial #(MAX_SIM_TIME) $finish;

initial begin
    adc_clk_i <= 0;
    forever #(CLK_PRD/2) adc_clk_i = ~adc_clk_i;
end

initial begin
    // Initial adc value
    adc_dat_a = 14'b00000000000000;
    adc_dat_b = 14'b00000000000000;

    // Wait 100 ns for global reset to finish
    #100;
    @(posedge adc_clk_i); // align with clock edge
    #HOLD_TIME; // offset a hold time
    repeat(2) #CLK_PRD;

    adc_dat_a = 14'b00000000000001;
    adc_dat_b = 14'b00000000100001;
    #CLK_PRD;
    adc_dat_a = 14'b00000000000010;
    adc_dat_b = 14'b00000000100010;
    #CLK_PRD;
    adc_dat_a = 14'b00000000000011;
    adc_dat_b = 14'b00000000100011;
    #CLK_PRD;
    adc_dat_a = 14'b00000000000100;
    adc_dat_b = 14'b00000000100100;

```

```
#CLK_PRD;
adc_dat_a = 14'b00000000000101;
adc_dat_b = 14'b00000000100101;
#CLK_PRD;
adc_dat_a = 14'b00000000000110;
adc_dat_b = 14'b00000000100110;
#CLK_PRD;
adc_dat_a = 14'b00000000000111;
adc_dat_b = 14'b00000000100111;
#CLK_PRD;
adc_dat_a = 14'b00000000001000;
adc_dat_b = 14'b00000000101000;
#CLK_PRD;
adc_dat_a = 14'b00000000001001;
adc_dat_b = 14'b00000000101001;
#CLK_PRD;
adc_dat_a = 14'b00000000001010;
adc_dat_b = 14'b00000000101010;
#CLK_PRD;
adc_dat_a = 14'b00000000001011;
adc_dat_b = 14'b00000000101011;
#CLK_PRD;
adc_dat_a = 14'b00000000001100;
adc_dat_b = 14'b00000000101100;
#CLK_PRD;
adc_dat_a = 14'b00000000001101;
adc_dat_b = 14'b00000000101101;
#CLK_PRD;
adc_dat_a = 14'b00000000001110;
adc_dat_b = 14'b00000000101110;
#CLK_PRD;
adc_dat_a = 14'b00000000001111;
```

```
adc_dat_b = 14'b00000000101111;  
#CLK_PRD;  
adc_dat_a = 14'b0000000010001;  
adc_dat_b = 14'b00000000110001;  
#CLK_PRD;  
adc_dat_a = 14'b0000000010010;  
adc_dat_b = 14'b00000000110010;  
#CLK_PRD;  
adc_dat_a = 14'b0000000000000;  
adc_dat_b = 14'b0000000000000;  
  
#CLK_PRD;  
  
adc_dat_a = 14'b0000000000001;  
adc_dat_b = 14'b0000010000001;  
#CLK_PRD;  
adc_dat_a = 14'b00000000000010;  
adc_dat_b = 14'b00000010000010;  
#CLK_PRD;  
adc_dat_a = 14'b00000000000011;  
adc_dat_b = 14'b00000010000011;  
#CLK_PRD;  
adc_dat_a = 14'b00000000000100;  
adc_dat_b = 14'b00000010000100;  
#CLK_PRD;  
adc_dat_a = 14'b00000000000101;  
adc_dat_b = 14'b00000010000101;  
#CLK_PRD;  
adc_dat_a = 14'b00000000000110;  
adc_dat_b = 14'b00000010000110;  
#CLK_PRD;  
adc_dat_a = 14'b00000000000111;
```

```
adc_dat_b = 14'b00000010000111;
#CLK_PRD
adc_dat_a = 14'b00000000001000;
adc_dat_b = 14'b00000010001000;
#CLK_PRD;
adc_dat_a = 14'b00000000001001;
adc_dat_b = 14'b00000010001001;
#CLK_PRD;
adc_dat_a = 14'b00000000001010;
adc_dat_b = 14'b00000010001010;
#CLK_PRD;
adc_dat_a = 14'b00000000001011;
adc_dat_b = 14'b00000010001011;
#CLK_PRD;
adc_dat_a = 14'b00000000001100;
adc_dat_b = 14'b00000010001100;
#CLK_PRD;
adc_dat_a = 14'b00000000001101;
adc_dat_b = 14'b00000010001101;
#CLK_PRD;
adc_dat_a = 14'b00000000001110;
adc_dat_b = 14'b00000010001110;
#CLK_PRD;
adc_dat_a = 14'b00000000001111;
adc_dat_b = 14'b00000010001111;
#CLK_PRD;
adc_dat_a = 14'b00000000010001;
adc_dat_b = 14'b00000010010001;
#CLK_PRD;
adc_dat_a = 14'b00000000010010;
adc_dat_b = 14'b00000010010010;
#CLK_PRD;
```



```
adc_dat_a = 14'b00000000000000;
adc_dat_b = 14'b00000000000000;

#CLK_PRD;

adc_dat_a = 14'b00000000000001;
adc_dat_b = 14'b00000000100001;
#CLK_PRD;

adc_dat_a = 14'b00000000000010;
adc_dat_b = 14'b00000000100010;
#CLK_PRD;

adc_dat_a = 14'b00000000000011;
adc_dat_b = 14'b00000000100011;
#CLK_PRD;

adc_dat_a = 14'b00000000000100;
adc_dat_b = 14'b00000000100100;
#CLK_PRD;

adc_dat_a = 14'b00000000000101;
adc_dat_b = 14'b00000000100101;
#CLK_PRD;

adc_dat_a = 14'b00000000000110;
adc_dat_b = 14'b00000000100110;
#CLK_PRD;

adc_dat_a = 14'b00000000000111;
adc_dat_b = 14'b00000000100111;
#CLK_PRD;

adc_dat_a = 14'b0000000001000;
adc_dat_b = 14'b00000000101000;
#CLK_PRD;

adc_dat_a = 14'b0000000001001;
adc_dat_b = 14'b00000000101001;
#CLK_PRD;
```

```
adc_dat_a = 14'b00000000001010;
adc_dat_b = 14'b00000000101010;
#CLK_PRD;
adc_dat_a = 14'b00000000001011;
adc_dat_b = 14'b00000000101011;
#CLK_PRD;
adc_dat_a = 14'b00000000001100;
adc_dat_b = 14'b00000000101100;
#CLK_PRD;
adc_dat_a = 14'b00000000001101;
adc_dat_b = 14'b00000000101101;
#CLK_PRD;
adc_dat_a = 14'b00000000001110;
adc_dat_b = 14'b00000000101110;
#CLK_PRD;
adc_dat_a = 14'b00000000001111;
adc_dat_b = 14'b00000000101111;
#CLK_PRD;
adc_dat_a = 14'b00000000010001;
adc_dat_b = 14'b00000000110001;
#CLK_PRD;
adc_dat_a = 14'b00000000010010;
adc_dat_b = 14'b00000000110010;
#CLK_PRD;
adc_dat_a = 14'b00000000000000;
adc_dat_b = 14'b00000000000000;

#CLK_PRD;

adc_dat_a = 14'b00000000000001;
adc_dat_b = 14'b00000100000001;
#CLK_PRD;
```

```
adc_dat_a = 14'b00000000000010;
adc_dat_b = 14'b00000010000010;
#CLK_PRD;
adc_dat_a = 14'b00000000000011;
adc_dat_b = 14'b00000010000011;
#CLK_PRD;
adc_dat_a = 14'b00000000000100;
adc_dat_b = 14'b00000010000100;
#CLK_PRD;
adc_dat_a = 14'b00000000000101;
adc_dat_b = 14'b00000010000101;
#CLK_PRD;
adc_dat_a = 14'b00000000000110;
adc_dat_b = 14'b00000010000110;
#CLK_PRD;
adc_dat_a = 14'b00000000000111;
adc_dat_b = 14'b00000010000111;
#CLK_PRD;
adc_dat_a = 14'b00000000001000;
adc_dat_b = 14'b00000010001000;
#CLK_PRD;
adc_dat_a = 14'b00000000001001;
adc_dat_b = 14'b00000010001001;
#CLK_PRD;
adc_dat_a = 14'b00000000001010;
adc_dat_b = 14'b00000010001010;
#CLK_PRD;
adc_dat_a = 14'b00000000001011;
adc_dat_b = 14'b00000010001011;
#CLK_PRD;
adc_dat_a = 14'b00000000001100;
adc_dat_b = 14'b00000010001100;
```

```
#CLK_PRD;
adc_dat_a = 14'b00000000001101;
adc_dat_b = 14'b00000010001101;
#CLK_PRD;
adc_dat_a = 14'b00000000001110;
adc_dat_b = 14'b00000010001110;
#CLK_PRD;
adc_dat_a = 14'b00000000001111;
adc_dat_b = 14'b00000010001111;
#CLK_PRD;
adc_dat_a = 14'b00000000010001;
adc_dat_b = 14'b00000010010001;
#CLK_PRD;
adc_dat_a = 14'b00000000010010;
adc_dat_b = 14'b00000010010010;
#CLK_PRD;
adc_dat_a = 14'b00000000000000;
adc_dat_b = 14'b00000000000000;

#CLK_PRD

adc_dat_a = 14'b00000000000001;
adc_dat_b = 14'b00000000100001;
#CLK_PRD;
adc_dat_a = 14'b00000000000010;
adc_dat_b = 14'b00000000100010;
#CLK_PRD;
adc_dat_a = 14'b00000000000011;
adc_dat_b = 14'b00000000100011;
#CLK_PRD;
adc_dat_a = 14'b00000000000100;
adc_dat_b = 14'b00000000100100;
```

```
#CLK_PRD;
adc_dat_a = 14'b00000000000101;
adc_dat_b = 14'b00000000100101;
#CLK_PRD;
adc_dat_a = 14'b0000000000110;
adc_dat_b = 14'b00000000100110;
#CLK_PRD;
adc_dat_a = 14'b0000000000111;
adc_dat_b = 14'b00000000100111;
#CLK_PRD;
adc_dat_a = 14'b0000000001000;
adc_dat_b = 14'b00000000101000;
#CLK_PRD;
adc_dat_a = 14'b0000000001001;
adc_dat_b = 14'b00000000101001;
#CLK_PRD;
adc_dat_a = 14'b0000000001010;
adc_dat_b = 14'b00000000101010;
#CLK_PRD;
adc_dat_a = 14'b0000000001011;
adc_dat_b = 14'b00000000101011;
#CLK_PRD;
adc_dat_a = 14'b0000000001100;
adc_dat_b = 14'b00000000101100;
#CLK_PRD;
adc_dat_a = 14'b0000000001101;
adc_dat_b = 14'b00000000101101;
#CLK_PRD;
adc_dat_a = 14'b0000000001110;
adc_dat_b = 14'b00000000101110;
#CLK_PRD;
adc_dat_a = 14'b0000000001111;
```

```
adc_dat_b = 14'b00000000101111;  
#CLK_PRD;  
adc_dat_a = 14'b0000000010001;  
adc_dat_b = 14'b00000000110001;  
#CLK_PRD;  
adc_dat_a = 14'b0000000010010;  
adc_dat_b = 14'b00000000110010;  
#CLK_PRD;  
adc_dat_a = 14'b00000000000000;  
adc_dat_b = 14'b00000000000000;  
  
#CLK_PRD;  
  
adc_dat_a = 14'b00000000000001;  
adc_dat_b = 14'b00000000100001;  
#CLK_PRD;  
adc_dat_a = 14'b00000000000010;  
adc_dat_b = 14'b00000000100010;  
#CLK_PRD;  
adc_dat_a = 14'b00000000000011;  
adc_dat_b = 14'b00000000100011;  
#CLK_PRD;  
adc_dat_a = 14'b00000000000100;  
adc_dat_b = 14'b00000000100100;  
#CLK_PRD;  
adc_dat_a = 14'b00000000000101;  
adc_dat_b = 14'b00000000100101;  
#CLK_PRD;  
adc_dat_a = 14'b00000000000110;  
adc_dat_b = 14'b00000000100110;  
#CLK_PRD;  
adc_dat_a = 14'b00000000000111;
```

```
adc_dat_b = 14'b00000000100111;
#CLK_PRD
adc_dat_a = 14'b00000000001000;
adc_dat_b = 14'b00000000101000;
#CLK_PRD;
adc_dat_a = 14'b00000000001001;
adc_dat_b = 14'b00000000101001;
#CLK_PRD;
adc_dat_a = 14'b00000000001010;
adc_dat_b = 14'b00000000101010;
#CLK_PRD;
adc_dat_a = 14'b00000000001011;
adc_dat_b = 14'b00000000101011;
#CLK_PRD;
adc_dat_a = 14'b00000000001100;
adc_dat_b = 14'b00000000101100;
#CLK_PRD;
adc_dat_a = 14'b00000000001101;
adc_dat_b = 14'b00000000101101;
#CLK_PRD;
adc_dat_a = 14'b00000000001110;
adc_dat_b = 14'b00000000101110;
#CLK_PRD;
adc_dat_a = 14'b00000000001111;
adc_dat_b = 14'b00000000101111;
#CLK_PRD;
adc_dat_a = 14'b00000000010001;
adc_dat_b = 14'b00000000110001;
#CLK_PRD;
adc_dat_a = 14'b00000000010010;
adc_dat_b = 14'b00000000110010;
#CLK_PRD;
```

```
adc_dat_a = 14'b00000000000000;
adc_dat_b = 14'b00000000000000;

#CLK_PRD;

adc_dat_a = 14'b00000000000001;
adc_dat_b = 14'b00000100000001;
#CLK_PRD;

adc_dat_a = 14'b00000000000010;
adc_dat_b = 14'b00000010000010;
#CLK_PRD;

adc_dat_a = 14'b00000000000011;
adc_dat_b = 14'b00000010000011;
#CLK_PRD;

adc_dat_a = 14'b00000000000100;
adc_dat_b = 14'b00000010000100;
#CLK_PRD;

adc_dat_a = 14'b00000000000101;
adc_dat_b = 14'b00000010000101;
#CLK_PRD;

adc_dat_a = 14'b00000000000110;
adc_dat_b = 14'b00000010000110;
#CLK_PRD;

adc_dat_a = 14'b00000000000111;
adc_dat_b = 14'b00000010000111;
#CLK_PRD;

adc_dat_a = 14'b00000000001000;
adc_dat_b = 14'b00000010001000;
#CLK_PRD;

adc_dat_a = 14'b00000000001001;
adc_dat_b = 14'b00000010001001;
#CLK_PRD;
```



```
adc_dat_a = 14'b00000000001010;
adc_dat_b = 14'b00000010001010;
#CLK_PRD;
adc_dat_a = 14'b00000000001011;
adc_dat_b = 14'b00000010001011;
#CLK_PRD;
adc_dat_a = 14'b00000000001100;
adc_dat_b = 14'b00000010001100;
#CLK_PRD;
adc_dat_a = 14'b00000000001101;
adc_dat_b = 14'b00000010001101;
#CLK_PRD;
adc_dat_a = 14'b00000000001110;
adc_dat_b = 14'b00000010001110;
#CLK_PRD;
adc_dat_a = 14'b00000000001111;
adc_dat_b = 14'b00000010001111;
#CLK_PRD;
adc_dat_a = 14'b00000000010001;
adc_dat_b = 14'b00000010010001;
#CLK_PRD;
adc_dat_a = 14'b00000000010010;
adc_dat_b = 14'b00000010010010;
#CLK_PRD;
adc_dat_a = 14'b00000000000000;
adc_dat_b = 14'b00000000000000;

#CLK_PRD

adc_dat_a = 14'b00000000000001;
adc_dat_b = 14'b00000000100001;
#CLK_PRD;
```

```
adc_dat_a = 14'b00000000000010;  
adc_dat_b = 14'b00000000100010;  
#CLK_PRD;  
adc_dat_a = 14'b00000000000011;  
adc_dat_b = 14'b00000000100011;  
#CLK_PRD;  
adc_dat_a = 14'b00000000000100;  
adc_dat_b = 14'b00000000100100;  
#CLK_PRD;  
adc_dat_a = 14'b00000000000101;  
adc_dat_b = 14'b00000000100101;  
#CLK_PRD;  
adc_dat_a = 14'b00000000000110;  
adc_dat_b = 14'b00000000100110;  
#CLK_PRD;  
adc_dat_a = 14'b00000000000111;  
adc_dat_b = 14'b00000000100111;  
#CLK_PRD  
adc_dat_a = 14'b00000000001000;  
adc_dat_b = 14'b00000000101000;  
#CLK_PRD;  
adc_dat_a = 14'b00000000001001;  
adc_dat_b = 14'b00000000101001;  
#CLK_PRD;  
adc_dat_a = 14'b00000000001010;  
adc_dat_b = 14'b00000000101010;  
#CLK_PRD;  
adc_dat_a = 14'b00000000001011;  
adc_dat_b = 14'b00000000101011;  
#CLK_PRD;  
adc_dat_a = 14'b00000000001100;  
adc_dat_b = 14'b00000000101100;
```

```
#CLK_PRD;
adc_dat_a = 14'b00000000001101;
adc_dat_b = 14'b00000000101101;
#CLK_PRD;
adc_dat_a = 14'b00000000001110;
adc_dat_b = 14'b00000000101110;
#CLK_PRD;
adc_dat_a = 14'b00000000001111;
adc_dat_b = 14'b00000000101111;
#CLK_PRD;
adc_dat_a = 14'b00000000010001;
adc_dat_b = 14'b00000000110001;
#CLK_PRD;
adc_dat_a = 14'b00000000010010;
adc_dat_b = 14'b00000000110010;
#CLK_PRD;
adc_dat_a = 14'b00000000000000;
adc_dat_b = 14'b00000000000000;

#CLK_PRD;

adc_dat_a = 14'b00000000000001;
adc_dat_b = 14'b00000000100001;
#CLK_PRD;
adc_dat_a = 14'b00000000000010;
adc_dat_b = 14'b00000000100010;
#CLK_PRD;
adc_dat_a = 14'b00000000000011;
adc_dat_b = 14'b00000000100011;
#CLK_PRD;
adc_dat_a = 14'b00000000000100;
adc_dat_b = 14'b00000000100100;
```

```
#CLK_PRD;
adc_dat_a = 14'b00000000000101;
adc_dat_b = 14'b00000000100101;
#CLK_PRD;
adc_dat_a = 14'b0000000000110;
adc_dat_b = 14'b00000000100110;
#CLK_PRD;
adc_dat_a = 14'b0000000000111;
adc_dat_b = 14'b00000000100111;
#CLK_PRD;
adc_dat_a = 14'b0000000001000;
adc_dat_b = 14'b00000000101000;
#CLK_PRD;
adc_dat_a = 14'b0000000001001;
adc_dat_b = 14'b00000000101001;
#CLK_PRD;
adc_dat_a = 14'b0000000001010;
adc_dat_b = 14'b00000000101010;
#CLK_PRD;
adc_dat_a = 14'b0000000001011;
adc_dat_b = 14'b00000000101011;
#CLK_PRD;
adc_dat_a = 14'b0000000001100;
adc_dat_b = 14'b00000000101100;
#CLK_PRD;
adc_dat_a = 14'b0000000001101;
adc_dat_b = 14'b00000000101101;
#CLK_PRD;
adc_dat_a = 14'b0000000001110;
adc_dat_b = 14'b00000000101110;
#CLK_PRD;
adc_dat_a = 14'b0000000001111;
```

```
adc_dat_b = 14'b00000000101111;  
#CLK_PRD;  
adc_dat_a = 14'b0000000010001;  
adc_dat_b = 14'b00000000110001;  
#CLK_PRD;  
adc_dat_a = 14'b0000000010010;  
adc_dat_b = 14'b00000000110010;  
#CLK_PRD;  
adc_dat_a = 14'b0000000000000;  
adc_dat_b = 14'b0000000000000;  
  
#CLK_PRD;  
  
adc_dat_a = 14'b0000000000001;  
adc_dat_b = 14'b0000010000001;  
#CLK_PRD;  
adc_dat_a = 14'b00000000000010;  
adc_dat_b = 14'b00000010000010;  
#CLK_PRD;  
adc_dat_a = 14'b00000000000011;  
adc_dat_b = 14'b00000010000011;  
#CLK_PRD;  
adc_dat_a = 14'b00000000000100;  
adc_dat_b = 14'b00000010000100;  
#CLK_PRD;  
adc_dat_a = 14'b00000000000101;  
adc_dat_b = 14'b00000010000101;  
#CLK_PRD;  
adc_dat_a = 14'b00000000000110;  
adc_dat_b = 14'b00000010000110;  
#CLK_PRD;  
adc_dat_a = 14'b00000000000111;
```

```
adc_dat_b = 14'b00000010000111;  
#CLK_PRD  
adc_dat_a = 14'b00000000001000;  
adc_dat_b = 14'b00000010001000;  
#CLK_PRD;  
adc_dat_a = 14'b00000000001001;  
adc_dat_b = 14'b00000010001001;  
#CLK_PRD;  
adc_dat_a = 14'b00000000001010;  
adc_dat_b = 14'b00000010001010;  
#CLK_PRD;  
adc_dat_a = 14'b00000000001011;  
adc_dat_b = 14'b00000010001011;  
#CLK_PRD;  
adc_dat_a = 14'b00000000001100;  
adc_dat_b = 14'b00000010001100;  
#CLK_PRD;  
adc_dat_a = 14'b00000000001101;  
adc_dat_b = 14'b00000010001101;  
#CLK_PRD;  
adc_dat_a = 14'b00000000001110;  
adc_dat_b = 14'b00000010001110;  
#CLK_PRD;  
adc_dat_a = 14'b00000000001111;  
adc_dat_b = 14'b00000010001111;  
#CLK_PRD;  
adc_dat_a = 14'b00000000010001;  
adc_dat_b = 14'b00000010010001;  
#CLK_PRD;  
adc_dat_a = 14'b00000000010010;  
adc_dat_b = 14'b00000010010010;  
#CLK_PRD;
```

```
adc_dat_a = 14'b0000000000000000;  
adc_dat_b = 14'b0000000000000000;
```

```
#CLK_PRD
```

```
adc_dat_a = 14'b0000000000000001;  
adc_dat_b = 14'b000000000100001;
```

```
#CLK_PRD;
```

```
adc_dat_a = 14'b0000000000000010;  
adc_dat_b = 14'b000000000100010;
```

```
#CLK_PRD;
```

```
adc_dat_a = 14'b0000000000000011;  
adc_dat_b = 14'b000000000100011;
```

```
#CLK_PRD;
```

```
adc_dat_a = 14'b0000000000000100;  
adc_dat_b = 14'b000000000100100;
```

```
#CLK_PRD;
```

```
adc_dat_a = 14'b0000000000000101;  
adc_dat_b = 14'b000000000100101;
```

```
#CLK_PRD;
```

```
adc_dat_a = 14'b0000000000000110;  
adc_dat_b = 14'b000000000100110;
```

```
#CLK_PRD;
```

```
adc_dat_a = 14'b0000000000000111;  
adc_dat_b = 14'b000000000100111;
```

```
#CLK_PRD
```

```
adc_dat_a = 14'b0000000000001000;  
adc_dat_b = 14'b000000000101000;
```

```
#CLK_PRD;
```

```
adc_dat_a = 14'b0000000000001001;  
adc_dat_b = 14'b000000000101001;
```

```
#CLK_PRD;
```

```
adc_dat_a = 14'b00000000001010;  
adc_dat_b = 14'b00000000101010;  
#CLK_PRD;  
adc_dat_a = 14'b00000000001011;  
adc_dat_b = 14'b00000000101011;  
#CLK_PRD;  
adc_dat_a = 14'b00000000001100;  
adc_dat_b = 14'b00000000101100;  
#CLK_PRD;  
adc_dat_a = 14'b00000000001101;  
adc_dat_b = 14'b00000000101101;  
#CLK_PRD;  
adc_dat_a = 14'b00000000001110;  
adc_dat_b = 14'b00000000101110;  
#CLK_PRD;  
adc_dat_a = 14'b00000000001111;  
adc_dat_b = 14'b00000000101111;  
#CLK_PRD;  
adc_dat_a = 14'b00000000010001;  
adc_dat_b = 14'b00000000110001;  
#CLK_PRD;  
adc_dat_a = 14'b00000000010010;  
adc_dat_b = 14'b00000000110010;  
#CLK_PRD;  
adc_dat_a = 14'b00000000000000;  
adc_dat_b = 14'b00000000000000;  
  
#CLK_PRD;  
  
adc_dat_a = 14'b00000000000001;  
adc_dat_b = 14'b00000000100001;  
#CLK_PRD;
```



```
adc_dat_a = 14'b00000000000010;
adc_dat_b = 14'b00000000100010;
#CLK_PRD;
adc_dat_a = 14'b00000000000011;
adc_dat_b = 14'b00000000100011;
#CLK_PRD;
adc_dat_a = 14'b00000000000100;
adc_dat_b = 14'b00000000100100;
#CLK_PRD;
adc_dat_a = 14'b00000000000101;
adc_dat_b = 14'b00000000100101;
#CLK_PRD;
adc_dat_a = 14'b00000000000110;
adc_dat_b = 14'b00000000100110;
#CLK_PRD;
adc_dat_a = 14'b00000000000111;
adc_dat_b = 14'b00000000100111;
#CLK_PRD;
adc_dat_a = 14'b0000000001000;
adc_dat_b = 14'b00000000101000;
#CLK_PRD;
adc_dat_a = 14'b0000000001001;
adc_dat_b = 14'b00000000101001;
#CLK_PRD;
adc_dat_a = 14'b0000000001010;
adc_dat_b = 14'b00000000101010;
#CLK_PRD;
adc_dat_a = 14'b0000000001011;
adc_dat_b = 14'b00000000101011;
#CLK_PRD;
adc_dat_a = 14'b0000000001100;
adc_dat_b = 14'b00000000101100;
```

```
#CLK_PRD;
adc_dat_a = 14'b00000000001101;
adc_dat_b = 14'b00000000101101;
#CLK_PRD;
adc_dat_a = 14'b00000000001110;
adc_dat_b = 14'b00000000101110;
#CLK_PRD;
adc_dat_a = 14'b00000000001111;
adc_dat_b = 14'b00000000101111;
#CLK_PRD;
adc_dat_a = 14'b00000000010001;
adc_dat_b = 14'b00000000110001;
#CLK_PRD;
adc_dat_a = 14'b00000000010010;
adc_dat_b = 14'b00000000110010;
#CLK_PRD;
adc_dat_a = 14'b00000000000000;
adc_dat_b = 14'b00000000000000;

#CLK_PRD;

adc_dat_a = 14'b00000000000001;
adc_dat_b = 14'b00000100000001;
#CLK_PRD;
adc_dat_a = 14'b000000000000010;
adc_dat_b = 14'b000000100000010;
#CLK_PRD;
adc_dat_a = 14'b000000000000011;
adc_dat_b = 14'b000000100000011;
#CLK_PRD;
adc_dat_a = 14'b000000000000100;
adc_dat_b = 14'b000000100000100;
```

```
#CLK_PRD;
adc_dat_a = 14'b00000000000101;
adc_dat_b = 14'b00000010000101;
#CLK_PRD;
adc_dat_a = 14'b00000000000110;
adc_dat_b = 14'b00000010000110;
#CLK_PRD;
adc_dat_a = 14'b00000000000111;
adc_dat_b = 14'b00000010000111;
#CLK_PRD;
adc_dat_a = 14'b00000000001000;
adc_dat_b = 14'b00000010001000;
#CLK_PRD;
adc_dat_a = 14'b00000000001001;
adc_dat_b = 14'b00000010001001;
#CLK_PRD;
adc_dat_a = 14'b00000000001010;
adc_dat_b = 14'b00000010001010;
#CLK_PRD;
adc_dat_a = 14'b00000000001011;
adc_dat_b = 14'b00000010001011;
#CLK_PRD;
adc_dat_a = 14'b00000000001100;
adc_dat_b = 14'b00000010001100;
#CLK_PRD;
adc_dat_a = 14'b00000000001101;
adc_dat_b = 14'b00000010001101;
#CLK_PRD;
adc_dat_a = 14'b00000000001110;
adc_dat_b = 14'b00000010001110;
#CLK_PRD;
adc_dat_a = 14'b00000000001111;
```

```
adc_dat_b = 14'b00000010001111;  
#CLK_PRD;  
adc_dat_a = 14'b00000000010001;  
adc_dat_b = 14'b00000010010001;  
#CLK_PRD;  
adc_dat_a = 14'b00000000010010;  
adc_dat_b = 14'b00000010010010;  
#CLK_PRD;  
adc_dat_a = 14'b00000000000000;  
adc_dat_b = 14'b00000000000000;
```

```
#CLK_PRD
```

```
adc_dat_a = 14'b00000000000001;  
adc_dat_b = 14'b00000000100001;  
#CLK_PRD;  
adc_dat_a = 14'b00000000000010;  
adc_dat_b = 14'b00000000100010;  
#CLK_PRD;  
adc_dat_a = 14'b00000000000011;  
adc_dat_b = 14'b00000000100011;  
#CLK_PRD;  
adc_dat_a = 14'b00000000000100;  
adc_dat_b = 14'b00000000100100;  
#CLK_PRD;  
adc_dat_a = 14'b00000000000101;  
adc_dat_b = 14'b00000000100101;  
#CLK_PRD;  
adc_dat_a = 14'b00000000000110;  
adc_dat_b = 14'b00000000100110;  
#CLK_PRD;  
adc_dat_a = 14'b00000000000111;
```

```
adc_dat_b = 14'b00000000100111;
#CLK_PRD
adc_dat_a = 14'b0000000001000;
adc_dat_b = 14'b00000000101000;
#CLK_PRD;
adc_dat_a = 14'b0000000001001;
adc_dat_b = 14'b00000000101001;
#CLK_PRD;
adc_dat_a = 14'b0000000001010;
adc_dat_b = 14'b00000000101010;
#CLK_PRD;
adc_dat_a = 14'b0000000001011;
adc_dat_b = 14'b00000000101011;
#CLK_PRD;
adc_dat_a = 14'b0000000001100;
adc_dat_b = 14'b00000000101100;
#CLK_PRD;
adc_dat_a = 14'b0000000001101;
adc_dat_b = 14'b00000000101101;
#CLK_PRD;
adc_dat_a = 14'b0000000001110;
adc_dat_b = 14'b00000000101110;
#CLK_PRD;
adc_dat_a = 14'b0000000001111;
adc_dat_b = 14'b00000000101111;
#CLK_PRD;
adc_dat_a = 14'b00000000010001;
adc_dat_b = 14'b00000000110001;
#CLK_PRD;
adc_dat_a = 14'b00000000010010;
adc_dat_b = 14'b00000000110010;
#CLK_PRD;
```

```
adc_dat_a = 14'b00000000000000;
adc_dat_b = 14'b00000000000000;

#CLK_PRD;

adc_dat_a = 14'b00000000000001;
adc_dat_b = 14'b00000000100001;
#CLK_PRD;

adc_dat_a = 14'b00000000000010;
adc_dat_b = 14'b00000000100010;
#CLK_PRD;

adc_dat_a = 14'b00000000000011;
adc_dat_b = 14'b00000000100011;
#CLK_PRD;

adc_dat_a = 14'b00000000000100;
adc_dat_b = 14'b00000000100100;
#CLK_PRD;

adc_dat_a = 14'b00000000000101;
adc_dat_b = 14'b00000000100101;
#CLK_PRD;

adc_dat_a = 14'b00000000000110;
adc_dat_b = 14'b00000000100110;
#CLK_PRD;

adc_dat_a = 14'b00000000000111;
adc_dat_b = 14'b00000000100111;
#CLK_PRD;

adc_dat_a = 14'b0000000001000;
adc_dat_b = 14'b00000000101000;
#CLK_PRD;

adc_dat_a = 14'b0000000001001;
adc_dat_b = 14'b00000000101001;
#CLK_PRD;
```

```
adc_dat_a = 14'b00000000001010;  
adc_dat_b = 14'b00000000101010;  
#CLK_PRD;  
adc_dat_a = 14'b00000000001011;  
adc_dat_b = 14'b00000000101011;  
#CLK_PRD;  
adc_dat_a = 14'b00000000001100;  
adc_dat_b = 14'b00000000101100;  
#CLK_PRD;  
adc_dat_a = 14'b00000000001101;  
adc_dat_b = 14'b00000000101101;  
#CLK_PRD;  
adc_dat_a = 14'b00000000001110;  
adc_dat_b = 14'b00000000101110;  
#CLK_PRD;  
adc_dat_a = 14'b00000000001111;  
adc_dat_b = 14'b00000000101111;  
#CLK_PRD;  
adc_dat_a = 14'b00000000010001;  
adc_dat_b = 14'b00000000110001;  
#CLK_PRD;  
adc_dat_a = 14'b00000000010010;  
adc_dat_b = 14'b00000000110010;  
#CLK_PRD;  
adc_dat_a = 14'b00000000000000;  
adc_dat_b = 14'b00000000000000;  
  
#CLK_PRD;  
  
adc_dat_a = 14'b00000000000001;  
adc_dat_b = 14'b00000100000001;  
#CLK_PRD;
```

```
adc_dat_a = 14'b00000000000010;
adc_dat_b = 14'b00000010000010;
#CLK_PRD;
adc_dat_a = 14'b00000000000011;
adc_dat_b = 14'b00000010000011;
#CLK_PRD;
adc_dat_a = 14'b00000000000100;
adc_dat_b = 14'b00000010000100;
#CLK_PRD;
adc_dat_a = 14'b00000000000101;
adc_dat_b = 14'b00000010000101;
#CLK_PRD;
adc_dat_a = 14'b00000000000110;
adc_dat_b = 14'b00000010000110;
#CLK_PRD;
adc_dat_a = 14'b00000000000111;
adc_dat_b = 14'b00000010000111;
#CLK_PRD;
adc_dat_a = 14'b00000000001000;
adc_dat_b = 14'b00000010001000;
#CLK_PRD;
adc_dat_a = 14'b00000000001001;
adc_dat_b = 14'b00000010001001;
#CLK_PRD;
adc_dat_a = 14'b00000000001010;
adc_dat_b = 14'b00000010001010;
#CLK_PRD;
adc_dat_a = 14'b00000000001011;
adc_dat_b = 14'b00000010001011;
#CLK_PRD;
adc_dat_a = 14'b00000000001100;
adc_dat_b = 14'b00000010001100;
```



```
#CLK_PRD;
adc_dat_a = 14'b00000000001101;
adc_dat_b = 14'b00000010001101;
#CLK_PRD;
adc_dat_a = 14'b00000000001110;
adc_dat_b = 14'b00000010001110;
#CLK_PRD;
adc_dat_a = 14'b00000000001111;
adc_dat_b = 14'b00000010001111;
#CLK_PRD;
adc_dat_a = 14'b00000000010001;
adc_dat_b = 14'b00000010010001;
#CLK_PRD;
adc_dat_a = 14'b00000000010010;
adc_dat_b = 14'b00000010010010;
#CLK_PRD;
adc_dat_a = 14'b00000000000000;
adc_dat_b = 14'b00000000000000;

#CLK_PRD

adc_dat_a = 14'b00000000000001;
adc_dat_b = 14'b00000000100001;
#CLK_PRD;
adc_dat_a = 14'b00000000000010;
adc_dat_b = 14'b00000000100010;
#CLK_PRD;
adc_dat_a = 14'b00000000000011;
adc_dat_b = 14'b00000000100011;
#CLK_PRD;
adc_dat_a = 14'b00000000000100;
adc_dat_b = 14'b00000000100100;
```

```
#CLK_PRD;
adc_dat_a = 14'b00000000000101;
adc_dat_b = 14'b00000000100101;
#CLK_PRD;
adc_dat_a = 14'b0000000000110;
adc_dat_b = 14'b00000000100110;
#CLK_PRD;
adc_dat_a = 14'b0000000000111;
adc_dat_b = 14'b00000000100111;
#CLK_PRD;
adc_dat_a = 14'b0000000001000;
adc_dat_b = 14'b00000000101000;
#CLK_PRD;
adc_dat_a = 14'b0000000001001;
adc_dat_b = 14'b00000000101001;
#CLK_PRD;
adc_dat_a = 14'b0000000001010;
adc_dat_b = 14'b00000000101010;
#CLK_PRD;
adc_dat_a = 14'b0000000001011;
adc_dat_b = 14'b00000000101011;
#CLK_PRD;
adc_dat_a = 14'b0000000001100;
adc_dat_b = 14'b00000000101100;
#CLK_PRD;
adc_dat_a = 14'b0000000001101;
adc_dat_b = 14'b00000000101101;
#CLK_PRD;
adc_dat_a = 14'b0000000001110;
adc_dat_b = 14'b00000000101110;
#CLK_PRD;
adc_dat_a = 14'b0000000001111;
```

```
adc_dat_b = 14'b00000000101111;
#CLK_PRD;
adc_dat_a = 14'b0000000010001;
adc_dat_b = 14'b00000000110001;
#CLK_PRD;
adc_dat_a = 14'b0000000010010;
adc_dat_b = 14'b00000000110010;
#CLK_PRD;
adc_dat_a = 14'b00000000000000;
adc_dat_b = 14'b00000000000000;

#CLK_PRD;

adc_dat_a = 14'b00000000000001;
adc_dat_b = 14'b00000000100001;
#CLK_PRD;
adc_dat_a = 14'b00000000000010;
adc_dat_b = 14'b00000000100010;
#CLK_PRD;
adc_dat_a = 14'b00000000000011;
adc_dat_b = 14'b00000000100011;
#CLK_PRD;
adc_dat_a = 14'b00000000000100;
adc_dat_b = 14'b00000000100100;
#CLK_PRD;
adc_dat_a = 14'b00000000000101;
adc_dat_b = 14'b00000000100101;
#CLK_PRD;
adc_dat_a = 14'b00000000000110;
adc_dat_b = 14'b00000000100110;
#CLK_PRD;
adc_dat_a = 14'b00000000000111;
```

```
adc_dat_b = 14'b00000000100111;
#CLK_PRD
adc_dat_a = 14'b0000000001000;
adc_dat_b = 14'b00000000101000;
#CLK_PRD;
adc_dat_a = 14'b0000000001001;
adc_dat_b = 14'b00000000101001;
#CLK_PRD;
adc_dat_a = 14'b0000000001010;
adc_dat_b = 14'b00000000101010;
#CLK_PRD;
adc_dat_a = 14'b0000000001011;
adc_dat_b = 14'b00000000101011;
#CLK_PRD;
adc_dat_a = 14'b0000000001100;
adc_dat_b = 14'b00000000101100;
#CLK_PRD;
adc_dat_a = 14'b0000000001101;
adc_dat_b = 14'b00000000101101;
#CLK_PRD;
adc_dat_a = 14'b0000000001110;
adc_dat_b = 14'b00000000101110;
#CLK_PRD;
adc_dat_a = 14'b0000000001111;
adc_dat_b = 14'b00000000101111;
#CLK_PRD;
adc_dat_a = 14'b00000000010001;
adc_dat_b = 14'b00000000110001;
#CLK_PRD;
adc_dat_a = 14'b00000000010010;
adc_dat_b = 14'b00000000110010;
#CLK_PRD;
```

```
adc_dat_a = 14'b0000000000000000;  
adc_dat_b = 14'b0000000000000000;
```

```
#CLK_PRD;
```

```
adc_dat_a = 14'b000000000000001;  
adc_dat_b = 14'b0000010000001;
```

```
#CLK_PRD;
```

```
adc_dat_a = 14'b000000000000010;  
adc_dat_b = 14'b00000010000010;
```

```
#CLK_PRD;
```

```
adc_dat_a = 14'b000000000000011;  
adc_dat_b = 14'b00000010000011;
```

```
#CLK_PRD;
```

```
adc_dat_a = 14'b000000000000100;  
adc_dat_b = 14'b00000010000100;
```

```
#CLK_PRD;
```

```
adc_dat_a = 14'b000000000000101;  
adc_dat_b = 14'b00000010000101;
```

```
#CLK_PRD;
```

```
adc_dat_a = 14'b000000000000110;  
adc_dat_b = 14'b00000010000110;
```

```
#CLK_PRD;
```

```
adc_dat_a = 14'b000000000000111;  
adc_dat_b = 14'b00000010000111;
```

```
#CLK_PRD
```

```
adc_dat_a = 14'b000000000001000;  
adc_dat_b = 14'b00000010001000;
```

```
#CLK_PRD;
```

```
adc_dat_a = 14'b000000000001001;  
adc_dat_b = 14'b00000010001001;
```

```
#CLK_PRD;
```

```
adc_dat_a = 14'b00000000001010;
adc_dat_b = 14'b00000010001010;
#CLK_PRD;
adc_dat_a = 14'b00000000001011;
adc_dat_b = 14'b00000010001011;
#CLK_PRD;
adc_dat_a = 14'b00000000001100;
adc_dat_b = 14'b00000010001100;
#CLK_PRD;
adc_dat_a = 14'b00000000001101;
adc_dat_b = 14'b00000010001101;
#CLK_PRD;
adc_dat_a = 14'b00000000001110;
adc_dat_b = 14'b00000010001110;
#CLK_PRD;
adc_dat_a = 14'b00000000001111;
adc_dat_b = 14'b00000010001111;
#CLK_PRD;
adc_dat_a = 14'b00000000010001;
adc_dat_b = 14'b00000010010001;
#CLK_PRD;
adc_dat_a = 14'b00000000010010;
adc_dat_b = 14'b00000010010010;
#CLK_PRD;
adc_dat_a = 14'b00000000000000;
adc_dat_b = 14'b00000000000000;

#CLK_PRD

adc_dat_a = 14'b00000000000001;
adc_dat_b = 14'b00000000100001;
#CLK_PRD;
```

```
adc_dat_a = 14'b00000000000010;
adc_dat_b = 14'b00000000100010;
#CLK_PRD;
adc_dat_a = 14'b00000000000011;
adc_dat_b = 14'b00000000100011;
#CLK_PRD;
adc_dat_a = 14'b00000000000100;
adc_dat_b = 14'b00000000100100;
#CLK_PRD;
adc_dat_a = 14'b00000000000101;
adc_dat_b = 14'b00000000100101;
#CLK_PRD;
adc_dat_a = 14'b00000000000110;
adc_dat_b = 14'b00000000100110;
#CLK_PRD;
adc_dat_a = 14'b00000000000111;
adc_dat_b = 14'b00000000100111;
#CLK_PRD;
adc_dat_a = 14'b0000000001000;
adc_dat_b = 14'b00000000101000;
#CLK_PRD;
adc_dat_a = 14'b0000000001001;
adc_dat_b = 14'b00000000101001;
#CLK_PRD;
adc_dat_a = 14'b0000000001010;
adc_dat_b = 14'b00000000101010;
#CLK_PRD;
adc_dat_a = 14'b0000000001011;
adc_dat_b = 14'b00000000101011;
#CLK_PRD;
adc_dat_a = 14'b0000000001100;
adc_dat_b = 14'b00000000101100;
```

```

#CLK_PRD;
adc_dat_a = 14'b00000000001101;
adc_dat_b = 14'b00000000101101;
#CLK_PRD;
adc_dat_a = 14'b00000000001110;
adc_dat_b = 14'b00000000101110;
#CLK_PRD;
adc_dat_a = 14'b00000000001111;
adc_dat_b = 14'b00000000101111;
#CLK_PRD;
adc_dat_a = 14'b00000000010001;
adc_dat_b = 14'b00000000110001;
#CLK_PRD;
adc_dat_a = 14'b00000000010010;
adc_dat_b = 14'b00000000110010;
#CLK_PRD;
adc_dat_a = 14'b00000000000000;
adc_dat_b = 14'b00000000000000;

repeat(40) #CLK_PRD;

$finish;

end
endmodule

```

red_Pitaya.xdc:

```

#
# $Id: red_pitaya.xdc 961 2014-01-21 11:40:39Z matej.oblak $
#
# @brief Red Pitaya location constraints.
#

```


@Author Matej Oblak

#

(c) Red Pitaya <http://www.redpitaya.com>

#

#####

IO constraints

#####

ADC

ADC data

set_property IOSTANDARD LVCMOS18 [get_ports {adc_dat_a[*]}]

set_property IOB TRUE [get_ports {adc_dat_a[*]}]

#set_property IOSTANDARD LVCMOS18 [get_ports {adc_dat_b[*]}]

#set_property IOB TRUE [get_ports {adc_dat_b[*]}]

ADC 0 data

set_property PACKAGE_PIN V17 [get_ports {adc_dat_a[0]}]

set_property PACKAGE_PIN U17 [get_ports {adc_dat_a[1]}]

set_property PACKAGE_PIN Y17 [get_ports {adc_dat_a[2]}]

set_property PACKAGE_PIN W16 [get_ports {adc_dat_a[3]}]

set_property PACKAGE_PIN Y16 [get_ports {adc_dat_a[4]}]

set_property PACKAGE_PIN W15 [get_ports {adc_dat_a[5]}]

set_property PACKAGE_PIN W14 [get_ports {adc_dat_a[6]}]

set_property PACKAGE_PIN Y14 [get_ports {adc_dat_a[7]}]

set_property PACKAGE_PIN W13 [get_ports {adc_dat_a[8]}]

set_property PACKAGE_PIN V12 [get_ports {adc_dat_a[9]}]

set_property PACKAGE_PIN V13 [get_ports {adc_dat_a[10]}]

set_property PACKAGE_PIN T14 [get_ports {adc_dat_a[11]}]

set_property PACKAGE_PIN T15 [get_ports {adc_dat_a[12]}]

```

set_property PACKAGE_PIN V15 [get_ports {adc_dat_a[13]}]
#set_property PACKAGE_PIN T16 [get_ports {adc_dat_i[0][14]}]
#set_property PACKAGE_PIN V16 [get_ports {adc_dat_i[0][15]}]

# ADC 1 data
set_property PACKAGE_PIN T17 [get_ports {adc_dat_b[0]}]
set_property PACKAGE_PIN R16 [get_ports {adc_dat_b[1]}]
set_property PACKAGE_PIN R18 [get_ports {adc_dat_b[2]}]
set_property PACKAGE_PIN P16 [get_ports {adc_dat_b[3]}]
set_property PACKAGE_PIN P18 [get_ports {adc_dat_b[4]}]
set_property PACKAGE_PIN N17 [get_ports {adc_dat_b[5]}]
set_property PACKAGE_PIN R19 [get_ports {adc_dat_b[6]}]
set_property PACKAGE_PIN T20 [get_ports {adc_dat_b[7]}]
set_property PACKAGE_PIN T19 [get_ports {adc_dat_b[8]}]
set_property PACKAGE_PIN U20 [get_ports {adc_dat_b[9]}]
set_property PACKAGE_PIN V20 [get_ports {adc_dat_b[10]}]
set_property PACKAGE_PIN W20 [get_ports {adc_dat_b[11]}]
set_property PACKAGE_PIN W19 [get_ports {adc_dat_b[12]}]
set_property PACKAGE_PIN Y19 [get_ports {adc_dat_b[13]}]
#set_property PACKAGE_PIN W18 [get_ports {adc_dat_i[1][14]}]
#set_property PACKAGE_PIN Y18 [get_ports {adc_dat_i[1][15]}]

#set_property IOSTANDARD DIFF_HSTL_I_18 [get_ports adc_clk_i[*]]
#set_property PACKAGE_PIN U18 [get_ports adc_clk_i[1]]
#set_property PACKAGE_PIN U19 [get_ports adc_clk_i[0]]

# Output ADC clock
#set_property IOSTANDARD LVCMOS18 [get_ports {adc_clk_o[*]}]
#set_property SLEW FAST [get_ports {adc_clk_o[*]}]
#set_property DRIVE 8 [get_ports {adc_clk_o[*]}]
#set_property IOB TRUE [get_ports {adc_clk_o[*]}]

```

```

#set_property PACKAGE_PIN N20 [get_ports {adc_clk_o[0]}]
#set_property PACKAGE_PIN P20 [get_ports {adc_clk_o[1]}]

# ADC clock stabilizer
#set_property IOSTANDARD LVCMOS18 [get_ports adc_cdcs_o]
#set_property PACKAGE_PIN V18 [get_ports adc_cdcs_o]
#set_property SLEW FAST [get_ports adc_cdcs_o]
#set_property DRIVE 8 [get_ports adc_cdcs_o]

### DAC

# data
set_property IOSTANDARD LVCMOS33 [get_ports {dac_dat_o[*]}]
set_property SLEW SLOW [get_ports {dac_dat_o[*]}]
set_property DRIVE 4 [get_ports {dac_dat_o[*]}]
#set_property IOB TRUE [get_ports {dac_dat_o[*]}]

set_property PACKAGE_PIN M19 [get_ports {dac_dat_o[0]}]
set_property PACKAGE_PIN M20 [get_ports {dac_dat_o[1]}]
set_property PACKAGE_PIN L19 [get_ports {dac_dat_o[2]}]
set_property PACKAGE_PIN L20 [get_ports {dac_dat_o[3]}]
set_property PACKAGE_PIN K19 [get_ports {dac_dat_o[4]}]
set_property PACKAGE_PIN J19 [get_ports {dac_dat_o[5]}]
set_property PACKAGE_PIN J20 [get_ports {dac_dat_o[6]}]
set_property PACKAGE_PIN H20 [get_ports {dac_dat_o[7]}]
set_property PACKAGE_PIN G19 [get_ports {dac_dat_o[8]}]
set_property PACKAGE_PIN G20 [get_ports {dac_dat_o[9]}]
set_property PACKAGE_PIN F19 [get_ports {dac_dat_o[10]}]
set_property PACKAGE_PIN F20 [get_ports {dac_dat_o[11]}]
set_property PACKAGE_PIN D20 [get_ports {dac_dat_o[12]}]
set_property PACKAGE_PIN D19 [get_ports {dac_dat_o[13]}]

```

```

# control

#set_property IOSTANDARD LVCMOS33 [get_ports dac_*_o]
#set_property SLEW FAST [get_ports dac_*_o]
#set_property DRIVE 8 [get_ports dac_*_o]
#set_property IOB TRUE [get_ports dac_*_o]

#set_property PACKAGE_PIN M17 [get_ports dac_wrt_o]
#set_property PACKAGE_PIN N16 [get_ports dac_sel_o]
#set_property PACKAGE_PIN M18 [get_ports dac_clk_o]
#set_property PACKAGE_PIN N15 [get_ports dac_rst_o]

### PWM DAC

#set_property IOSTANDARD LVCMOS18 [get_ports {dac_pwm_o[*]}]
#set_property SLEW FAST [get_ports {dac_pwm_o[*]}]
#set_property DRIVE 12 [get_ports {dac_pwm_o[*]}]
#set_property IOB TRUE [get_ports {dac_pwm_o[*]}]

#set_property PACKAGE_PIN T10 [get_ports {dac_pwm_o[0]}]
#set_property PACKAGE_PIN T11 [get_ports {dac_pwm_o[1]}]
#set_property PACKAGE_PIN P15 [get_ports {dac_pwm_o[2]}]
#set_property PACKAGE_PIN U13 [get_ports {dac_pwm_o[3]}]

### XADC

#set_property IOSTANDARD LVCMOS33 [get_ports {vinp_i[*]}]
#set_property IOSTANDARD LVCMOS33 [get_ports {vinn_i[*]}]
#AD0
#set_property PACKAGE_PIN C20 [get_ports {vinp_i[1]}]
#set_property PACKAGE_PIN B20 [get_ports {vinn_i[1]}]
#AD1
#set_property PACKAGE_PIN E17 [get_ports {vinp_i[2]}]
#set_property PACKAGE_PIN D18 [get_ports {vinn_i[2]}]
#AD8

```

```

#set_property PACKAGE_PIN B19 [get_ports {vinp_i[0]}]
#set_property PACKAGE_PIN A20 [get_ports {vinn_i[0]}]
#AD9
#set_property PACKAGE_PIN E18 [get_ports {vinp_i[3]}]
#set_property PACKAGE_PIN E19 [get_ports {vinn_i[3]}]
#V_0
#set_property PACKAGE_PIN K9 [get_ports {vinp_i[4]}]
#set_property PACKAGE_PIN L10 [get_ports {vinn_i[4]}]

### Expansion connector
#set_property IOSTANDARD LVCMOS33 [get_ports {exp_p_io[*]}]
#set_property IOSTANDARD LVCMOS33 [get_ports {exp_n_io[*]}]
#set_property SLEW    FAST    [get_ports {exp_p_io[*]}]
#set_property SLEW    FAST    [get_ports {exp_n_io[*]}]
#set_property DRIVE   8       [get_ports {exp_p_io[*]}]
#set_property DRIVE   8       [get_ports {exp_n_io[*]}]

#set_property PACKAGE_PIN G17 [get_ports {exp_p_io[0]}]
#set_property PACKAGE_PIN G18 [get_ports {exp_n_io[0]}]
#set_property PACKAGE_PIN H16 [get_ports {exp_p_io[1]}]
#set_property PACKAGE_PIN H17 [get_ports {exp_n_io[1]}]
#set_property PACKAGE_PIN J18 [get_ports {exp_p_io[2]}]
#set_property PACKAGE_PIN H18 [get_ports {exp_n_io[2]}]
#set_property PACKAGE_PIN K17 [get_ports {exp_p_io[3]}]
#set_property PACKAGE_PIN K18 [get_ports {exp_n_io[3]}]
#set_property PACKAGE_PIN L14 [get_ports {exp_p_io[4]}]
#set_property PACKAGE_PIN L15 [get_ports {exp_n_io[4]}]
#set_property PACKAGE_PIN L16 [get_ports {exp_p_io[5]}]
#set_property PACKAGE_PIN L17 [get_ports {exp_n_io[5]}]
#set_property PACKAGE_PIN K16 [get_ports {exp_p_io[6]}]
#set_property PACKAGE_PIN J16 [get_ports {exp_n_io[6]}]
#set_property PACKAGE_PIN M14 [get_ports {exp_p_io[7]}]

```

```
#set_property PACKAGE_PIN M15 [get_ports {exp_n_io[7]}]
```

```
#set_property PULLDOWN TRUE [get_ports {exp_p_io[0]}]
```

```
#set_property PULLDOWN TRUE [get_ports {exp_n_io[0]}]
```

```
#set_property PULLUP TRUE [get_ports {exp_p_io[7]}]
```

```
#set_property PULLUP TRUE [get_ports {exp_n_io[7]}]
```

```
### SATA connector
```

```
#set_property IOSTANDARD LVCMOS18 [get_ports {daisy_p_o[*]}]
```

```
#set_property IOSTANDARD LVCMOS18 [get_ports {daisy_n_o[*]}]
```

```
#set_property IOSTANDARD LVCMOS18 [get_ports {daisy_p_i[*]}]
```

```
#set_property IOSTANDARD LVCMOS18 [get_ports {daisy_n_i[*]}]
```

```
#set_property PACKAGE_PIN T12 [get_ports {daisy_p_o[0]}]
```

```
#set_property PACKAGE_PIN U12 [get_ports {daisy_n_o[0]}]
```

```
#set_property PACKAGE_PIN U14 [get_ports {daisy_p_o[1]}]
```

```
#set_property PACKAGE_PIN U15 [get_ports {daisy_n_o[1]}]
```

```
#set_property PACKAGE_PIN P14 [get_ports {daisy_p_i[0]}]
```

```
#set_property PACKAGE_PIN R14 [get_ports {daisy_n_i[0]}]
```

```
#set_property PACKAGE_PIN N18 [get_ports {daisy_p_i[1]}]
```

```
#set_property PACKAGE_PIN P19 [get_ports {daisy_n_i[1]}]
```

```
### LED
```

```
#set_property IOSTANDARD LVCMOS33 [get_ports {led_o[*]}]
```

```
#set_property SLEW SLOW [get_ports {led_o[*]}]
```

```
#set_property DRIVE 4 [get_ports {led_o[*]}]
```

```
#set_property PACKAGE_PIN F16 [get_ports {led_o[0]}]
```

```
#set_property PACKAGE_PIN F17 [get_ports {led_o[1]}]
```

```
#set_property PACKAGE_PIN G15 [get_ports {led_o[2]}]
```

```
#set_property PACKAGE_PIN H15 [get_ports {led_o[3]}]
```

```
#set_property PACKAGE_PIN K14 [get_ports {led_o[4]}]
```

```

#set_property PACKAGE_PIN G14 [get_ports {led_o[5]}]
#set_property PACKAGE_PIN J15 [get_ports {led_o[6]}]
#set_property PACKAGE_PIN J14 [get_ports {led_o[7]}]

#####

# Clock constraints                                     #
#####

#NET "adc_clk" TNM_NET = "adc_clk";
#TIMESPEC TS_adc_clk = PERIOD "adc_clk" 125 MHz;
create_clock -period 8.000 -name adc_clk [get_ports adc_clk_i]

#set_input_delay -clock adc_clk 3.400 [get_ports adc_dat_i[*][*]]
#create_clock -period 4.000 -name rx_clk [get_ports daisy_p_i[1]]

#set_false_path -from [get_clocks adc_clk] -to [get_clocks dac_clk_o]
#set_false_path -from [get_clocks adc_clk] -to [get_clocks dac_clk_2x]
#set_false_path -from [get_clocks adc_clk] -to [get_clocks dac_clk_2p]
#set_false_path -from [get_clocks clk_fpga_0] -to [get_clocks adc_clk]
#set_false_path -from [get_clocks clk_fpga_0] -to [get_clocks dac_clk_1x]
#set_false_path -from [get_clocks clk_fpga_0] -to [get_clocks dac_clk_2x]
#set_false_path -from [get_clocks clk_fpga_0] -to [get_clocks dac_clk_2p]
#set_false_path -from [get_clocks clk_fpga_0] -to [get_clocks ser_clk]
#set_false_path -from [get_clocks clk_fpga_0] -to [get_clocks pdm_clk]
#set_false_path -from [get_clocks dac_clk_o] -to [get_clocks dac_clk_2x]
#set_false_path -from [get_clocks dac_clk_o] -to [get_clocks dac_clk_2p]

```